

# Izrada trodimenzionalnog modela laboratorija

---

**Radić, Ivan**

**Master's thesis / Diplomski rad**

**2022**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:235:947189>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-12-19**

*Repository / Repozitorij:*

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)





SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# **DIPLOMSKI RAD**

**Ivan Radić**

Zagreb, 2022.



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# **DIPLOMSKI RAD**

**Izrada trodimenzionalnog modela laboratorija**  
**Creation of a three-dimensional model of the laboratory**

Mentor:  
Prof. dr. sc. Tomislav Stipančić

Student:  
Ivan Radić

Zagreb, 2022.

Izjavljujem da sam ovaj rad izradio samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem se svojem mentoru doc.dr.sc. Tomislavu Stipančiću na tome što je uvijek bio dostupan i spreman pomoći tokom izrade ovoga rada te na uloženom vremenu i trudu.

Zahvaljujem se i asistentu mag.ing.mech. Leonu Korenu koji je nesebično odvajao svoje vrijeme da bi mi pomogao oko snimanja laboratorija te me sa svojim savjetima usmjeravao prema uspješnom završetku ovoga rada.

Također, velika zahvala kolegi Ivanu Tišljaru koji je posudio svoj iPad Pro za snimanje laboratorija.

Posebna zahvala mojim roditeljima, djevojci Luciji i prijateljima koji su mi pružali podršku kroz sretne, a i one teške trenutke studiranja.

Ivan Radić



SVEUČILIŠTE U ZAGREBU  
**FAKULTET STROJARSTVA I BRODOGRADNJE**



Središnje povjerenstvo za završne i diplomske ispite  
Povjerenstvo za diplomske radove studija strojarstva za smjerove:  
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment,  
inženjerstvo materijala te mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum:	Prilog:
Klasa:	602-14/22-6/1
Ur. broj:	15-1703-22-

## DIPLOMSKI ZADATAK

Student: **IVAN RADIĆ** Mat. br.: 0035210937

Naslov rada na hrvatskom jeziku: **Izrada trodimenzionalnog modela laboratorija**

Naslov rada na engleskom jeziku: **Creation of a three-dimensional model of the laboratory**

Opis zadatka:

Proces kreiranja 3D prikaza nekog objekta, plohe, linije ili točke u virtualnom 3D okolišu koristeći oblak točaka, odnosno poligona, zove se 3D modeliranje. Ono se koristi u različitim područjima ljudskog djelovanja uključujući inženjerstvo, industriju igara, oglašavanje, aditivne tehnologije, medicinu, različite grane znanosti i sl.

Prilikom izrade 3D modela prostora moguće je koristiti stereo vizijski sustav koji će snimiti cijelu prostoriju dio po dio te na osnovu snimki upotrebom određene programske podrške napraviti oblak točaka (engl. Point Cloud). Oblak točaka se u daljnjem koraku koristi za definiranje smislenih cjelina u sklopu novog 3D modela prostora.

U sklopu rada potrebno je:

1. Odabrati prikladnu stereo vizijsku kameru za provođenje akvizicije prostora Laboratorija za projektiranje izradbenih i montažnih sustava.
2. Odabrati i upoznati se s odgovarajućom programskom podrškom za izradu 3D modela prostora.
3. Virtualni laboratorij kreirati da bude što vjernija slika stvarnog Laboratorija.
4. Integrirati dobiveni prostor u odgovarajuću okolinu za virtualnu stvarnost koristeći neki od dostupnih radnih okvira za rad s virtualnom stvarnošću (npr. Unreal Engine, Blender ili Unity).


Razvijeno cjelovito rješenje potrebno je eksperimentalno evaluirati u sklopu Laboratorija za projektiranje izradbenih i montažnih sustava.

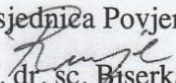
U radu je potrebno navesti korištenu literaturu te eventualno dobivenu pomoć.

Zadatak zadan:  
29. rujna 2022.

Rok predaje rada:  
1. prosinca 2022.

Predvideni datum obrane:  
12. prosinca do 16. prosinca 2022.

Zadatak zadao:   
doc. dr. sc. Tomislav Stipančić

Predsjednica Povjerenstva:  
  
prof. dr. sc. Biserka Runje

## Sadržaj

POPIS SLIKA.....	II
SAŽETAK .....	IV
SUMMARY .....	V
1. UVOD.....	1
2. ODABIR STEREOVIZIJSKE KAMERE .....	4
2.1. Intel RealSense D435.....	5
2.2. Xbox 360 Kinect .....	7
2.3. Apple-ov LiDAR sustav .....	9
3. SNIMANJE PROSTORA.....	12
3.1. Kalibriranje kamera .....	12
3.2. SLAM pristup .....	16
3.3. Evaluacija snimki sa Intel RealSense D435 stereo kamerom.....	21
3.4. Evaluacija snimki sa Xbox360 Kinect kamerom.....	28
3.5. Evaluacija snimki sa iPad Pro LiDAR sustavom.....	40
4. USPOREDBA REZULTATA I ODABIR NAJBOLJEG MODELA .....	44
5. POPRAVLJANJE ODABRANOG 3D MODELA .....	46
6. IMPLEMENTIRANJE 3D MODELA U VIRTUALNO OKRUŽENJE .....	52
7. ZAKLJUČAK.....	58
LITERATURA .....	59

## POPIS SLIKA

Slika 1. VR naočale u upotrebi s rukavicama [1] .....	2
Slika 2. Intel RealSense D435 stereo kamera [3] .....	5
Slika 3. Osnovni dijelovi Intel RealSense D435 stereo kamere [3].....	6
Slika 4. Xbox 360 Kinect kamera [5] .....	7
Slika 5. Primjer aplikacije sa primjenom proširene stvarnosti [7].....	9
Slika 6. Sustav dvije kamere i lidara na tabletu iPad Pro [6].....	10
Slika 7. Rezultati točnosti LiDAR skeniranja mobilnim uređajem iPhone 12 Pro [8].....	11
Slika 8. Rezultati kalibracije dubine na primjeru zida [9] .....	13
Slika 9. Distorzijska slika Xbox 360 Kinect kamere .....	14
Slika 10. Distorzijska slika Intel RealSense D435 stereo kamere .....	15
Slika 11. Mapa kretanja kamere u prostoru .....	16
Slika 12. Algoritam rada RTAB-Map-a [11].....	17
Slika 13. Prikaz povezivanja lokacija SLAM pristupom [11] .....	18
Slika 14. Dupliciranje objekta kao rezultat nezatvaranja petlje.....	19
Slika 15. Kreiranje nove baze podataka u programskom sučelju RTAB-Map.....	21
Slika 16. Odabir kamere kojom će se snimati u programu RTAB-Map.....	22
Slika 17. Odabir opcije Preferences .....	23
Slika 18. Umetanje kalibracijske datoteke u preferencije programa .....	23
Slika 19. Gumb za pokretanje snimanja .....	24
Slika 20. Prozor programskog sučelja RTAB-Map-a tokom snimanja prostorije.....	24
Slika 21. Pogled 1- Oblak točaka proizašao iz snimki s kamerom Intel RealSense D435 .....	26
Slika 22. Pogled 2- Oblak točaka proizašao iz snimki s kamerom Intel RealSense D435 .....	27
Slika 23. Udaljeni pogled na kreirani oblak točaka laboratorija.....	27
Slika 24. Odabir kamere Kinect u programu RTAB-Map.....	28
Slika 25. Spremanje oblaka točaka .....	29
Slika 26. Značajke odabrane prije spremanja oblaka točaka .....	30
Slika 27. Umetanje oblaka točaka u programu MeshLab .....	30
Slika 28. Otvaranje oblaka točaka .....	31
Slika 29. Pogled 1- Oblak točaka proizašao iz snimki sa kamerom Xbox360 Kinect.....	31
Slika 30. Pogled 2- Oblak točaka proizašao iz snimki sa kamerom Xbox360 Kinect.....	32
Slika 31. Pogled na oblak točaka prostorije sa veće udaljenosti.....	33
Slika 32. Brisanje viška točaka u programu MeshLab .....	34
Slika 33. Konačni izgled 3D oblaka točaka nakon brisanja.....	34
Slika 34. Odabir opcije računanja normala za proizvoljni skup točaka.....	35
Slika 35. Odabir veličine skupa točaka i broja iteracija za računanje normala .....	36
Slika 36. Odabir metode rekonstrukcije površine: Screened Poisson.....	37
Slika 37. Rekonstrukcija površine Screened Poisson unos ulaznih argumenata .....	38
Slika 38. Poruka iz statusne trake .....	38
Slika 39. Pogled 1- Izgled kreiranog 3D modela laboratorija .....	39
Slika 40. Pogled 2- Izgled kreiranog 3D modela laboratorija .....	39
Slika 41. Ikona i naziv aplikacije .....	40
Slika 42. Pokretanje skeniranja prostora LiDAR-om .....	40
Slika 43. Zaslona aplikacije tokom snimanja prostora .....	41

Slika 44. Parametri procesiranja skeniranih podataka prostorije.....	42
Slika 45. Procesiranje podataka na osnovu prethodno zadanih parametara .....	43
Slika 46. Kreirani 3D model iz skeniranja LiDAR-om .....	45
Slika 47. Ikona aplikacije Blender .....	46
Slika 48. Učitavanje 3D modela laboratorija u programu Blender.....	46
Slika 49. 3D model bez teksture .....	47
Slika 50. 3D model sa primijenjenom teksturom.....	47
Slika 51. Bliži pogled na 3D model prostorije.....	48
Slika 52. Pogled iz drugog kuta na 3D model laboratorija.....	49
Slika 53. Promjena načina rada u programskom sučelju Blender-a .....	49
Slika 54. Izgled 3D modela u načinu rada za uređivanje.....	50
Slika 55. Promjena načina odabira u odabir rubova .....	50
Slika 56. Popravljen 3D mreža modela.....	51
Slika 57. Pokretanje programa Unreal Engine.....	52
Slika 58. Pokretanje novog projekta .....	52
Slika 59. Kreiranje novog projekta sa opcijom virtualne stvarnosti.....	53
Slika 60. Mapa predložka u projektu virtualne stvarnosti.....	53
Slika 61. Brisanje nepotrebnih virtualnih objekata na standardnom predlošku.....	54
Slika 62. Dodavanje modela u virtualno okruženje .....	55
Slika 63. Otvaranje 3D modela laboratorija u .fbx formatu .....	55
Slika 64. Izgled virtualne scene nakon implementiranja 3D modela laboratorija .....	56
Slika 65. Postavljena početna pozicija korisnika .....	57
Slika 66. Povezivanje virtualnog okruženja sa VR naočalama.....	57
Slika 67. Početni prikaz na VR naočalama .....	57



## SAŽETAK

U ovome radu biti će objašnjen postupak kreiranja 3D prikaza prostorije Laboratorija za projektiranje izradbenih i montažnih sustava. Upotrebom stereo kamere Intel RealSense D435 navedeni prostor je snimljen i iz dobivenih snimki se pomoću programa RTAB-Map kreira oblak točaka (engl. *Point Cloud*). U cilju generiranja što vjernijeg prikaza laboratorija bilo je potrebno otkloniti višak točaka koje su raspršene u prostoru te ne predstavljaju nikakvu funkcionalnu cjelinu prostora laboratorija nego su one šum nastao tijekom snimanja prostorije. Otklanjanje šuma i generiranje 3D mreže u takvom novom oblaku točaka te optimiranje 3D mreže primjenom programa MeshLab također su teme kojih će se ovaj rad dotaknuti. Cilj ovog rada je dobiti takav 3D model laboratorija koji će biti što sličniji onom stvarnom. Da bi se taj pokušaj prikaza stvarnog prostora u virtualnom svijetu mogao evaluirati, model će biti integriran u okolinu za virtualnu stvarnost. Upotrebom VR naočala moći će se steći stvarni doživljaj kvalitete integracije realnog svijeta u onaj virtualni.

Ključne riječi: stereo kamera, virtualni svijet, 3D model, RTAB-Map, oblak točaka, MeshLab, VR naočale

## **SUMMARY**

In this thesis 3D view creation procedure of Laboratory for designing production and assembly systems will be explained. Using stereo camera Intel RealSense D435 room is recorded and point cloud is created in software RTAB-Map from obtained recordings. In order to generate as faithful a representation of laboratory as possible it's needed to exclude excess points which are dispersed in space and they don't represent any functional part of laboratory room, they are the noise created while recording of the room. Noise removing and generating 3D mesh in this new point cloud and optimising 3D mesh in software MeshLab are subjects that will be subject of this thesis too. Goal of this thesis is to create 3D model of laboratory which will be as similar as possible to the real one. In order to evaluate this attempt to represent the real space in the virtual world, the model will be integrated into the virtual reality environment. By using VR glasses, it will be possible to get a real experience of the quality of the integration of the real world into the virtual one.

Key words: stereo camera, virtual world, 3D model, RTAB-Map, point cloud, MeshLab, VR glasses

## 1. UVOD

Tehnološki razvoj u zadnjem stoljeću, prije svega razvoj vizijskih sustava i računalnih komponenti (procesori i grafičke kartice) doveo je do preduvjeta za pojavu virtualne stvarnosti. Virtualna stvarnost je sve češći pojam koji se može čuti u svakodnevnom razgovoru, a definitivno pronalazi svoje mjesto i u industriji gdje se sve više primjenjuje.

U industriji računalnih igara ovaj način simuliranja svijeta u kojem se korisnik nalazi je preuzeo veliki dio tržišta te se kvaliteta doživljaja unaprjeđuje iz dana u dan. Korisnik se nalazi u virtualnom okruženju koje se može kreirati u programima poput Unreal Engine-a ili Unity-a. To su programi koji su prvotno služili za izradu računalnih igrica, ali kao i svi na tržištu, morali su pratiti trendove te se sada njihove platforme koriste i za kreiranje virtualnih okruženja. Virtualna okruženja su inspirirana stvarnim svijetom te pokušavaju što vjernije simulirati i korisniku približiti stvarni doživljaj. Navedeni zadatak je jako težak i u rješavanju istog se nailazi na pregršt prepreka koje se moraju premostiti. Glavna prepreka je kompleksnost stvarnog svijeta u kojem se nalazi pregršt detalja koje treba moći vjerno prikazati. Što je veće virtualno okruženje u kojeg se želi smjestiti korisnika to će biti potrebno više ljudskog i procesorskog rada da bi se to postiglo. U velikom broju slučajeva, stvara se prazno okruženje u kojem se onda iz jednostavnih elemenata poput kocke, kugle, valjka i sličnih geometrijskih tijela grade virtualni elementi iz kojih se nadalje kroje cjeline koje upotpunjuju cijelo virtualno okruženje i to je zapravo ono što će korisnik vidjeti kada pogleda u VR naočale.

VR (engl. *Virtual Reality*) naočale su uređaj koji je predviđen da korisnik stavi na glavu i da gleda kroz dva stereoskopska monitora koja se poklapaju sa pozicijom očiju. Kroz ta dva monitora osoba može vidjeti projekciju virtualnog okruženja koje je kreirano na način kako je to prethodno definirano. Također, uređaj sadrži akcelerometar i žiroskop koji služe za praćenje pozicije korisnika unutar virtualnog svijeta. Virtualno okruženje nije kreirano da bi se samo posmatralo iz neke udaljene točke nego se čovjek može kretati po virtualnom svijetu i čak u nekim slučajevima i obavljati neke radnje pomoću tzv. rukavica (engl. *data gloves*). Rukavice služe za interakciju između čovjeka u realnom svijetu i virtualnog svijeta, moguća je manipulacija gibanja virtualnim objektima upotrebom rukavica koje su izgledom kao klasične rukavice, ali uz to sadrže pregršt senzora koji šalju signale o poziciji ruku i prstiju koji su bitni za izvršenje traženih radnji unutar virtualnog svijeta.



**Slika 1. VR naočale u upotrebi s rukavicama [1]**

Tehnike i kompleksnost interakcije čovjeka i virtualnog svijeta se konstantno razvijaju i siguran sam da će u narednom vremenu virtualna realnost doživjeti još veću primjenu, ne samo u gaming industriji, nego i u vojnoj industriji, turizmu, medicini ili u bilo kojoj grani gdje je potrebno obaviti određeni trening prije izvršavanja same akcije, u slučaju medicine to je operacija. Trend razvijanja ove grane će definitivno pratiti trend razvijanja industrije, vidimo da se s vremenom razvijaju sve bolje i jače komponente, specifikacije računala se unaprjeđuju rapidno što vidimo po pojavi tzv. radnih stanica (engl. *workstations*). To su računala sa puno jačom specifikacijom od normalnih PC (engl. *Personal Computer*) računala i koriste se uglavnom na profesionalnoj razini za obavljanje kompleksnijih zadataka. Takva računala se mogu nositi sa zahtjevnim geometrijskim i grafičkim operacijama iza kojih se kriju složene matematičke operacije koje računalo mora izvršiti, a potrebne su da bi se kreirali virtualni objekti i virtualno okruženje. Potrebno je u ovome radu pokazati način dobivanja virtualnog okruženja koji je drukčiji od prethodno navedenog. U ovom slučaju virtualno okruženje će se dobiti na temelju snimke realnog prostora. Temelj tomu će biti oblak točaka (engl. *point cloud*) koji će biti rezultat toga

snimanja. Iz dobivenog oblaka točaka kreirati će se 3D model sustava koji će naknadno biti implementiran u virtualno okruženje. U ostvarenju navedenog trebati će se odabrati adekvatni vizijski sustav o čemu će biti riječi u narednom poglavlju. Kakav god da bude odabir, dobiveni oblak točaka neće biti savršen, a samim time i model koji će se generirati iz istoga u programu MeshLab će se trebati doraditi primjenom optimizacijskih algoritama. Nakon što se procjeni da je model prihvatljiv, tj. da vjerno prikazuje prostoriju Laboratorija za projektiranje izradbenih i montažnih sustava, rezultat će se evaluirati tako da se model postavi u virtualno okruženje.

## 2. ODABIR STEREOVIZIJSKE KAMERE

Da bi se ostvario prethodno navedeni zadatak te iz snimanja prostora dobio oblak točaka, potrebno je koristiti adekvatan stereo vizijski sustav koji će programu za mapiranje prostora moći proslijediti poziciju i vrijednost piksela u prostoru. Pod vrijednošću piksela smatram RGB (engl. *Red, Green, Blue*) vrijednosti koje definiraju boju piksela koji u konačnici tvore teksturu cijelog prostora. Obradom tih podataka program može konstruirati oblak točaka snimljenog prostora.

Klasične kamere koje se koriste u fotoaparatom ili u mobitelima ne mogu se koristiti u ovome slučaju. Slika koja se generira na fotoosjetljivom čipu ovakvih kamera je dvodimenzionalna i ona ne sadrži vrijednosti udaljenosti piksela od leće kamere. Bez toga podatka nemoguće je generirati oblak točaka te se iz toga razloga za ovu svrhu koriste stereo kamere. Potrebno je navesti da noviji modeli mobilnih uređaja marke iPhone sadrže lidar u svom sustavu kamera pa takvi uređaji ne spadaju u navedenu skupinu koja se ne može koristiti u svrhu generiranja oblaka točaka.

Stereo kamera je sustav koji se sastoji od dva fotoosjetljiva senzora sa lećom međusobno razmaknuta otprilike 6.35 cm što odgovara udaljenosti između ljudskih očiju. Slike se dobivaju istovremeno sa oba dvije kamere te se nalaze zajedničke točke na njima. Pomoću tih zajedničkih točki vrši se triangulacija, odnosno pronalaženje koordinata tih zajedničkih točaka u prostoru. U razmatranje će se uzeti dva modela stereo kamera. To su Intel RealSense D435 i Kinect Xbox 360. Ove dvije kamere posjeduju laboratorij u kojem se vrše snimanja i izrada modela stoga su one u izboru. Osim tih kamera, kao kandidati za snimanje biti će i Appleovi iPad Pro i iPhone 13 Pro koji sadrže LiDAR sustav koji će posebno biti objašnjen u potpoglavlju 2.3.

## 2.1. Intel RealSense D435

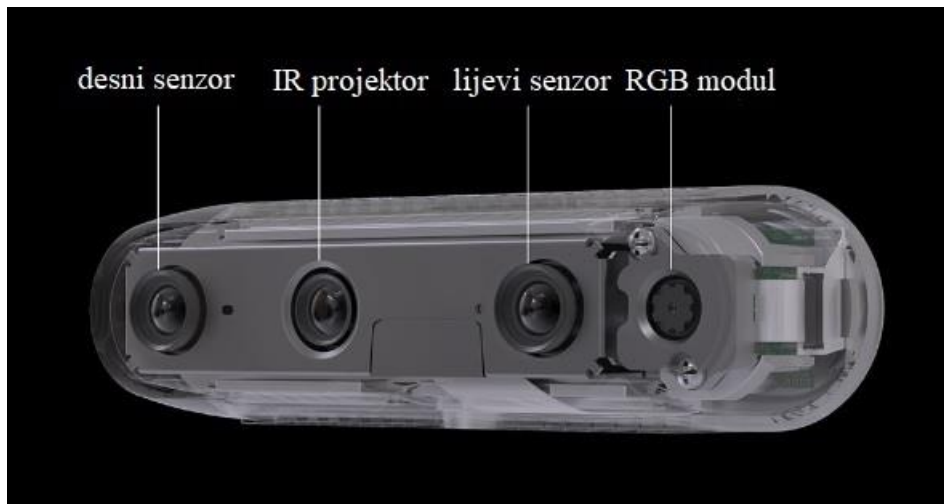
Ova kamera spada u područje RGB-D stereo kamera. RGB je prethodno u ovom poglavlju pojašnjeno što znači, a dio oznake sa slovom D predstavlja (engl. *Depth*) podatak o „dubini“, tj. udaljenosti detektirane točke u odnosu na kameru. Stereo kamera ovog tipa se sastoji od dvije standardne leće međusobno razmaknute kako je prethodno već opisano, uz to u kućištu kamere se nalazi i IR (engl. *Infrared*) projektor koji služi kao izvor infracrvenog svjetla koje osvjetljava scenu, objekt ili osobu u svrhu skupljanja podataka o dubini. Također, u kućištu se nalazi i RGB modul koji služi za prikupljanje RGB vrijednosti za svaki piksel.



**Slika 2. Intel RealSense D435 stereo kamera [3]**

Ovakav tip kamere ima najšire vidno polje (engl. *Field Of View*) od svih Intel-ovih kamera. Uz to, dohvaćanje podataka o pikselima vezanim za dubinu obavlja se tehnikom globalnog zatvaranja (engl. *Global Shutter*). To znači da se uzimanje slike obavlja od jednom za sve piksele, a ne od gornjeg reda piksela prema donjem kao što je to u pojedinim slučajevima. Prednosti ovog pristupa izuzimanja slika je u tome što se mogu pratiti pokreti većih brzina pri kojem je spriječeno zamućivanje dubinskih slika. Također, olakšano je i snimanje u situacijama u kojem je slabo osvjetljenje.

Bitno je napomenuti da se može koristiti i u vanjskom okruženju, tj. nesređenoj okolini koja nema neke granice poput zidova kod snimanja unutarnjih okruženja. Rezolucija slike koja se postiže kod dubinskih snimanja je 1280x720 piksela, a maksimalni broj slika u sekundi je 90. Kod RGB snimanja maksimalna rezolucija slike koja se postiže je 1920x1080 piksela, a broj slika u sekundi je 30. Rezolucija samog senzora iznosi 2 MP (engl. *Megapixels*).



**Slika 3. Osnovni dijelovi Intel RealSense D435 stereo kamere [3]**

Cijena uređaja je pristupačna, a dimenzije i masa su relativno male što ne predstavlja probleme za implementiranje kamere u konstrukcije gdje je potreban ovakav vizijski sustav. Zbog ovakvih performansi ovaj tip kamere se koristi u robotici gdje se primjenjuje za navođenje robota. Prepoznavanje objekata je isto vrlo važan zadatak koji je dio robotike, a obavlja se uz upotrebu ove kamere. U ovome radu, kamera će poslužiti za snimanje prostorijske laboratorija, a zbog prethodno navedenih prednosti smatra se kao odličan izbor za dobivanje što točnijeg oblaka točaka.



## 2.2. Xbox 360 Kinect

Da bi se rezultati snimanja sa stereo kamerom Intel RealSense D435 mogli usporediti s nečime, u okviru ovoga rada koristiti će se još jedna stereo kamera, a to je Xbox 360 Kinect stereo kamera. To je kamera koja se nalazi u laboratoriju te će dobro poslužiti da se i s njom obavi procedura snimanja prostorijske i nakon toga na osnovu tih snimki generiranje oblaka točaka. Usporedbom dobivenih oblaka točaka moći će se procijeniti koja kamera je bolja za ovu namjenu te će se s tim kvalitetnijim oblakom točaka kreirati 3D model.

Treba navesti kako je ova kamera razvijena od strane Microsofta davne 2010. godine, a služila je za praćenje pokreta korisnika igraće konzole Xbox. Sustav se sastoji od RGB kamere, IR projektora i specijalnog mikročipa koji generira mrežu koja služi za određivanje pozicije najbližeg objekta u prostoru. Također, sadrži i mikrofoni koji omogućuje akustičnu interakciju između korisnika i konzole. Prvotno raspoznavanje glasa bilo dostupno isključivo za par zemalja svijeta (Velika Britanija, Sjedinjene Američke Države, Japan i Kanada) nakon čega se ova usluga proširila na veliki dio Europe, Australiju i Meksiko. Također ovaj sustav mikrofona omogućuje igračoj konzoli da lokalizira akustični izvor i suzbijanje buke.

Kamera se nalazi na postolju koje sadrži zglob koji je motoriziran, a svrha toga je podešavanje kuta nagiba kamere po mjeri korisnika. To su usluge koje neće biti od pomoći i neće se koristiti u ovome radu, ali ih je definitivno zgodno spomenuti da se ukaže na to što se sve s ovim sustavom može raditi. Značajke u kojima se Kinect također iskazao su prepoznavanje lica i gestikulacija korisnika. Nakon dobrih rezultata u Xbox svijetu, ova kamera se počela primjenjivati i za druge svrhe, a jedna od tih će biti dobivanje 3D oblaka točaka što neće biti problem za ovaj sustav.



**Slika 4. Xbox 360 Kinect kamera [5]**

Dimenzije Kinect-a su dosta veće od Intel RealSense stereo kamere pa će snimanje prostora biti malo teže jer je prostor velik pa je potrebno nositi prijenosno računalo i kameru u isto vrijeme. Osim toga, zbog motora kojeg koristi Xbox360 Kinect za pozicioniranje, nije moguće direktno spojiti kameru na prijenosno računalo, nego se kamera spaja na priključak koji je spojen sa adapterom, a iz toga priključka izlazi konektor koji se spaja na prijenosno računalo. Manipulacija kamerom je teža u odnosu na prethodni primjer i zbog veće količine kablova. Maksimalna rezolucija koja se može postići je 1280x1024 piksela, a maksimalni broj slika po sekundi je 30.

### 2.3. Apple-ov LiDAR sustav

LiDAR (engl. Light detection and ranging) je optički sustav koji emitira svjetlosnu zraku te uslijed toga mjeri koliko je potrebno vremena toj zruci da se vrati nakon odbijanja od objekta. Konkretno, u ovom slučaju lidarski dio sustava Apple-ovog uređaja radi sa posebnim procesorom koji služi za obradu signala slika, Apple taj procesor naziva ISP (engl. *Image Signal Processor*). Upotreba ovog procesora omogućava točnije slike u području fotografija snimljenih u okruženjima sa slabim osvjetljenjem i općenito izuzimanje slika je brže. Domet LiDAR-a je po podacima proizvođača 5 metara što je sasvim dovoljno za onu upotrebu koju korisnici mobilnog uređaja mogu ostvariti. Pod tim smatram hobistički pristup skeniranjima prostorija, mjerenjima udaljenosti i sl. Apple-ov cilj implementiranja LiDAR-a u njihove sustave je zabava njihovih korisnika. Jedan od primjera je proširena stvarnost (engl. *Augmented Reality*) koji ju moguće ostvariti nakon 3D mapiranja prostora, do kojeg se u ovom slučaju dolazi LiDAR skeniranjem, pa se tako na internetu može pronaći sve više videa i aplikacija na kojima korisnici Apple u realnom okruženju mogu dodavati virtualne elemente po želji te tako doživjeti iskustvo proširene stvarnosti.



Slika 5. Primjer aplikacije sa primjenom proširene stvarnosti [7]

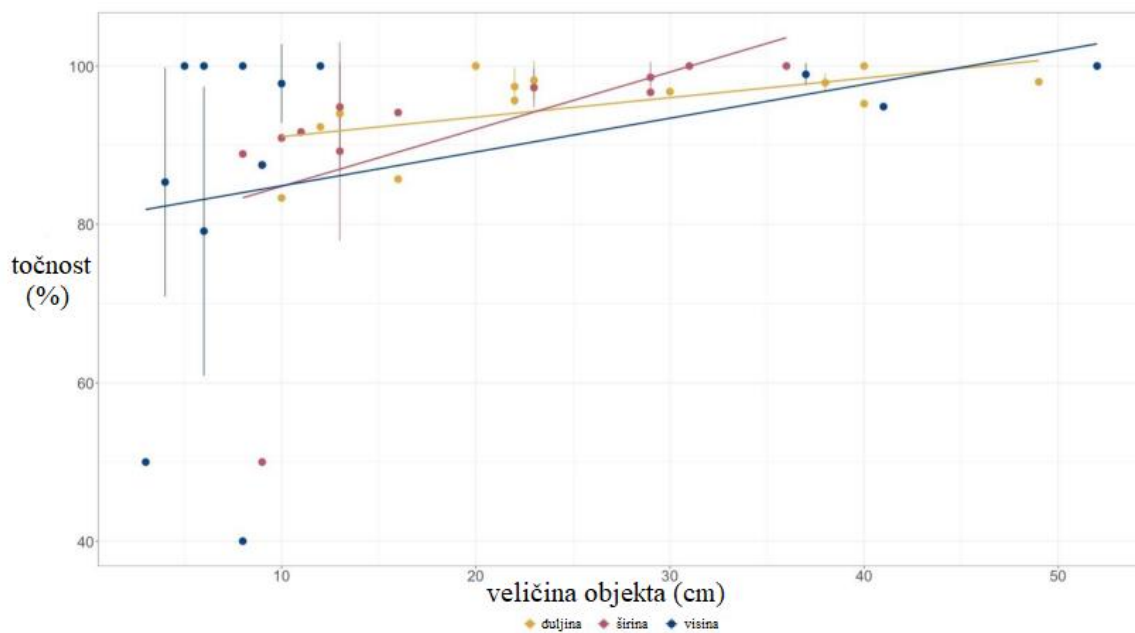
Ova aplikacija koristi LiDAR za skeniranje prostorije i svih prepreka u njoj. Na osnovu skenirane mape, korisnik može u prostoriju dodavati virtualne objekte na bilo koju lokaciju u prostoru pa tako može i sakriti virtualni objekt iza onog stvarnog. Očekivanja su da će se u nadolazećem vremenu pojaviti sve više ovakvih aplikacija koje su temeljene na proširenoj stvarnosti i mapiranju prostora LiDAR skeniranjem, a Apple definitivno prednjači u odnosu na ostale proizvođače te se njihov potez sa implementiranjem ovog sustava ne čini uzaludnim nego će pronaći svoju svrhu na tržištu.

2020. godine Apple je napravio ogromnu inovaciju na tržištu kada je u sustave kamera na svojim mobitelima i tabletima ugrađen LiDAR. LiDAR je sastavni dio modela tableta iPad Pro i mobilnih uređaja iPhone 12 Pro, iPhone 13 Pro i iPhone 14 Pro. Ovo su jedini modeli na tržištu koji imaju LiDAR u svom sustavu kamera. Od ostalih proizvođača mobilnih uređaja i tableta nitko još nije pokušao staviti ovu značajku u svoju ponudu.



**Slika 6. Sustav dvije kamere i lidara na tabletu iPad Pro [6]**

Tehnika korištenja LiDAR-a je izrazito točna, s toga se očekuju i takvi rezultati upotrebe iPad Pro-a u svrhu rješavanja zadatka ovoga rada. Ukoliko se žele dobiti izrazito točni rezultati koristiti će se profesionalna oprema, a to je u slučaju određenih geoloških ili sličnih istraživanja. LiDAR se masovno koristi u geologiji i arheologiji za dobivanje 3D modela prirodnih ljepota poput stijena, vodopada, planina, građevina i sl. U te svrhe se koriste profesionalni LiDAR-i koji su izrazito skupi i mogu ostvariti visoku preciznost. Istraživanje o točnosti LiDAR-a iPhone-a 12 Pro je pokazalo da ona ovisi o veličini objekta koji se skenira. Rezultati su prikazani u grafu.



**Slika 7. Rezultati točnosti LiDAR skeniranja mobilnim uređajem iPhone 12 Pro [8]**

Skeniranje je obavljeno na objektima poznatih dimenzija te se na osnovu dimenzija dobivenih skeniranjem odredila točnost u postocima. Isti objekti su skenirani pet puta. Može se zaključiti da je u najlošijem slučaju točnost dobivenih dimenzija iznad 80% što je prihvatljivo za zadatak ovog rada. Također, s porastom dimenzija objekata iz snimanog prostora, točnost raste gdje je čak na određenim dimenzijama točnost iznosila 100% što znači da će dimenzija u 3D modelu odgovarati toj istoj dimenziji u stvarnom svijetu što zvuči impresivno. U tri točke je došlo do točnosti niže od 60% pri malim dimenzijama objekta, što bi svakako moglo pokvariti izgled modela, ali to je mali broj devijacija i sigurno se te iste mogu ispraviti u određenim programima za grafičko uređivanje poput npr. Blender-a.

### 3. SNIMANJE PROSTORA

Nakon odabira kamera za akviziciju prostora Laboratorija za projektiranje izradbenih i montažnih sustava može se preći na odabir programa u kojem će snimanje biti obavljeno. RTAB-Map je program u kojem će se obaviti snimanje i čiji će algoritam iz snimki kreirati oblak točaka. Kreiranje oblaka točaka temelji se na SLAM principu koji će biti detaljno opisan u nastavku rada. Prije snimanja potrebno je kalibrirati kamere da bi se smanjila vjerojatnost raspršivanja oblaka točaka ili nepoklapanja prostora nakon obrade slike.

#### 3.1. Kalibriranje kamera

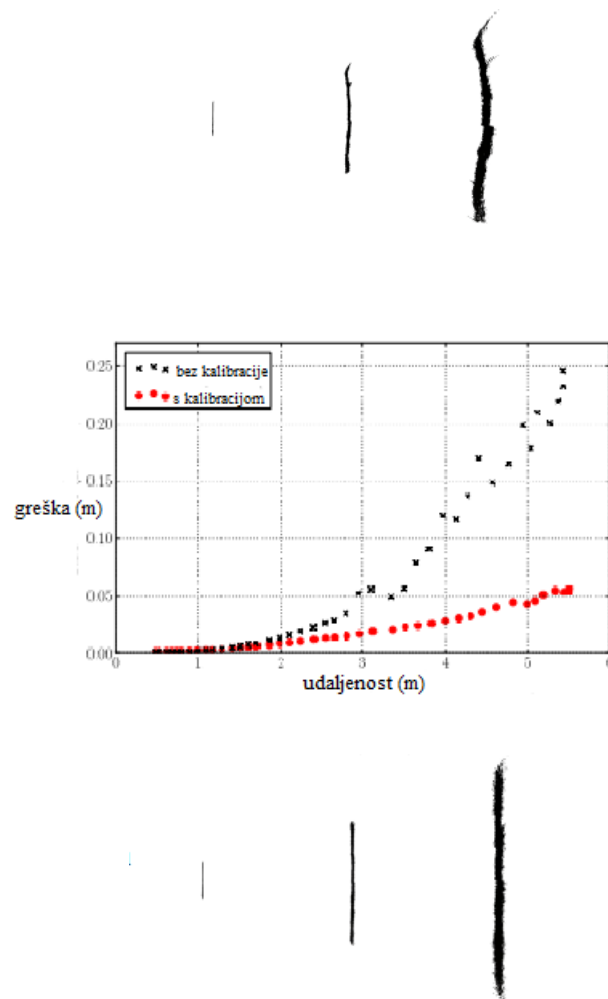
Za razliku od klasičnih kalibracija kamera koje se obavljaju slikanjem šahovske ploče ili specijalnih kalibracijskih ploča iz različitih kutova te se na osnovu njih računaju kalibracijski faktori, u ovome radu koristiti će se potpuno drukčiji pristup. Kalibracija kamere odnosno kalibracija dubine (engl. *Depth calibration*) kako je to definirano u programu RTAB-Map temelji se na CLAMS (engl. Calibration, Localizing and Mapping, Simultaneously) pristupu. Ovaj pristup je zaista jednostavan jer nisu potrebna nikakva kalibracijska pomagala poput prethodno navedenih ploča, niti su potrebna ikakva mjerenja. Uvjet koji je potreban da je kamerom i programom koji se koristi moguće koristiti SLAM pristup što je u ovom slučaju moguće. Ovom metodom se kalibriraju intrinzični parametri senzora dubine kamere. Metoda kalibracije intrinzičnih parametara se temelji na teoriji „kratkovidnosti“ dubinskih intrinzičnih parametara, tj. da njihova distorzija linearno raste sa dometom kamere, odnosno udaljenošću kamere od onoga što snima.

Cilj kalibracije je pronaći parametre distorzije za sve domete koje kamera može postići na toj sceni. Preporuka je da se snimanje obavlja u onom okruženju u kojem će se naknadno koristiti. Snimanje se obavlja tako da se na određenoj udaljenosti od npr. zida ili bilo koje površine unutar prostorije započne snimanje te da se laganom kretnjom i kružnim pomicanjem kamere snimatelj približava zidu. Kružnim pomicanjem kamere, cilj je postići da se scena snimi iz različitih pozicija jer će tako algoritam dobiti više točaka za računanje, ali će konačni rezultat biti točniji, a kalibracija preciznija. Nakon što se snimatelj približi zidu na otprilike pola metra, potrebno je vratiti se sličnom putanjom nazad na početnu točku uz također kružno pomicanje kamere. Preporučeno je da se ovaj postupak ponovi četiri ili pet puta što traje otprilike četiri i

pol do pet minuta snimanja. Ono što algoritam odradi je to što zanemari sve snimke na većoj udaljenosti od dva metra. Na slikama koje nisu zanemarene, traže se zajedničke točke iz kojih se kreira oblak točaka.

U obzir se uzimaju navedene snimke zato što se po teoriji „kratkovidnosti“ svi podatci o dubini unutar dva metra smatraju točnim, odnosno greška im je toliko mala da je prihvatljiva. Na osnovu kreiranog oblaka točaka u usporedbu se uzimaju slike koje su bile prethodno zanemarene te se na osnovu točaka detektiranih na tim slikama računaju multiplikatori dubine za svaku udaljenost veću od dva metra.

Multiplikator dubine je korekcijski faktor s kojim će se pomnožiti svaka detektirana dubina na tom mjestu na slici na dometu na kojem je multiplikator dubine izračunat.

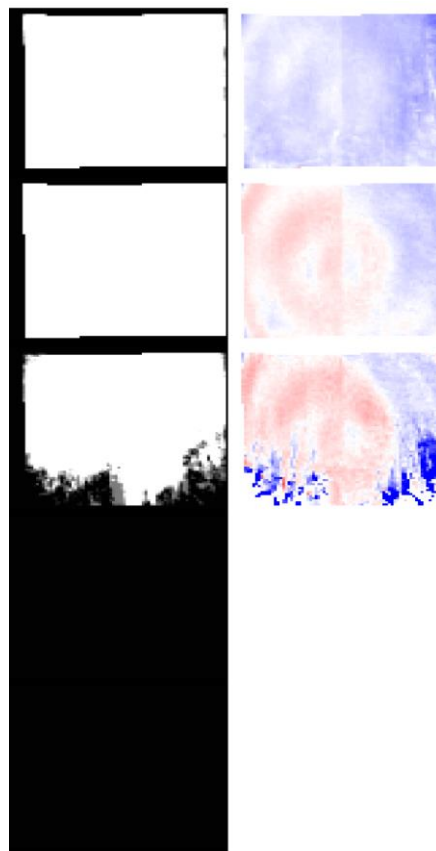


**Slika 8. Rezultati kalibracije dubine na primjeru zida [9]**

Iz grafa se može shvatiti o čemu teorija „kratkovidnosti“ govori. Vidi se da je do udaljenosti od dva metra greška mala i zanemariva i da se stvarno detektirane udaljenosti točaka u tom dometu mogu definirati kao točne vrijednosti. Ovaj graf potvrđuje ono na čemu se temelji CLAMS metoda kalibracije. Iznad grafa je projekcija oblaka točaka zida u jednoj ravnini koji je generiran izuzimanjem slika nekalibriranom stereo kamerom, a ispod projekcija oblaka točaka nastalih iz slika sa kalibriranom stereo kamerom. Udaljenosti oblaka točaka odgovaraju poziciji na kojoj se nalaze u odnosu na os apscisa grafa.

Nakon završene kalibracije, vrijednosti se spremaju u datoteku sa ekstenzijom .bin. Prilikom generiranja oblaka točaka, datoteka se prilaže i onda program iz nje čita kalibracijske faktore s kojima množi detektiranu dubinu točaka. Također, uz kalibracijsku datoteku, spremljene su i fotografije dubinske distorzije gdje se može vidjeti kolike su izračunati multiplikatori dubina na određenim dometima.

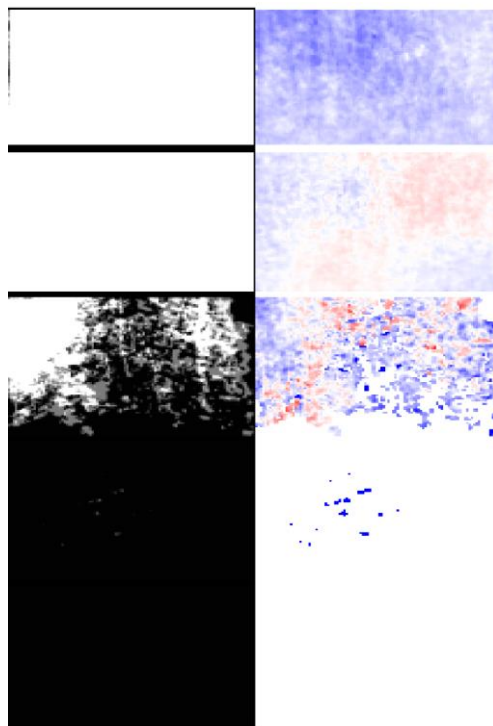
Snimanja sa obje prethodno navedene kamere su obavljena u Laboratoriju za za projektiranje izradbenih i montažnih sustava jer je to okruženje u kojem se kamere planiraju koristiti. Rezultati će biti prikazani na sljedećim slikama.



**Slika 9. Distorzijska slika Xbox 360 Kinect kamere**



Dio slike koji je u boji predstavlja izračunate multiplikatore dubina po svim pikselima. Gornja slika je na doometu od dva metra, sljedeća ispod je od četiri i takvim nizom do kraja. Ova kamera nema domet veći od 6 metara s toga se tu nalaze samo tri distorzijske slike. Bijelo područje distorzijske slike u boji je ono u kojoj je multiplikator jednak 1, a to znači da u tim dijelovima detektirana vrijednost dubine odgovara onoj stvarnoj. Na mjestima gdje je boja žarko crvena, a to je samo na jednom mjestu, multiplikator je 1.1, a na mjestima gdje je boja žarko plava, iznos multiplikatora dubine je 0.9. Nijanse između si možemo predočiti estimacijom između navedenih vrijednosti.



**Slika 10. Distorzijska slika Intel RealSense D435 stereo kamere**

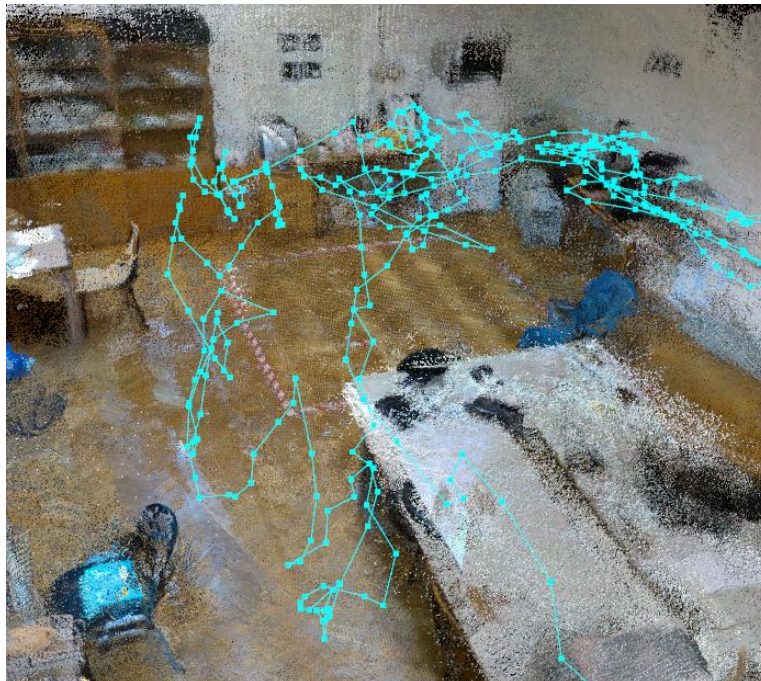
Kod ovog modela se pojavljuju multiplikatori dubine i na dometu od osam metara zato što ova stereo kamera ima veći domet od one prethodne. Kalibracijom navedenih kamera smanjena je vjerojatnost pojavljivanja raspršenih točaka u prostoru. Ukoliko dođe do te pojave, površine koje se generiraju iz oblaka točaka neće biti ravne nego će biti grbave i neće vjerno prikazivati prostor koji bi trebale. Ovo je jedan od bitnijih postupaka, rekao bih čak i postupak koji se ne smije izbjeći u procesu dobivanja 3D modela prostora.

### 3.2. SLAM pristup

RTAB-Map (engl. *Real-Time Appearance-Based Mapping*) je program u kojem će se obavljati snimanja sa kamerama Intel RealSense D435 i Xbox360 Kinect. Ovaj program se temelji na takozvanom SLAM (engl. *Simultaneous localization and mapping*) pristupu koji će biti detaljno objašnjen u ovom potpoglavlju. Snimanja se mogu obavljati sa kamerom koja se drži u ruci, a gibanje koje se ostvaruje sa istom je slobodno u prostoru tj. kamera ima šest stupnjeva slobode gibanja.

Drugi način je korištenjem mobilnog robota koji je opremljen laserskim mjeracom udaljenosti te se na taj način ostvaruje mapiranje sa tri stupnja slobode gibanja. U ovome radu snimanja će se izvršiti sa slobodnim gibanjem kamere u prostoru navođene rukom snimatelja. SLAM princip se temelji na kreiranju i ažuriranju mape kretanja kamere unutar radnog okruženja u nepoznatom prostoru na osnovu prethodno posjećene lokacije.

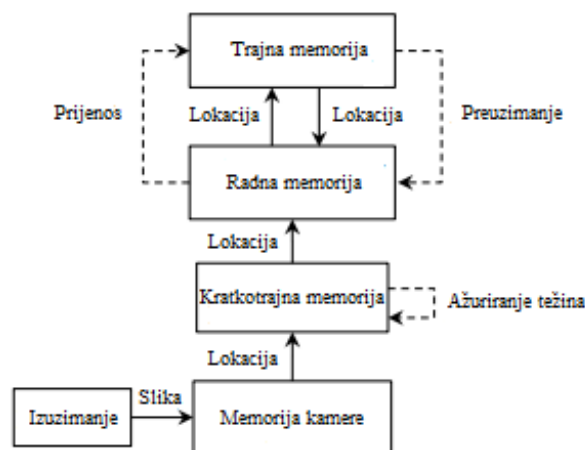
U većim prostorijama dolazi do kreiranja velike mape te se pri tom računanje novo zapaženih povećava s vremenom. Ukoliko dođe do situacije da je računanje lokacije dulje od vremena akvizicije, dolazi do kašnjenja te se ažuriranje mape kretanja više ne izvršava u realnom vremenu što kao rezultat stvara poteškoće pri preklapanju oblaka točaka, tj. izračunata lokacija kamere nije ta s koje je slika izuzeta.



**Slika 11. Mapa kretanja kamere u prostoru**

Na prethodnoj slici je svijetlo plavom bojom estimirana putanja kamere u snimanom prostoru. Plavim točkama su označene izračunate lokacije kamere. Točke su povezane linijama radi jednostavnijeg prikaza, jasno je da nije moguće ljudskom rukom ostvariti savršena pravocrtna gibanja kao na slici.

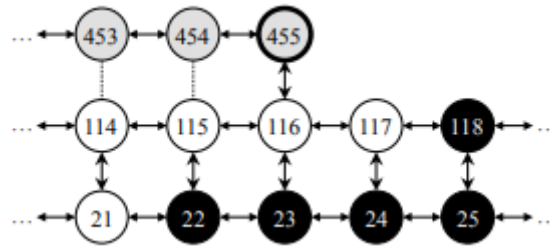
Da bi se izbjegla situacija sa slanjem krive lokacije kamere u prostoru, a samim time i pojave greške pri preklapanju oblaka točaka, razvijen je poseban način korištenja memorije. Način korištenja se temelji na tome da se najnovija i često posjećivane lokacije drže u radnoj memoriji (engl. *Working Memory*), a ostale lokacije su spremljene u „trajnu“ memoriju (engl. *Long-Term Memory*). Ova ideja je potekla iz psihologije koja tvrdi da ljudi pamte više mjesta na kojima provode većinu svog vremena u odnosu na ona koja ne posjećuju toliko često. Po tome se i u memoriji za računanje čuvaju ona mjesta, tj. lokacije na kojima je kamera bila više puta jer se tako povećava vjerojatnost da će se novopridošla lokacija poklopiti sa tim lokacijama. Takvim pristupom se i štedi procesorsko vrijeme jer ne mora raditi sa ogromnom količinom podataka nego sa samo određenim skupom. Nakon povezivanja novopridošle lokacije sa nekom od navedenih iz tzv. radne memorije, pridružene lokacije se premještaju u „trajnu“ memoriju te tako dodatno smanjuju skup s kojima se uspoređuje.



**Slika 12. Algoritam rada RTAB-Map-a [11]**

Nakon izuzimanja slike sa trenutne lokacije kamere, u memoriji kamere se izuzeta slika obrađuje te se radi smanjenja podataka na slici pronalaze korisne značajke te se nakon toga definira lokacija slike koja se šalje u kratkotrajnu memoriju.

U kratkotrajnoj memoriji se odvija proces ažuriranja težina na temelju kojih će se tražiti podudaranje između trenutne lokacije i lokacija kojima je kamera već prolazila kao što je prethodno objašnjeno. Težine se ažuriraju zato što se slike uzimaju jedna za drugom i relativno su slične, kao i njihova lokacija. Te dvije lokacije se spajaju u novu te se težina poveća i takva novokreirana lokacija se šalje u radnu memoriju. Proces upravljanja radnom i trajnom memorijom radi uštede procesorskog vremena je detaljno objašnjen.



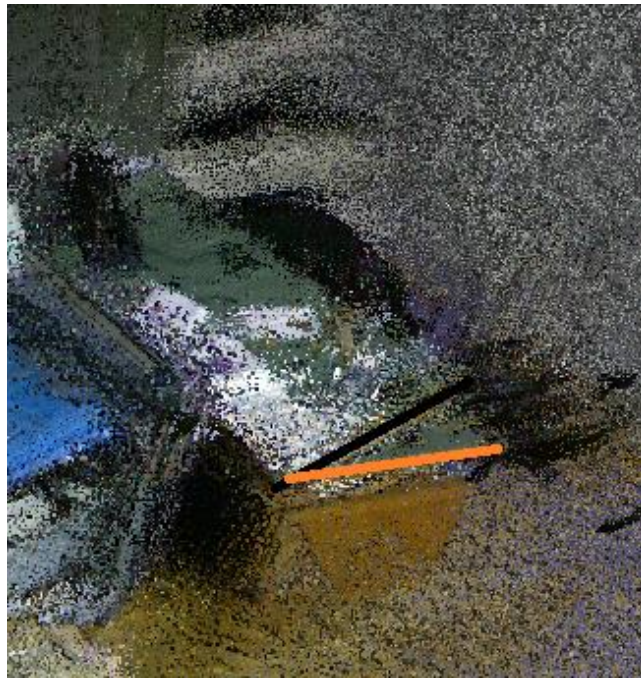
**Slika 13. Prikaz povezivanja lokacija SLAM pristupom [11]**

Na ovoj slici je grafički prikazan način na koji algoritam povezuje lokacije snimanja prostorije s kamerom. Krug predstavlja sliku i lokaciju na kojoj je slika izuzeta, a broj u krugu je broj izuzete slike i lokacije na kojoj je ta slika izuzeta od početka snimanja. Krugovi su smješteni u tri retka, a svaki redak predstavlja jedan prolaz unutar prostorije. Crni krugovi predstavljaju one lokacije koje su spremljene u trajnu memoriju, a bijeli krugovi u radnu memoriju. Sivi krugovi predstavlja one lokacije koje se nalaze u kratkotrajnoj memoriji što znači da su se za njih u trenutku u kojem je nastao prikaz ažurirale težine. Sivi krug sa brojem 455 je trenutna lokacija na kojoj je izuzeta slika. Krugovi unutar retka su međusobno povezani horizontalnim strelicama što ih definira susjednim lokacijama. Između redova se nalaze i vertikalne strelice koje također povezuju spremljene lokacije. Na tim mjestima su spojene značajke sa slike iz prvog prolaza i iz drugoga te je tako detektirano zatvaranje petlje (engl. *loop closure*), npr. krug sa brojem 114 i 21. Na mjestima na kojima je vertikalna linija točkasta, nije ostvareno zatvaranje petlje, tj. značajke na slikama se ne poklapaju i lokacija na slici 453 ne odgovara onoj na 114.

Ovakav pristup omogućava mapiranje prostora u realnom vremenu što je i cilj ovoga rada. Prednosti mapiranja prostora u realnom vremenu su te što se tokom snimanja može pratiti 3D mapa koja se generira ovim pristupom. Ukoliko se vidi da određeni dio prostorije u oblaku točaka nije dobro kreiran ili ga uopće nema, može se sa kamerom dodatno proći po tom dijelu

te tako poboljšati kvalitetu oblaka točaka koji će se generirati na toj lokaciji. To su benefiti koje ovaj pristup omogućava sa svojim algoritmom i upravljanjem memorijom.

Treba napomenuti da se s više prolaza može postići veća točnost i kvaliteta 3D mape, ali ne mora nužno biti. U slučaju da ne dolazi do pronalaska zajedničkih značajki između snimki, a samim time i zatvaranjima petlje, radna memorija se puni sa sve više lokacija te se računanje usporava i može doći do situacije koja je ranije spomenuta u ovome potpoglavlju, a to je da računanje lokacije bude sporije od akvizicije i samim time mapa neće vjerno prikazivati kretnju kamere, što za konačni rezultat daje oblak točaka u kojem neće biti podudaranja između određenih cjelina prostorije. Kao primjer se može navesti dupliciranje objekata ili bridovi objekta nisu ravni, a trebali bi biti i sl.



**Slika 14. Dupliciranje objekta kao rezultat nezatvaranja petlje**

Rezultat lošeg snimanja se upravo može vidjeti na ovoj slici. Usljed jednog od probnih nije došlo do zatvaranja petlje tijekom više prolaza oko uredskog stola iz laboratorija. Iz slike se vidi da je došlo do velike disperzije oblaka točaka, teško je uopće razaznati da se radi o uredskom stolu. Do toga je došlo jer se nije dogodila detekcija zajedničkih značajki i nakon toga zatvaranje petlje te u tom slučaju umjesto zajedničkih točaka u završnom globalnom oblaku točaka, na ovoj lokaciji u prostoru se sadrže sve točke iz svih prolaza.



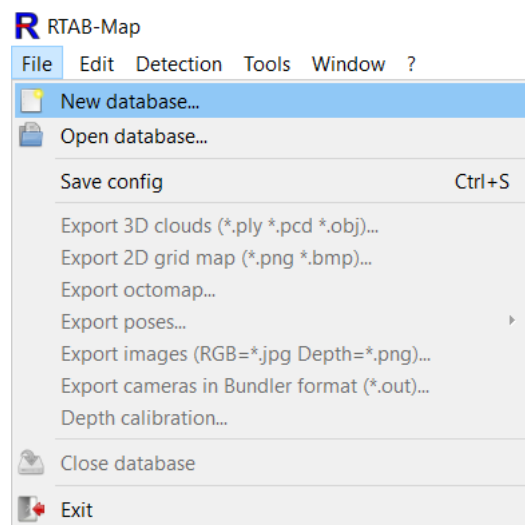
Što predstavlja jako veliki broj točaka što se može i zaključiti iz slike. Druga stvar koja se pojavljuje, a dokazuje nezatvaranje petlje je ta što se stol duplicirao. Na slici je naznačen crnom bojom stvarni brid stola koji postoji u oblaku točaka, a narančastom bojom je označen brid koji se može razaznati iz oblaka točaka, a nastao je u drugom prolazu snimanja i zarotiran je za određeni kut u odnosu na stvarni. Također je generirana druga ploha stola koja se nalazi malo niže od stvarne plohe stola koja se vidi na slici.

Dakle, u drugom prolazu nije bilo zatvaranja petlje te se izračunata lokacija kamere zarotirala i spustila u odnosu na stvarnu lokaciju što je dovelo do dupliciranja objekta i lošeg oblaka točaka. Iz ovakvog oblaka točaka se ne može generirati prihvatljiv 3D model koji će donekle vjerno prikazivati prostor Laboratorija za projektiranje izradbenih i montažnih sustava. Snimanje većih prostora traje dugo, a ukoliko se ne pazi na ove stvari, cijeli postupak mapiranja prostora se mora ponoviti. Ukoliko snimak ne uspije i dođe do pojave raspršenosti ili dupliciranja objekata, optimizacijski algoritam to neće moći riješiti, a brisanjem objekata ili dorađivanjem u editoru poput Blender-a će se izgubiti više vremena nego što ponovljeno snimanje traje.

Najbolji način za snimanje je da se snima u krug kroz prostoriju i da se ne hoda velikom brzinom jer je cilj ostvariti što više zatvaranja petlji. U programskom sučelju RTAB-Map-a se može pratiti tokom snimanja kada je došlo do zatvaranja petlje i između kojih lokacija se to ostvarilo.

### 3.3. Evaluacija snimki sa Intel RealSense D435 stereo kamerom

Postupak snimanja kreće tako da se otvori program RTAB-Map i u padajućem izborniku *File* odabere se opcija *New database*. To je datoteka sa *.db* ekstenzijom, a sadrži sve snimke i kreirane mape te oblake točaka koji su generirani analizom snimki. Nakon snimanja prostora je najbolje spremiti ovu bazu podataka jer i nakon što se zatvori program, može se naknadno u toj bazi podataka nekada kasnije raditi na oblaku točaka ili se može obaviti dodatni prolaz koji bi mogao detaljizirati izgled ukupnog oblaka točaka ili kreirane mape.



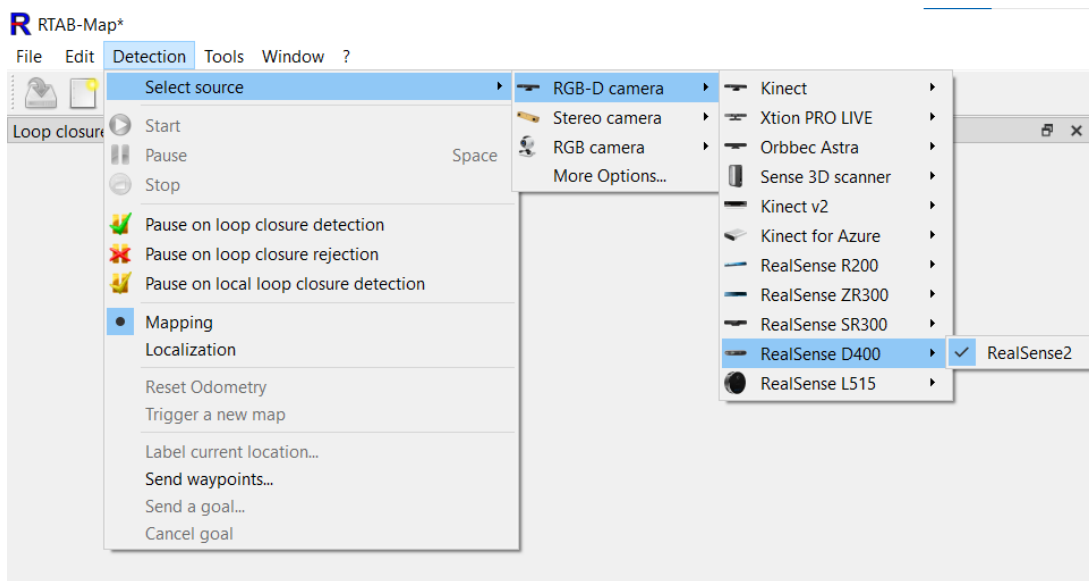
**Slika 15. Kreiranje nove baze podataka u programskom sučelju RTAB-Map**

Nakon kreiranja nove baze podataka, potrebno je odabrati izvor, odnosno kameru kojom će se snimati prostor i na osnovi čijih slika u vremenu će se generirati oblak točaka što se izvršava na prethodno opisani način.

Postupak odabira kamere ide tako da se na alatnoj traci odabere *Detection*, iz padajućeg izbornika se odabere *Select source* nakon čega se otvara još jedan izbornik u kojem se odabire *RGB-D camera* jer je to tip kamere koja se koristi u nadolazećem snimanju. U izborniku koji se nakon toga prikaže odabire se *RealSense D400* i na kraju se treba uvjeriti da se nalazi kvačica pored izbora *RealSense2*. Na toj lokaciji nema drugog izbora, ali ukoliko se ne nalazi kvačica kao na slici 16. potrebno je pritisnuti taj izvor. Možda je malo zbunjujuće što se odabire tip kamere RealSense D400, a ne točno naziv modela kamere koji se koristi, a to je RealSense D435. Međutim, RealSense D400 je cijela serija proizvoda u koju spada jako puno modela kamera i zbog toga je povezivanje na njih moguće na isti način. Osim toga, najbitnija stvar je

imati instaliran upravljački program (engl. *driver*) za Intel RealSense kameru. Upravljački program je besplatan i dostupan je na internetu za svakog korisnika.

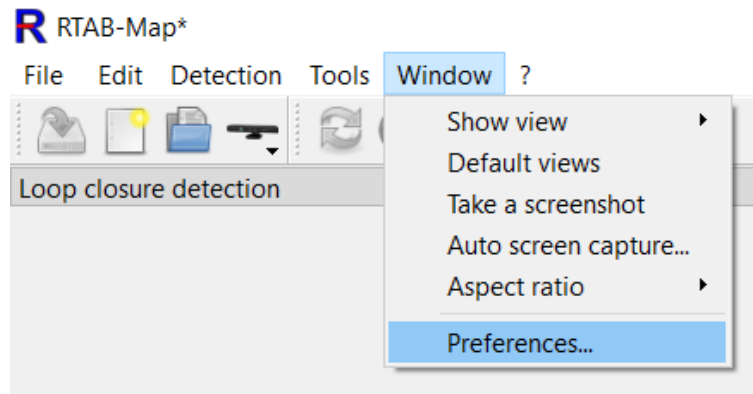
Ukoliko korisnik želi neke dodatne značajke, a Intel ih nudi, kao npr. dodatne opcije filtriranja i sl. mora ih nadoplatiti. Ukoliko nije instaliran upravljački program od proizvođača kamere koji odgovara modelu kamere koja će se koristiti, računalo ne može prepoznati koja kamera je spojena na USB priključku i uzaludno je onda odabiranje izvora u RTAB-Mapu, kamera se neće moći povezati s programom i postupak snimanja se neće pokrenuti.



**Slika 16. Odabir kamere kojom će se snimati u programu RTAB-Map**

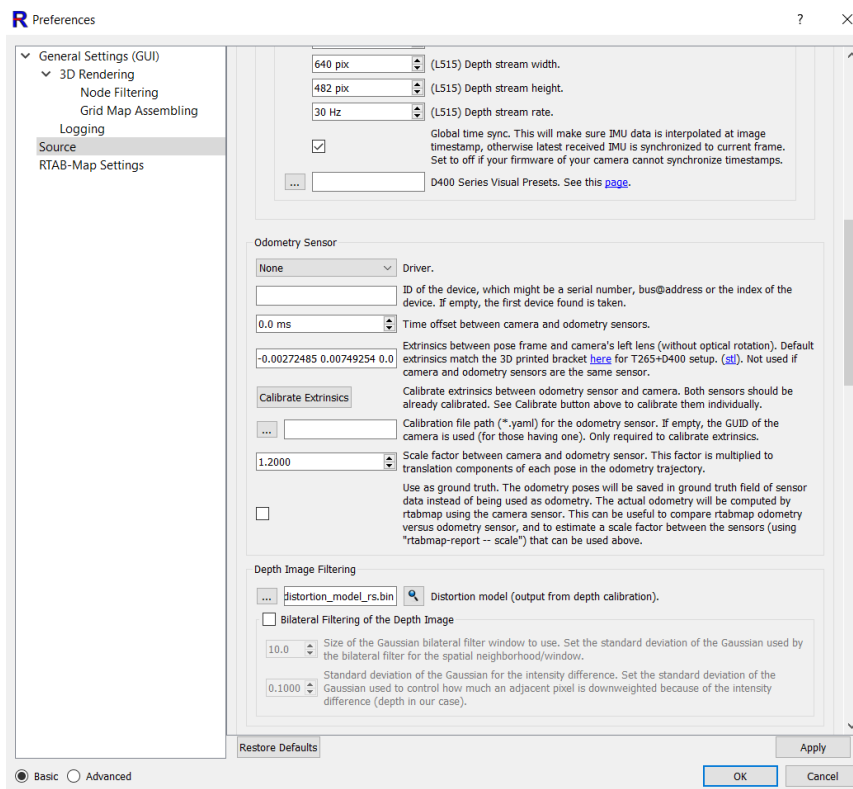
Prije početka snimanja je potrebno priložiti kalibracijsku datoteku koja je rezultat postupka kalibracije koji je opisan na početku ovoga poglavlja. Program će čitati korekcijske faktore iz kalibracijske datoteke s kojima mora množiti udaljenosti detektiranih točki što će dovesti do točnijeg oblaka točaka, a samim time i boljim i kvalitetnijim preklapanjem izuzetih slika. Potrebno je pritisnuti na tipku *Window* i zatim iz padajućeg izbornika odabrati *Preferences*.





Slika 17. Odabir opcije *Preferences*

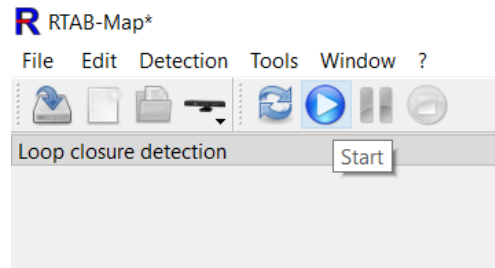
Nakon toga se pojavljuje novi prozor u kojem se u izborniku *Source* može priložiti kalibracijska datoteka. Na slici 18. se može vidjeti na kojem mjestu je priložena kalibracijska datoteka *distortion\_model\_rs.bin* koja je spremljena u memoriji računala.



Slika 18. Umetanje kalibracijske datoteke u preferencije programa

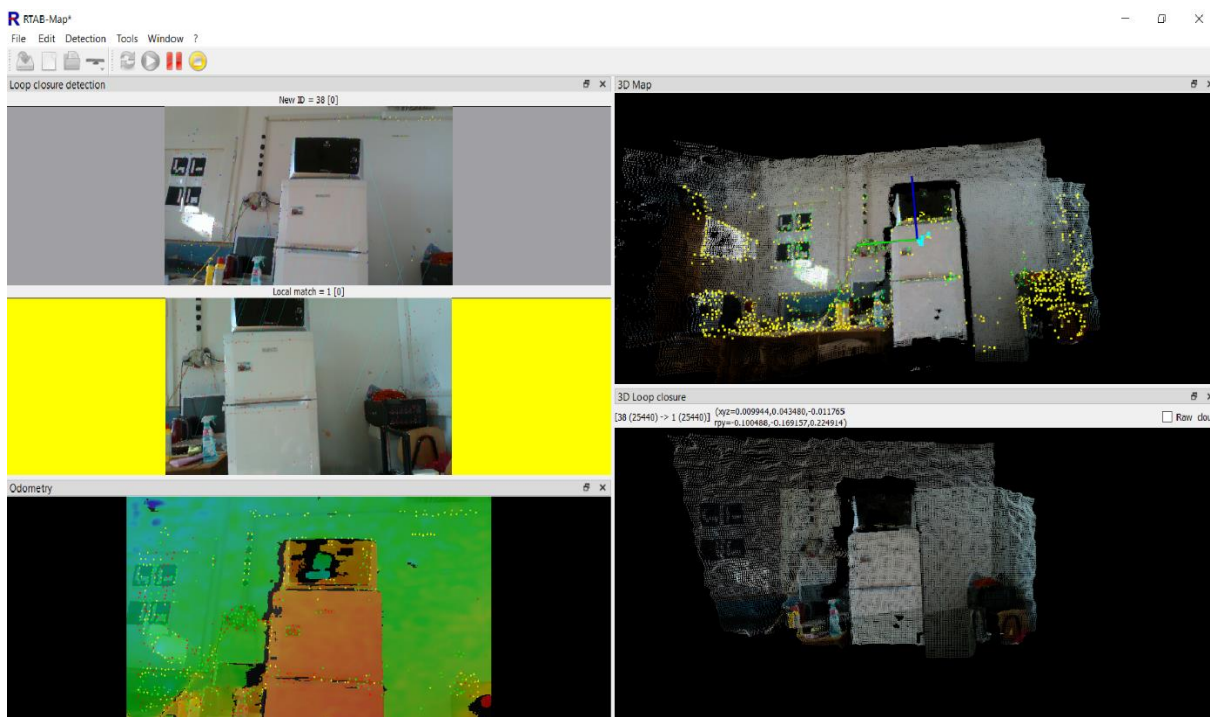
Ukoliko su ispunjeni svi prethodno navedeni koraci i instalirani svi potrebni paketi, snimanje u programskom sučelju RTAB-Map se može pokrenuti. Pokreće se tako da se na alatnoj traci pritisne plavi gumb *Start*. U slučaju da izvor nije odabran ili da konekcija sa kamerom nije uspostavljena, gumb bi bio sive boje i ne bi se mogao pritisnuti. Nakon pritiska na gumb,

nakon par sekundi se pokrene proces izuzimanja slika te se može pratiti na ekranu. Također se može pratiti i 3D mapa koja se kreira te 3D mapa koja nastaje uslijed zatvaranja petlje. To je mapa sastavljena od lokacija koje se nalaze u radnoj memoriji sustava, ona se mijenja tokom vremena ovisno o tome koje lokacije završe u trajnoj memoriji, a koje ostanu ili se stvore u radnoj memoriji. Ovaj princip je prikazan algoritmom na slici 12.



**Slika 19. Gumb za pokretanje snimanja**

Slika 20. prikazuje standardni izgled prozora RTAB-Map-a programskog sučelja tijekom snimanja prostorije Laboratorija za projektiranje izradbenih i montažnih sustava. Sastoji se od četiri manja prozora koji služe za praćenje snimanja, detekciju zajedničkih značajki na slikama, kreiranje 3D mape te zatvaranja petlje.



**Slika 20. Prozor programskog sučelja RTAB-Map-a tokom snimanja prostorije**

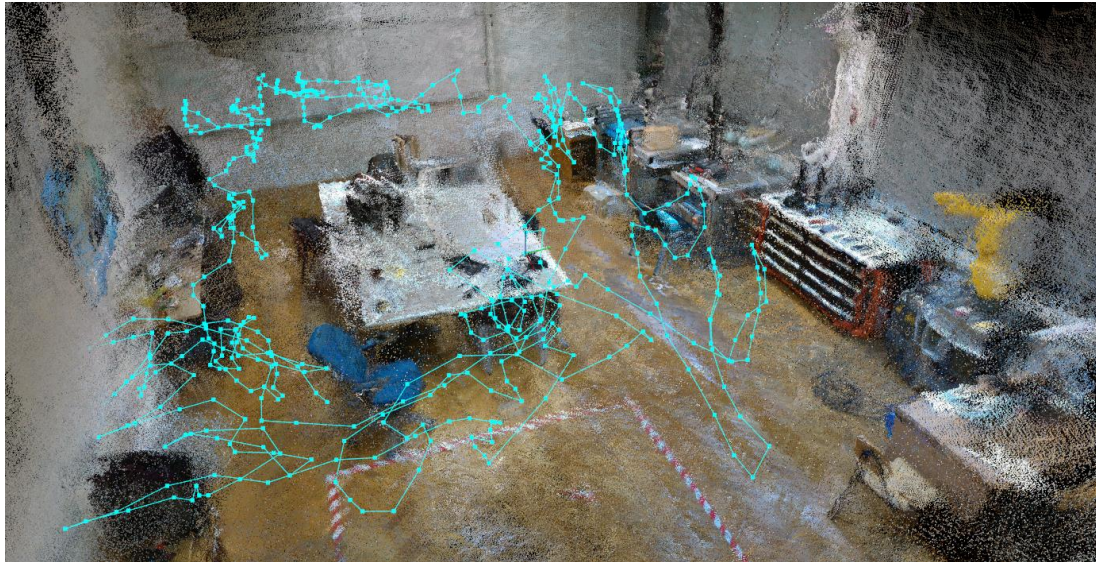
Snimka zaslona je izuzeta u trenutku kada je program pronašao lokalni spoj između slike koja je izuzeta kamerom u tom trenutku i prve slike koja je izuzeta od trenutka kada je snimanje krenulo što se vidi u prozoru sa žutom bojom koja korisniku privlači pažnju i informira ga da je došlo do detekcije zajedničkih značajki. Na slici na kojoj je nađen spoj, lokacije značajki koje su iste na slikama su spojene sa plavim linijama. Ispod toga prozora nalazi se prozor sa odometrijom koji u spektru boja pokazuje koliko su točke udaljene od kamere. Crvenom bojom su prikazane one točke koje se nalaze jako blizu kamere, a plavom one koje se nalaze jako daleko.

Desni dio glavnog prozora se sastoji od prozora sa 3D mapom oblaka točaka i to je prozor u kojem se može vidjeti kako će izgledati oblak točaka koji se može spremirati. Naravno, spremljeni oblak točaka ne mora nužno tako izgledati jer može doći do određenih filtriranja i čišćenja prilikom spremanja oblaka, a taj postupak će kasnije biti prikazan. Prozor ispod ovoga se nalazi 3D mapa zatvaranja petlje koja predstavlja trenutne točke koje se nalaze u radnoj memoriji programa. Te točke su okarakterizirane programom kao točke sa najvećom vjerovatnošću da će sa njima doći do detekcije. To je najčešće 3D mapa kreirana od nekoliko posljednjih spojeva značajki. U trenutku kada se pojedine lokacije prebace u trajnu memoriju, dio ove mape nestane, a u nekim slučajevima nestane i cijela mapa. To nije znak da se taj dio izgubio nego da je prešao u trajnu memoriju da ne bi dodatno opterećivao procesor računala.

Prolaskom kamerom kroz cijelu sobu, 3D kreirani oblak točaka bi trebao obuhvatiti cijelu sobu, a snimanje se može zaustaviti na tipku stop. Ta tipka se nalazi dva gumba desno od tipke *Start* koja se nalazi na prethodnoj slici. Ukoliko neki dijelovi prostorije nisu obuhvaćeni ili se na 3D mapi može zaključiti da nisu detaljno prikazani, moguće je napraviti još jedan prolaz. To se pokreće tako da se ponovno stisne tipka *Start* nakon čega je potrebno ostvariti zatvaranje petlje, tj. na slici izuzetoj u ovom prolazu, program mora pronaći zajedničke značajke sa slikom iz prethodnog prolaza. U trenutku kada se to dogodi, pozicija i orijentacija kamere je određena u prostoru i snimanje detalja se može započeti. Ovo je značajka koja je velika prednost ovog pristupa, međutim tokom isprobavanja iste nije polučila najbolje rezultate.

Na mjestima gdje se pokušao dobiti detaljniji oblak točaka koji će kasnije rezultirati jasnijim izgledom objekta u 3D modelu, došlo je do raspršenja točaka, nepoklapanja i čak do lakog gubljenja orijentacije kamere. U tom slučaju se snimanje zaustavlja, a u pravilu se oblak točaka totalno rasprši na tom mjestu da bude neupotrebljiv. To je iz razloga jer program dobiva kao ulaz krivu lokaciju kamere i onda zapravo točke s neke druge lokacije stavlja u mapu na

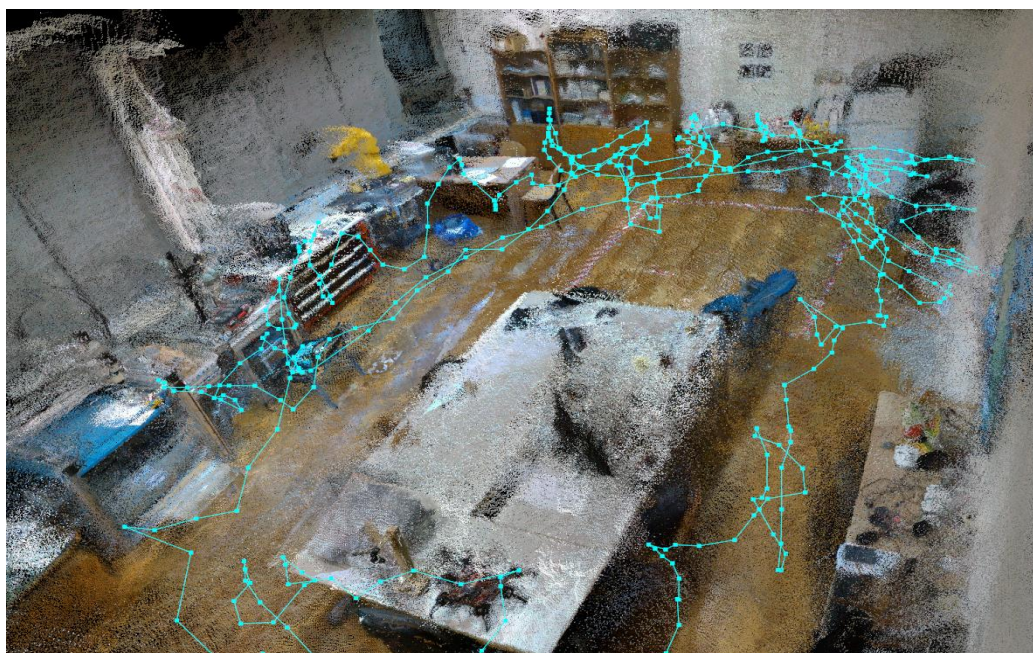
poziciju na koju ne bi trebao. Na slikama 21. i 22. prikazan je oblak točaka u dva pogleda koji je generiran iz snimke prostora sa Intel RealSense D435 kamerom. Generirani oblak točaka pokazuje nedostatke SLAM pristupa za primjenu snimanja laboratorija.



**Slika 21. Pogled 1- Oblak točaka proizašao iz snimki s kamerom Intel RealSense D435**

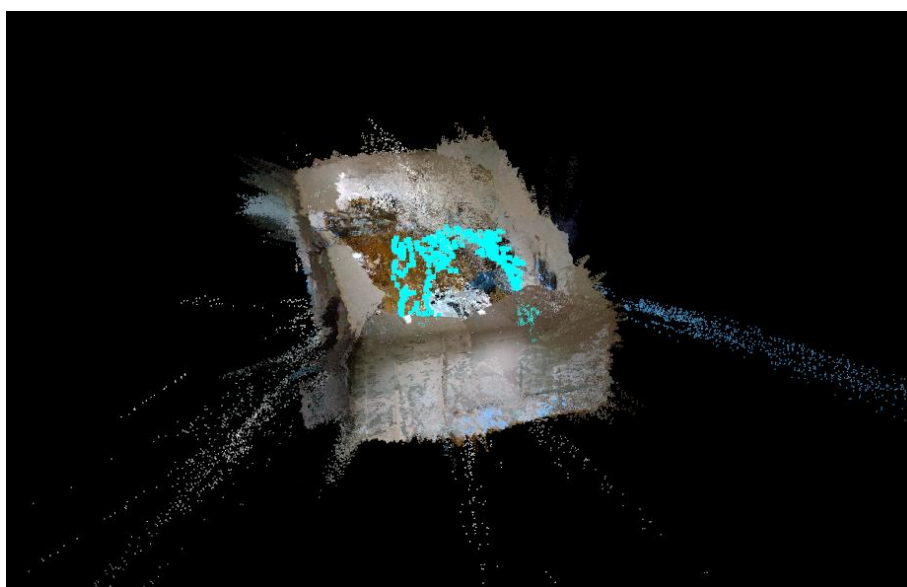
Na mjestima gdje se u snimanom prostoru nalazi veći broj objekata ili značajki, rezultati su izrazito dobri. Pod evaluacijom dobrog se smatra da je oblak točaka uniforman i da samim time vjerno prikazuje objekt koji je snimljen na toj lokaciji u prostoriji. Mjesta na kojima nema toliko značajki ili ih nema uopće su kritična mjesta za ovaj pristup. To su prije svega bijeli zidovi i zavjese koje su jednobojne bez ikakvih karakterističnih značajki na sebi. Također, snimanje takvih površina i objekata je zahtjevno zato što ukoliko se snimaju sa male udaljenosti, kamera se izgubi jer ne može naći zajedničke značajke iz prethodnih slika te tako ne može odrediti ni lokaciju na kojoj se nalazi u prostoru što stvara problem pri mapiranju prostora. Osim što je teško snimati, rezultat koji nastane iz tih snimki je raspršenost oblaka točaka. To se događa zato što se konstruira iz slika sa veće udaljenosti. Premda je dubina kalibrirana, opet na jednobožnim površinama bez značajki, teško je naći zajedničke točke koje će napraviti zatvaranje petlje na tome mjestu i iz tog razloga se dobiva raspršeni oblak točaka. Ovaj problem se reflektira dalje na kreiranje 3D modela jer ukoliko se iz takvog oblaka točaka pokuša konstruirati 3D model, zidovi i zavjese će biti blago rečeno grbave, a pokušajem zaglađivanja će se gubiti ostali detalji u prostoriji.





**Slika 22. Pogled 2- Oblak točaka proizašao iz snimki s kamerom Intel RealSense D435**

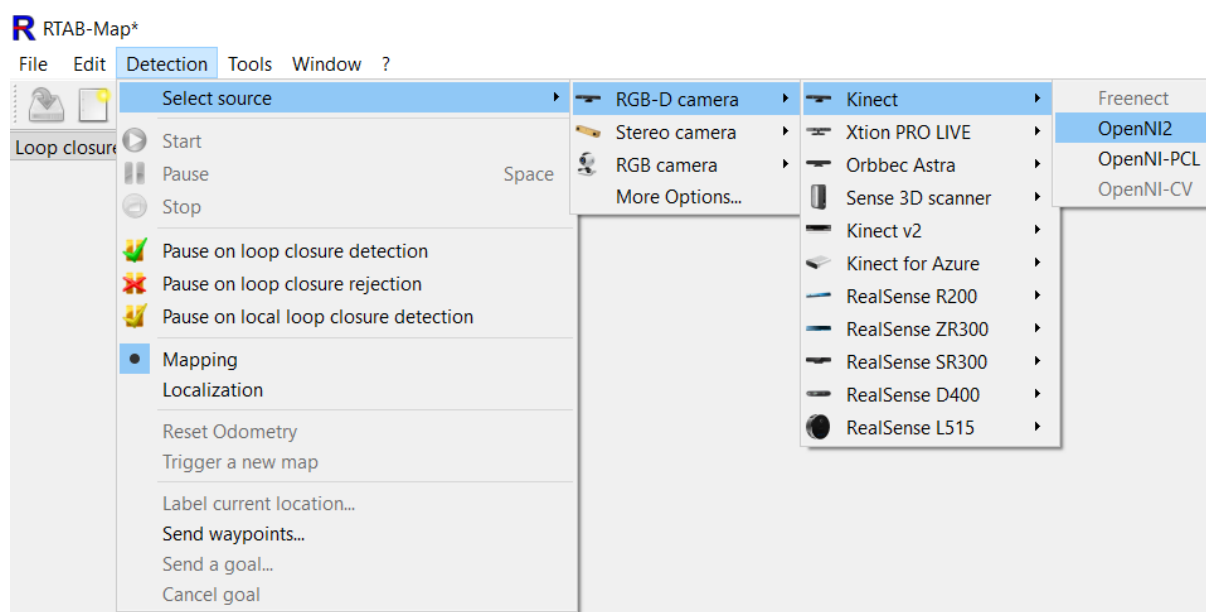
Iz udaljenog pogleda na oblak točaka prostorije Laboratorija za projektiranje izradbenih i montažnih sustava dodatno se vidi koliko je na nekim dijelovima došlo do raspršivanja oblaka točaka tj. gdje je lokacija kamere bila potpuno izgubljena. Po lokacijama se može zaključiti da je to kao što je prethodno spominjano na zidovima tj. površinama na kojima nema toliko značajki na kojima bi se mogle detektirati zajedničke točke. Raspršene točke izvan stvarnih dimenzija promatranog prostora se mogu obrisati u programu MeshLab tako da one neće moći utjecati na izgled 3D modela nakon pretvaranja oblaka točaka u model.



**Slika 23. Udaljeni pogled na kreirani oblak točaka laboratorija**

### 3.4. Evaluacija snimki sa Xbox360 Kinect kamerom

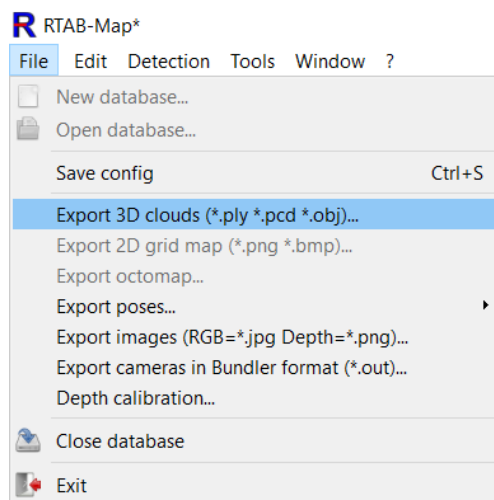
Snimanje ovom kamerom se obavlja na isti način kao i sa Intel RealSense D435 u programu RTAB-Map. Jedina razlika je u tome što se za izvor odabire druga kamera. Ostatak procedure je identičan s toga nije potrebno ponovno pokazivati istu stvar. U ovom potpoglavlju biti će prikazano kako se sprema oblak točaka iz programa RTAB-Map i kako se taj oblak točaka implementira u program MeshLab. MeshLab je besplatni program koji je dostupan svim korisnicima, a služi za obradu i uređivanje 3D trokutnih mreža. U ovom slučaju mreža se generira iz oblaka točaka koji se implementiraju u programu. U programu se nalazi pregršt alata za uređivanje, čišćenje, popravljanje i pretvaranje mreža. Također, sadrži i značajke kojima se modeli mogu pripremiti za 3D ispis. Za okvire ovog zadatka biti će potrebno generiranje modela iz 3D oblaka točaka te popravljanje istog.



Slika 24. Odabir kamere Kinect u programu RTAB-Map

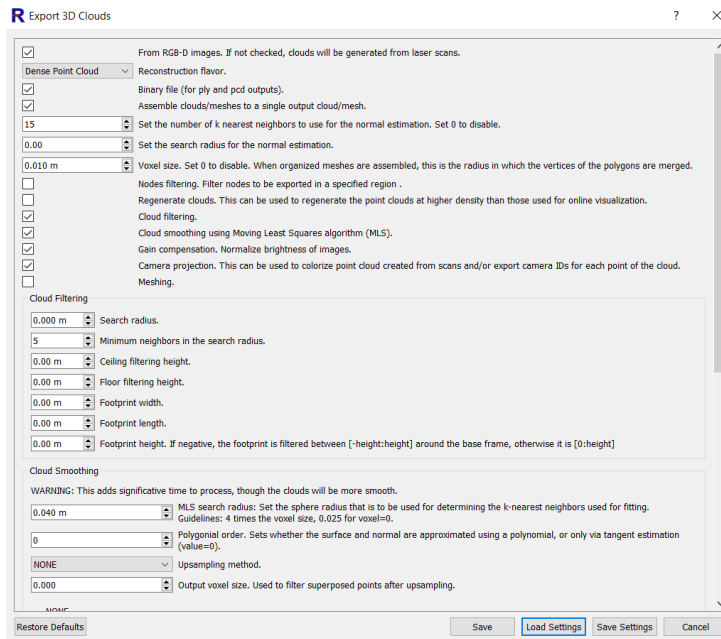
Po završetku postupka snimanja i nakon karakteriziranjem oblaka točaka kao prihvatljivim, oblak točaka se može spremiti u datoteku sa .ply (engl. *Polygon File Format*) ekstenzijom. Datoteka u tom formatu može ići na daljnju obradu u programe koji iz oblaka točaka generiraju 3D model. Jedan od takvih je MeshLab i on će se koristiti u ovome radu. Spremanje oblaka točaka u navedenom formatu ostvariti će se tako da se u programskom sučelju RTAB-Map-a

stisne *File* i u padajućem izborniku odabere *Export 3D clouds*. Odabirom ove opcije, otvara se novi prozor u kojem je potrebno odabrati značajke koje će se koristiti pri spremanju oblaka točaka. Odabrane značajke se mogu vidjeti na slici 26. Filtriranje i zaglađivanje oblaka točaka i kompenzacija pojačanja neke su od značajke koje su odabrane van standardnih postavki koje program nudi. Projekcija kamere je odabrana zbog dodavanja boje točkama što i piše u objašnjenju značajke. Također, moguće je da i RTAB-Map kreira 3D mrežu na osnovu oblaka točaka, a to se aktivira značajkom *Meshing*. Međutim, ukoliko generirana mreža bude loše kvalitete, opet je potreban neki program u kojem će se moći doraditi, a to je MeshLab.



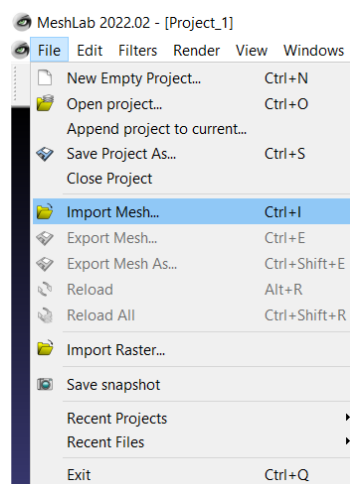
**Slika 25. Spremanje oblaka točaka**

Datoteka oblaka točaka se spremi na željenu lokaciju na osobnom računalu i nakon toga se otvara program MeshLab u kojem će se prethodno spremljena datoteka implementirati. U programskom sučelju MeshLab pritisne se na *File* i nakon toga u padajućem izborniku se odabere naredba *Import Mesh*. Dolazi do otvaranja novog prozora u kojem je potrebno naći datoteku sa ekstenzijom *.ply* koja odgovara oblaku točaka snimljenog sa Xbox360 Kinect kamerom. Nakon što se pronađe datoteka, ona se označi i pritisne se gumb *Open*. Možda je zbunjujuće što se u ovoj naredbi spominje mreža (engl. *Mesh*), ali to nije jedini oblik modela koji se može ubaciti nego se kao i u ovom slučaju mogu ubaciti datoteke sa ekstenzijom *.ply* ili npr. *.stl*, *.obj*, *.fbx* i mnogi drugi formati.



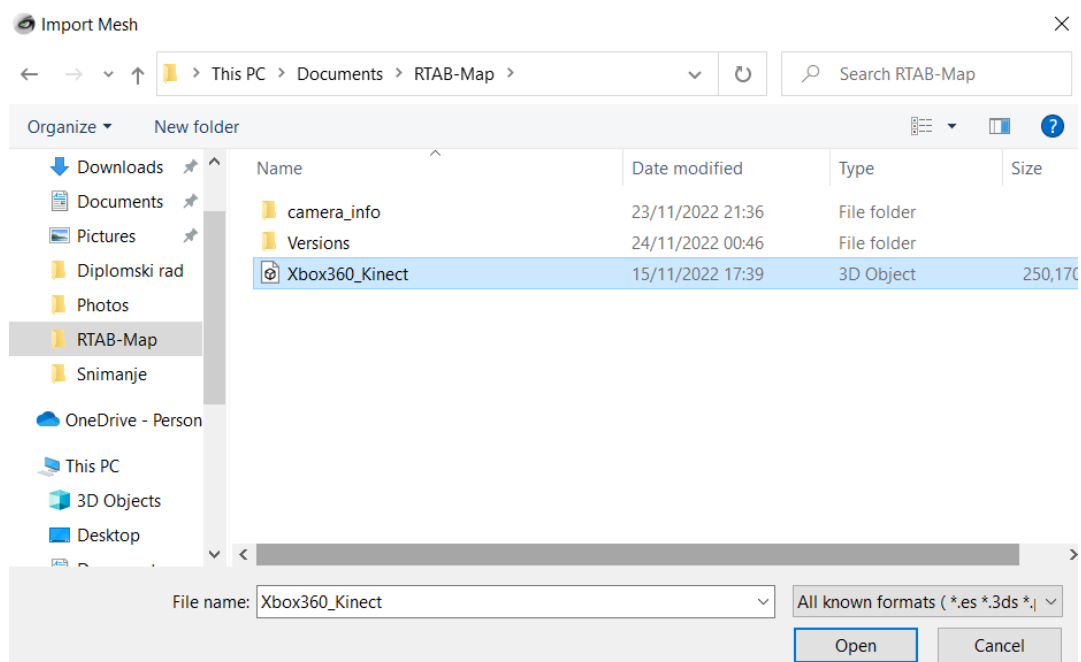
**Slika 26. Značajke odabrane prije spremanja oblaka točaka**

Datoteka oblaka točaka se spremi na željenu lokaciju na osobnom računalu i nakon toga se otvara program MeshLab u kojem će se prethodno spremljena datoteka implementirati. U programskom sučelju MeshLab pritisne se na *File* i nakon toga u padajućem izborniku se odabere naredba *Import Mesh*. Dolazi do otvaranja novog prozora u kojem je potrebno naći datoteku sa ekstenzijom *.ply* koja odgovara oblaku točaka snimljenog sa Xbox360 Kinect kamerom.



**Slika 27. Umetanje oblaka točaka u programu MeshLab**





**Slika 28. Otvaranje oblaka točaka**

Nakon što se pronađe datoteka, ona se označi i pritisne se gumb *Open*. Možda je zbunjujuće što se u ovoj naredbi spominje mreža (engl. *Mesh*), ali to nije jedini oblik modela koji se može ubaciti nego se kao i u ovom slučaju mogu ubaciti datoteke sa ekstenzijom *.ply* ili npr. *.stl*, *.obj*, *.fbx* i mnogi drugi formati.



**Slika 29. Pogled 1- Oblak točaka proizašao iz snimki sa kamerom Xbox360 Kinect**

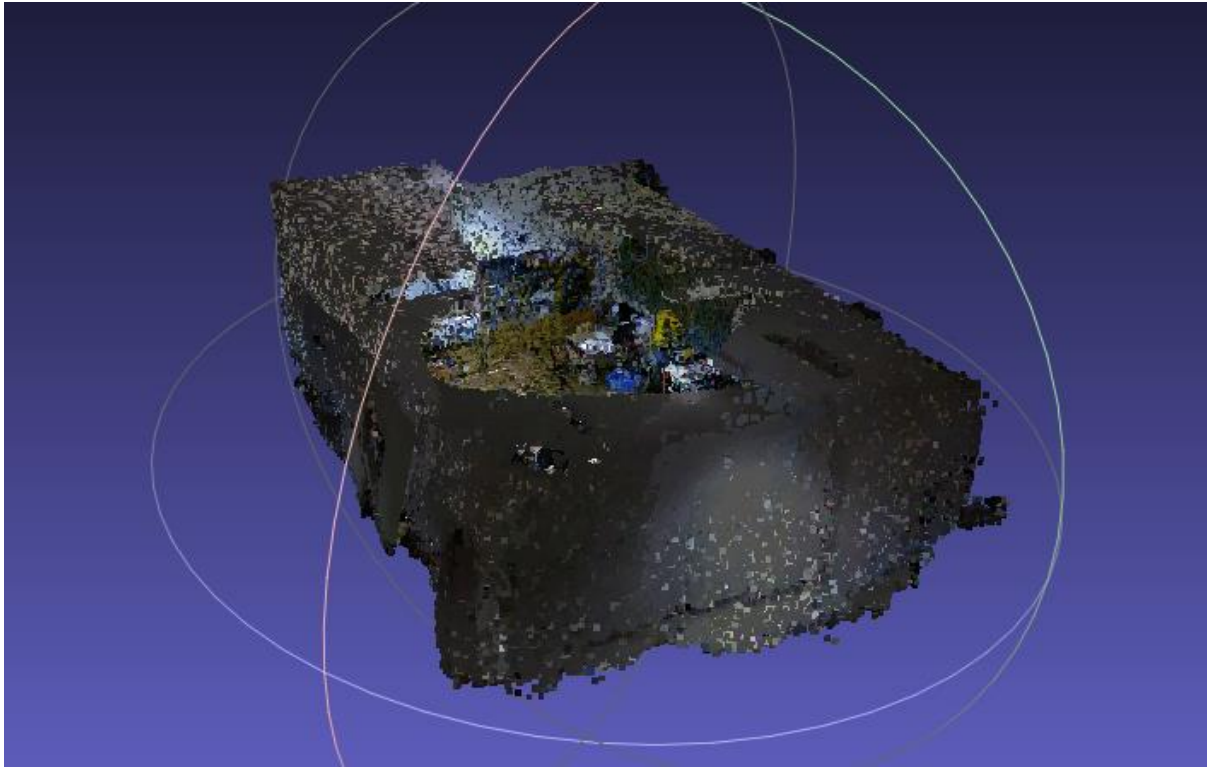
Učitani oblak točkaca izgleda manje raspršen od onog iz prethodnog potpoglavlja. Također, zavjese i zidovi se jasnije vide, a površine objekata su ravnije premda ima nekih mjesta koji su potpuno nejasni.



**Slika 30. Pogled 2- Oblak točkaca proizašao iz snimki sa kamerom Xbox360 Kinect**

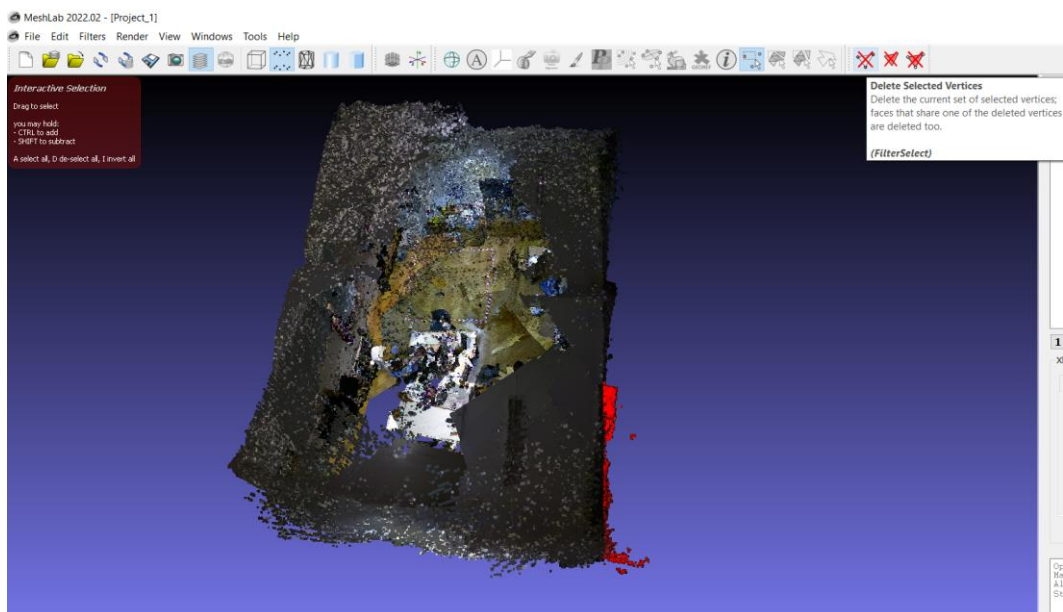
Na podu i u kutovima se nalaze rupe u oblaku točkaca zato što se na tim mjestima kamera gubila, odnosno program nije mogao odrediti lokaciju kamere iz izuzetih slika. Ukoliko bi se snimatelj približio tom praznom mjestu, kamera bi potpuno izgubila vezu sa mapiranjem prostora. Povratak u mapu bi se dogodio tek kada bi se ponovno dogodio neki lokalni spoj između trenutne slike i slike koja se nalazi u radnoj memoriji, a to je ponekad teško pronaći. Treba napomenuti da su snimke rađene više puta i da snimanja mogu biti zaista mukotrpana pogotovo u prethodno navedenom slučaju kada se želi ostvariti potpuno pokrivanje mape na nekim najudaljenijim mjestima na kojima nema specifičnih točki koje se mogu povezati sa prethodnima. U svakom slučaju je ovaj 3D oblak točkaca puno bolji od prethodnog, detaljniji je, ali se također na njemu vide nedostaci SLAM pristupa za mapiranje prostorija. Nedostaci se očituju u tim nepristupačnim mjestima poput kutova koji su većinom mračni što još dodatno otežava pronalazak značajki na slici.





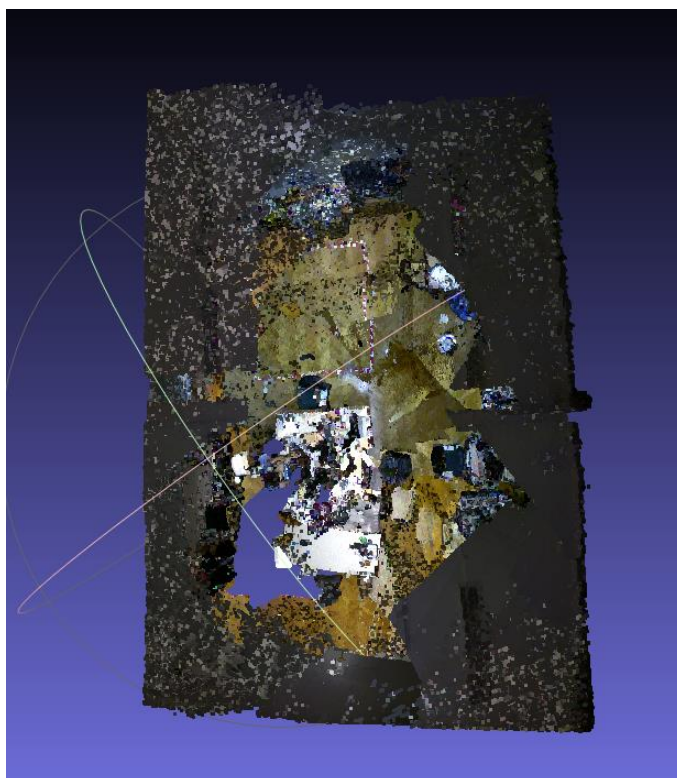
**Slika 31. Pogled na oblak točaka prostorije sa veće udaljenosti**

Iz ovog pogleda se vidi da iako model ima rupa u svojoj strukturi, gledajući izvana, raspršenih točaka nema kao što je to bio slučaj u prethodnom potpoglavlju. Rezultat tomu je bolje poklapanje značajki sa izuzetih slika i samim time i bolji izgled mape tj. 3D oblaka točaka. S obzirom da je izgled oblaka točaka prihvatljiv, u narednom dijelu potpoglavlja biti će prikazana metoda dobivanja 3D modela iz oblaka točaka. Dobiveni model je potrebno što više uljepšati i srediti da što vjernije prikazuje stvarni prostor. Prije svega će višak točaka biti obrisano primjenom metode *Delete Select Vertices*. Prije toga je potrebno označiti te točke, a to se obavlja pomoću naredbe *Select Vertices*. Klikom miša i rastezanjem pravokutnika obuhvaćaju se točke koje korisnik želi označiti. Brisanje viška točaka je korisna stvar koja se može napraviti nakon što se oblak točaka učita u program MeshLab. Ukoliko se to učini, računalo se manje opterećuje jer radi sa manje točaka, a još bitnija stavka je ta što prilikom generiranja 3D mreže, točke koje su izvan prostora mogu prouzročiti generiranje potpuno krivih oblika što će otežati kasnije proces popravljavanja i zaglađivanja površina. Brisanjem točaka koje ne predstavljaju ništa definitivno se taj proces vremenski smanjuje.



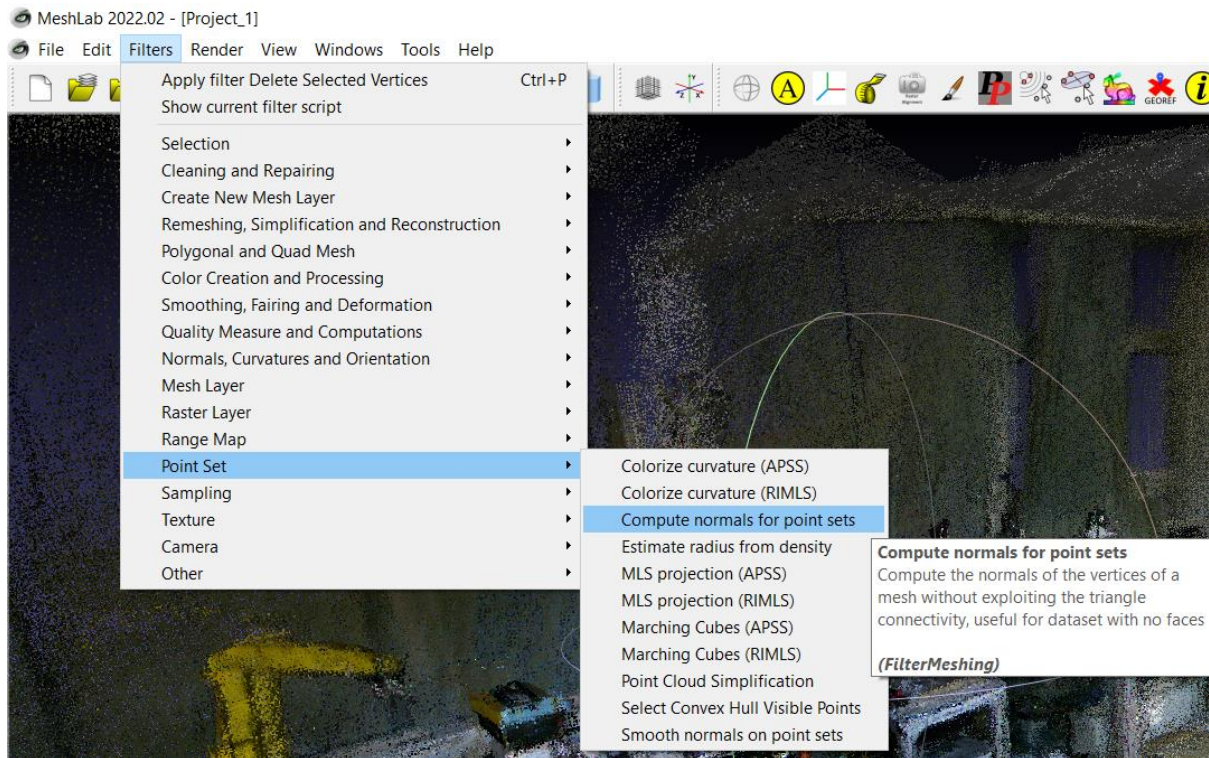
**Slika 32. Brisanje viška točaka u programu MeshLab**

Postupak je ponovljen za višak točaka na svim rubovima modela prostorije. Treba biti oprezan tokom brisanja točaka jer ukoliko se slučajno označe točke unutar modela, npr. točke koje opisuju konturu zida, na tim mjestima će biti rupa u modelu. Negativna strana ovog programa je što ne postoji naredba za poništavanje prethodnog koraka (engl. *Undo*).



**Slika 33. Konačni izgled 3D oblaka točaka nakon brisanja**

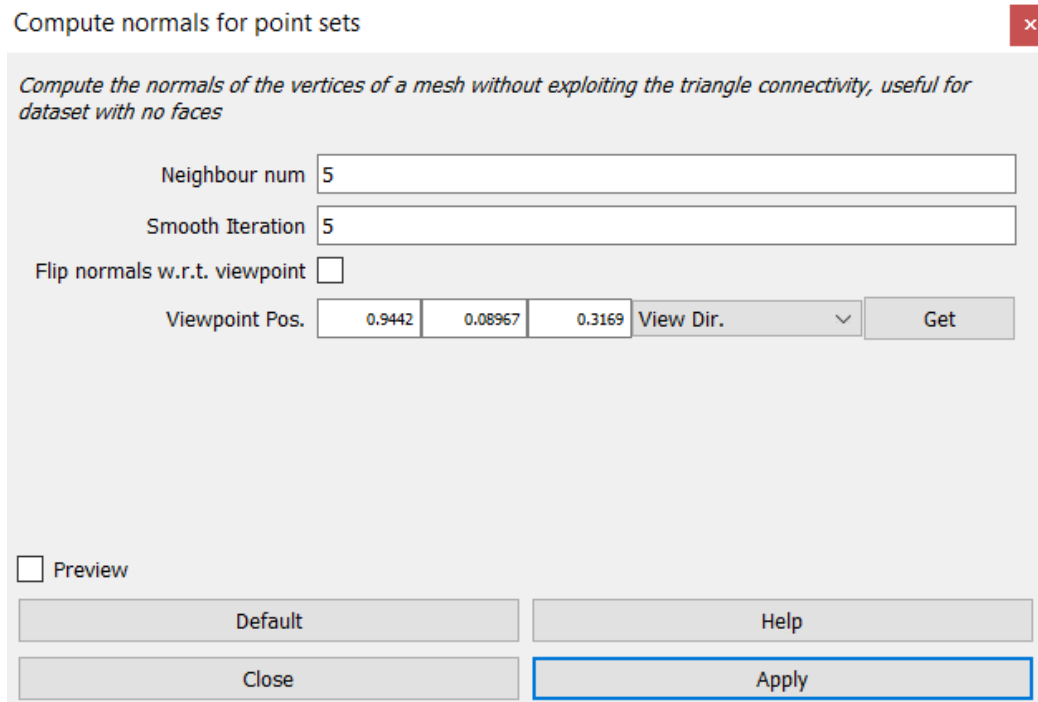
Nakon što su nepotrebne točke uklonjene, može se početi sa generiranjem 3D modela iz novog, počišćenog oblaka točaka. Za generiranje će se koristiti metoda rekonstrukcije površine *Screened Poisson* koja se može odabrati u programu MeshLab, a ulazni podatci koji su potrebni metodi su orijentirani skupovi točaka. Da bi se postupak mogao pokrenuti, potrebno je izračunati normale za skup točaka veličine koju korisnik programa određuje.



**Slika 34. Odabir opcije računanja normala za proizvoljni skup točaka**

Odabirom prikazane opcije otvara se novi prozor u kojem se mogu odabrati parametri izračuna normala za skup točaka. Parametri koji se traže su broj susjeda (engl. *Neighbour number*) i broj iteracija zaglađivanja (engl. *Smooth iteration*). Ukoliko se odabere veći broj susjeda, kreirana površina iz ovih izračunatih normala će imati manje konačnih elemenata koji će biti većih dimenzija. To će rezultirati time da površine neće biti glatke, a detalji se gotovo pa i neće moći razaznati na modelu. Maksimalni broj susjeda koji metoda dozvoljava je 99. Odabirom jako malog broja susjeda, površine generirane iz ovoga skupa biti će prikazane sa većim brojem konačnih elemenata što znači da će biti detaljnije i glađe, međutim to računanje će više opteretiti računalo u konačnici. Ukoliko se zadaju preteški zahtjevi za računalo, program neće

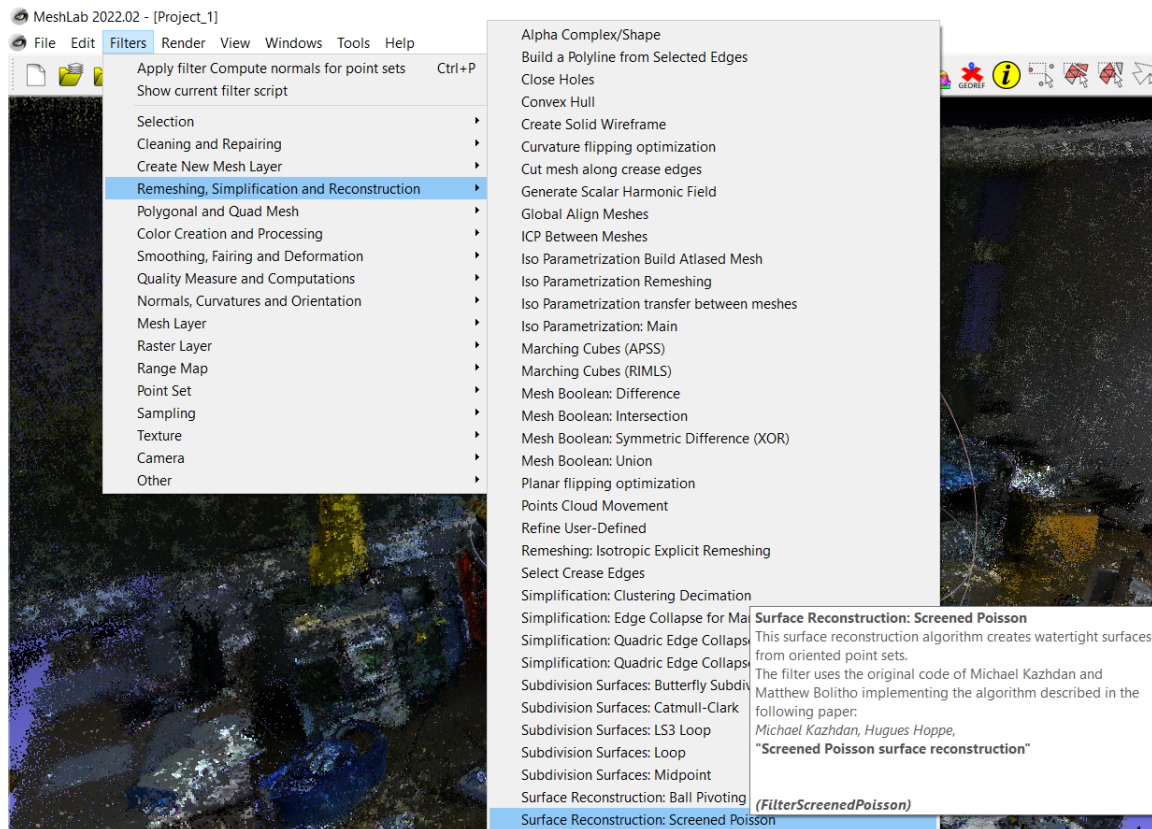
moći generirati površinu iz toga nego će javiti grešku da radna memorija računala to ne može podnijeti. Ukoliko uzmemo u obzir da ovaj oblak točaka sadrži preko osam milijuna točaka ako se odabere broj susjeda jedan, to će biti 4 milijuna računskih operacija što je jako puno za neko prosječno računalo. Iteracije zaglađivanja je broj dodatnih računanja koje će poboljšati izračun normala u snimljenom prostoru.



**Slika 35. Odabir veličine skupa točaka i broja iteracija za računanje normala**

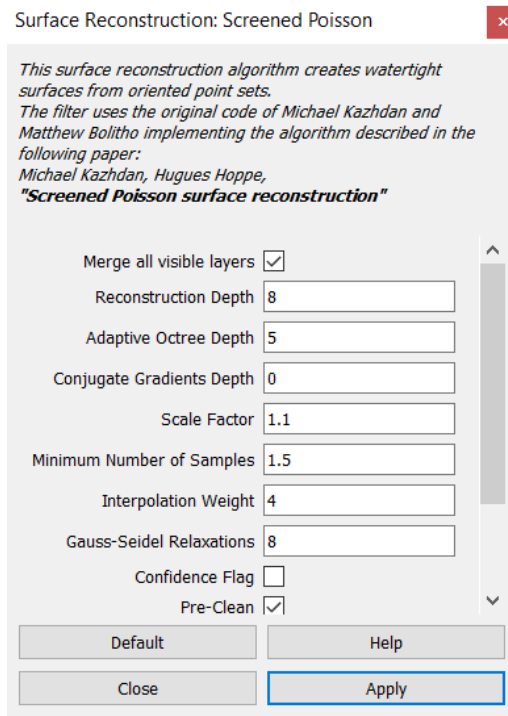
Pritiskom na tipku Primjeni (engl. *Apply*) računalo započinje računanje na cijelom skupu točaka po zadanim ulaznim podacima. Odabrano je računanje sa pet susjeda i pet iteracija zaglađivanja što je neki optimum za računalo koje se koristi. To je neki kompromis između kvalitete generiranja modela i vremena potrebnog za generiranje i uopće mogućnosti generiranja što je prethodno objašnjeno. Nakon što se normale izračunaju, rezultati su spremljeni i mogu se implementirati u metodu generiranja 3D mreže *Screened Poisson*. Pokretanje metode se pokreće odabirom *Filters* nakon čega u padajućem izborniku treba odabrati *Remeshing, Simplification and Reconstruction* te iz narednog padajućeg izbornika opciju *Surface reconstruction: Screened Poisson*. Ukoliko normale za skup točaka nisu izračunate program će javiti grešku da nije moguće pokrenuti rekonstrukciju površine.





**Slika 36. Odabir metode rekonstrukcije površine: Screened Poisson**

Ulazni parametri ove metode su postavljeni na standardne, jedino je uključena dodatna opcija predčišćenja u kojem metoda može isključiti pojedine normale koje smatra nebitnima za generiranje 3D modela. Ovu značajku je korisno uključiti u slučajevima kada je oblak točaka raspršen kao u ovom slučaju. Ova metoda se temelji na rekonstrukciji površine *Screened Poisson* koja je poboljšanje Poissonove metode rekonstrukcije površine. Pogodna je za kreiranje 3D modela iz raspršenih oblaka točaka. Temelji se na razmatranju da se polje normala granica čvrstog tijela može interpretirati kao gradijent indikatorske funkcije čvrstog tijela. Postupak obrade orijentiranog skupa točaka se odvija na taj način da se prvo one transformiraju u kontinuirano vektorsko polje u 3D prostoru, nakon toga se nalazi funkcija čiji gradijenti najbolje odgovaraju vektorskom polju. Na osnovu te funkcije se generiraju prikladne površine. Najbitniji, a i najkompleksniji dio ove metode je dio računanja funkcije kojom se opisuju površine i taj dio metode se unaprijeđivao kroz godine i bio je tema mnogih znanstvenih radova.



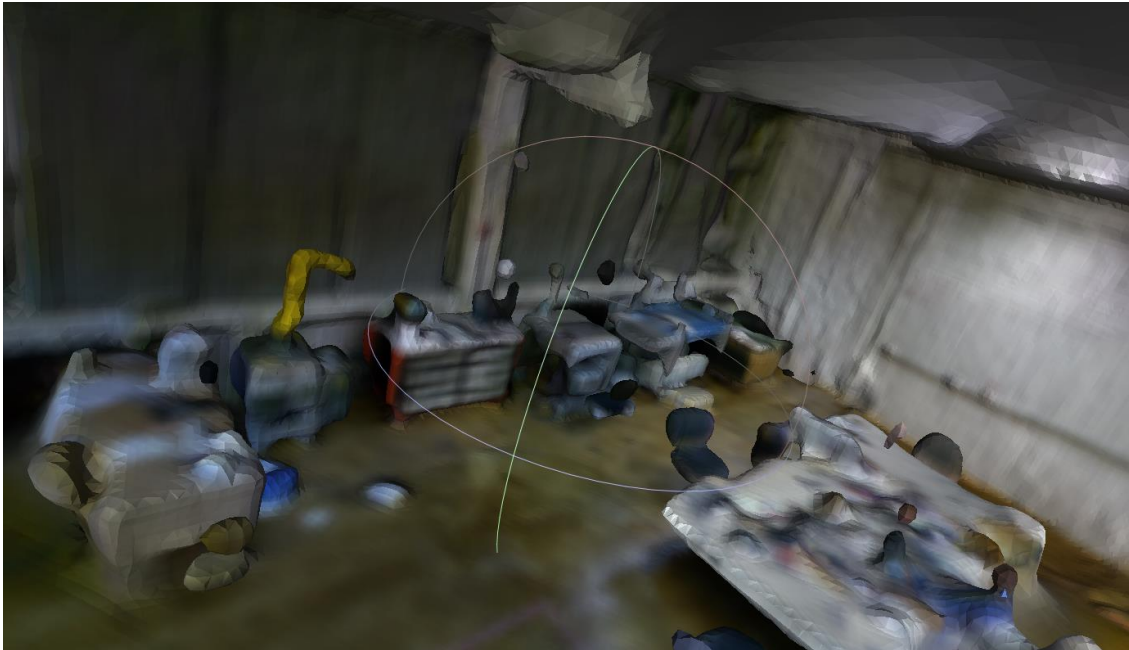
**Slika 37. Rekonstrukcija površine Screened Poisson unos ulaznih argumenata**

Postupak generiranja površina iz oblaka točaka se pokreće pritiskom na tipku Primjeni (engl. *Apply*). Vrijeme potrebno da se ova metoda izvrši može biti duže i može se dogoditi da program neko vrijeme uopće ne reagira (engl. *Not responding*). Nakon što se potrebno računanje i generiranje 3D modela izvrši, na statusnoj traci se pojavi poruka da da je filter primijenjen i koliko vremena je za to trebalo. U slučaju u kojem su odabrane standardne postavke iz programa za generiranje površina primjenom ove metode, vrijeme potrebno za to je bilo 11 sekundi.

```
Applied filter Surface Reconstruction  
Screened Poisson in 11059 msec
```

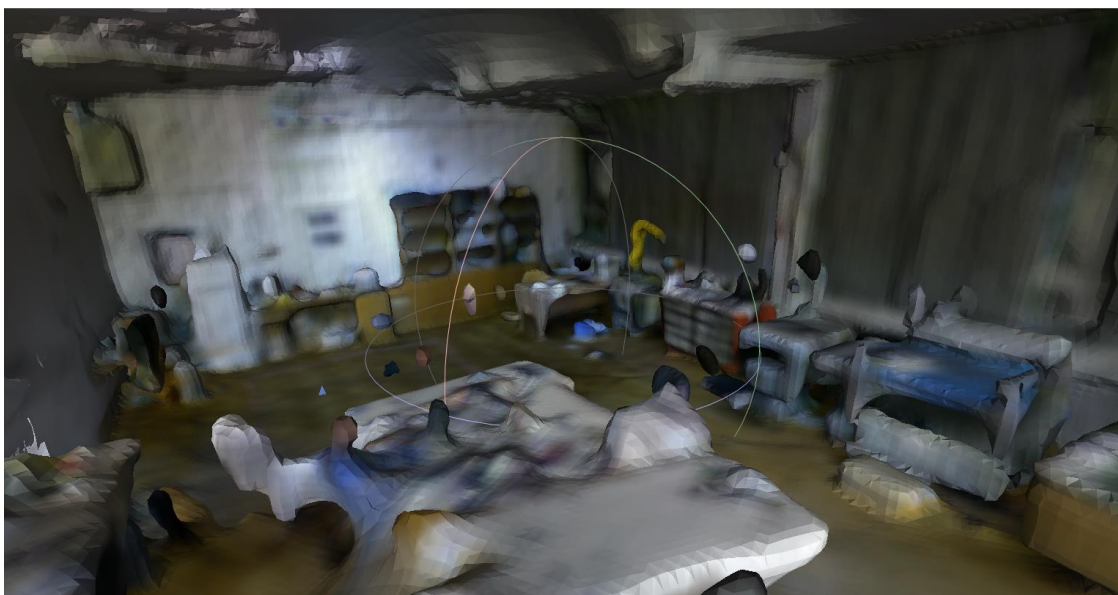
**Slika 38. Poruka iz statusne trake**





**Slika 39. Pogled 1- Izgled kreiranog 3D modela laboratorija**

Ovaj model je pokazatelj prednosti metode kreiranja 3D modela iz oblaka točaka *Screened Poisson* u slučaju raspršenog oblaka točaka. Rupe koje su se nalazile u oblaku točaka su popunjene, a model izgleda relativno prihvatljivo. 3D model prostora nije vjerni prikaz prostora, ali definitivno može poslužiti kao baza za daljnji rad u programu poput Blendera gdje je moguće kreirati virtualne objekte i zamijeniti ih sa postojećima, npr. model robota zamijeniti sa kreiranim i sl.



**Slika 40. Pogled 2- Izgled kreiranog 3D modela laboratorija**

### 3.5. Evaluacija snimki sa iPad Pro LiDAR sustavom

Na iOS platformi nije moguće koristiti program RTAB-Map, ali i sama tehnika snimanja se razlikuje od one sa prethodne dvije kamere. U ovom slučaju se koristi LiDAR, kao što je prethodno objašnjeno, koji ispušta zraku svjetlosti te na osnovu mjerenja vremena potrebnog da se zraka reflektira od objekta estimira udaljenost od objekta. U svrhu izuzimanja snimki i kreiranja modela koristiti će se aplikacija koja je razvijena isključivo za iOS, a naziva se 3D Scanner App. iOS je operativni sustav koji se koristi na Apple-ovim uređajima.



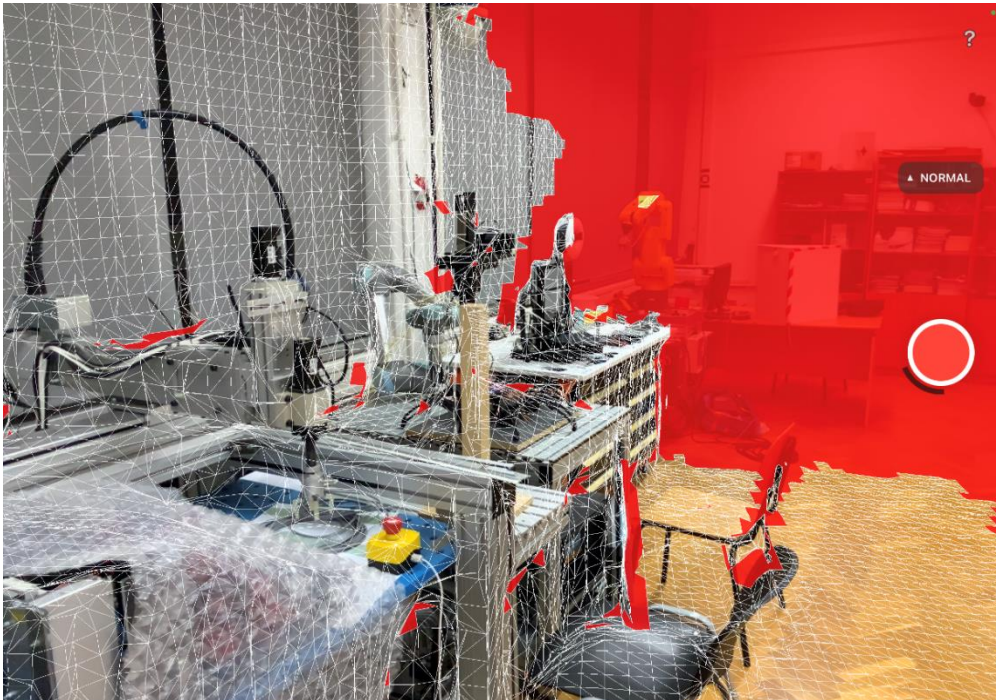
Slika 41. Ikona i naziv aplikacije

3D Scanner App je aplikacija koja koristi skeniranja sa LiDAR-a te na osnovu njih kreira 3D model skeniranog prostora ili objekta. Aplikacija je potpuno besplatna i dostupna svim korisnicima iOS-a. Skidati ju mogu svi, ali pokrenuti je mogu samo uređaji koji sadrže LiDAR u svom sustavu kamera. Rukovanje aplikacijom je zaista jednostavno, odabirom opcije *Scan*, otvara se kamera i pritiskom na crveni gumb pokreće se skeniranje prostora.



Slika 42. Pokretanje skeniranja prostora LiDAR-om

Nakon pokretanja snimanja, na zaslonu se prikazuje 3D kreirana mreža koja se mijenja u realnom vremenu ovisno o tome koliko se koji dio detaljno snimi. Također, dijelovi koji još nisu skenirani, prikazani su crvenom bojom na ekranu. Ovakav način omogućava korisniku da bolje prati trenutno stanje snimke i koje dijelove bi trebao detaljnije snimiti u kojem dijelu prostora. Prikaz je dosta jednostavan i interaktivan kao i cijela aplikacija koja na vrlo lak način omogućava komunikaciju između programa i korisnika. Nije potrebno kao u prethodnim slučajevima kalibrirati kameru, optimirati određene parametre ili instalirati neke programske pakete, ovdje je sve spremno na pritisak jednog gumba.

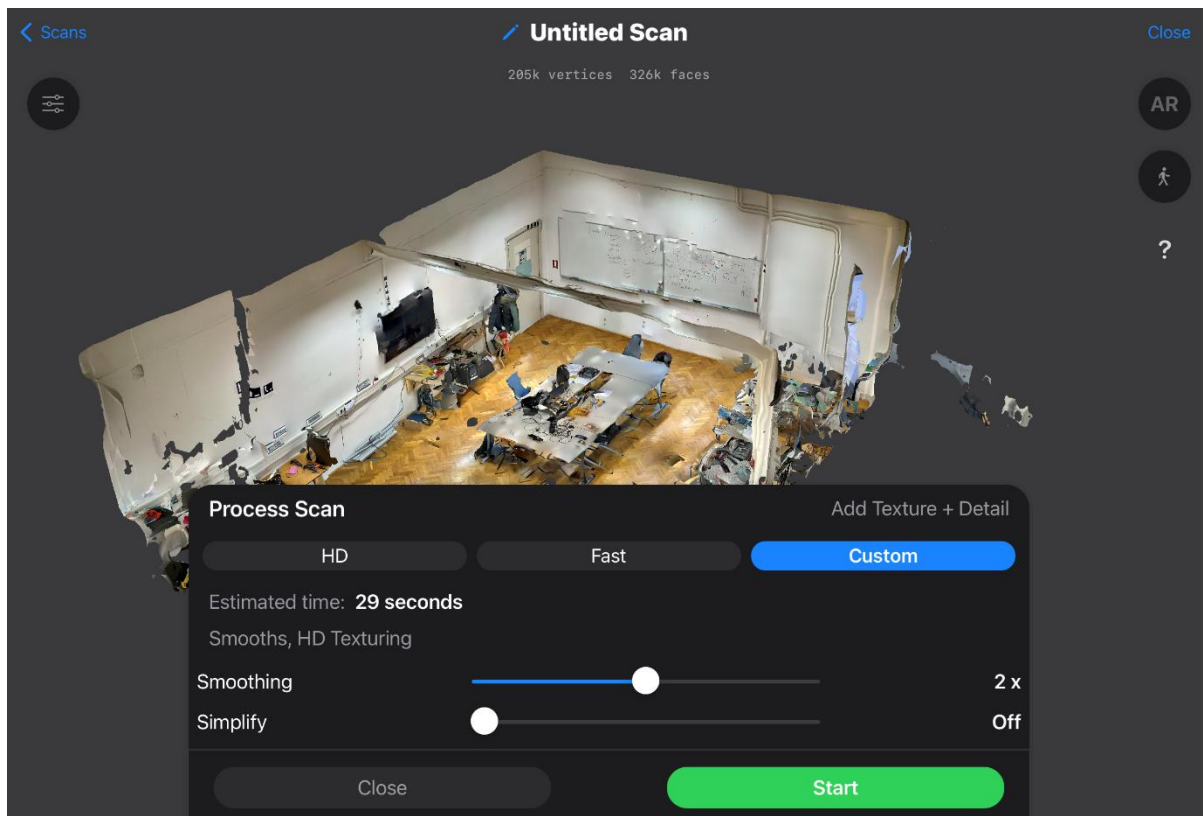


**Slika 43. Zaslona aplikacije tokom snimanja prostora**

Tokom snimanja, cilj je eliminirati crvena područja na ekranu, to je moguće usmjeravanjem LiDAR-a prema tom području. Za razliku od pristupa u snimanjima iz prethodnog dijela rada, ovim pristupom je lako snimati i kutove i ravne površine bez značajki kao npr. bijele zidove dok je to bio ogroman problem za SLAM pristup, u nekim slučajevima čak i nerješiv. Snimanje je također moguće i iz bližih udaljenosti od objekta, što prethodno nije bilo moguće. Brzina kojom se kamera pomiče je irelevantna, koliko god brzo da se kamera giba, neće doći do gubljenja kamere ili u konačnici zaustavljanja snimanja prostora.



Treba imati na umu ako se kamera brže giba, manje detalja izuzima i moglo bi doći do rupa u modelu. S obzirom na tu činjenicu, snimanje prostora može biti brže u usporedbi sa prethodnima. Ova tvrdnja se može potkrijepiti i iskustvom snimanja prostorije sa tri različite kamere i dva različita pristupa tokom izrade ovog rada. Može se estimirati da je otprilike trebalo duplo manje vremena za snimiti istu prostoriju laboratorija ovim pristupom u odnosu na SLAM.



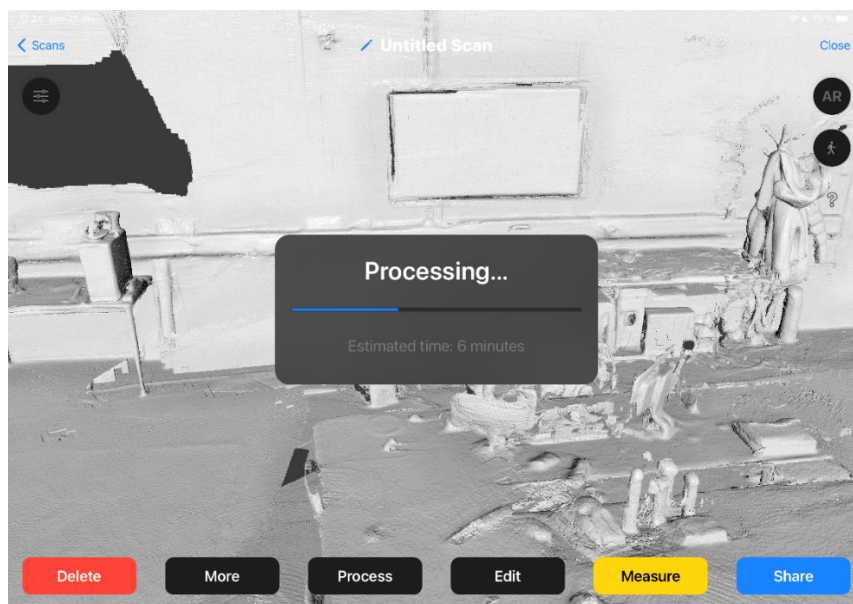
**Slika 44. Parametri procesiranja skeniranih podataka prostorije**

Procesiranje skeniranih podataka se obavlja onoliko detaljno koliko to korisnik aplikacije zatraži. Postoje tri opcije: *HD*, *Fast* i *Custom*. Odabir opcija i parametara tih opcija se obavlja postavljanjem interaktivnih klizača na željenu poziciju. Parametri su iskazani jednostavnim brojkama koje predstavljaju stupanj jačine funkcije zaglađivanja (engl. *Smoothing*) i pojednostavljenja (engl. *Simplify*). Korisnik ne mora znati veliki broj parametara koji se kriju iza ove dvije funkcije koje npr. u RTAB-Mapu i MeshLab-u mora znati da bi dobio što bolji rezultat. Aplikacija je maksimalno pojednostavila ovaj postupak.

*HD* je opcija u kojoj se naglasak stavlja na kvalitetu modela, odnosno oblaka točaka, ovisno u kojem formatu će se spremati. Na ovaj način će biti potrebno više vremena da se generira 3D model ili oblak točaka i više će se opteretiti procesor.

Opcija *Fast* je ona u kojoj se naglasak stavlja na brzinu procesiranja, kvaliteta modela koji će se spremati je manja od prethodne opcije, ali će se zato prije spremati i manje će opteretiti u ovom slučaju tablet.

Opcija koja je korištena za kreiranje 3D modela snimke prostora laboratorija LiDAR-om je *Custom*. To je opcija gdje se nalazi kompromis između prethodne dvije metode generiranja 3D modela. Odabrano je dvostruko zaglađivanje i bez pojednostavljenja zato što je cilj dobiti što detaljniji model. Ukoliko se stavi preveliko zaglađivanje, također se gube detalji jer bridovi neće biti dovoljno oštri i objekti neće biti toliko naglašeni koliko bi trebali da 3D model bude vjerni prikaz stvarnoga laboratorija.



**Slika 45. Procesiranje podataka na osnovu prethodno zadanih parametara**

Procesiranje podataka je trajalo oko šest minuta što i nije toliko puno s obzirom da je prostor velik i da je sustav LiDAR-a pronašao jako puno točaka u snimanju. Ako se ovaj podatak uspoređi sa MeshLab procesiranjem podataka, a to je računanje normala pa zatim kreiranje 3D modela, rezultati su u prosjeku isti.

## 4. USPOREDBA REZULTATA I ODABIR NAJBOLJEG MODELA

U prethodnim poglavljima je detaljno prikazan postupak dobivanja 3D modela prostorije Laboratorija za projektiranje izradbenih i montažnih sustava. Tokom rada prikazane su dvije metode dobivanja 3D modela, a snimke su se obavljale upotrebom tri različite kamere, odnosno dvije kamere i LiDAR-om.

Dobiveni modeli, odnosno oblaci točaka se mogu usporediti na temelju osobnog utiska vjernosti prikaza stvarnog prostora. Odabrat će se jedan model koji će se dodatno obraditi te će takav novi, popravljeni i uljepšani model biti integriran u virtualnom okruženju pomoću programa Unreal Engine 5.

3D oblaci točaka i modeli koji se generiraju iz tih oblaka kreiranih na osnovu SLAM principa imaju određene nedostatke koji uvelike utječu na kvalitetu konačnog modela. Oblak točaka kreiran iz snimaka sa Intel RealSense D435 stereo kamerom je potpuno raspršen, a objekti koji se nalaze u prostoriji, poput robota, računala ili stolice nisu konzistentni. Treba istaknuti i činjenicu da prezentirani oblak točaka sa slike 21. nije nastao iz samo jednog pokušaja nego je to najbolji rezultat u izboru od pet pokušaja snimanja prostora. Puno vremena je uloženo za dobivanje prihvatljivog rezultata koji i dalje vjerno ne prikazuje prostor, s toga se ovaj rezultat izbacuje iz izbora te se u radu nije ni kreirao model iz ovog oblaka točaka jer se moglo zaključiti da generirani model neće biti prihvatljiv.

Model kreiran na osnovu snimki sa Xbox360 Kinect kamere je bolji i konzistentniji u odnosu na prethodni primjer, ali također nije ni blizu vjernog prikaza prostorije. Definitivno se mogu primijeniti određeni filteri koji mogu malo popraviti izgled, ali ne toliko dobro da bi se ovaj model koristio za integraciju u virtualno okruženje. Premda, na slici 39. se vidi da generirani model ima potencijala i mogao bi poslužiti kao skelet iz kojeg bi se izradio model u virtualnom okruženju.

Prethodna dva slučaja koristila su SLAM pristup koji je kroz dobivene rezultate koji su pokazali određene nedostatke toga pristupa. Pokušaj sa LiDAR skeniranjem generirao je model koji puno vjernije prikazuje stvarni prostor laboratorija u odnosu na prethodna dva. Na modelu se vide svi objekti, a čak i detalji poput slova na ploči, malih kutija na stolu i sl. Osim što je prikaz točniji i detaljniji, snimanje je brže, a zadavanje ulaznih parametara procesiranja snimke jednostavno.



**Slika 46. Kreirani 3D model iz skeniranja LiDAR-om**

Još jedna od prednosti ovoga modela je što je skeniran i strop prostorije. U prethodna dva slučaja skeniranje cijelog stropa je jako teško, gotovo pa i nemoguće. Na slici se ne vidi strop i dva zida koja se nalaze na strani od koje dolazi pogled na prostoriju, ali te površine i objekti postoje, samo su u ovom slučaju prozirne da bi korisnik mogao vidjeti unutrašnjost prostorije.

Konačna procjena je da se za integraciju modela u virtualno okruženje odabire model kreiran iz snimaka sa LiDAR-om iPad Pro-a. Na nekoliko mjesta se nalaze manje rupe u modelu koje će se u sljedećem dijelu rada popuniti. Rupe nastaju na mjestima gdje nije snimljeno dovoljno detalja, a uzrok tomu je što aplikacija ima limit koliko memorije može zauzeti snimak, a prostorija je dosta velika pa se trebalo snimati bržim tempom da se uspije proći cijeli laboratorij. Nedostatci odnosno rupe u modelu nisu zabrinjavajuće i u domeni su problema koji se lako rješavaju.

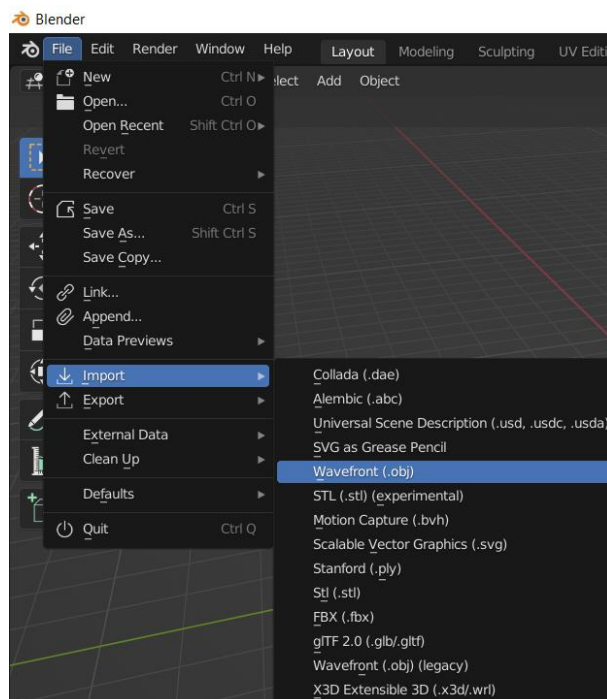
## 5. POPRAVLJANJE ODABRANOG 3D MODELA

Odabrani model ima određenih nedostataka, a glavni je taj što se na određenim mjestima pojavljuju rupe u modelu odnosno dijelovi u kojima nije kreirana 3D mreža na osnovu oblaka točaka. Za popravljjanje modela koristiti će se program Blender koji je besplatan i dostupan svim korisnicima.



Slika 47. Ikona aplikacije Blender

Blender je program koji služi za grafičko modeliranje, kreiranje, renderiranje i simulaciju 3D scena. Program ima široku primjenu i u zadnje vrijeme se čak koristi za uređivanje video zapisa te kreiranje video igara. Ovaj program sadrži pregršt značajki koje se mogu koristiti i pomoći korisniku u rješavanju prethodno navedenih zadataka. U ovome radu koristiti će se osnovne naredbe iz ovoga programa koje će služiti za popunjavanje rupa u kreiranoj 3D mreži modela. Mogućnosti programa su daleko veće od onoga što će biti prezentirano u ovome dijelu rada. Prije svega je potrebno učitati 3D model prostorije u scenu programa Blender.



Slika 48. Učitavanje 3D modela laboratorija u programu Blender



Datoteka sa ekstenzijom .obj je 3D model koji za razliku od istoga sa ekstenzijom .stl sadrži podatke o teksturi modela. Učitavanjem modela, učitava se i datoteka sa teksturom, uz uvjet da se te dvije datoteke nalaze u istoj mapi. Nakon što se model učitava u Blenderu, on je sive boje, odnosno na njemu nije prikazana tekstura.



**Slika 49. 3D model bez teksture**

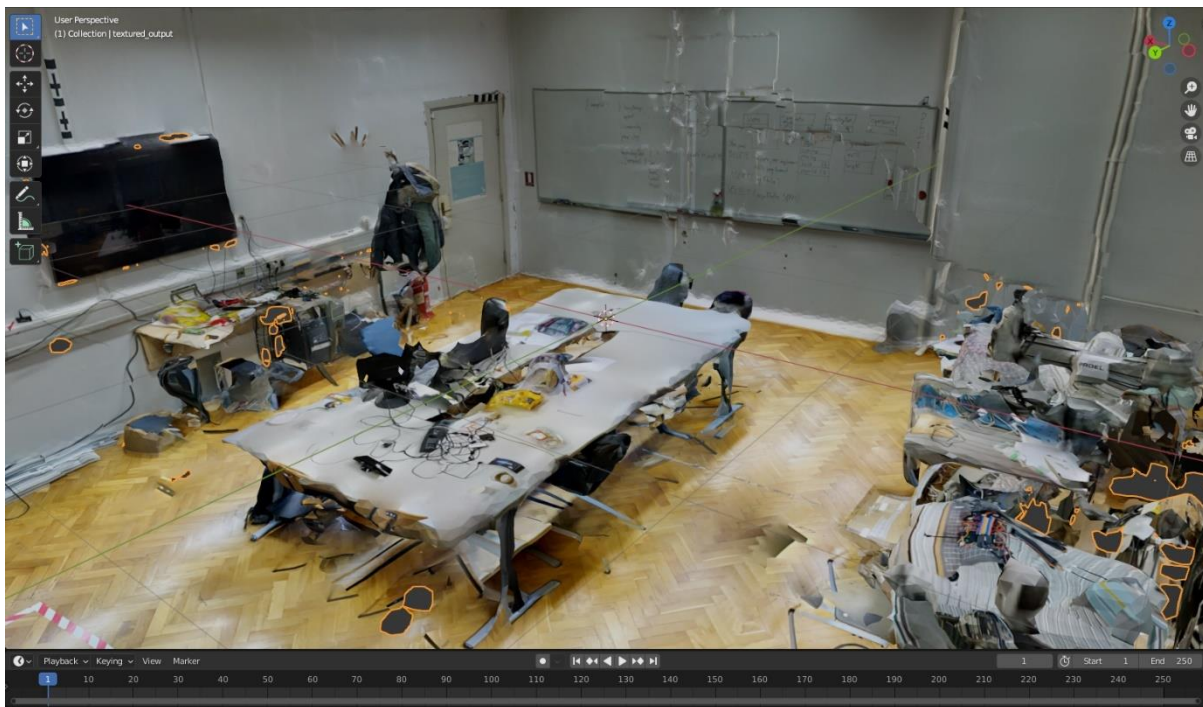
Odabirom naredbe *Viewport Shading* tekstura će se prikazati na 3D modelu. Ukoliko model nije u istoj mapi kao i datoteka teksture ili ona uopće ne postoji, ništa se neće dogoditi, model bi u ovom slučaju izgledao isto kao i na prethodnoj slici.



**Slika 50. 3D model sa primijenjenom teksturom**

U ovakvom pogledu se model vidi iz daljine i ne mogu se primijetiti nedostaci modela, odnosno da pojedini dijelovi fale. Također, može se vidjeti da model sadrži i strop, o čemu se govorilo prilikom evaluacije rezultata snimke sa LiDAR-a, ali nije bio prikazan, u ovom slučaju se može i vidjeti.

Da bi se pogled premjestio unutar prostorije, potrebno je kotačićem miša približiti pogled na lokaciju u prostoriji koja odgovara korisniku. Nakon što se približi pogled, mogu se vidjeti nedostaci unutar modela, čak će ih program naglasiti narančastim obrubom.



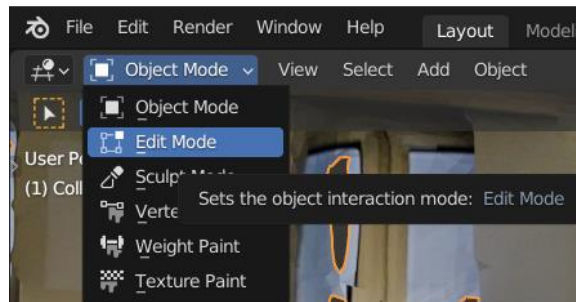
**Slika 51. Bliži pogled na 3D model prostorije**

Na ovom dijelu prostorije se pojavljuju rupe koje su relativno male, ali ih nema puno. Na drugoj strani prostorije, nalaze se malo veće rupe koje se nalaze na zavjesama i oko stupa, ali također njihov broj nije zabrinjavajući. Iz ovog bližeg pogleda se može vidjeti koliko je kreirani model detaljniji i čišći te realniji u odnosu na ostala dva koja su razmatrana u ovome radu. Detalji poput kablova po zidovima, cijevi na zidovima ili slova na ploči zaista pokazuju koliko je ova metoda superiornija u odnosu na SLAM metodu.



**Slika 52. Pogled iz drugog kuta na 3D model laboratorija**

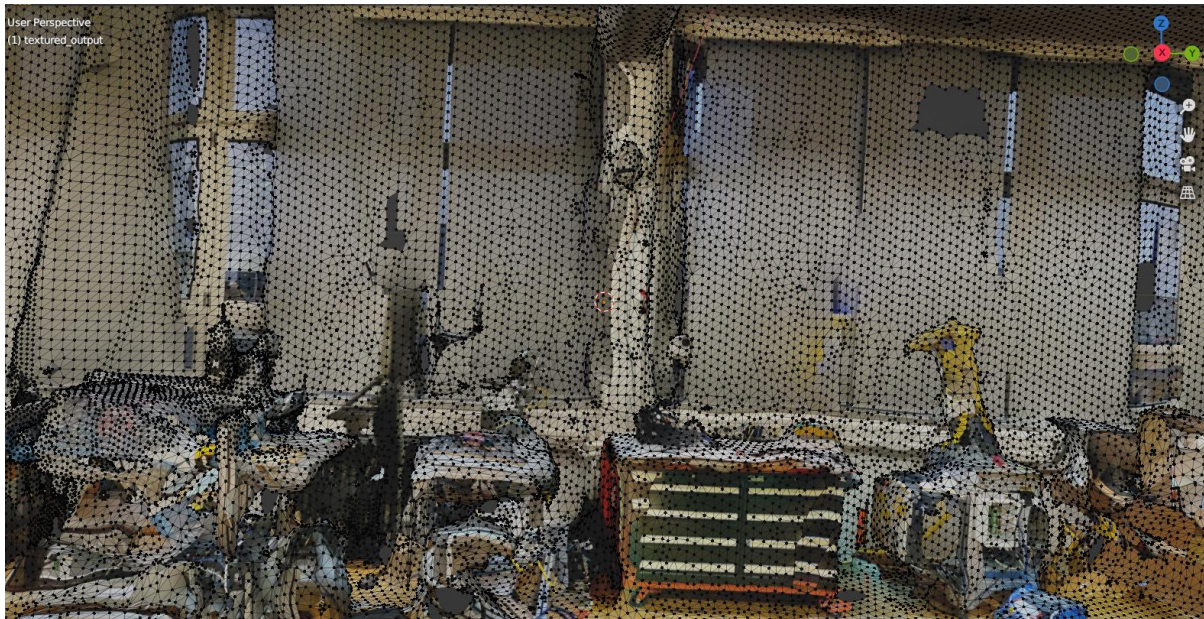
Da bi se popunile rupe koristiti će se jednostavna naredba koja je u sklopu paketa naredbi programa Blender, a zove se *Fill*. Prije toga je potrebno zamijeniti način rada (engl. *Mode*) iz objektnog načina rada (engl. *Object Mode*) u način rada za uređivanje (engl. *Edit Mode*).



**Slika 53. Promjena načina rada u programskom sučelju Blender-a**

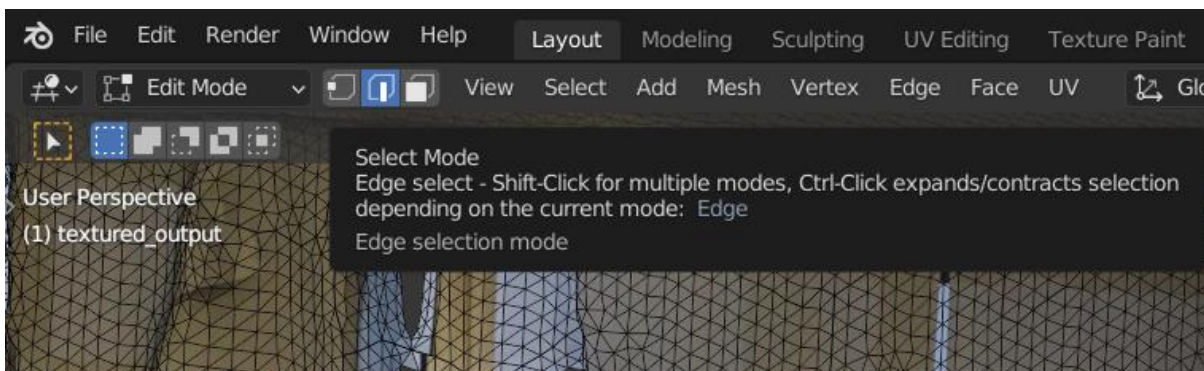
U ovom načinu rada, 3D model će biti prikazan u 3D mreži trokutnih konačnih elemenata. Na vrhovima trokuta će biti crne točke pomoću kojih se može manipulirati sa mrežom. Moguće je premještati točke pa tako izvlačiti element ili zaoštriti određeni dio objekta i sl. Također, mogu se dodavati i osnovni objekti poput kugle, kocke, valjka ili kvadra koji se nakon dodavanja mogu dodatno uređivati. Sve te akcije se mogu izvršiti u ovome načinu rada. U slučaju da određeni dio prostorije, odnosno objekt koji se u tom dijelu nalazi, ne ispadne onako kako stvarno izgleda, uvijek se na ovaj način može zamijeniti onim virtualnim.





**Slika 54. Izgled 3D modela u načinu rada za uređivanje**

Nakon što se prikaz promijenio u prikaz mreže konačnih elemenata moguće je pokrenuti postupak popunjavanja rupa unutar cijelog skupa 3D mreže prostora. Prije toga je potrebno promijeniti način odabira, standardno je postavljen na odabir vrhova (engl. *Vertex select*), a proces korigiranja rupa u 3D mreži će biti puno lakši i brži ukoliko se postavi način odabira na odabir rubova (engl. *Edge select*).



**Slika 55. Promjena načina odabira u odabir rubova**

Na slici se vidi da se na 3D mreži više ne nalaze crne točke koje predstavljaju vrhove, nego se samo vide rubovi trokutastih konačnih elemenata. Kada se pritisne na određenu lokaciju na mreži koja se želi označiti, biti će označen rub koji je najbliži kursoru miša. U prethodnom načinu, označena bi bila najbliža točka.



Popunjavanje praznina u 3D mreži modela se vrši tako da se uz konstantno držanje tipke *Ctrl* pritisnutom, označe svi rubovi mreže oko rupe na tom dijelu modela koji se promatra. Tipka *Ctrl* se drži da bi se mogli označiti i zapamtiti svi rubovi, ukoliko se ne bi držala, konstantno bi se označavao samo jedan rub, a odabir prethodnog bi se poništio. Nakon što se označe svi rubovi koji omeđuju rupu u modelu, potrebno je pritisnuti tipku *F* na tipkovnici koja označava naredbu ispuni (engl. *Fill*).



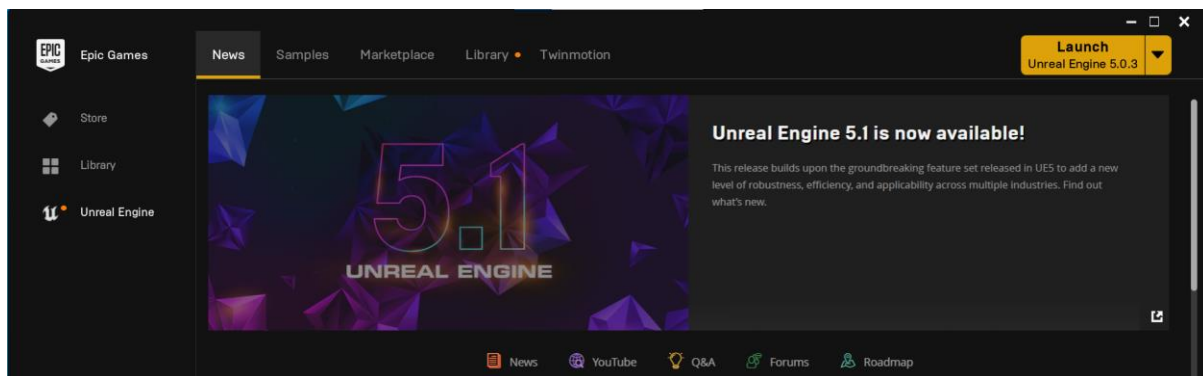
**Slika 56. Popravljena 3D mreža modela**

Potrebno je ponoviti objašnjeni postupak na svim dijelovima mreže prostorije na kojima se nalaze diskontinuiteti. Zbog jednostavnog principa rada programa ovaj postupak se obavi dosta brzo, naravno ovisi o broju rupa o modelu, ali konkretno u ovom slučaju popravljavanje modela nije iziskivalo puno uloženog vremena. Boje na dijelu na kojem se mreža popunila će se odrediti iz datoteke s teksturom.

## 6. IMPLEMENTIRANJE 3D MODELA U VIRTUALNO OKRUŽENJE

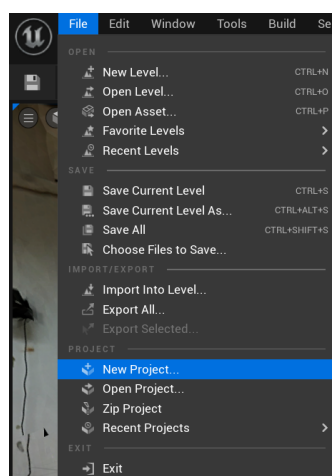
Ovaj dio rada će se obavljati na radnoj stanici (engl. *Workstation*) koja se nalazi u laboratoriju. Unreal Engine 5 je program koji će se koristiti za kreiranje virtualnog okruženja. Osnovni zahtjevi ovog programa su daleko iznad onih koje prijenosno računalo koje se koristilo do sada u ovome radu može ispuniti. Općenito, bilo kakvo kreiranje virtualnog okruženja je nemoguće bez jakih specifikacija.

Unreal Engine 5 se pokreće tako da se otvori Epic Games Launcher program u čije se izborniku nalazi Unreal Engine, potrebno je odabrati ovu opciju. Nakon toga u desnom kutu se nalazi narančasti gumb za pokretanje (engl. *Launch*). Pritiskom na ovaj gumb, program će se pokrenuti.



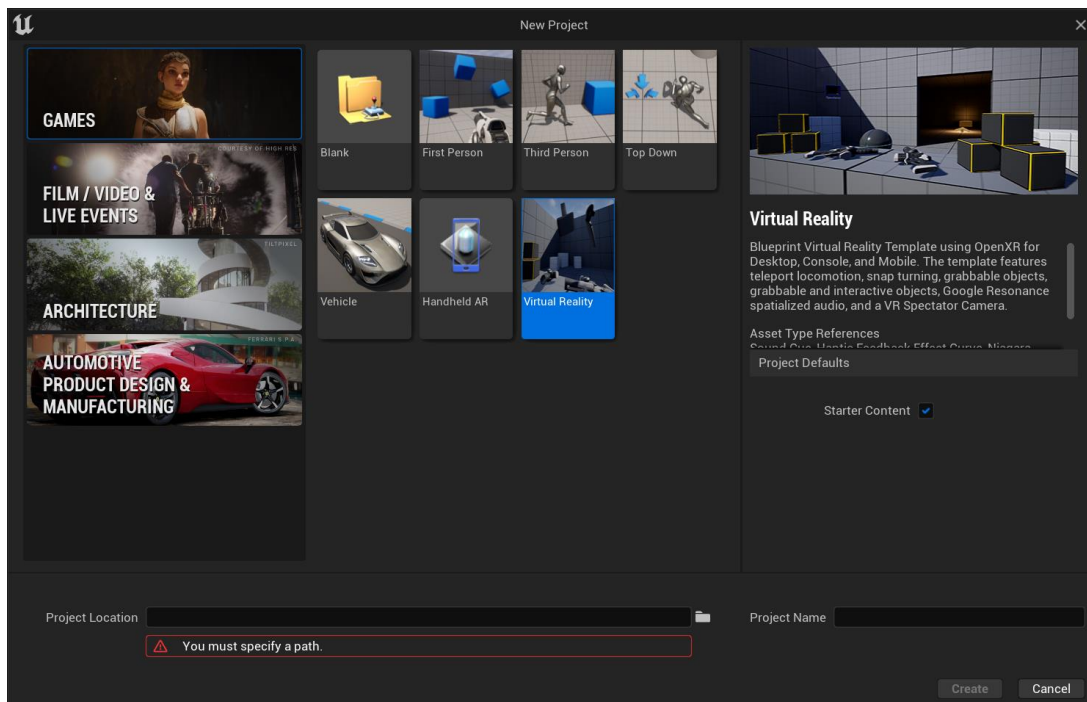
Slika 57. Pokretanje programa Unreal Engine

Potrebno je malo vremena da se program otvori nakon čega je potrebno izabrati novi projekt (engl. *New Project*) u kojem će se moći dodati 3D model prostorije koji je prethodno popavljen u Blender-u. Odabir opcije *New Project* se odabire iz padajućeg izbornika *File*.



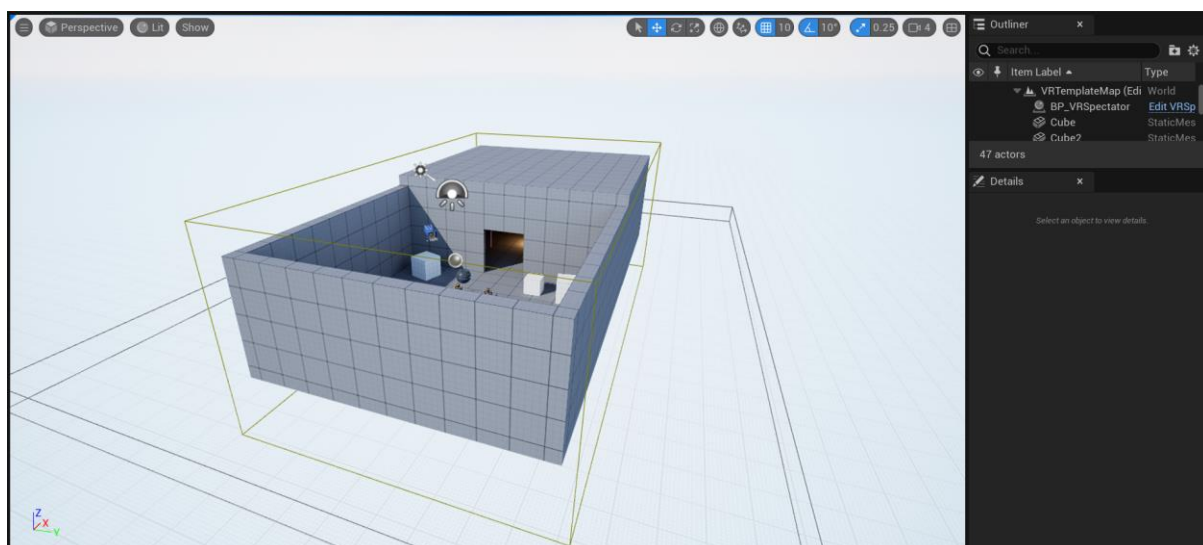
Slika 58. Pokretanje novog projekta

U novom prozoru koji se otvori nakon pritiska na opciju novi projekt, potrebno je u ponuđenim opcijama odabrati virtualnu stvarnost (engl. *Virtual Reality*). Krajnji cilj implementiranja 3D modela laboratorija u virtualno okruženje je povezivanje tog modela sa VR naočalama pa je ovaj izbor adekvatan za tu namjenu.



**Slika 59. Kreiranje novog projekta sa opcijom virtualne stvarnosti**

Nakon što se odredi lokacija projekta (engl. *Project Location*) na računalu u koju se projekt želi spremiti te ime projekta (engl. *Project Name*), otvoriti će se osnovni model koji služi kao predložak u ovome okruženju.

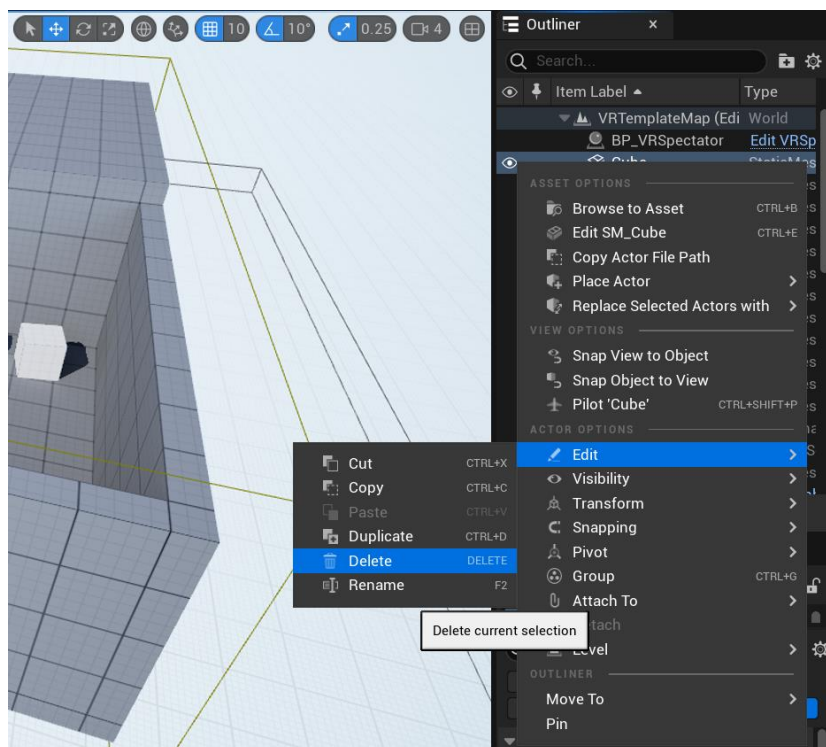


**Slika 60. Mapa predložka u projektu virtualne stvarnosti**



Iz mape predložka mogu se iskoristiti određeni virtualni objekti, a višak je potrebno obrisati. Od objekata koji se mogu iskoristiti je definitivno pod (engl. *Floor*). Učitavanjem popravljene modela u virtualno okruženje, 3D model će se pojaviti u virtualnom okruženju. Ukoliko bi se u tom slučaju odredila početna pozicija VR korisnika i nakon toga bi se takvo okruženje spojilo na VR naočale, korisnik bi nakon pokretanja simulacije na VR naočalama krenuo propadati kroz mapu i beskonačno bi propadao dok ne bi skinuo naočale. To se događa zato što nije definiran pod jer program ne može zaključiti gdje se pod nalazi na implementiranom modelu, što rezultira time da onaj koji razvija virtualno okruženje mora odrediti pod i na njemu graditi cijelu mapu, kao što je to i u predlošku napravljeno.

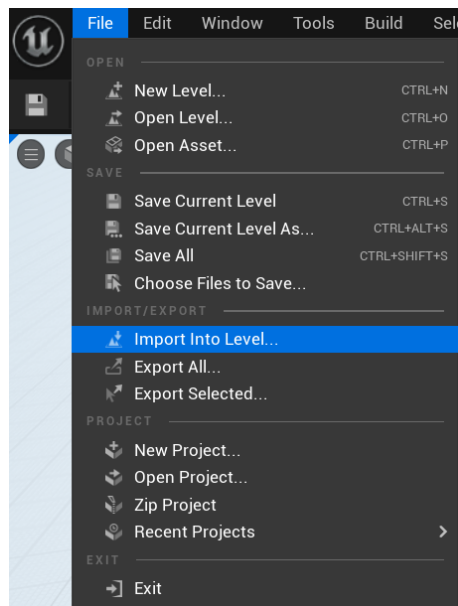
Brisanje virtualnih objekata se izvršava tako da se u popisu objekata (engl. *Item Label*) odabere na objekt koji je potrebno obrisati te se pritiskom na desnu tipku miša otvara padajući izbornik iz kojega se odabire opcija uredi (engl. *Edit*) i iz sljedećeg padajućeg izbornika se odabire opcija brisanja (engl. *Delete*). Postupak se ponavlja za sve objekte koji nisu potrebni.



**Slika 61. Brisanje nepotrebnih virtualnih objekata na standardnom predlošku**

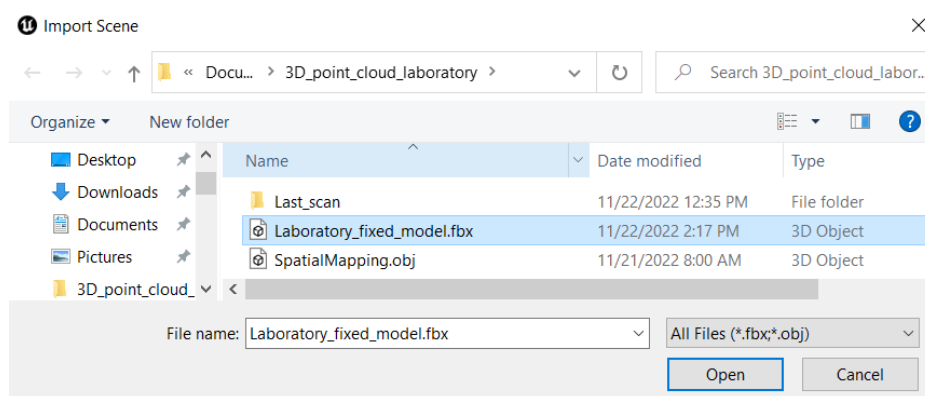
Nakon što su svi nepotrebni objekti izbrisani, može se započeti postupak integriranja 3D modela Laboratorija za projektiranje izradbenih i montažnih sustava. Prije toga je bitno napomenuti da se nakon što se model popravi u programu Blender, model treba spremiti sa ekstenzijom .fbx jer je to ekstenzija koju Unreal Engine podržava.

Model se implementira tako što se odabere opcija *File* te iz padajućeg izbornika opcija dodaj unutar razine (engl. *Import Into Level*).



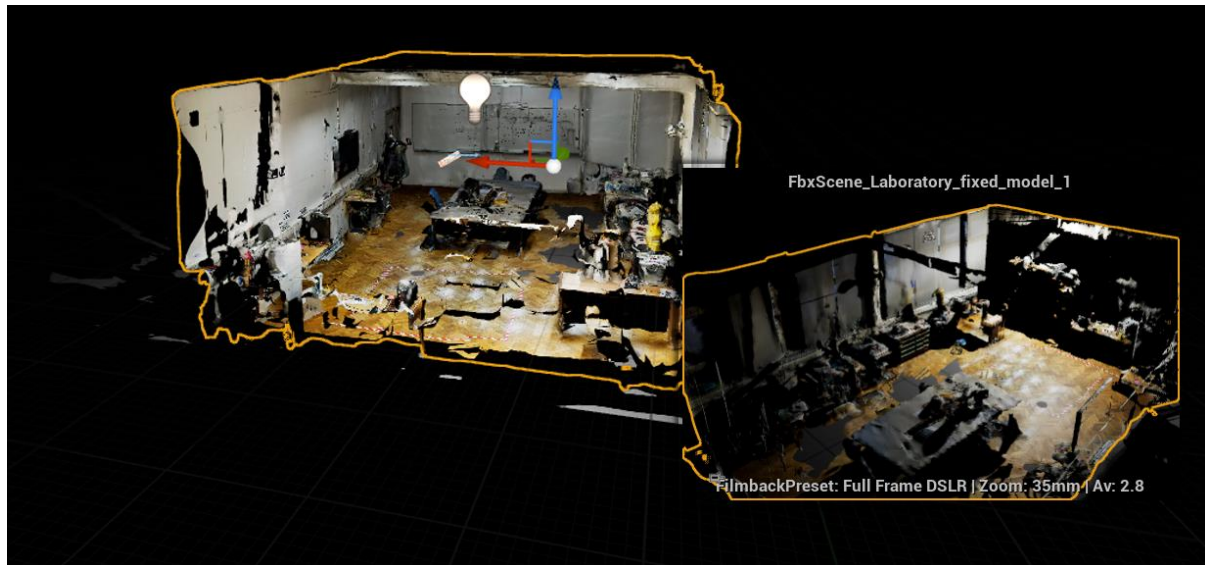
**Slika 62. Dodavanje modela u virtualno okruženje**

Nakon toga se mora pronaći lokacija na računalu na kojoj je spremljena datoteka sa 3D modelom laboratorija u .fbx formatu. Nakon što se pronađe ta lokacija odabere se opcija otvori (engl. *Open*). Odabirom te opcije pojaviti će se još jedan prozor u kojem će se tražiti od korisnika da odredi lokaciju na kojoj želi spremiti podatke o sceni, ta lokacija može biti bilo gdje od ponuđenih unutar računala, pritiskom na *OK* odabir se sprema.



**Slika 63. Otvaranje 3D modela laboratorija u .fbx formatu**

Nakon odabrane lokacije na kojoj će se spremati podatci o sceni, 3D model laboratorija će se implementirati u virtualnu scenu. Model će biti smješten na neku nasumičnu lokaciju u virtualnom okruženju. Pritiskom na model, pojaviti će se koordinatni sustav koji služi za manipuliranje s modelom unutar virtualnog okruženja. Potrebno je stisnuti na plavu strelicu koja predstavlja z os te pomaknuti model u ravninu sa podom.



**Slika 64. Izgled virtualne scene nakon implementiranja 3D modela laboratorija**

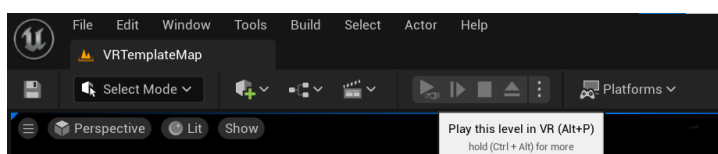
Također, isti postupak je potrebno odraditi sa svjetlom (engl. *Light*). To je virtualni objekt u obliku sijalice koji predstavlja osvijetljene scene. Virtualnom sijalicom se manipulira na isti način kao i sa modelom sobe. Postavlja se kao što je i na slici 64. prikazano u razini stropa. Ukoliko osvjetljenje nije dovoljno, moguće ih je postaviti i više.

Početna pozicija gdje će se korisnik stvoriti u virtualnom okruženju kada pogleda kroz VR naočale se postavlja na isti način. Virtualni objekt koji se naziva *Player Start* se također postavlja na isti način kao i model i osvjetljenje. Također je moguće postaviti i početnu orijentaciju koju će korisnik imati prilikom početka prikazivanja 3D prikaza prostorije. Najbolje je prvi puta tokom testiranja postaviti poziciju na suprotni kraj sobe od onoga gdje se korisnik s VR naočalama nalazi. Nakon što se nekoliko puta isproba i shvati kako funkcionira interakcija čovjeka u stvarnom svijetu sa onim virtualnim, korisnik se može postaviti na istu lokaciju u stvarnom i virtualnom svijetu. Postoji opasnost u tom slučaju jer ukoliko korisnik nekontrolirano hoda po virtualnom svijetu može se zabiti u stvarne objekte te se ozlijediti ili oštetiti inventar prostorije u stvarnom svijetu, što u virtualnom ne može na što treba obratiti pozornost.



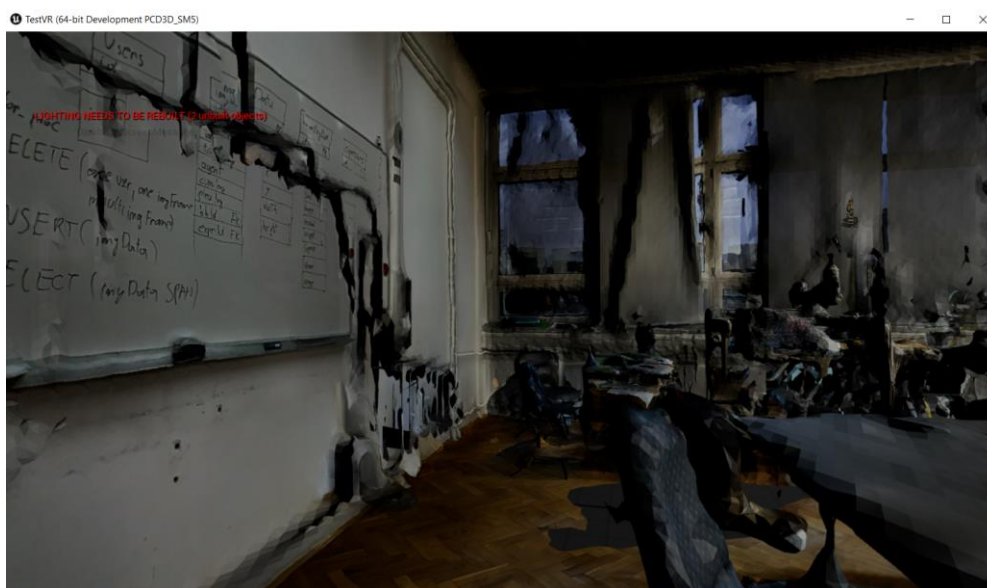
**Slika 65. Postavljena početna pozicija korisnika**

Nakon pokretanja pogleda na VR naočala možda visina na kojoj će se korisnik stvoriti neće biti dobra, u tom slučaju je potrebno spustiti ili podići koordinatni sustav sa slike 65. ovisno o potrebi korisnika.



**Slika 66. Povezivanje virtualnog okruženja sa VR naočalama**

Pokretanje na VR naočalama se izvršava tako što se u izborniku *Platforms* odabere model VR naočala koji je spojen na radnu stanicu i nakon toga pritisne gumb pokreni (engl. *Play*), ikona je izgleda poput klasičnog izgleda tipke pokreni uz dodatak naočala.



**Slika 67. Početni prikaz na VR naočalama**

## 7. ZAKLJUČAK

Ovim radom prikazana je strukturna analiza dvije metode izuzimanja oblaka točaka, SLAM i LiDAR skeniranje prostorije Laboratorija za projektiranje izradbenih i montažnih sustava. Tokom rada navedene su sve prednosti i nedostaci spomenutih postupaka uz konačnu analizu i odabir metode sa boljim rezultatima. Rezultat postupka je 3D model snimanog prostora, a procjena rezultata je subjektivna na temelju umjetničkog dojma vjernijeg prikaza stvarnog prostora.

Zaključeno je da je tehnika LiDAR skeniranja upotrebom sustava kamere iPad Pro tableta daleko ispred SLAM pristupa i da je superiornija metoda sa kojom se SLAM ne može nositi. Rezultati su jednostavno puno čišći, jasniji i detaljniji što je i cilj ovoga rada, kreirati 3D prikaz prostora Laboratorija za projektiranje izradbenih i montažnih sustava i integrirati ga u virtualno okruženje.

Negativna strana ovoga pristupa je što aplikacija koja se koristi u svrhu LiDAR skeniranja na iPad-u ima ograničenje u vidu memorije koju snimka može zauzeti što je i dovelo do pojave rupa u nekim dijelovima modela koje su kasnije otklonjene upotrebom Blender-a. U slučaju da nema ovog ograničenja, prilikom snimanja bi se svakom pojedinom objektu u prostoru snimatelj mogao više posvetiti te bi svaki dio prostorije ispao još detaljnije nego što trenutno jest. Također, na App Store-u se nalazi i profesionalna verzija ove aplikacija koja vjerojatno nema ovo ograničenje.

Evaluacija kvalitete modela se vršila isprobavanjem modela u virtualnom okruženju upotrebom VR naočala koja je pokazala impresivne rezultate. Na nekim mjestima je doživljaj prostora bio izrazito realan, kao npr. stajanjem pred pločom sa koje se moglo čitati ono što je bilo napisano u trenutku kada se prostor snimao.

Završetkom ovoga rada, model u virtualnom okruženju ostaje kao temelj za daljnji razvoj i poboljšanje modela u kojem bi se moglo naći mjesta za virtualne objekte, digitalne blizance i sve što ljudskoj mašti padne na pamet se može implementirati u ovaj 3D model prostora koji je vjerna replika onog stvarnog.

## LITERATURA

- [1] [https://hr.wikipedia.org/wiki/Virtualni\\_svijet](https://hr.wikipedia.org/wiki/Virtualni_svijet), Zadnje pristupljeno: studeni 2022.
- [2] [https://en.wikipedia.org/wiki/Stereo\\_camera#:~:text=The%20distance%20between%20the%20lenses,produces%20more%20extreme%203%2Ddimensionality](https://en.wikipedia.org/wiki/Stereo_camera#:~:text=The%20distance%20between%20the%20lenses,produces%20more%20extreme%203%2Ddimensionality), Zadnje pristupljeno: studeni 2022.
- [3] <https://www.intelrealsense.com/depth-camera-d435/>, Zadnje pristupljeno: studeni 2022.
- [4] Intel® RealSense™ Product Family D400 Series, Revision 013, April 2022
- [5] [https://en.wikipedia.org/wiki/Kinect#Kinect\\_for\\_Xbox\\_360\\_\(2010\)](https://en.wikipedia.org/wiki/Kinect#Kinect_for_Xbox_360_(2010)), Zadnje pristupljeno: studeni 2022.
- [6] <https://www.apple.com/ipad-pro/>, Zadnje pristupljeno: studeni 2022.
- [7] <https://www.cnet.com/tech/mobile/lidar-is-one-of-the-iphone-ipad-coolest-tricks-its-only-getting-better/#:~:text=Lidar%20stands%20for%20light%20detection,flight%2C%20of%20the%20light%20pulse.>, Zadnje pristupljeno: studeni, 2022.
- [8] Gregor Luetzenburg, Aart Kroon & AndersA. Bjørk. Evaluation of the Apple iPhone 12 Pro LiDAR for an Application in Geosciences
- [9] Alex Teichman, Stephen Miller, Sebastian Thrun. Unsupervised intrinsic calibration of depth sensors via SLAM
- [10] <http://introlab.github.io/rtabmap/>, Zadnje pristupljeno: studeni 2022.
- [11] Mathieu Labbe, Francois Michaud. Appearance-Based Loop Closure Detection for Online Large-Scale and Long-Term Operation
- [12] <https://www.meshlab.net/>, Zadnje pristupljeno: studeni 2022.
- [13] Michael Kazhdan, Hugues Hoppe. Screened Poisson Surface Reconstruction
- [14] <https://3dscannerapp.com/>, Zadnje pristupljeno: studeni 2022.
- [15] <https://www.blender.org/>, Zadnje pristupljeno: studeni 2022.