

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

Evidenčné číslo: FEI-104376-104245

**REKONŠTRUKCIA POVRCHU Z MRAČIEN BODOV  
DIPLOMOVÁ PRÁCA**

**2024**

**Bc. Matúš Jankech**

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

Evidenčné číslo: FEI-104376-104245

**REKONŠTRUKCIA POVRCHU Z MRAČIEN BODOV  
DIPLOMOVÁ PRÁCA**

Študijný program: Robotika a kybernetika

Názov študijného odboru: kybernetika

Školiace pracovisko: Ústav robotiky a kybernetiky

Vedúci záverečnej práce: Ing. Martin Dekan, PhD.

**Bratislava 2024**

**Bc. Matúš Jankech**



## ZADANIE DIPLOMOVEJ PRÁCE

Študent: **Bc. Matúš Jankech**

ID študenta: 104245

Študijný program: robotika a kybernetika

Študijný odbor: kybernetika

Vedúci práce: Ing. Martin Dekan, PhD.

Vedúci pracoviska: prof. Ing. František Duchoň, PhD.

Miesto vypracovania: Ústav robotiky a kybernetiky

Názov práce: **Rekonštrukcia povrchu z mračien bodov**

Jazyk, v ktorom sa práca vypracuje: slovenský jazyk

Specifikácia zadania:

Výsledkom geodetického mapovania pomocou laserového diaľkomeru vzniká mračno bodov vysokej hustoty. Takéto mračno bodov môže dosahovať až desiatky gigabytov priestoru, čo značne komplikuje prácu s ním. Často používaným riešením je jeho podvzorkovanie, čo ale môže spôsobiť stratu detailov. Jedným z riešení je navrhnuté adaptívne prevzorkovanie, ktoré v prípade členitých povrchov zachová plné rozlíšenie mračna bodov a v prípade plôch nastáva výraznejšie vynechanie bodov. Aby nevyzeralo novovzniknuté mračno bodov nespojité, tak sa vytvorí z tohto mračna takzvaný mesh, ktorý reprezentuje povrch nasnímaných objektov. Cieľom tejto práce je navrhnuté práve takéto podvzorkovanie a rekonštrukciu povrchu, ktorá výrazne zníži pamäťové nároky na uchovávanie bez straty detailov. Práca bude vypracovaná v spolupráci so spoločnosťou Lidarett.

### Úlohy

1. Naštudujte si algoritmy spracovania mračien bodov.
2. Naštudujte si algoritmy rekonštrukcie povrchu.
3. Navrhnite a implementujte riešenie prevzorkovania a rekonštrukcie povrchu z mračna bodov.
4. Navrhnuté riešenie overte a spracujte do práce.
5. Vyhodnoťte dosiahnuté výsledky.

Termín odovzdania diplomovej práce: 10. 05. 2024  
Dátum schválenia zadania diplomovej práce: 15. 01. 2024  
Zadanie diplomovej práce schválil: prof. Ing. Jarmila Pavlovičová, PhD. – garantka študijného programu

# SÚHRN

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Študijný program:	Robotika a kybernetika
Autor:	Bc. Matúš Jankech
Diplomová práca:	Rekonštrukcia povrchu z mračien bodov
Vedúci záverečnej práce:	Ing. Martin Dekan, PhD.
Miesto a rok predloženia práce:	Bratislava 2024

Diplomovej práca sa zaobrá problematikou mračien bodov a ich efektívnym uchovaním prostredníctvom rekonštrukcie povrchu. Úvod práce je venovaný preskúmaniu existujúcich metód a algoritmov na spracovanie mračien bodov, ako aj na rekonštrukciu povrchu. Cieľom práce je zníženie pamäťových nárokov na uchovanie 3D informácií o nasnímanom priestore, pričom sú zachované detaily. Naša metóda začína očistením mračna bodov od odľahlých bodov, čím zachová iba relevantné informácie. Pokračuje separáciou bodov zeme a objektov, zhlukovaním bodov do supervoxellov a ich následnou segmentáciou, ktorá umožní individuálny prístup ku jednotlivým objektom. Ďalším krokom je adaptívne podvzorkovanie bodov objektov, po ktorom je vykonaná samotná rekonštrukcia povrch pomocou Poissonovej rekonštrukčnej metódy. Na záver je práca overená na niekoľkých súboroch dát získaných v reálnych podmienkach, pričom dosahuje vysoké zníženie pamäťových nárokov a súčasne zachovanie detailov.

Kľúčové slová: mračno bodov, rekonštrukcia povrchu, segmentácia, podvzorkovanie

# ABSTRACT

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA  
FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION TECHNOLOGY

Study Programme:	Robotics and cybernetics
Author:	Bc. Matúš Jankech
Master's thesis:	Surface reconstruction from point clouds
Supervisor:	Ing. Martin Dekan, PhD.
Place and year of submission:	Bratislava 2024

The diploma thesis deals with the issue of point clouds and their efficient storage through surface reconstruction. The introduction of the thesis is dedicated to exploring existing methods and algorithms for point cloud processing, as well as surface reconstruction. The aim of the thesis is to reduce the memory requirements for storing 3D information about the scanned space, while preserving details. Our method begins by cleaning the point cloud from outlier points, retaining only relevant information. It continues with the separation of ground and object points, clustering points into supervoxels, and their subsequent segmentation, which allows individual access to each object. The next step is adaptive subsampling of object points, followed by the surface reconstruction itself using the Poisson reconstruction method. At the end, the work has been validated on several datasets obtained under real conditions, achieving a significant reduction in memory requirements while preserving details.

Keywords: point cloud, surface reconstruction, segmentation, subsampling

## **Pod'akovanie**

Týmto by som sa chcel pod'akovat' svojmu vedúcemu práce Ing. Martinovi Dekanovi, PhD. za jeho cenné rady a odborný prístup, ktorý ponúkol pri tvorbe tejto práce.

# Obsah

<b>Úvod</b>	<b>1</b>
<b>1 Mračná bodov a ich spracovanie</b>	<b>2</b>
1.1 Dátové štruktúry pre spracovanie mračna bodov . . . . .	3
1.1.1 Voxelová mriežka . . . . .	3
1.1.2 K-d strom . . . . .	4
1.1.3 Oktálový strom . . . . .	5
1.2 Filtrácia mračna bodov . . . . .	6
1.2.1 Štatisticky založené metódy . . . . .	6
1.2.2 Metódy založené na susednosti . . . . .	7
1.3 Podvzorkovanie mračna bodov . . . . .	8
1.3.1 Bežné podvzorkovacie metódy . . . . .	8
1.3.2 Adaptívne podvzorkovacie metódy . . . . .	10
1.4 Segmentácia . . . . .	11
1.4.1 Segmentačné metódy . . . . .	11
1.4.2 DBSCAN . . . . .	13
1.4.3 K-means . . . . .	13
1.4.4 RANSAC . . . . .	14
<b>2 Rekonštrukcia a spracovanie povrchu</b>	<b>16</b>
2.1 Metódy pre rekonštrukciu povrchu . . . . .	17
2.1.1 Marching cubes . . . . .	17
2.1.2 Poisson . . . . .	18
2.1.3 Ball pivoting . . . . .	19
2.1.4 MLS . . . . .	19
2.1.5 MPU . . . . .	21
2.2 Metódy pre zjednodušenie povrchu . . . . .	22
2.2.1 Lokálne metódy . . . . .	22
2.2.2 Globálne metódy . . . . .	23
<b>3 Špecifikácia cieľov</b>	<b>25</b>
<b>4 Použité súbory dát</b>	<b>26</b>
<b>5 Realizácia riešenia</b>	<b>29</b>

5.1	Odstránenie odľahlých bodov . . . . .	29
5.2	Separácia bodov zeme a objektov . . . . .	31
5.3	Zhlukovanie bodov objektov do supervoxelov . . . . .	33
5.4	Segmentácia mračna bodov . . . . .	35
5.5	Podvzorkovanie mračna bodov . . . . .	37
5.5.1	Podvzorkovanie bodov objektov . . . . .	37
5.5.2	Podvzorkovanie bodov zeme . . . . .	39
5.6	Rekonštrukcia povrchu . . . . .	40
5.6.1	Výber metódy . . . . .	40
5.6.2	Odstránenie nedostatkov . . . . .	42
5.7	Textúrovanie povrchu . . . . .	45
<b>6</b>	<b>Overenie riešenia</b>	<b>49</b>
6.1	Overenie separácia bodov zeme . . . . .	49
6.2	Overenie generovania supervoxelov a segmentácie . . . . .	53
6.3	Overenie adaptívneho podvzorkovania . . . . .	55
6.4	Overenie rekonštrukcie povrchu . . . . .	58
6.5	Overenie textúrovania povrchu . . . . .	61
<b>Záver</b>		<b>63</b>
<b>Zoznam použitej literatúry</b>		<b>64</b>
<b>Prílohy</b>		<b>68</b>
<b>A Technická dokumentácia</b>		<b>69</b>
<b>B Odkaz na zdrojový kód</b>		<b>70</b>
<b>C Ukážka rekonštruovaného povrchu</b>		<b>71</b>

# Zoznam obrázkov a tabuliek

Obrázok 1.1	Ukážka mračna bodov . . . . .	2
Obrázok 1.2	Voxelová reprezentácia mračna bodov - (a) Originálne mračno bodov (b) Voxelová reprezentácia [1] . . . . .	3
Obrázok 1.3	Reprezentácia k-d stromu v 2D priestore [5] . . . . .	4
Obrázok 1.4	Tradičná reprezentácia oktálového stromu [7] . . . . .	5
Obrázok 1.5	Optimalizovaná reprezentácia oktálového stromu (Bunky obsahujúce body označené šedou farbou) [6] . . . . .	6
Obrázok 1.6	Princíp filtrovania odľahlých bodov pomocou SOR [10] . . . . .	7
Obrázok 1.7	Princíp filtrovania odľahlých bodov pomocou ROR v 2D . . . . .	8
Obrázok 1.8	Ukážka rôznych metód podvzorkovania [11] . . . . .	9
Obrázok 1.9	Ukážka klasifikácie dôležitých bodov pomocou metódy založenej na rovinnosti [12] . . . . .	10
Obrázok 1.10	Rozdelenie segmentačných metód . . . . .	11
Obrázok 1.11	Porovnanie segmentácie pomocou DBSCAN a K-means algoritmu [18] . . . . .	14
Obrázok 1.12	Postup RANSAC algoritmu pre odhad parametrov modelu priamky .	15
Obrázok 2.1	Ukážka rekonštrukcie povrchu - (a) mračno bodov (b) zrekonštrovaný povrch . . . . .	16
Obrázok 2.2	Rôzne možnosti konfigurácie voxelov [22] . . . . .	17
Obrázok 2.3	Ukážka Poissonovej rekonštrukcie v 2D [23] . . . . .	18
Obrázok 2.4	Ukážka ball-pivoting rekonštrukcie v 2D - (a) Ideálny prípad (b) mála a neuniformná hustota bodov (c) príliš veľké zakrivenie (väčšie ako $1/\rho$ ) [23] . . . . .	19
Obrázok 2.5	Ukážka princípu pre viacero variantov MLS v 2D. Miestne aproxi-mácie vypočítané pre hodnotiaci bod $x$ sú znázornené zelenou farbou. Oranžové krivky zodpovedajú rekonštruovaným izo-kontúram. [25] . . . . .	20
Obrázok 2.6	Ukážka rekonštrukcie pomocou MPU implicit. Ľavá časť modelu je ofarbená podľa hĺbky rozdelenia, s tým že stúpa od modrej až po červenú. Modely v pozadí sú rekonštruované so stúpajúcou chybou v odhade. [26] . . . . .	21
Obrázok 2.7	Ukážka decimácie vrcholu v 2D . . . . .	22

Obrázok 2.8	Ukážka kolapsu strany v 2D . . . . .	23
Obrázok 2.9	Ukážka zhlukovania vrcholov v 2D. Originálne vrcholy (čierne), po úprave (modré), pre viditeľnosť posunuté od stredu mriežky . . . . .	24
Obrázok 4.1	Mračno bodov mestskej časti s bytovkami - <i>SD1</i> (Optech CL-360) . . . . .	26
Obrázok 4.2	Mračno bodov mestskej časti s kruhovým odjazdom - <i>SD2</i> (Optech CL-360) . . . . .	27
Obrázok 4.3	Mračno bodov parkoviska pri obchodoch - <i>SD3</i> (Hesai XT-32) . . . . .	27
Obrázok 4.4	Mračno bodov ulice so trolejbusovou zastávkou - <i>SD4</i> (Hesai XT-32) . . . . .	28
Obrázok 4.5	Mračno bodov dedinskej časti - <i>SD5</i> (Hesai XT-32) . . . . .	28
Obrázok 5.1	Výsledné mračná bodov (pôvodná farba) po odstránení odľahlých bodov (biele farba) pomocou SOR metódy ( <i>SD1</i> hore, <i>SD2</i> dole) . . . . .	30
Obrázok 5.2	Mračná bodov objektov (pôvodná farba) po separácii bodov zeme (biele farba) pomocou approximácie morfologického filtra ( <i>SD1</i> hore, <i>SD2</i> dole) . . . . .	32
Obrázok 5.3	Mračná bodov objektov zoskupené do supervoxelov rôznej farby ( <i>SD1</i> hore, <i>SD2</i> dole) . . . . .	34
Obrázok 5.4	Mračná bodov segmentované na jednotlivé objekty, označené rôznou farbou ( <i>SD1</i> hore, <i>SD2</i> dole) . . . . .	36
Obrázok 5.5	Ukážka princípu fungovania DoN na zakrivenom povrchu [31] . . . . .	37
Obrázok 5.6	(a) Mračno diferencií normál (b) Klasifikované mračno na základe dôležitosti bodov v vzostupnom poradí červené, zelené, modré a najdôležitejšie tyrkysové (c) Výsledné podvzorkované mračno bodov objektov ( <i>SD1</i> ) . . . . .	38
Obrázok 5.7	Mračno bodov zeme pred (hore) a po (dole) podvzorkovaní. ( <i>SD1</i> ) . . . . .	39
Obrázok 5.8	Porovnanie rôznych metód rekonštrukcie povrchu na objekte auta - (a) Podvzorkované mračno bodov (b) GPT (c) Marching cubes (d) Poissonová rekonštrukcia ( <i>SD1</i> ) . . . . .	41
Obrázok 5.9	Podvzorkované mračno bodov (hore), plocha s nadbytočnými pologónmi (dole) ( <i>SD1</i> ) . . . . .	42
Obrázok 5.10	Rekonštrukcia celého povrchu naraz (hore), rekonštrukcia individuálnych objektov (dole) ( <i>SD1</i> ) . . . . .	43
Obrázok 5.11	(a) Nesprávny odhad normál (b) Rekonštrukcia zo nesprávneho odhadu normál (c) Správny odhad normál (d) Rekonštrukcia zo správneho odhadu normál ( <i>SD1</i> ) . . . . .	44

Obrázok 5.12	Porovnanie originálneho objektu (vľavo) so zdecimovaným (vpravo) pri znížení pamäťovej náročnosti o 70% . . . . .	44
Obrázok 5.13	(a) Textúrovanie povrchu z najbližšej kamery (b) Textúrovanie z najblížej kamery, iba ak kamera skutočne vidí povrch (červené polygóny nevidené žiadou kamerou) (c) Odstránenie statickej prekážky auta na ceste ( <i>SD2</i> ) . . . . .	47
Obrázok 5.14	Ukážka kamerovej fotky s označeným autom, pomocou zeleného pologónu ( <i>SD2</i> ) . . . . .	48
Obrázok 6.1	Body cesty klasifikované ako body objektov ( <i>SD3</i> ) . . . . .	49
Obrázok 6.2	Body cesty klasifikované ako body objektov (detail šumu) ( <i>SD3</i> ) . .	50
Obrázok 6.3	Odolnosť morfologickej separácie bodov zeme na šum - (a) Mračno bodov objektov (pôvodná farba), body zeme (biela farba) (b) Body objektov obsahujú aj body zeme (c) Body zeme obsahujú strechy budov ( <i>SD5</i> ) . . . . .	51
Obrázok 6.4	Body kolies auta vyhodnotená ako body zeme ( <i>SD1</i> ) . . . . .	52
Obrázok 6.5	Málo hustne body áut vyhodnotené ako body zeme ( <i>SD1</i> ) . . . . .	52
Obrázok 6.6	Prechod supervoxelov medzi bodmi stĺpu a bodmi stromu ( <i>SD4</i> ) . .	53
Obrázok 6.7	Body viacerých objektov segmentované ako jeden objekt ( <i>SD4</i> ) . .	54
Obrázok 6.8	Segmentovanie bodov veľkej budovy na viacero objektov ( <i>SD3</i> ) . .	54
Obrázok 6.9	Segmentovanie bodov stromu a auta na viacero objektov ( <i>SD3</i> ) . .	55
Obrázok 6.10	Mračná bodov a ich príslušný povrch - (a) Nepodvzorkované mračno bodov (b) Adaptívne podvzrokované mračno bodov (c) Rovnomerne podvzorkované mračno bodov pomocou voxelovej mriežky ( <i>SD1</i> ) .	56
Obrázok 6.11	Mračná bodov a ich príslušný povrch - (a) Nepodvzorkované mračno bodov (b) Adaptívne podvzrokované mračno bodov (c) Rovnomerne podvzorkované mračno bodov pomocou voxelovej mriežky ( <i>SD3</i> ) .	57
Obrázok 6.12	Vznik drsného povrchu pri nesprávnej separácii bodov zeme (Obr. 6.3) a rekonštrukcií zo zašumených dát ( <i>SD5</i> ) . . . . .	58
Obrázok 6.13	Hladký prechod medzi viacerými povrchmi zeme, aj napriek nesprávnej separácii zeme (Obr. 6.1) ( <i>SD3</i> ) . . . . .	59
Obrázok 6.14	Prepojenie viacerých objektov spolu, na základe zlej segmentácie (Obr. 6.7) ( <i>SD4</i> ) . . . . .	59
Obrázok 6.15	Výskyt trhlín na základe zlého odhadu normál ( <i>SD3</i> ) . . . . .	60

Obrázok 6.16	Problém Poissonovej rekonštrukčnej metódy s rekonštrukciou úzkych objektov (trolejbusové vedenie) ( <i>SD4</i> ) . . . . .	60
Obrázok 6.17	Problém Poissonovej rekonštrukčnej metódy s dôveryhodnou rekonštrukciou stromov ( <i>SD1</i> ) . . . . .	61
Obrázok 6.18	(a) Mozaikové textúrovanie kvôli zlej reprezentácií polygónov pomocou oktálového stromu (sledovanie lúčov) (b) Ostrý prechod farby medzi viacerými snímkami (c) Neodstránenie dynamických objektov (auto na ceste) ( <i>SD2</i> ) . . . . .	62
Tabuľka 5.1	Porovnanie veľkosti pôvodných a filtrovaných mračien bodov, pre jednotlivé súbory dát . . . . .	31
Tabuľka 5.2	Porovnanie pomeru bodov zeme a objektov, pre jednotlivé súbory dát	33
Tabuľka 5.3	Porovnanie zmenšenia počtu bodov pred (vid'. Tab. 5.2) a po podvzorkovaní mračien bodov . . . . .	40
Tabuľka 5.4	Výsledné hodnoty rekonštruovaných povrchov pre jednotlive súbory dát . . . . .	45

# Zoznam skratiek

<b>CPC</b>	Constrained Planar Cuts
<b>DBSCAN</b>	Density-Based Spatial Clustering of Applications with Noise
<b>DoN</b>	Difference of Normals
<b>GPT</b>	Greedy Projection Triangulation
<b>KNN</b>	K-nearest neighbors
<b>LIDAR</b>	Light detection and ranging
<b>MLS</b>	Moving least squares
<b>MPU</b>	Multi-level Partition of Unity
<b>PCL</b>	Point cloud library
<b>RANSAC</b>	Random Sample Consensus
<b>ROR</b>	Radius outlier removal
<b>SOR</b>	Statistical outlier removal
<b>VCCS</b>	Voxel Cloud Connectivity Segmentation

# Úvod

V posledných rokoch prudký nárast vývoja výpočtovej techniky a cenové sprístupnenie technológií, viedlo ku zvýšeniu využívania 3D dát vo forme mračien bodov, ktoré sú často získavané pomocou techník, ako sú LIDAR, fotogrametria alebo stereo kamera. Mračná bodov dokážu veľmi presne zachytávať geometrickú reprezentáciu rôznorodých prostredí, čo ich robí dobrou vol'bou v oblastiach, ako sú robotika, mapovanie prostredí a medzi inými aj vo výrobe.

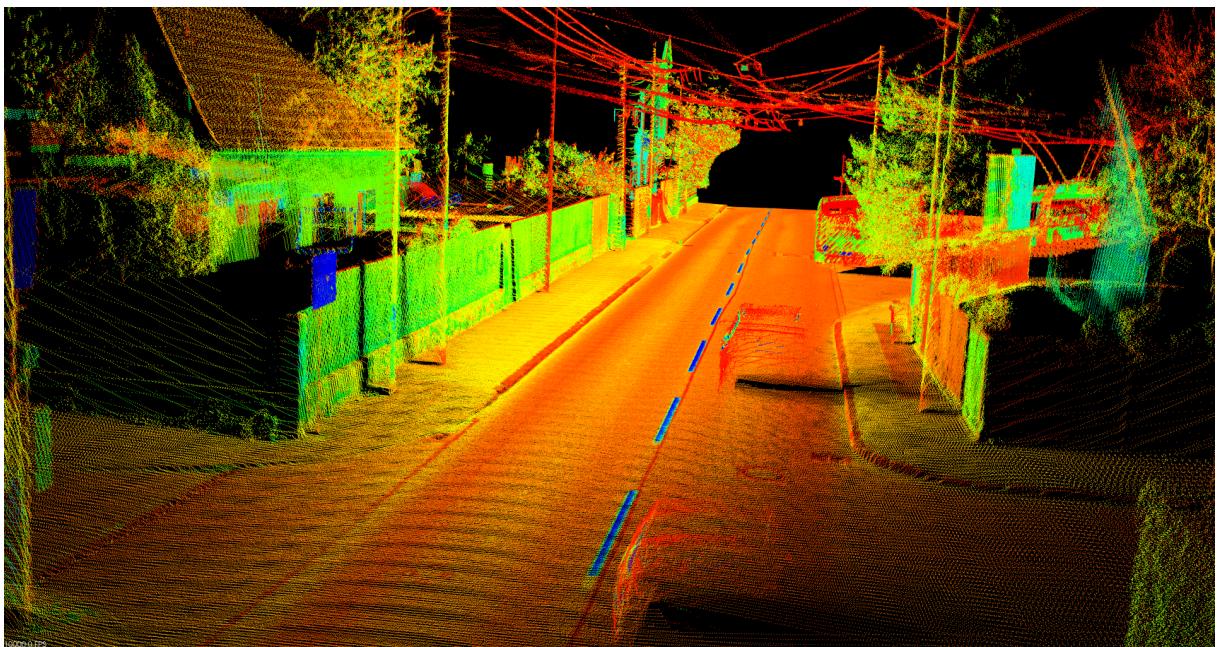
Mračná bodov často obsahujú milióny bodov a v neupravenom stave predstavujú výzvu z hľadiska efektívneho ukladania, spracovania a prístupu ku požadovanej časti priestoru. Jedným spôsobom pre odstránenie tohto problému, je zníženie celkového počtu bodov pomocou podvzorkovania, nakoľko vo väčšine prípadov bývajú mračná bodov príliš husté a tú istú informáciu zachytávajú niekol'ko krát. Týmto spôsobom znížime pamäťové a výpočtové nároky, ale pri bežných metódach sa stretнемe bud' s nízkou úrovňou podvzorkovania alebo stratou detailov. Je preto potrebné navrhnúť adaptívny prístup podvzorkovania, ktorý bude brat' do úvahy dôležitosť jednotlivých bodov.

Samotné mračná bodov predstavujú iba jednotlivé body povrchu a po aplikovaní podvzorkovania, môže dôjsť ku problematickej reprezentácii pôvodných objektov. Je preto potrebné, vytvoriť z mračna bodov povrch v podobe tzv. mesh-u, ktorý bude reprezentovať prepojenia bodov pomocou sady polygónov, čím odstráni vzniknuté diery. Týmto spôsobom sa taktiež vytvorí efektívnejšia štruktúra, ktorá pri menšej pamäťovej náročnosti bude zachytávať ten istý povrch.

V tejto práci sa preto budeme zaoberať preskúmaním existujúcich metód a algoritmov pre spracovanie mračien bodov, ako aj rekonštrukciu povrchu. Následne navrhнемe naše riešenie, ktoré spracuje mračno bodov, vytvorí reprezentáciu jeho povrchu a zároveň zníži celkové pamäťové nároky.

# 1 Mračná bodov a ich spracovanie

Mračno bodov je kolekcia bodov v 3D priestore, kde každý bod je reprezentovaný pomocou XYZ súradníc, okrem ktorých môže obsahovať doplňujúcu informáciu o ich farbe, intenzite a v prípade potreby, aj ďalších špecifických atribútoch. Spájaním veľkého množstva týchto bodov, do jedného súboru zo spoločným súradnicovým systémom, je možné zachytiť úplnú geometriu akéhokoľvek 3D objektu. Vďaka schopnosti poskytovať priestorové súradnice povrchov objektov sa mračná bodov stali preferovaným typom dát pre 3D vizualizácia rôznorodých prostredí. Mračna bodov sa osvedčili, ako jedny z najvhodnejších zdrojov údajov, najmä pri mapovaní rozsiahlych mestských prostredí, pretože získané 3D body obsahujú priame priestorové súradnice geometrických povrchov, čo výrazne zjednodušuje procesy modelovania povrchov a geometrickej rekonštrukcie. [1]



Obr. 1.1: Ukážka mračna bodov

Mračna bodov d'alej často nachádzajú využitie v rôznych oblastiach, ako je počítačové videnie, modelovanie terénu, robotika či v rámci medicíny. Spôsoby získavania mračna bodov zahŕňajú rôzne metódy a techniky, pričom každá ponúka jedinečné výhody. Medzi najpopulárnejšie prístupy v oblasti robotiky a priemyslu patrí využitie LIDAR-u, stereo kamery a hĺbkovej kamery.

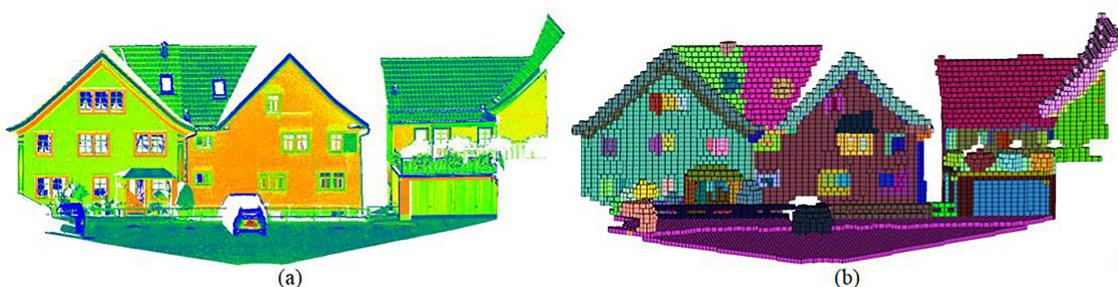
## 1.1 Dátové štruktúry pre spracovanie mračna bodov

Neupravené mračná bodov sú obvykle neštruktúrované, majú nerovnomernú hustotu bodov a navyše, ich kvalita môže byť odlišná v rôznych častiach, čo býva považované za nevhodné pre ich adekvátnu použiteľnosť. Okrem toho pri mračnách bodov získaných z viacerých zdrojov je zvyčajne veľké množstvo prekrývajúcich bodov, čo zbytočne zvyšuje pamäťové a výpočtové nároky. [1] [2]

Pri mnohých algoritmoch je kritické získať informácie o susedných bodoch, avšak v prípade veľkých a neštruktúrovaných mračien je táto úloha časovo a výpočtovo náročná. Preto pre tento účel existujú dátové štruktúry, ktoré umožňujú organizáciu mračna bodov do efektívnejších usporiadania.

### 1.1.1 Voxelová mriežka

Voxelová mriežka (z ang. Voxel grid) je zložená z voxelov, ktoré sa dajú charakterizovať, ako elementárne bloky 3D priestoru. Môžeme si ich predstaviť, ako 3D analógiu pixelov. Majú súčasť tvar kvádrov so stenami, rohmi a hranami, ale dátovo nie sú nikdy uložené ako celé kvádre. Namiesto toho sa vo voxelovej reprezentácii používa bud' stredný bod alebo rohové body (t.j. vrcholy) kvádra pre jeho celú reprezentáciu. Keď sa dva voxelovia dotýkajú stenou, rohom alebo hranou, sú považované za susedné. Celé mračno bodov je takto rozdelené na osobitné voxelovia, pričom body vo vnútri daného voxelu sú reprezentované týmto voxelom a zároveň daný voxel je reprezentovaný týmto bodmi. To znamená, že každý voxel je charakterizovaný pomocou počtu, farby, intenzity a rozloženia daných bodov, z čoho sa dajú vyvodíť užitočné informácie. [1]



Obr. 1.2: Voxelová reprezentácia mračna bodov - (a) Originálne mračno bodov (b) Voxelová reprezentácia [1]

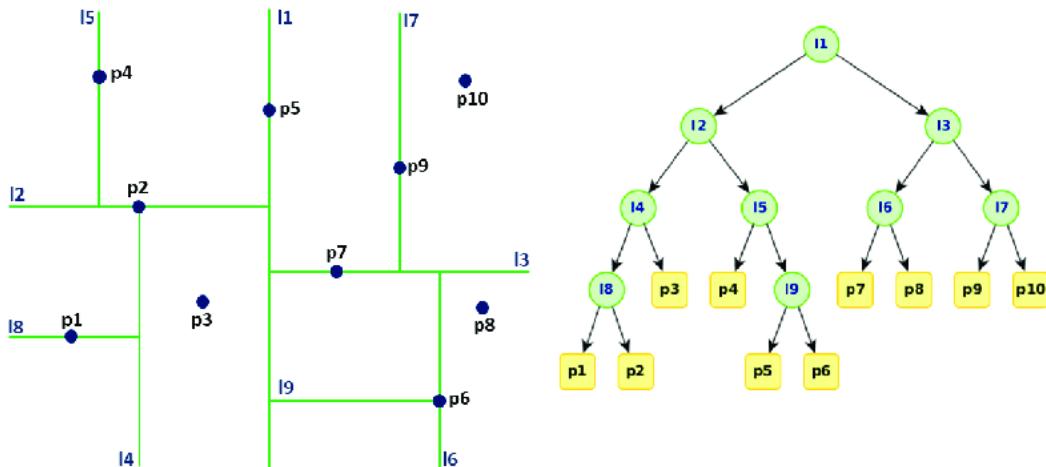
Vo všeobecnosti, pri voxelizácii mračien bodov existujú štyri kroky. Prvý krok spočíva v výpočte ohraničenia mračna bodov, ktoré definuje priestor, ktorý sa má rozdeliť. V druhom kroku je definovaný priestor rasterovaný do pravidelných buniek tvaru kvádra s určitou veľkosťou. Potom je mračno bodov segmentované na malé časti týmto bunkami. Nakoniec sú rozdelené podmnožiny bodov reprezentované voxelmi, pričom sa z týchto bodov môžu vypočítať

príslušné vlastnosti. [1]

Vďaka svojím vlastnostiam nachádzajú voxelové mriežky uplatnenie v mnohých oblastiach, ako sú predspracovanie údajov, modelovanie, registrácia, segmentácia a klasifikácia mračien bodov.

### 1.1.2 K-d strom

K-d strom (k-dimenzionálny strom), je špecializovaná dátová štruktúra používaná na efektívnu organizáciu viacozmerných dát. K-d strom funguje tak, že organizuje viacozmerný priestor do hierarchickej binárnej stromovej štruktúry. Primárne sa používa na rýchle vyhľadávanie najbližšieho suseda, čo je kritická operácia v rôznych aplikáciách, ako je robotika, LIDAR odometria a mapovanie a práca s mračnami bodov. [3] [4]



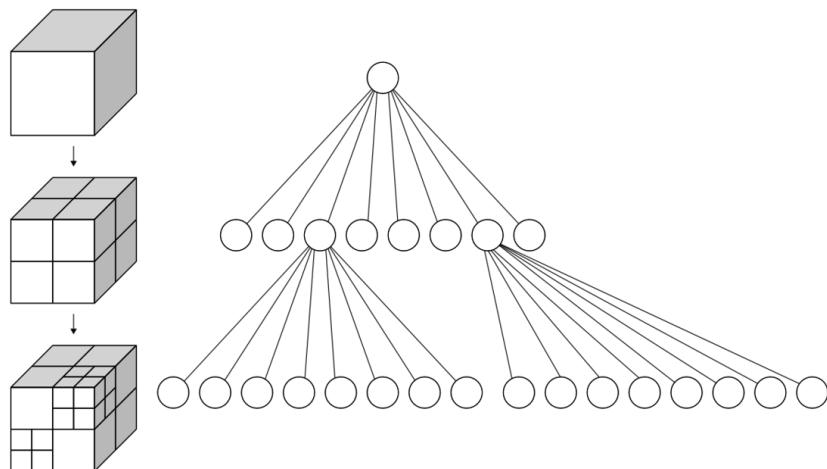
Obr. 1.3: Reprezentácia k-d stromu v 2D priestore [5]

Vytvorenie k-d stromu spočíva v postupnom delení priestoru pozdĺž všetkých dimensií, pričom ako bod delenia volíme hodnotu mediánu bodov v danom intervale. Tento spôsob vedie ku vzniku vyváženého stromu, až ku vzniku stromu, kde každá podmnožina obsahuje len jeden bod. Pri vytváraní stromu pomocou mediánu je časová zložitosť  $O(n \log_2 n)$ . Je možné použiť aj iné spôsoby rozdelenia ako medián, avšak tie nemusia vždy zabezpečiť rovnovážne vytvorenie stromu. [4]

Ako už bolo spomínané hlavným využitím k-d stromov v oblasti mračien bodov je vyhľadávanie najbližších susedov. Ak je už k-d strom vytvorený, algoritmus začína v koreni stromu a rekurzívne ním prechádza, pričom si vyberá vetvu, ktorá je najbližšie k zadanému bodu. Tento proces pokračuje, pokým sa nedosiahne listový uzol. Najbližší sused sa následne určí spätným pohybom po jednotlivých vetvách a kontrolovaním bližších susedov v ostatných vetvách. [4]

### 1.1.3 Oktálový strom

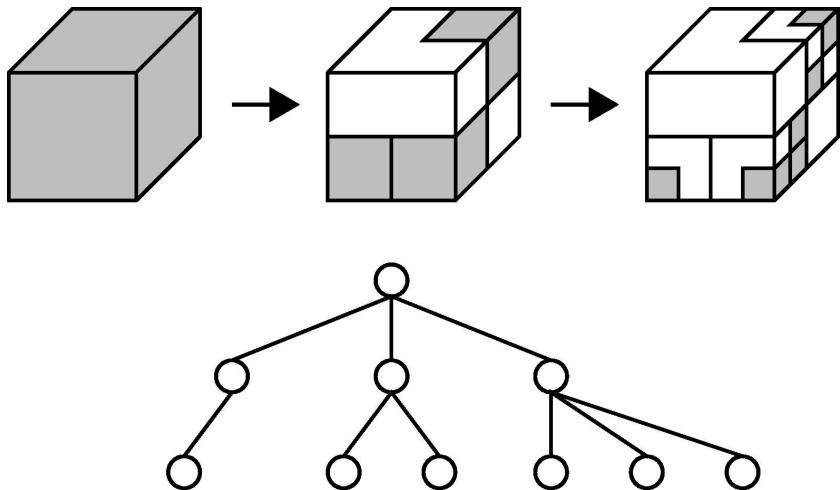
Oktálový strom je dátová štruktúra navrhnutá na manipuláciu s 3D dátami, najmä mračnami bodov získaných zo zariadení ako laserové skenery. Organizuje priestor do kociek (alebo kvádrov), pričom každá kocka môže byť rozdelená na ďalších osem menších kociek. Oktálový strom je analógiou pre kvadrátový strom využívaný v 2D priestore. Kľúčovým konceptom tejto štruktúry je efektívne ukladanie dát prostredníctvom delenia priestoru len tam, kde sú existujúce body. [6]



Obr. 1.4: Tradičná reprezentácia oktálového stromu [7]

V procese tvorby oktálového stromu dochádza ku rekurzívному deleniu 3D priestoru na osem menších kociek, ktoré pokračuje až do splnenia určených zastavovacích podmienok. Tieto podmienky môžu zahŕňať dosiahnutie maximálnej úrovne delenia, čo zodpovedá najmenšiemu rozmeru kocky, alebo môžu byť viazané na minimálny počet bodov, pri ktorom delenie pokračuje. Kocky pre ktoré bolo delenie ukončené, sú označené ako tzv. listy a obsahujú informácie o všetkých príslušných bodoch. [6]

Aj napriek tomu, že oktalový strom je považovaný za pamäťovo efektívnejšiu verziu voxellovej mriežky, má niekoľko nedostatkov, ktoré je možné vylepšiť. Optimalizovaná implementácia oktálového stromu eliminuje zbytočné dáta v pamäti a využíva ju efektívnejšie, pričom zachováva rýchle prístupové operácie. Na rozdiel od tradičných implementácií, ktoré ukladajú redundantné informácie o prázdných bunkách, táto optimalizovaná verzia minimalizuje využívanie pamäte tým, že dynamicky pridáva pamäť len pre potrebné uzly a listy. Napriek zníženiu miesta v pamäti udržiava efektívne prístupové operácie z časovou zložitosťou  $O(\log n)$ . [6]



Obr. 1.5: Optimalizovaná reprezentácia oktálového stromu (Bunky obsahujúce body označené šedou farbou) [6]

Vďaka svojím vlastnostiam a efektívnej štruktúre nachádzajú oktálové stromy využitie v mnohých oblastiach, ako sú počítačové videnie, robotika, grafika a vizualizácia. V prípade robotiky a práce s mračnami bodov sa využívajú pre nájdenie najbližšieho suseda, ale aj kompresiu dát a podvzorkovanie.

## 1.2 Filtrácia mračna bodov

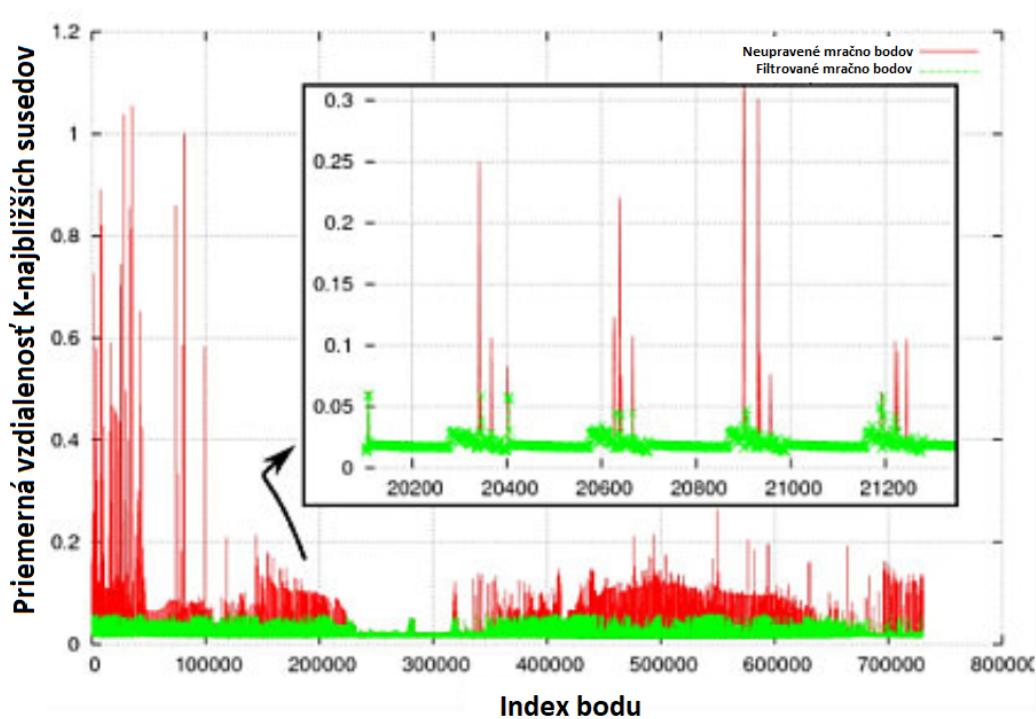
Mračná bodov získavané z rôznych typov senzorov, či už cenovo dostupných alebo drahých, často trpia prítomnosťou šumu a odľahlých hodnôt, ktoré sú výsledkom obmedzení samotných senzorov, vplyvu osvetlenia, či charakteru povrchu. Tieto nepresnosti vyžadujú využitie rôznych filtračných operácií, aby sa zabezpečila presnosť a vhodnosť pre ich ďalšie spracovanie. Prístupy filtrácie mračien bodov možno kategorizovať do niekoľkých skupín, medzi ktoré patria štatisticky založené metódy alebo metódy založené na susednosti. [8]

### 1.2.1 Štatisticky založené metódy

Štatistické metódy detekcie odľahlých hodnôt spočívajú v použití štatistických konceptov na identifikáciu výnimcočných (odlišných) hodnôt v súbore údajov. Využívajú koncepty, ako sú priemer, medián, štandardná odchýlka, kvartály a vzdialenosť, na identifikáciu hodnôt, ktoré sa výrazne odchylujú od očakávanej distribúcie, čo naznačuje, že patria medzi odľahlé hodnoty. Medzi bežné metódy detekcie odľahlých hodnôt patria metódy ako Z-skóre, Chauvenetovo kritérium, Grubbsov test a Dixonov Q-test.

Populárnu metódou pre filtrovanie mračien bodov je SOR (Statistical outlier removal). Metóda spočíva vo vypočítaní priemernej vzdialenosť medzi každým bodom a jeho  $K$  najbližšími susedmi pomocou KNN algoritmu. Predpokladajúc, že tieto vzdialenosť nasledujú nor-

málne rozdelenie, algoritmus aplikuje sigma pravidlo na základe zvoleného násobiteľa štandardnej odchýlky  $N$ . A teda, ak priemerná vzdialenosť bodu a jeho susedov presahuje  $N$  štandardných odchýlok priemernej vzdialosti celého súboru, potom je bod považovaný za odľahlú hodnotu. Úpravou parametrov, ako je počet susedov  $K$  a násobiteľ štandardnej odchýlky  $N$ , je možné jemne nastaviť citlivosť algoritmu na odľahlé hodnoty. [9]

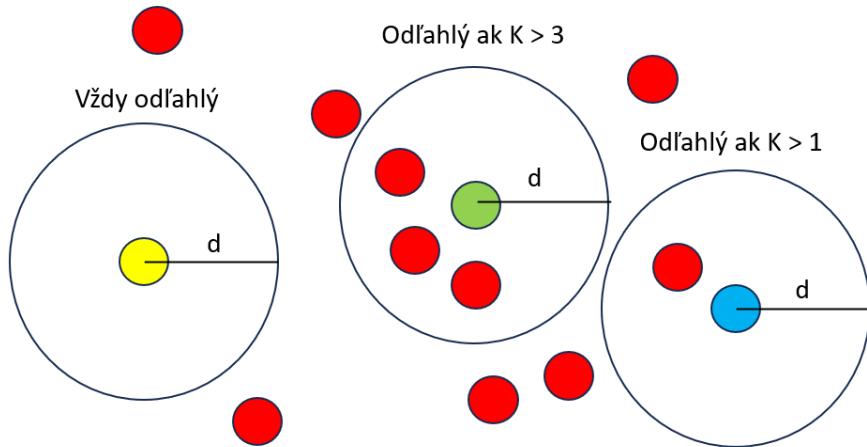


Obr. 1.6: Princíp filtrovania odľahlých bodov pomocou SOR [10]

### 1.2.2 Metódy založené na susednosti

Metódy filtrovania založené na susedstve určujú vhodnosť bodu na základe jeho podobnosti s okolitými bodmi, čo výrazne ovplyvňuje účinnosť a efektivitu filtračného prístupu. Podobnosť medzi susednými bodmi môže byť definovaná ich vzájomnou vzdialenosťou, polohou, podobnosťou intenzity, farby alebo inými užitočnými vlastnosťami. [8]

Jednou zo základných metód založených na susedstve je ROR (Radius outlier removal). Táto metóda je založená na určení minimálneho počtu susedov a polomeru v ktorom budú vyladená. Ak je počet nájdených susedov menší ako  $K$ , je bod považovaný za odľahlý. Na Obr. 1.7 môžeme vidieť princíp tejto metódy.



Obr. 1.7: Princíp filtrovania odľahlých bodov pomocou ROR v 2D

### 1.3 Podvzorkovanie mračna bodov

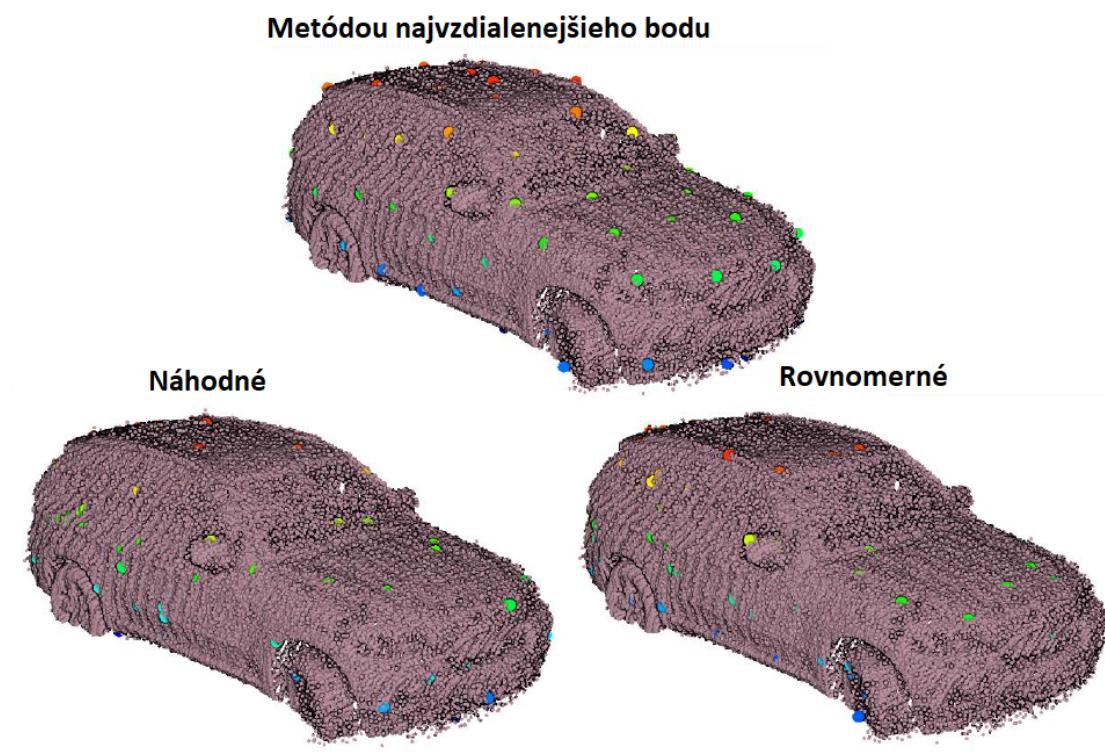
Metódy pre redukciu mračien bodov sú kľúčovým krokom pre zníženie pamäťových požiadaviek, zlepšenie výpočtovej efektívnosti a redukciu nežiaduceho šumu. Tieto metódy vo výsledku zmenšujú veľkosť mračien bodov, čo ich robí jednoduchšie pre manipuláciu, vizualizáciu a spracovanie. Snažia sa dosiahnuť rovnováhu medzi znižovaním zložitosti a uchovaním kľúčových informácií, čo je podstatná vlastnosť pre ich ďalšie spracovanie. Vieme ich rozdeliť do dvoch základných skupín, bežné metódy a adaptívne metódy pre podvzorkovanie mračien bodov.

#### 1.3.1 Bežné podvzorkovacie metódy

Bežné podvzorkovacie metódy obvykle vykonávajú rovnomerné alebo pravidelné zmenšenie počtu bodov. Tieto metódy nerozlišujú, akú úlohu napĺňajú jednotlivé body, čo môže viesť k potenciálnemu strateniu dôležitých čít alebo informácií v oblastiach, ktoré by mohli vyžadovať presnejšie podvzrokovanie.

- 1. Podvzorkovanie pomocou voxelovej mriežky** - Táto metóda rozdeľuje vstupné mračno bodov do voxelovej mriežky preddefinovaných rozmerov. Následne prechádza cez všetky vytvorené voxely, pričom ak voxel obsahuje aspoň jeden bod, bod najbližší ku tazisku alebo samotné tazisko reprezentuje všetky body vo voxelu. Vstupným parametrom tejto metódy je veľkosť strán voxelu, čo ovplyvňuje úroveň zmenšenia a zachovaných detailov. Táto metóda dosahuje rovnomerné rozdelenie bodov v mračne, ale nezachováva potrebné detaily.

2. **Náhodné podvzorkovanie** - Metóda náhodného podvzorkovania je založená na náhodnom výbere bodov z pôvodného mračna bodov, až do dosiahnutia stanoveného počtu. Napriek tomu, že táto metóda je jednoduchá a rýchla, nedosahuje rovnomerné rozdelenie bodov a nezachováva detaily, čo je obzvlášť pravda, ak má vstupné mračno bodov nerovnomernú hustotu.
3. **Rovnomerné podvzorkovanie** - Metóda spočíva v systematickom výbere bodov z pôvodného mračna s definovaným intervalom. Teda z celého mračna bodov vyberáme každý k-ty bod, čo znamená, že táto metóda je závislá od uloženého poradia bodov, a preto je vhodná, len pre určité mračná bodov. V prípade, že má vstupné mračno bodov nerovnomernú hustotu bodov, výstupné mračno bude mať nerovnomerné rozdelenie.
4. **Podvzorkovanie na základe najvzdialenejšieho bodu** - Táto metóda sa zameriava na dodržaní najväčšieho možného rozostupu bodov. Ako prvý sa vyberie náhodný bod, po čom sa iteratívne vyberá najvzdialenejší bod od aktuálneho vybraného, až dokým sa nedosiahne požadovaný počet bodov. Táto metóda dosahuje rovnomerné rozdelenie bodov a pri porovnaní s predošlými metódami, najlepšie opisuje geometriu objektov.

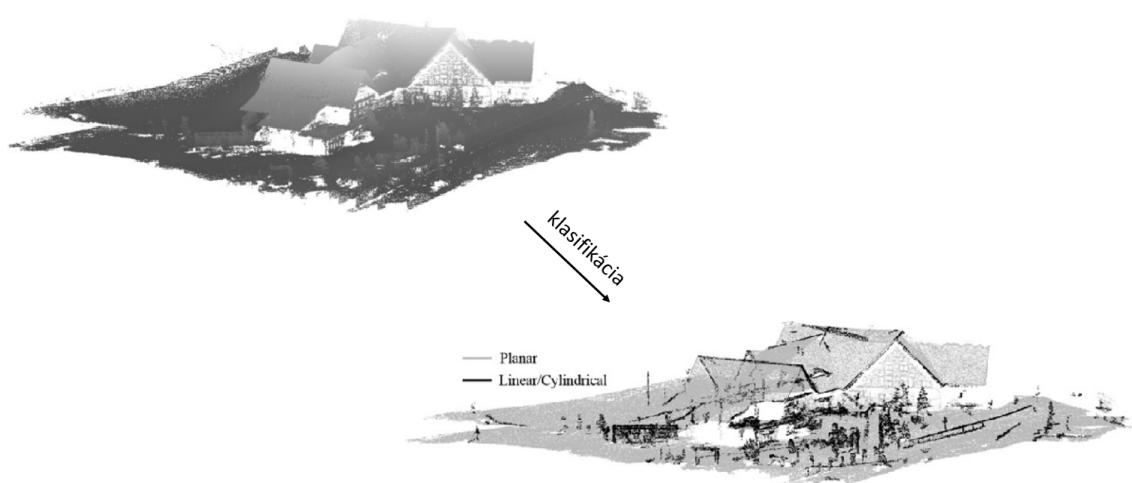


Obr. 1.8: Ukážka rôznych metód podvzorkovania [11]

### 1.3.2 Adaptívne podvzorkovacie metódy

Adaptívne metódy podvzorkovania sa snažia zachovať dôležité body, ktoré opisujú detaily a zároveň efektívne odstrániť tie menej dôležité. Výber dôležitých bodov robia na základe viacerých charakteristík, pričom zohľadňujú faktory, ako lokálna hustota bodov, zakrivenie alebo iné geometrické vlastnosti.

- 1. Podvzorkovanie na základe krivosti** - Táto metóda využíva informácie o normálach alebo krivosti povrchu na určenie ich dôležitosti. Body vo rovnejších alebo menej zakrivených oblastiach môžu byť odstránené agresívnejšie, ako tie v oblastiach s vyššou krivostou. Posudzovaním týchto geometrických vlastností je možné selektívne uchovávať body, ktoré prispievajú viac k celkovému tvaru alebo štruktúre objektu a zachovať dôležité detaily.
- 2. Podvzorkovanie na základe rovinnosti/susednosti** - Táto metóda využíva PCA (Principal Component Analysis) na charakterizáciu lokálnych susedstiev v mračnách bodov. PCA analyzuje vlastné čísla kovariančných matíc a zaraďuje tieto susedstvá do troch hlavných kategórií: rovinné, lineárne/cylindrické alebo drsné. Po kategorizácii sa metóda zameriava na odhad hustoty bodov v každom type regiónu, pričom používa matematické vzorce prispôsobené individuálne pre každú kategóriu. Následne, na základe znalostí týchto susedstiev, môže selektívne preferovať odstránenie bodov v oblastiach, ktoré vykazujú rovinný charakter a majú hustotu bodov vyššiu, ako stanovenú užívateľom, zatiaľ čo v ostatných oblastiach aplikujeme menej agresívne podvzorkovanie. Pre lepšie pochopenie fungovania tejto metódy sa odkazujte na priložený článok. [12]



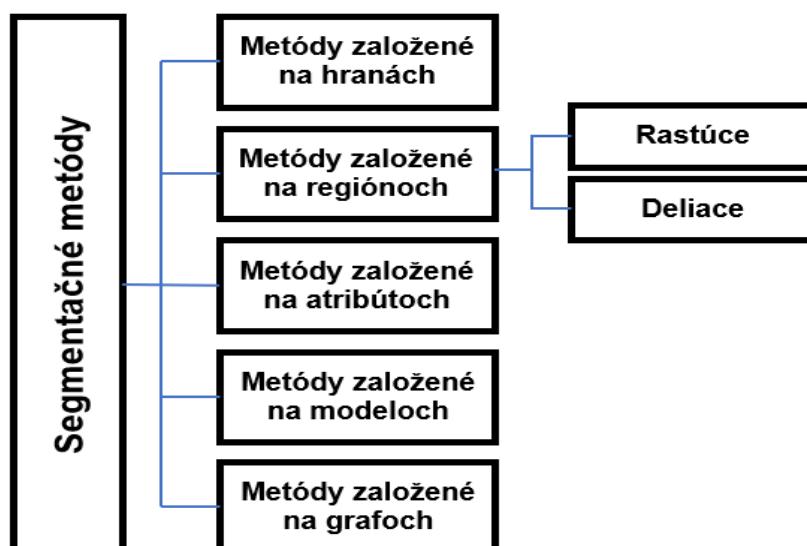
Obr. 1.9: Ukážka klasifikácie dôležitých bodov pomocou metódy založenej na rovinnosti [12]

## 1.4 Segmentácia

Segmentácia je proces rozdelenia väčšieho celku údajov na menšie významovo odlišné časti, na základe určitých charakteristík alebo vlastností. V kontexte mračien bodov alebo počítačového videnia, segmentácia rozdeľuje údaje na jednotlivé časti, na základe vlastnosti, ako sú farba, textúra, geometria a ďalšie. V prípade mračien bodov pomáha segmentácia rozlíšiť rôzne objekty alebo povrhy v scéne, čím pomáha pri úlohách, ako klasifikácia objektov, rekonštrukcia povrchu, navigácia robotov. Celkovo segmentácia slúži, ako základný krok pri extrahovaní významných informácií zo scény, čo umožňuje hlbšiu analýzu, porozumenie a využitie segmentovaných komponentov pre ďalšie účely.

### 1.4.1 Segmentačné metódy

V tejto podkapitole preskúmame rôzne prístupy navrhnuté pre segmentáciu mračien bodov, pričom ich vieme rozdeliť do piatich základných kategórií (Obr. 1.10).



Obr. 1.10: Rozdelenie segmentačných metód

1. **Metódy založené na hranách** - Tieto metódy sa zameriava na detekciu hrán, pomocou rôznych prístupov, ako sú identifikácia výrazných zmien intenzity, zmien orientácie normálových vektorov alebo výpočtu gradientu. Tieto metódy zvyčajne pozostávajú z dvoch fáz. Prvou je detekcia hrán, kde sa vytvárajú obrysy hraníc regiónov na základe lokálnych zmien vlastností povrchu, pričom v druhej fáze sa vnútorne body zoskupia, čím sa vytvoria samostatné segmenty. Aj keď tieto metódy umožňuje rýchlu segmentáciu, často sa stretávajú s problémami presnosti kvôli citlivosti na šum a variáciám hustoty mračien bodov. V 3D priestore môžu detegovať nespojité hrany, čo komplikuje identifikáciu uzav-

retých segmentov bez ďalších dodatočných krokov. [13]

2. **Metódy založené na regiónoch** - Metódy založené na regiónoch využívajú informáciu o okolitých bodoch s podobnými vlastnosťami na vytvorenie izolovaných regiónov a následne zistujú odlišnosti medzi týmito regiónmi. Metódy založené na regiónoch sú odolnejšie voči šumu, ako metódy založené na hranách, avšak majú problém s nadmerným alebo nedostatočným segmentovaním a presným určovaním hraníc regiónov. Tieto metódy delíme do dvoch kategórií: [13]
  - **Rastúce** (z ang. seeded-region) začínajú výberom určitého počtu začiatočných bodov, a z týchto bodov následne každý región rastie pridaním susedných bodov, ak splňajú určité kritériá kompatibility. Medzi tieto kritéria patrí vzdialenosť, intenzita alebo rovnosť susedných bodov. [13]
  - **Deliace** (z ang. unseeded-region) najprv zoskupia všetky body do jedného regiónu. Potom začína proces delenia na menšie regióny a pokračuje, až pokým zvolená miera vhodnosti presahuje stanovený prah. [13]
3. **Metódy založené na atribútoch** - Tieto metódy využívajú vypočítané charakteristiky (atribúty) jednotlivých bodov, ako sú priestorové súradnice, hustota, povrchová textúra a ďalšie, na zoskupovanie podobných bodov. Tieto atribúty usmerňujú zhlukovacie algoritmy, ktoré vytvárajú odlišné segmenty na základe podobnosti atribútov s cieľom izolovať rôzne povrhy alebo objekty v mračne bodov. Aj keď sú tieto metódy flexibilné a presné, ich presnosť silno závisí od presnosti odvodnených atribútov.
4. **Metódy založené na modeloch** - Tieto metódy sa snažia nájsť a kategorizovať body na základe ich reprezentácie pomocou základných tvarov. Využívajú základné geometrické tvary, ako guľa, kužeľ, rovina a valec na zoskupovanie bodov s podobnými matematickými vlastnosťami. Sú rýchle a robustné voči šumu, ale môžu sa stretnúť s problémami pri práci s rôznorodými alebo komplexnými mračnami bodov. [14]
5. **Metódy založené na grafoch** - Tieto metódy vnímajú body, ako vzájomne prepojené vrcholy pomocou hrán, pričom hrany reprezentujúce vzťahy medzi susediacimi bodmi. Využívajú štruktúry grafov, kde zohľadňujú rôzne kritériá, ako sú farba, normály alebo pravdepodobnostné modely. Tieto metódy sú známe svojou presnosťou a efektívnosťou, najmä v robotických aplikáciach. Napriek tomu, že sú schopné pracovať so šumom alebo rôznorodými dátami, môžu čeliť problémom s spracovaním v reálnom čase a často vyžadujú predchádzajúce znalosti alebo špecializované nastavenia senzorov. [13][15]

### 1.4.2 DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) je populárny zhlukovací algoritmus, používaný pre analýzu priestorových dát a segmentáciu mračien bodov. Tento algoritmus využíva princíp toho, že zhluky sú oblasti s vysokou hustotou, oddelené miestami s nižšou hustotou bodov. Algoritmus dynamicky identifikuje zhluky na základe hustoty bodov, bez predpokladu fixného počtu zhlukov alebo konkrétnych tvarov. [16]

DBSCAN pracuje s dvomi vstupnými parametrami:  $Eps$  a  $MinPts$ .  $Eps$  definuje polomer prehľadávaného okolia, zatiaľ čo  $MinPts$  určuje minimálny počet bodov v rámci  $Eps$  potrebných na vytvorenie hustej oblasti. Nastavenie týchto parametrov je kľúčové pre správne fungovanie algoritmu, a preto existuje heuristika, ktorá ich určuje na základe “najtenšieho” zhluku, ktorý nie je považovaný za šum. [16]

Princíp fungovania algoritmu vieme rozdeliť do niekol'kých bodov:

1. Pre každý bod zo súboru sa vypočíta počet bodov ležiacich v okolí určeným  $Eps$  hodnotou. Na základe tejto hodnoty sú body zaradené do troch kategórií.
2. Body, ktoré majú tento počet väčší, ako hraničná hodnota  $MinPts$  sú považované za jadrové, naopak tie ktoré ju majú menšiu, ale zároveň sú v  $Eps$  okolí iného jadrového bodu, sú považované za hraničné. Zvyšné body sú zaradené medzi šumové.
3. Následne sa vytvárajú zhluky pomocou spájania jadrových bodov, ktoré ležia vo svojom  $Eps$  okolí, pričom sú ohraničené hraničnými bodmi. Body šumu nie sú zaradené do žiadnych zhlukov.

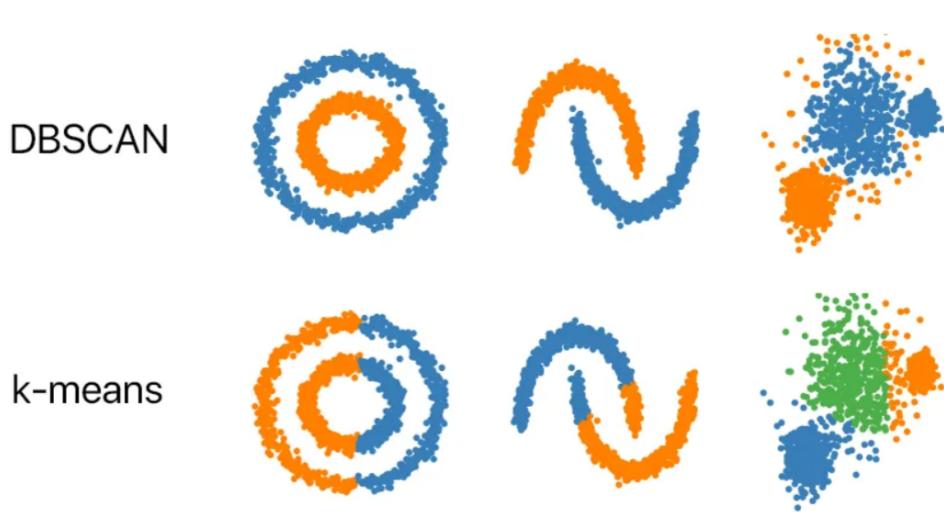
### 1.4.3 K-means

K-means je populárny algoritmus pre zhlukovanie súborov dát, ktorý rozdeľuje dáta do  $K$  odlišných a neprekryvajúcich sa zhlukov. Jeho cieľom je minimalizovať rozptyl vzdialenosí vo vnútri zhlukov prostredníctvom iteratívneho priradovania bodov ku najbližšiemu centru, pričom centrá zhlukov, sú aktualizované, až do dosiahnutia konvergencie. [17]

Princíp fungovania algoritmu vieme rozdeliť do niekol'kých bodov:

1. Výber  $K$  náhodných bodov zo súboru a označenie ich za centrá.
2. Následne sa všetky body priradujú ku svojmu najbližšiemu centru.
3. Zo priemeru združených bodov sa vypočíta nové centrum.
4. Predošlé body sa opakujú až dokým jednotlivé centrá nezkonvergujú do svojej optimálnej polohy (ich poloha sa už výrazne nemení).

Aj napriek tomu, že K-means algoritmus má jednoduchý princíp fungovania, náhodný výber počiatočných centier môže viesť ku uväzneniu v lokálnom minime, a pre optimálny výsledok, je potrebné, aby bol algoritmus spustený niekoľkokrát. Určenie správneho počtu zhlukov  $K$  predstavuje ďalší problém. S rastúcim  $K$  klesá rozptyl, ktorý sa snažíme minimalizovať. Aby sa zabránilo jeho nekonečnému nárastu existuje tzv. lakt'ová metóda (z ang. elbow method), pri ktorej z grafu pre rôzne hodnoty  $K$  vieme nájsť zlom, pri ktorom rozptyl začne klesať výrazne pomalšie. [17]



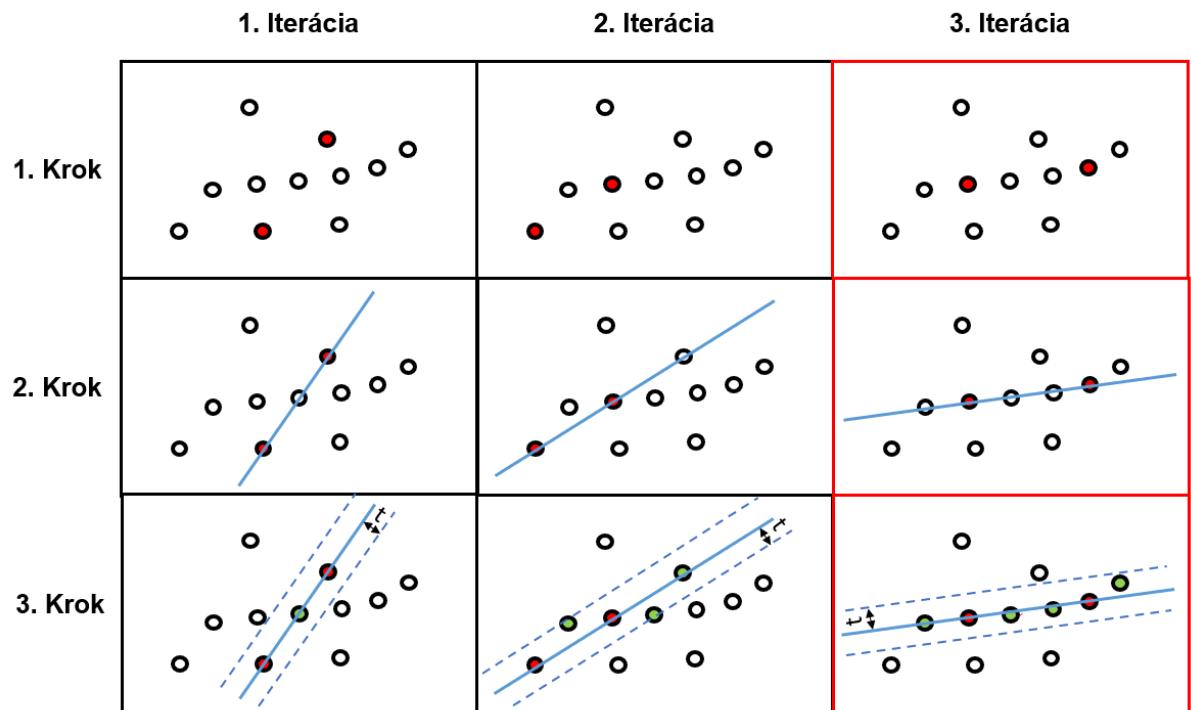
Obr. 1.11: Porovnanie segmentácie pomocou DBSCAN a K-means algoritmu [18]

#### 1.4.4 RANSAC

RANSAC (Random Sample Consensus) je metóda používaná na odhad parametrov matematického modelu zo súboru dát, ktorý môže obsahovať odľahlé hodnoty alebo chyby. Na rozdiel od tradičných metód, ktoré sa snažia zahrnúť väčšinu údajov a eliminovať odľahlé hodnoty, RANSAC funguje tým, že začína s minimálnym množstvom údajov a postupne ho rozširuje o konzistentné údaje na vytvorenie spoľahlivého modelu. [19]

Metóda začína výberom  $N$  bodov zo celého súboru, pričom  $N$  predstavuje najmenší potrebný počet bodov pre vyjadrenie modelu. Následne sa pre vybrané body vypočítava zvolený model (napr. priamka, krivka, plocha). Pre daný model sa identifikujú body, ktoré súhlasia s modelom v rámci definovanej tolerancie a zaznamenáva sa ich počet. Tieto kroky sa iteratívne opakujú, až do dosiahnutia definovaného počtu iterácií, alebo až kým nie je splnená ukončovacia podmienka (napr. nájdenie modelu s dostatočným počtom bodov). Ak je identifikácia ukončená nájdením dostatočného počtu bodov, RANSAC využíva vyhľadzovaciu techniku, ako

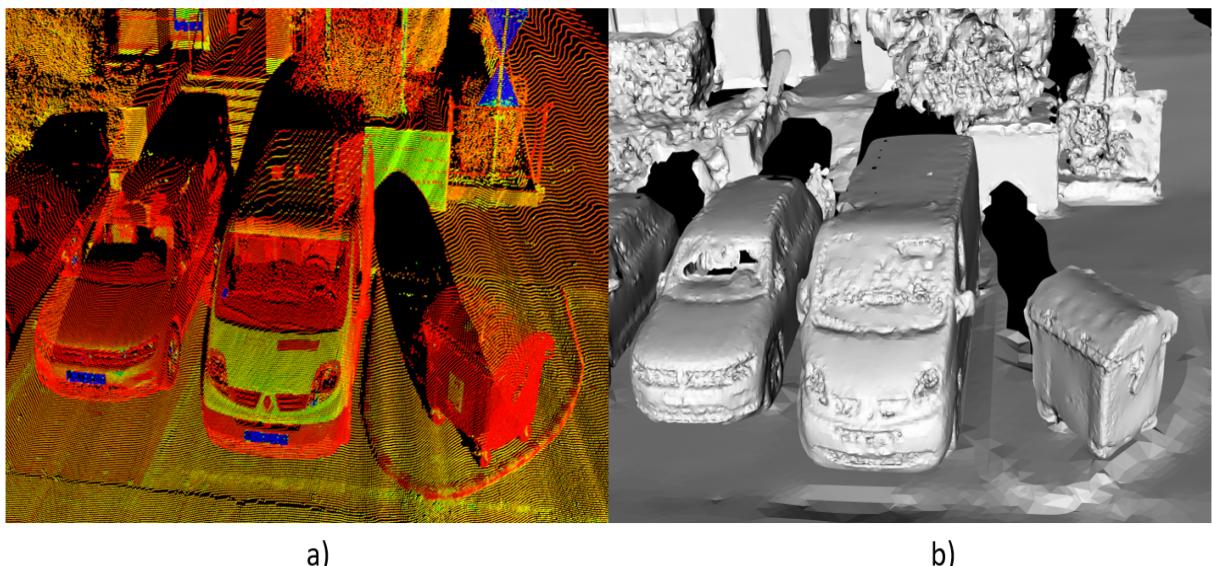
je napríklad metóda najmenších štvorcov, na výpočet vylepšeného odhadu parametrov modelu zo všetkých príslušných bodov. [19]



Obr. 1.12: Postup RANSAC algoritmu pre odhad parametrov modelu priamky

## 2 Rekonštrukcia a spracovanie povrchu

Rekonštrukcia povrchu, či už z fotiek alebo mračien bodov, sa stala v posledných rokoch populárnu témou vďaka rýchlemu vývoju výpočtovej techniky, ale aj cenovému sprístupneniu skenerov, ktoré ponúkajú slušné výsledky. Tento proces nachádza vysoké využitie v oblastiach, ako sú robotika, počítačová vizia, mapovanie, ale aj kultúrne dedičstvo. V týchto oblastiach sa stretávame so scénami, ktoré obsahujú objekty so rôznymi rozmermi, a zachovanie presnosti a zníženie pamäťovej náročnosti je náročnou a dôležitou úlohou. Pre tento účel sa vyvinulo mnoho rôznych metód, ktoré majú svoje výhody aj nevýhody.



Obr. 2.1: Ukážka rekonštrukcie povrchu - (a) mračno bodov (b) zrekonštruovaný povrch

Jednotlivé metódy vieme rozdeliť do dvoch základných skupín na základne ich prístupu, ku generovaniu povrchu.

1. **Explicitné metódy** - Využívajú explicitnú reprezentáciu povrchu (primitívne tvary), čo znamená, že vytvorený povrch bude ležať priamo na bodech mračna. Ich nevýhodou je vysoká citlosť na šum, ako aj časová náročnosť na výpočet.
2. **Implicitné metódy** - Využívajú implicitnú reprezentáciu povrchu, a teda povrch je charakterizovaný funkiami, ktorých izo-kontúry tesne approximujú body mračna. Rekonštrukcia implicitných plôch spočíva vo vyhľadávaní funkcie, ktorá najlepšie vyhovuje vstupným bodom, pričom následne vyžaduje dodatočné spracovanie pre vizualizáciu. Metóda Marching Cubes patrí medzi najpoužívanejšie metódy pre generovanie trojuholníkových plôch z implicitnej reprezentácie. [20]

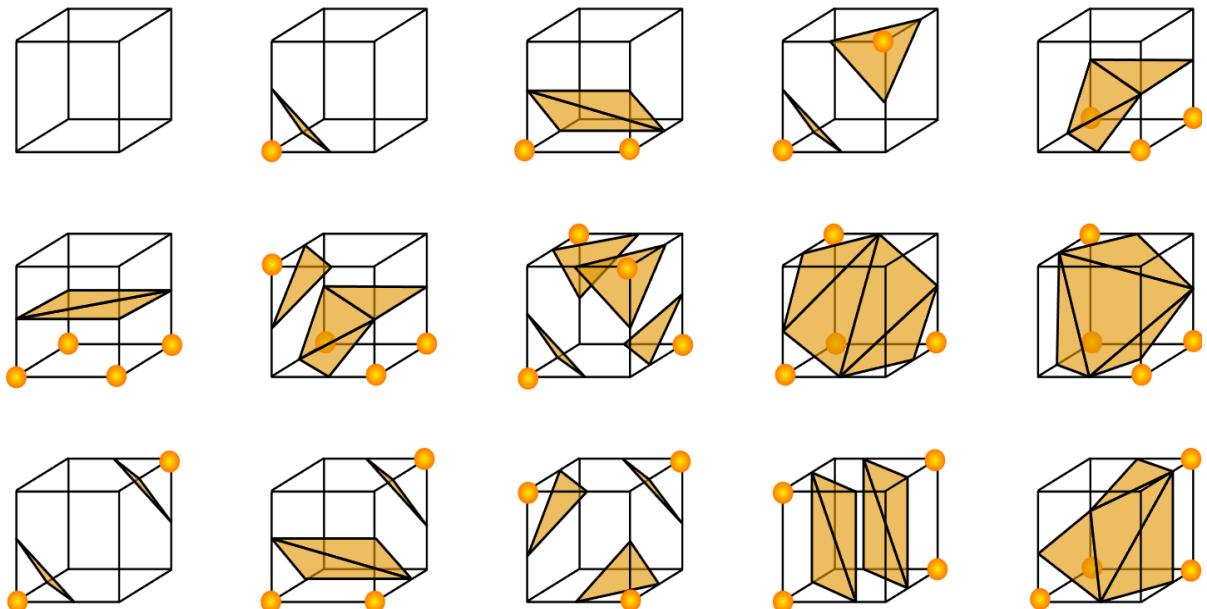
## 2.1 Metódy pre rekonštrukciu povrchu

V tejto podkapitole sa zameriame na principiálne fungovanie konkrétnych metód pre rekonštrukciu povrchu. Ako už bolo povedané, metódy delíme na explicitné a implicitné, pričom väčšina z uvedených metód bude implicitná, napäťko sú odolnejšie voči šumu a výpočtovo menej náročné.

### 2.1.1 Marching cubes

Metóda marching cubes je jednou z najpoužívanejších metód pre rekonštrukciu povrchu. Táto metóda je založená rozdelení priestoru do voxelovej mriežky, pričom každý voxel tvorí osem vrcholov. Každému vrcholu je priradená hodnota podľa toho, či sa nachádza vnútri (hodnota 1) alebo mimo prehľadávaného povrchu (hodnota 0), podľa hraničnej úrovne nastavenej od užívateľa. Pre osem vrcholov existuje 256 rôznych konfigurácií, čo môže byť pomerne zložité na spracovanie. Preto sa tieto konfigurácie redukujú na 14 vzorov (Obr. 2.2) vďaka symetrickým vlastnostiam, čím sa zároveň minimalizuje pravdepodobnosť chýb. [21]

Výhodou tejto metódy je, že pre svoju funkciu nepotrebuje odhad normál, ale naopak, pôvodná verzia ponúka výpočet normál pomocou gradientu. Tieto normály je možno následne využiť pri vizualizácii a tieňovaní výsledného povrchu. Na druhú stranu jednou z hlavných nevýhod je využitie voxelovej mriežky, čo môže pri veľkých mračnach bodov viest' ku rýchlemu naplneniu RAM pamäte. Pre odstránenie tohto problému je nutná úprava pôvodnej metódy, ktorá rozdelí prehľadávaný priestor na viacero častí a na konci ich spojí.

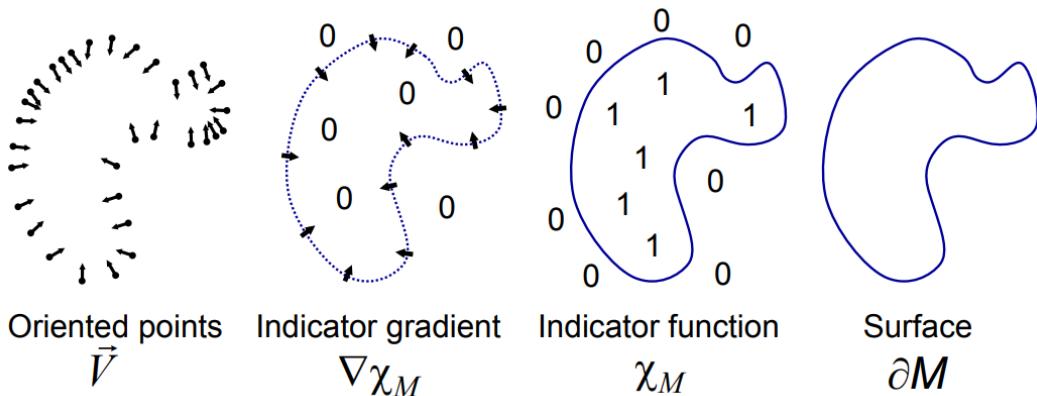


Obr. 2.2: Rôzne možnosti konfigurácie voxelov [22]

## 2.1.2 Poisson

Táto metóda ukazuje, že rekonštrukciu povrchu z orientovaných bodov môže byť vyriešená, ako priestorový Poissonov problém. Formulácia pomocou Poissonovej metódy zohľadňuje všetky body naraz, bez použitia heuristického priestorového delenia alebo prelínania, a je preto vysoko odolná voči šumu. [23]

Táto metóda využíva implicitný prístup pre reprezentáciu povrchu, konkrétnie sa zameriava na výpočet 3D indikačnej funkcie  $\chi$  (definovanej, ako 1 pre body vnútri modelu a 0 pre body mimo modelu vidieť Obr. 2.3) a následne získanie zrekonštruovanej plochy extrahovaním vhodnej izo-kontúry. Dôležitým poznatkom je, že medzi orientovanými bodmi získanými z povrchu modelu a indikačnou funkciou modelu, existuje integrálny vzťah. Konkrétnie, gradient indikačnej funkcie je vektorové pole, ktoré je nulové takmer všade (keďže funkcia indikátora je konštantná takmer všade), okrem bodov blízkych povrchu, kde je rovné normálam smerujúcim do vnútra povrchu. S tohto dôvodu, vzorky orientovaných bodov možno považovať za vzorky gradientu indikačnej funkcie modelu. [23]



Obr. 2.3: Ukážka Poissonovej rekonštrukcie v 2D [23]

Úloha výpočtu indikačnej funkcie sa zjednodušuje na hľadanie skalárnej funkcie  $\chi$ , ktorej gradient najlepšie approximuje dané vektorové pole  $\vec{V}$ , definované množinou vzoriek. Keď sa použije operátor divergencie, táto úloha sa pretransformuje do štandardného Poissonového problému, kde je cieľom vypočítať  $\chi$ , tak aby jeho Laplačián (divergencia gradientu), bol rovný divergencii vektorového pola  $\vec{V}$ . [23]

$$\Delta \chi \equiv \nabla \cdot \nabla \chi = \nabla \cdot \vec{V} \quad (2.1)$$

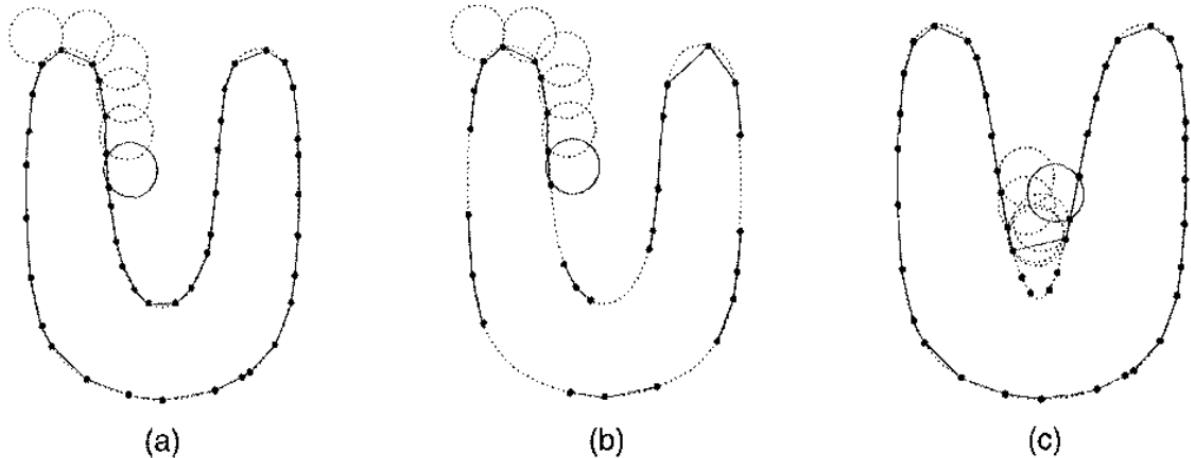
Ako už bolo skôr uvedené, táto metóda vytvára globálne riešenie, ktoré využíva všetky body naraz. To jej umožňuje vytvoriť vodotesný a hladký povrch, čo môže byť výhoda oproti

iným metódam. Naopak metóda potrebuje na vstup orientované body a je teda nutné mať informáciu o normálach, čo môže byť problém, ak je rekonštruovaný objekt nasnímaný z viacerých snímkov.

### 2.1.3 Ball pivoting

Hlavný princíp fungovania tejto metódy, je celkom jednoduchý. Uvažujme body v priestore, na ktoré položíme  $\rho$ -guľu (guľa s radiom  $\rho$ ), pričom uvažujeme, že hustota bodov je dostatočne veľká na to, aby cez nich guľa nevedela prejест' bez dotyku. Metóda začína rekonštrukciu položením  $\rho$ -gule na povrch, tak aby sa dotýkala troch náhodných bodov. Následne sa guľa udržuje v kontakte s dvoma pôvodnými bodmi a obtáča sa (z ang. pivot), až dokým nenarazí na ďalší bod. Trojica bodov, s ktorými je guľa v kontakte, tvorí nový trojuholník a pridáva sa ku existujúcemu povrchu. Proces sa opakuje prechádzaním po okraji súčasnej hranice povrchu, až dokým neboli prejedené všetky hranice a nepríbudiol žiadny nový trojuholník. [24]

Aj napriek jednoduchosti a priamočiarosti, má táto metóda svoje nedostatky. Ako môžeme vidieť na Obr. 2.4, metóda funguje dobre, ak pracuje v ideálnych podmienkach. V prípade, ak sú body neuniformné (častí prípad mračien bodov), guľa nedokáže premostiť ku ďalšiemu bodu a v povrchu vznikajú diery. Rovnaký prípad nastáva, aj keď je polomer gule príliš veľký a rekonštruovaný objekt obsahuje detaily s príliš veľkým zakrivením.



Obr. 2.4: Ukážka ball-pivoting rekonštrukcie v 2D - (a) Ideálny prípad (b) mála a neuniformná hustota bodov (c) príliš veľké zakrivenie (väčšie ako  $1/\rho$ ) [23]

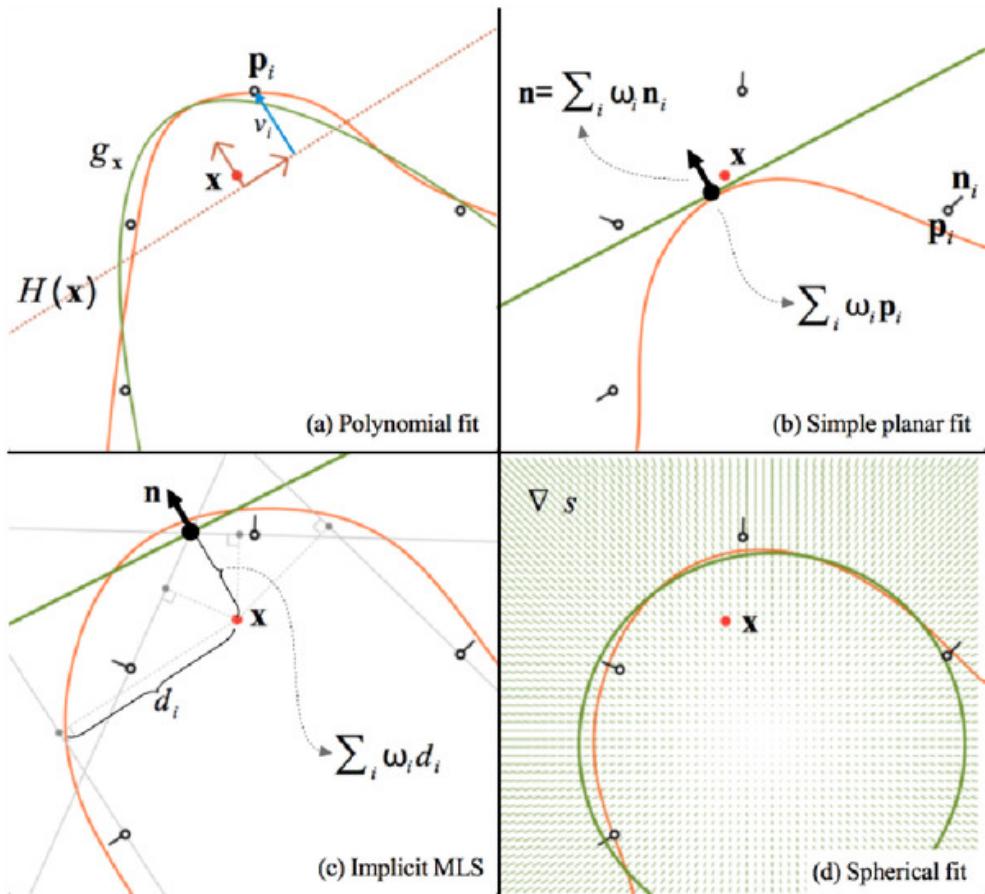
### 2.1.4 MLS

MLS (Moving least squares) je trieda, ktorá zastrešuje viacero metód, ktoré pristupujú ku rekonštrukcií povrchu, tak že approximuje jednotlivé body pomocou polynómu nízkeho rádu (kubický, kvadratický), čo zároveň ponúka možnosť podvzorkovania alebo nadvzorkovania. [25]

Za predpokladu, že skalárna hodnota  $v_i$ , je priradená ku každej vstupnej vzorke  $p_i$ , je potom zrekonštruovaná plocha v  $x$  definovaná, ako hodnota v  $x$  viacrozmersného polynómu  $g_x$ , ktorý najlepšie aproximuje susedstvo bodu  $x$  v zmysle váženej metódy najmenších štvorcov [25]:

$$g_x = \arg \min_g \sum \omega(||x - p_i||)(g(p_i) - v_i)^2 \quad (2.2)$$

kde  $\omega$  je váhová funkcia, ktorá priraduje väčší vplyv bodom, ktoré sú bližšie ku aktuálne hodnotenému bodu. Táto váhová funkcia taktiež pôsobí, ako forma dolnopriepustného filtra, pomáhajúc zmierniť vplyv šumu na vstupných dátach. Okrem toho, sa MLS metódy vedia prispôsobiť nerovnomerným hustotám vzorkovania, tým že podľa potreby využijú rozdielnú váhovú funkciu, ktorá sa vie adaptívne prispôsobiť vzorkovacej hustote. Táto prispôsobivost', je dôležitá pre zachovanie presnosti pri práci s neuniformnými dátami, čo je častý prípad mračien bodov. [25]



Obr. 2.5: Ukážka princípu pre viaceru variantov MLS v 2D. Miestne aproximácie vypočítané pre hodnotiaci bod  $x$  sú znázornené zelenou farbou. Oranžové krivky zodpovedajú rekonštruovaným izo-kontúram. [25]

## 2.1.5 MPU

MPU (Multi-level partition of unity) je metóda, ktorá bola špecificky vytvorená na rýchlu a presnú rekonštrukciu povrchu s veľkého súboru orientovaných bodov (body, ktoré obsahujú jednotkové normály). Názov metódy vyplýva zo podstaty fungovania MPU, a to že sa skladá zo viacerých vážených funkcií, ktoré sa spočítavajú do jedna pre všetky body v doméne. Táto metóda ponúka adaptívny chybovo kontrolovaný odhad funkcie vzdialenosť od povrchu, čo znamená, že odhad je presný v blízkosti povrchu a hrubý d'aleko od povrchu. [26]

Pre vytvorenie implicitnej reprezentácie povrchu sa najprv všetky body rozdelia pomocou oktálového stromu. V každej bunke stromu sa vytvorí po častiach kvadratická funkcia (lokálna tvarová funkcia), ktorá sa snaží napasovať na body v bunke. Hodnota týchto funkcií nadobúda hodnotu 0 v blízkosti bodov, pozitívnu hodnotu vo vnútri a negatívnu (vonku) d'aleko od bodov. V prípade, že dosiahnutá presnosť je stále malá, bunka v oktálovom strome sa znova rozdelí a celý proces sa opakuje, až dokým nie je dosiahnutá požadovaná presnosť (vid'. Obr. 2.6). Na záver v miestach, kde sa stretávajú hranice buniek, sú tvarovacie funkcie spojené dokopy na základe váh z funkcií rozdelenia jednoty (z ang. partition of unity functions). [26]



Obr. 2.6: Ukážka rekonštrukcie pomocou MPU implicit. Ľavá časť modelu je ofarbená podľa hĺbkely rozdelenia, s tým že stúpa od modrej až po červenú. Modely v pozadí sú rekonštruované so stúpajúcou chybou v odhade. [26]

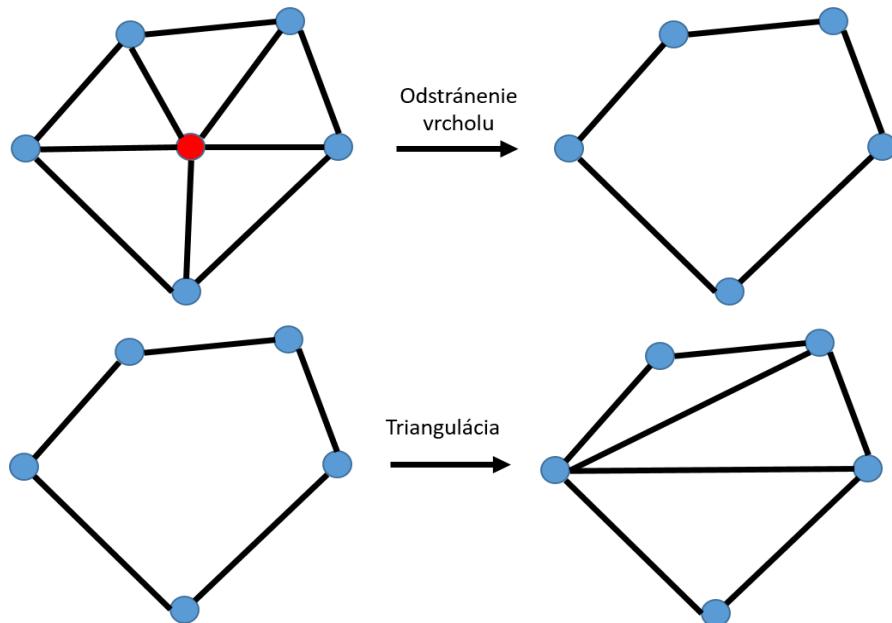
## 2.2 Metódy pre zjednodušenie povrchu

V mnohých prípadoch sa po rekonštrukcií povrchu z mračien bodov stretávame s tým, že povrch daného modelu obsahuje zbytočne veľa polygónov, pričom by ho bolo možné vyjadriť menším počtom a zároveň zachovať celkový tvar. V tejto podkapitole sa preto budeme zaoberať metódami, ktoré sa zameriavajú na zjednodušenie topológie povrchu, pričom ich delíme na lokálne a globálne metódy.

### 2.2.1 Lokálne metódy

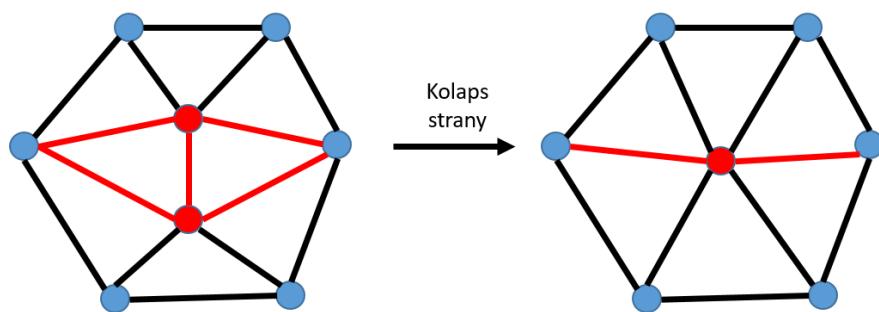
Lokálne metódy sú oproti globálnym bežnejšie, a preto sa budeme zaoberať hlavne nimi. Tieto metódy pristupujú k zjednodušovaniu iteratívou aplikáciou lokálneho operátora na malú sadu bodov, čím vytvoria povrch s menším počtom prvkov. Využívajú chybové alebo nákladové (z ang. cost) funkcie, ktoré usmerňujú proces zjednodušovania, a taktiež ponúkajú presnú kontrolu výsledných atribútov povrchu (počet polygónov, hranice chýb pre vrcholy). [27]

1. **Decimácia vrcholov (z ang. Vertex decimation)** - Tento prístup pracuje s jedným vrcholom, ktorý sa vymaže a následne sa snaží zaplátiať vytvorenú dieru na základe klasifikačnej schémy o susedstve. Výber vrcholu, ktorý bude vymazaný, je spojený so vzdialenosťou od roviny, ktorá je aproximovaná pomocou spriemerovania normál, t'azísk a plôch okolitych trojuholníkov. Vzdialenosť vrcholu od tejto roviny slúži, ako miera chyby pre jeho odstránenie, pričom vrcholy ležiace na hladkých častiach sú uprednostňované pred tými, ktoré opisujú ostré zmeny (detailsy). [27]



Obr. 2.7: Ukážka decimácie vrcholu v 2D

2. **Kolaps strán (z ang. Edge collapse)** - V tejto metóde pracujeme s dvoma vrcholmi naraz, pričom ich výber môže byť vykonávaný ako v predošej metóde. Táto metóda funguje na základe odstraňovania strán medzi vybranými vrcholmi, a teda zlučuje ich do jedného, pričom vždy ostani dva trojuholníky. Ukázalo sa, že výber pozície nového vrcholu nie je až tak jasný, nakoľko uloženie do stredu medzi pôvodnými vrcholmi nie je optimálne. Vyvinulo sa preto niekoľko prístupov, ktoré rozhodujú o novom uložení a medzi najpopulárnejšie patrí prístup na základe kvadratickej chyby, ktorý je založený na umocnenej vzdialosti vrcholov od approximovanej roviny okolitých trojuholníkov. Tento prístup nájde optimálnu polohu nového vrcholu, tak aby sa zachovali aj ostré zmeny v povrchu.

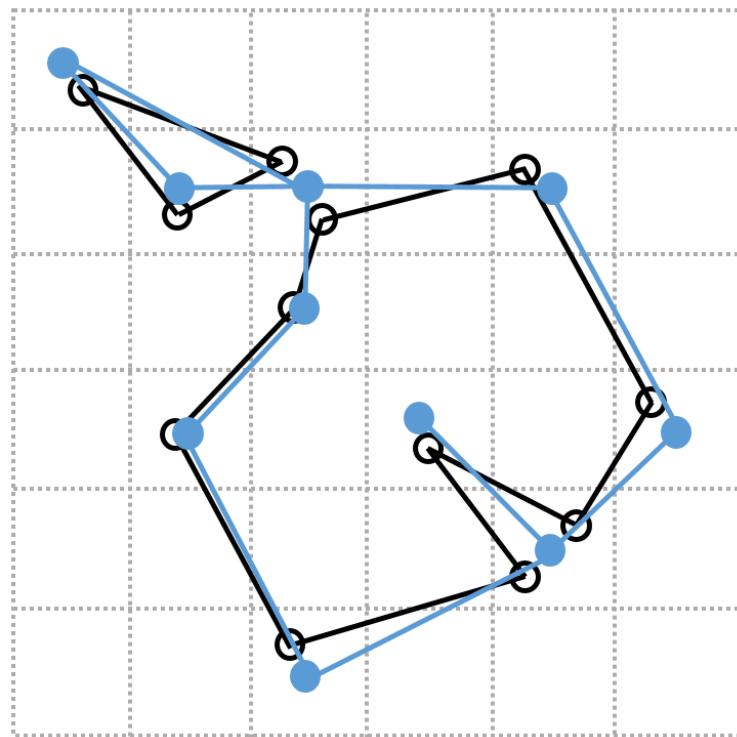


Obr. 2.8: Ukážka kolapsu strany v 2D

### 2.2.2 Globálne metódy

Ako to už vyplýva z mena, globálne metódy pristupujú ku spracovaniu povrchu ako celku a na rozdiel od lokálnych metód, ktoré využívajú iteratíve procesy, globálne metódy analyzujú celý povrch naraz, aby dosiahli jeho zjednodušenie.

1. **Zhlukovanie vrcholov (z ang. Vertex clustering)** - Táto metóda je založená na priradení váhy každému vrcholu, na základe jeho dôležitosti. Pre vrcholy susedné ku veľkým trojuholníkom a vrcholy v zakrivených oblastiach, je priradená väčšia váha, ako pre vrcholy v hladkých oblastiach a vedľa malých trojuholníkov. Celý priestor v ktorom sa povrch nachádza sa rozdelí pomocou voxelovej mriežky. Na záver sa všetky vrcholy v danej bunke mriežky zoskupia do vrcholu s najväčšou váhou, čím sa zníži celkový počet vrcholov. Táto metóda je výpočtovo veľmi efektívna, jej presnosť sa dá upravovať pomocou rozlíšenia voxelovej mriežky, ale taktiež vie zmeniť topológiu povrchu nepredvídateľným spôsobom. [27]



Obr. 2.9: Ukážka zhľukovania vrcholov v 2D. Originálne vrcholy (čierne), po úprave (modré), pre viditeľnosť posunuté od stredu mriežky

### **3 Špecifikácia ciel'ov**

Po úspešnom zhrnutí rôznych prístupov pre spracovanie mračien bodov a rekonštrukciu povrchu v predošlých kapitolách, môžeme prejsť na návrh nášho riešenia. Navrhovaný algoritmus využije skôr spomínané metódy na predspracovanie mračna bodov, aby potom následná rekonštrukcia povrchu dosahovala lepšie výsledky, zatiaľ čo sa pokúsime o zníženie celkovej pamäťovej náročnosti na uloženie modelov.

Tento proces môžeme rozdeliť do následujúcich bodov:

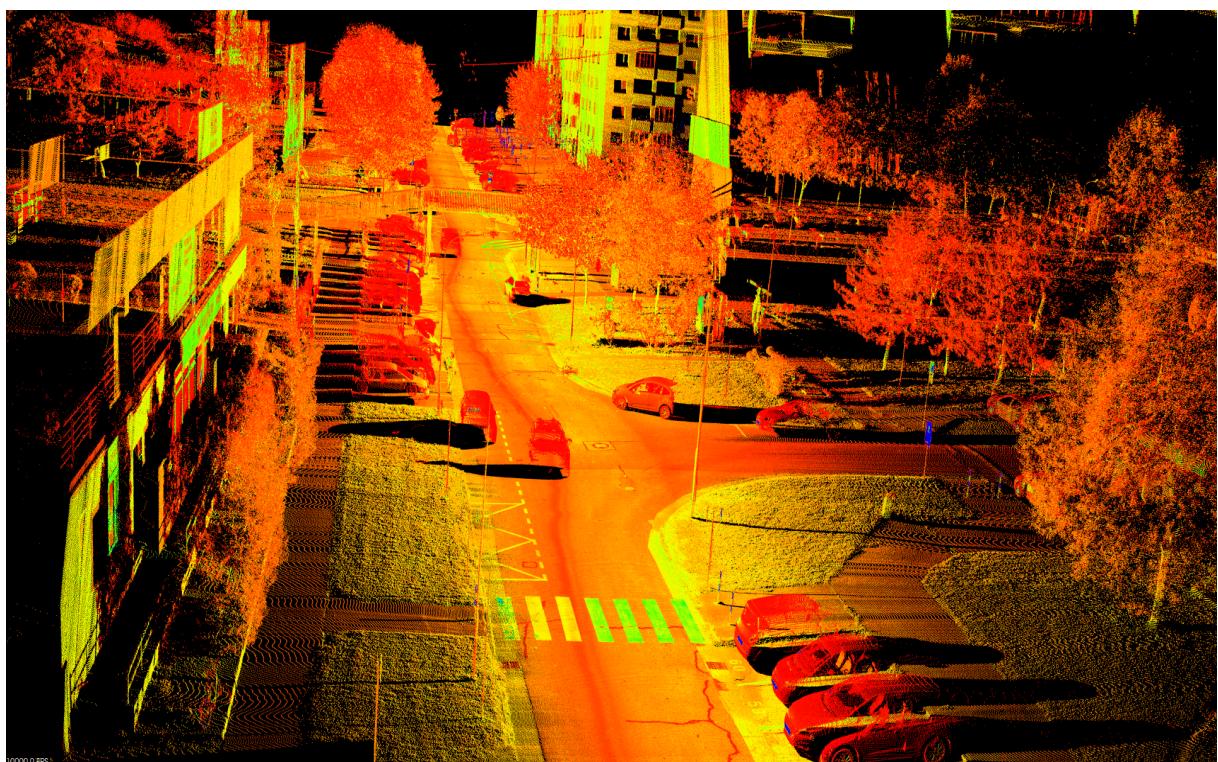
1. Odstránenie odľahlých bodov
2. Separácia bodov zeme a objektov
3. Zhlukovanie bodov objektov do supervoxelov
4. Segmentácia objektov
5. Podvzorkovanie mračna bodov
6. Rekonštrukcia povrchu
7. Textúrovanie povrchu

## 4 Použité súbory dát

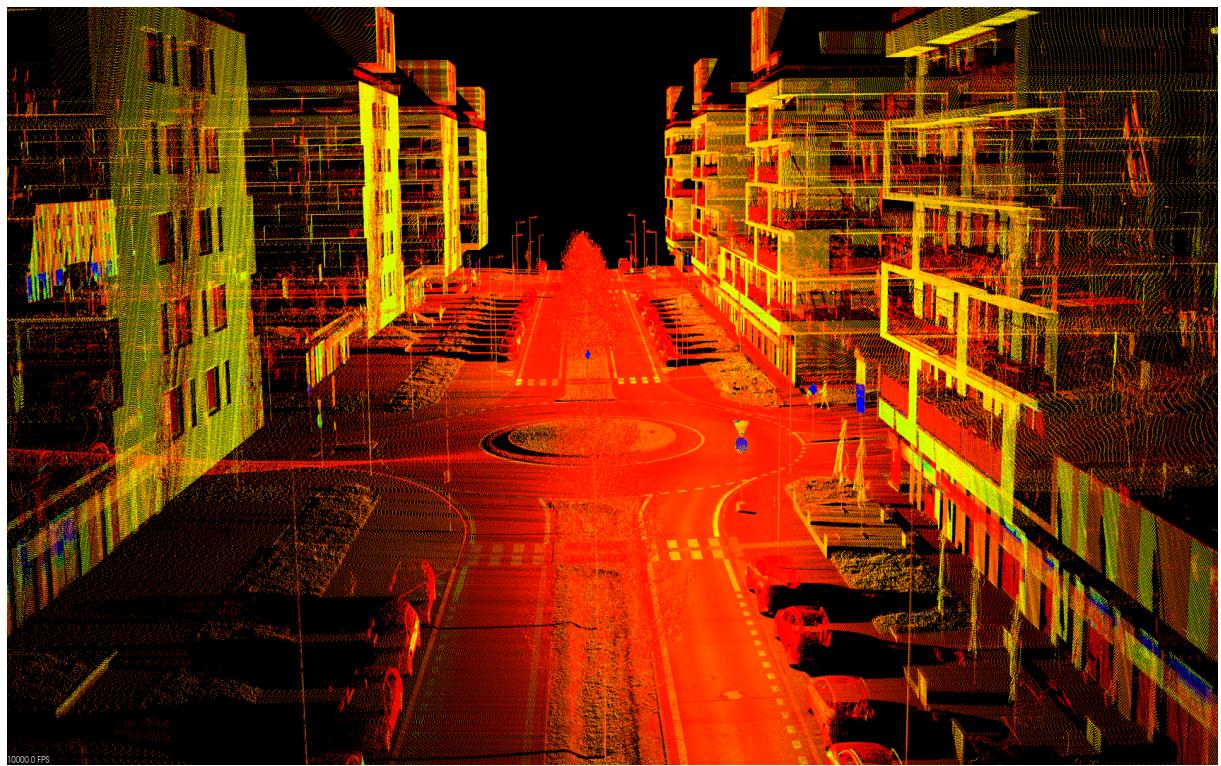
Pred samotnou implementáciou navrhnutého riešenia si v krátkosti predstavíme mračná bodov, s ktorými budeme pracovať. Pre účely tejto práce nám bolo poskytnutých päť rozdielnych mračien bodov, pričom dve boli použité na vývoj a testovanie, a zvyšné boli použité na validáciu a overenie riešenia.

Pre získanie poskytnutých mračien bodov boli použité dva rozdielne LIDAR-y. Mračno bodov mestskej časti s bytovkami (Obr. 4.1) a mračno bodov mestskej časti s kruhovým odjazdom (Obr. 4.2) boli namerané pomocou Optech CL-360 LIDAR-u, zatiaľ čo mračno bodov parkoviska pri obchodoch (Obr. 4.3), mračno bodov ulice so trolejbusovou zastávkou (Obr. 4.4) a mračno bodov dedinskej časti (Obr. 4.5) boli namerané pomocou Hesai XT-32 LIDAR-u. V prípade prvých štyroch súborov dát bol LIDAR pripojený na streche auta, zatiaľ čo v poslednom bol pripojený na dronovi a dátá boli získane zo vzduchu.

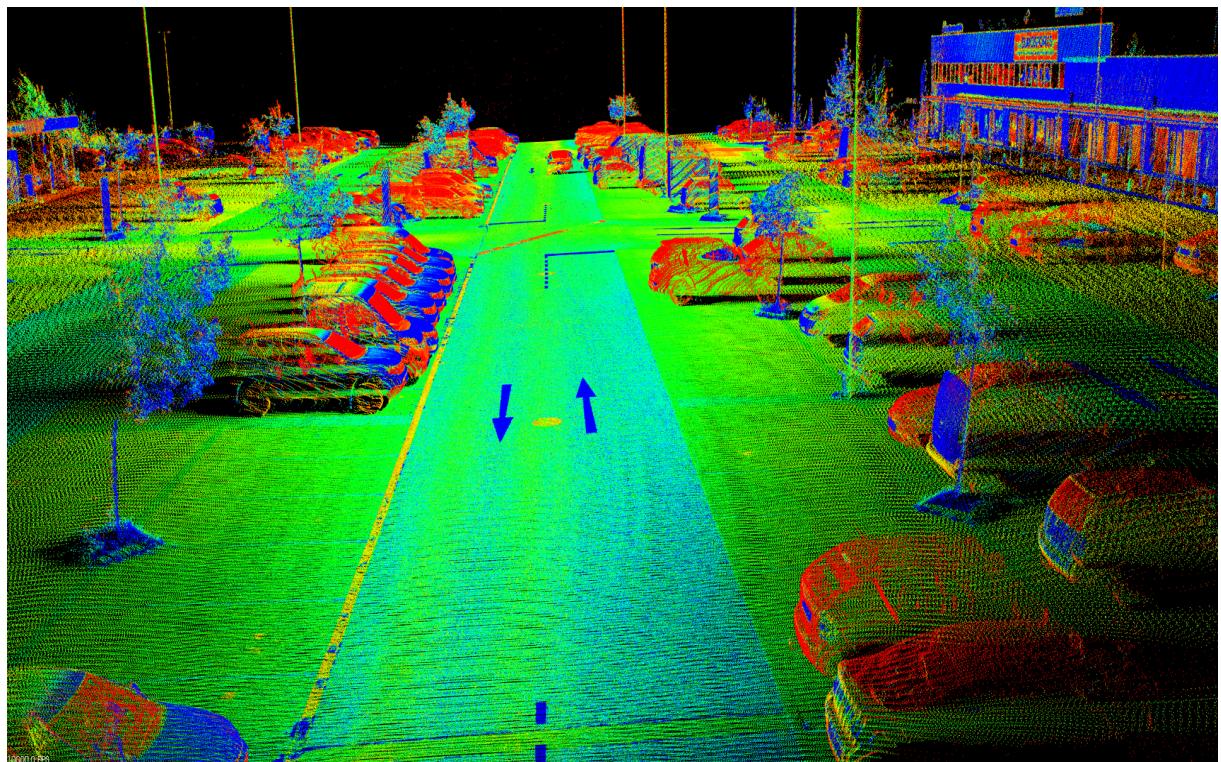
Pre jednoduchosť sa budú po zvyšok práce jednotlivé mračná bodov označovať, ako  $SDx$ , kde  $x$  je číslo súboru dát.



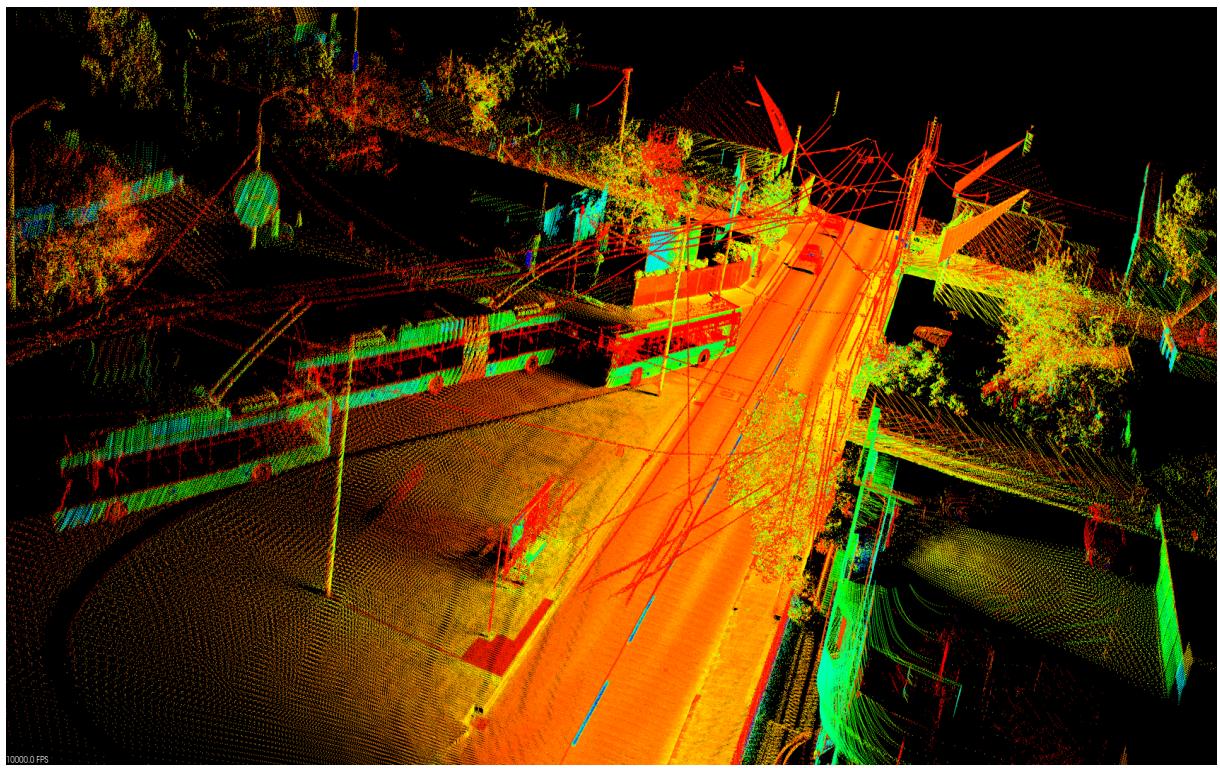
Obr. 4.1: Mračno bodov mestskej časti s bytovkami -  $SD1$  (Optech CL-360)



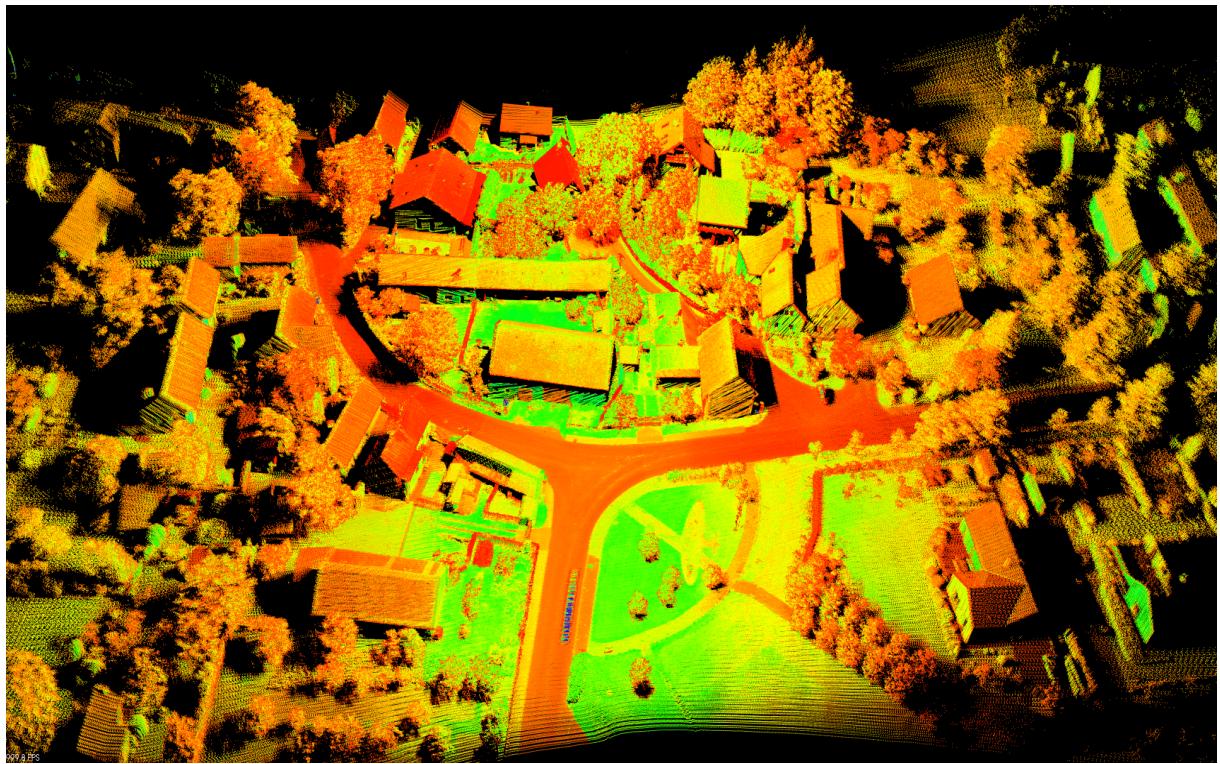
Obr. 4.2: Mračno bodov mestskej časti s kruhovým odjazdom - SD2 (Optech CL-360)



Obr. 4.3: Mračno bodov parkoviska pri obchodoch - SD3 (Hesai XT-32)



Obr. 4.4: Mračno bodov ulice so trolejbusovou zastávkou - SD4 (Hesai XT-32)



Obr. 4.5: Mračno bodov dedinskej časti - SD5 (Hesai XT-32)

# 5 Realizácia riešenia

V tejto kapitole sa budeme zaoberať konkrétnou implementáciou vyššie špecifikovaných cieľov. Na implementáciu budeme používať programovací jazyk C++ v prostredí Visual Studio, pričom sa hlavne zameriame na využitie existujúcich implementácií algoritmov z PCL knižnice.

## 5.1 Odstránenie odľahlých bodov

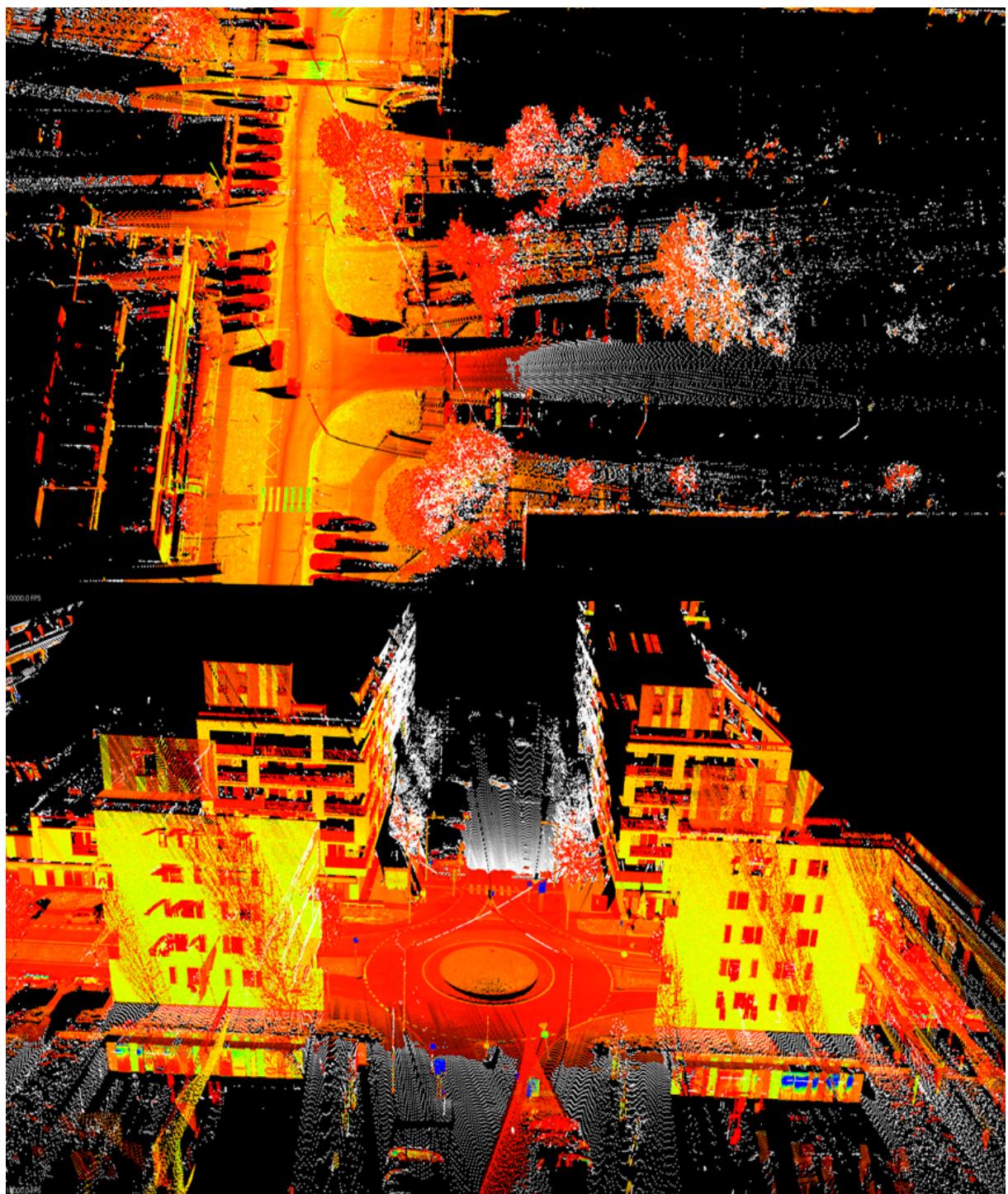
Ako už bolo uvedené v úvodnej kapitole, mračná bodov často obsahujú body, ktoré nereprezentujú reálny povrch objektov a nazývame ich odľahlé. Tieto body vznikajú z rôznych dôvodov, ako sú šum v systéme snímača, prekrytie snímača, lesklé alebo priehľadné povrhy objektov a mnoho ďalších. Odfiltrovanie týchto bodov je dôležitým krokom pri spracovaní mračien bodov, nakol'ko ich odstránením nie len znížime celkový počet bodov, ale aj zabezpečíme, že nasledovné kroky budú mať vyššiu úspešnosť, či už ide o segmentáciu alebo samotnú rekonštrukciu povrchu.

Pre samotné odstránenie odľahlých bodov nám PCL knižnica ponúka dve rýchle a efektívne metódy, ROR (Radial outlier removal) a SOR (Statistical outlier removal). Prvá s týchto metód nám ponúka odstránenie bodov, ktoré nemajú minimálny počet susedov v definovanom rádiuse. Táto metóda je ale parametricky závislá od hustoty mračna bodov a pre rozdielne súbory dát nefunguje rovnako, a preto sme sa rozhodli, že budeme pokračovať so SOR metódou. Táto metóda počíta so priemernou vzdialenosťou každého bodu a jeho K najbližších susedov, na základe čoho vyhodnocuje, ktoré body sú odľahlé (presnejšie vysvetlené v kapitole 1.2.1).

Metóda nám ponúka nastavenie počtu najbližších susedov a násobiteľ prahu štandardnej odchýlky. Tieto dva parametre nám umožňujú nastavenie agresívnosti filtru a po vyskúšaní viačerých kombinácií, sme dospeli ku hodnotám:

- Počet najbližších susedov (z ktorých sa počíta priemer) = 20
- Násobiteľ prahu štandardnej odchýlky = 2,0

pre ktoré sme dosiahli dobrý pomer odstránenia nepotrebných (riedkych) bodov a zachovania podstatných bodov. Ukážku dosiahnutých výsledkov môžeme vidieť na Obr. 5.1.



Obr. 5.1: Výsledné mračná bodov (pôvodná farba) po odstránení odľahlých bodov (biele farba) pomocou SOR metódy (*SD1* hore, *SD2* dole)

	<b>Pôvodný počet bodov</b>	<b>Nový počet bodov</b>	<b>Zmenšenie mračna bodov [%]</b>	<b>Potrebná pamäť [MB]</b>
<b>SD1</b>	15 667 351	15 477 814	1,21	221,70
<b>SD2</b>	5 297 511	5 144 894	2,89	176,20
<b>SD3</b>	9 924 233	9 797 094	1,29	140,20
<b>SD4</b>	5 462 647	5 344 810	2,16	76,40
<b>SD5</b>	7 544 836	7 297 335	3,28	104,70

Tabuľka 5.1: Porovnanie veľkosti pôvodných a filtrovaných mračien bodov, pre jednotlivé súbory dát

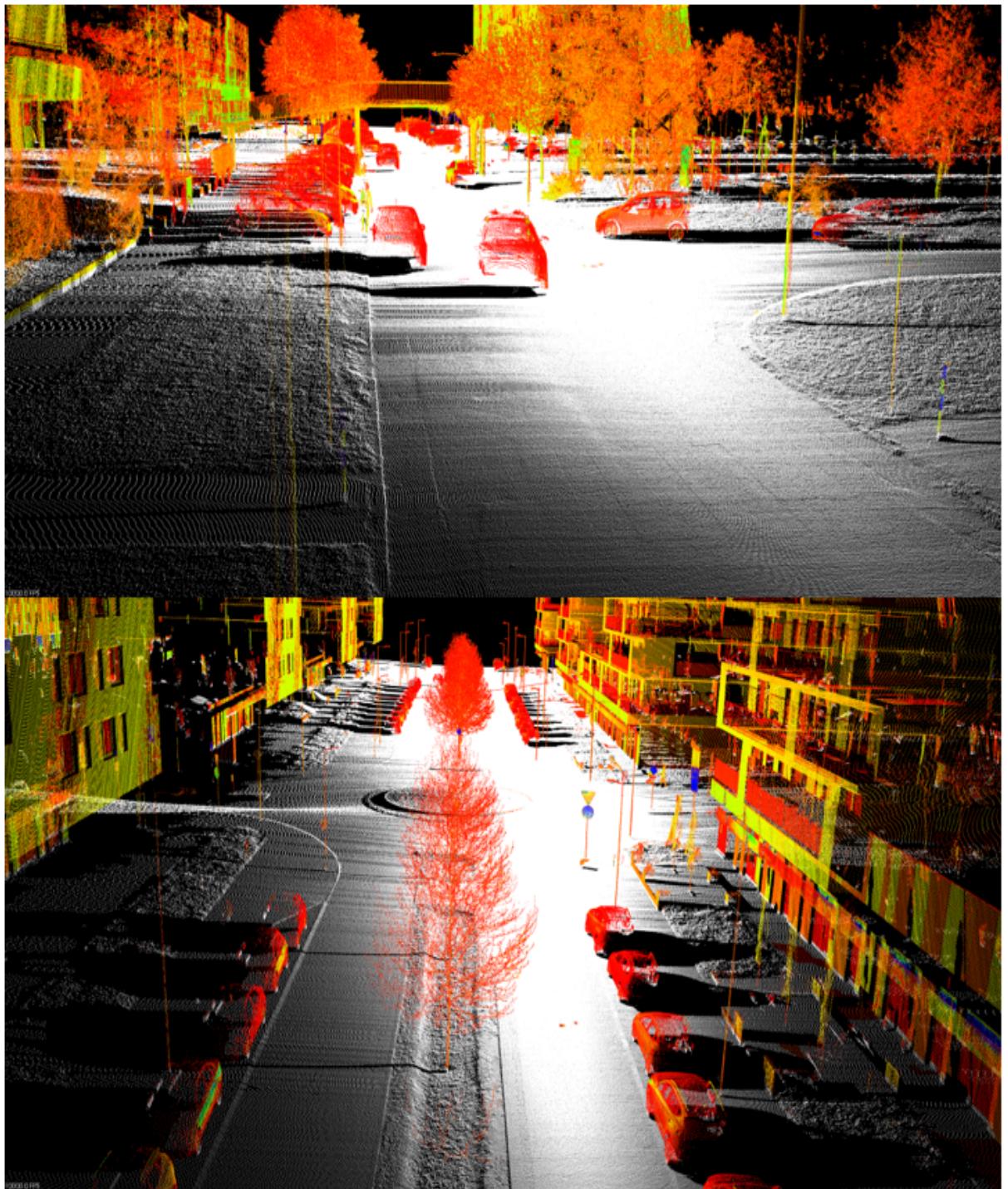
## 5.2 Separácia bodov zeme a objektov

Po odfiltrovaní odľahlých bodov je ďalším hlavným cieľom segmentovať mračno bodov na jednotlivé objekty. Pred samotnou segmentáciou je vhodné separovať body zeme (terénu) od bodov jednotlivých objektov, nakoľko v druhej väčšine prípadov, sú tieto body navzájom spojené iba so samotnou zemou.

Pre tento účel využijeme progresívny morfologický filter. Princíp tohto filtra spočíva vo využití matematickej operácie otvárania (kombinácia erózie a následnej dilatácie), ktorá sa aplikuje na body v tzv. "okne". Ak je toto okno väčšie ako objekt reprezentovaný bodmi vo vnútri, objekt je odstránený, zatiaľ čo väčšie objekty sú zachované. Táto metóda začína vytvorením mriežky s počiatočnou veľkosťou okna, pričom odstraňuje menšie objekty. Postupným zväčšovaním tohto okna odstraňuje stále väčšie objekty, až kým nedosiahne maximálnu veľkosť okna, kedy by mali byť zachované len body zeme. [28]

V našej práci využijeme implementáciu z PCL knižnice, konkrétnie využijeme jeho aproximáciu, keďže jeho plná verzia je výpočtovo a časovo veľmi náročná. Metóda ponúka viačero nastaviteľných parametrov, pričom pre následujúcu konfiguráciu sme dosiahli najlepšie výsledky (vid'. Obr. 5.2).

- Maximálna veľkosť okna = 18
- Veľkosť bunky = 0,1
- Sklon = 2,85
- Počiatočná vzdialenosť (nad parametrizovaným povrchom) = 0,1
- Maximálna vzdialenosť (nad parametrizovaným povrchom) = 2,5



Obr. 5.2: Mračná bodov objektov (pôvodná farba) po separácii bodov zeme (biele farba) pomocou approximácie morfologického filtra (*SD1* hore, *SD2* dole)

	<b>Počet bodov zeme</b>	<b>Počet bodov objektov</b>	<b>Pomer bodov zeme a objektov[%]</b>
<b>SD1</b>	8 455 280	7 022 534	54,63 - 45,37
<b>SD2</b>	3 075 670	2 069 224	59,78 - 40,22
<b>SD3</b>	5 260 628	4 536 466	53,70 - 46,30
<b>SD4</b>	3 321 577	2 023 233	62,15 - 37,85
<b>SD5</b>	3 206 050	4 091 285	43,93 - 56,07

Tabuľka 5.2: Porovnanie pomeru bodov zeme a objektov, pre jednotlivé súbory dát

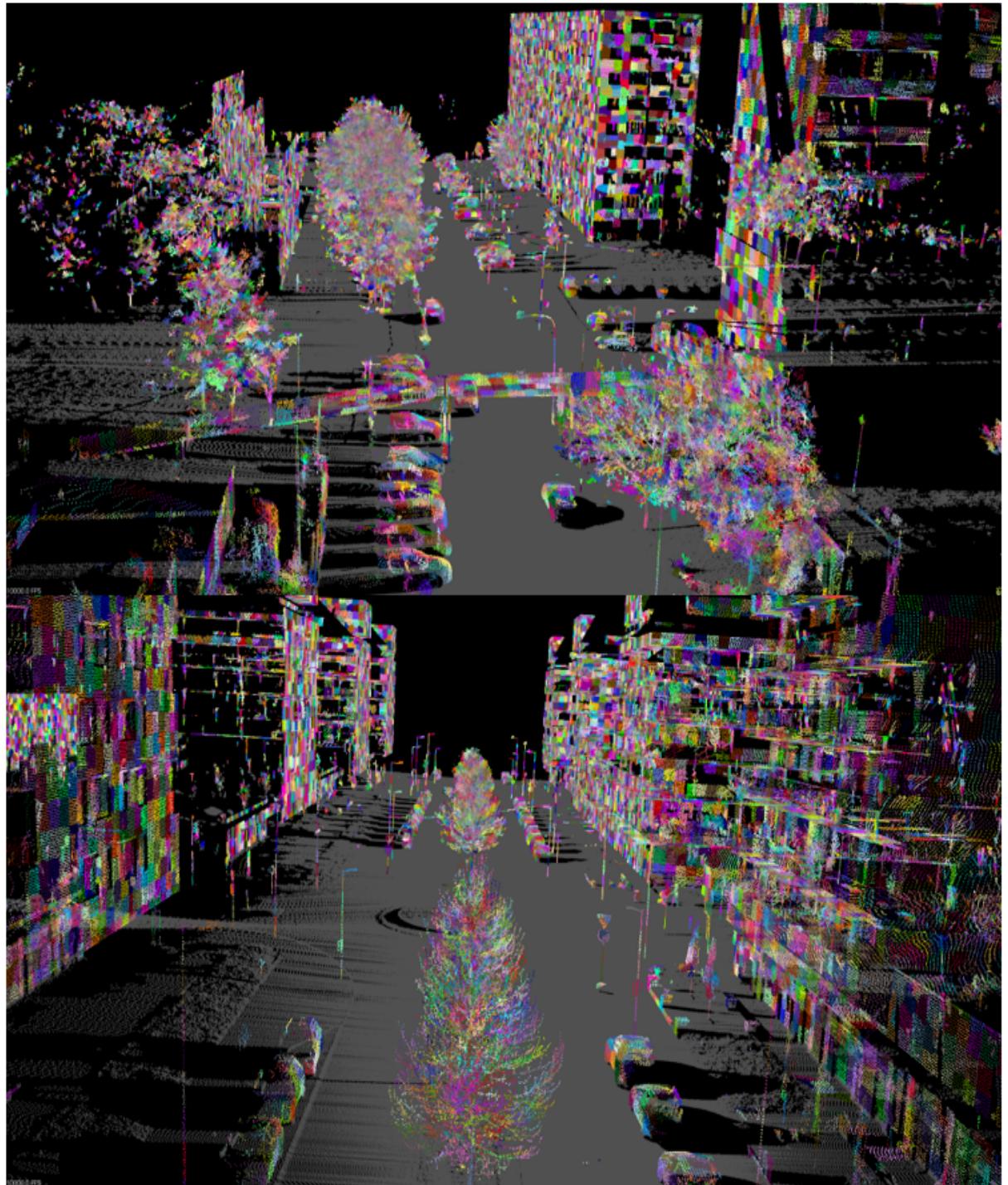
### 5.3 Zhlukovanie bodov objektov do supervoxelov

Pre účely použitia nasledujúcej segmentačnej metódy, je potrebne mračno bodov zoskupiť do tzv. supervoxelov, ktoré predstavujú významovo väčšie celky, ktoré majú podobné vlastnosti, zatiaľ čo sa snažia nepresahovať hranice objektov. Tieto zhluky taktiež predstavujú vyššiu úroveň reprezentácie mračna bodov, čím redukujú výpočtovú náročnosť ďalších procesov, zatiaľ čo uchovávajú dôležité štrukturálne informácie.

Zoskupenie mračna bodov vykonáme pomocou PCL knižnice, ktorá obsahuje triedu Supervoxel Clustering. Táto trieda vykonáva zhlukovanie využitím VCCS (Voxel Cloud Connectivity Segmentation), čo je nová metóda, ktorá využíva rastový variant k-means zhlukovania, pričom body označuje priamo do voxelov štruktúry oktálového stromu. Pomocou oktálového stromu efektívne uchováva graf súsednosti 26 susedov (spojený hranou, tvárou alebo vrcholom), pričom tento graf je vo veľkej mieri použitý pre rast regiónov zhlukov, ako aj pre uchovanie súsednosti výsledných supervoxelov. Rast supervoxelov je riadený normalizovanými priestorovými, farebnými a normálovými vzdialosťami, pričom prvá meria priestorový rozsah, druhá meria rozdiel farby v RGB priestore a posledná meria uhol medzi normálami povrchu. [29]

Trieda Super-voxel Clustering ponúkala nastavenie piatich parametrov, pričom hodnota rozlíšenia voxelov, rozlíšenia počiatočných bodov a dôležitosť normál hrala najvýznamnejšiu úlohu. Výsledná konfigurácia parametrov je nasledovná:

- Rozlíšenie voxelov = 0,1
- Rozlíšenie počiatočných bodov (z ang. seed) = 1,0
- Dôležitosť normál = 1,0
- Dôležitosť farby = 0,2
- Priestorová dôležitosť = 0,2



Obr. 5.3: Mračná bodov objektov zoskupené do supervoxelov rôznej farby (SD1 hore, SD2 dole)

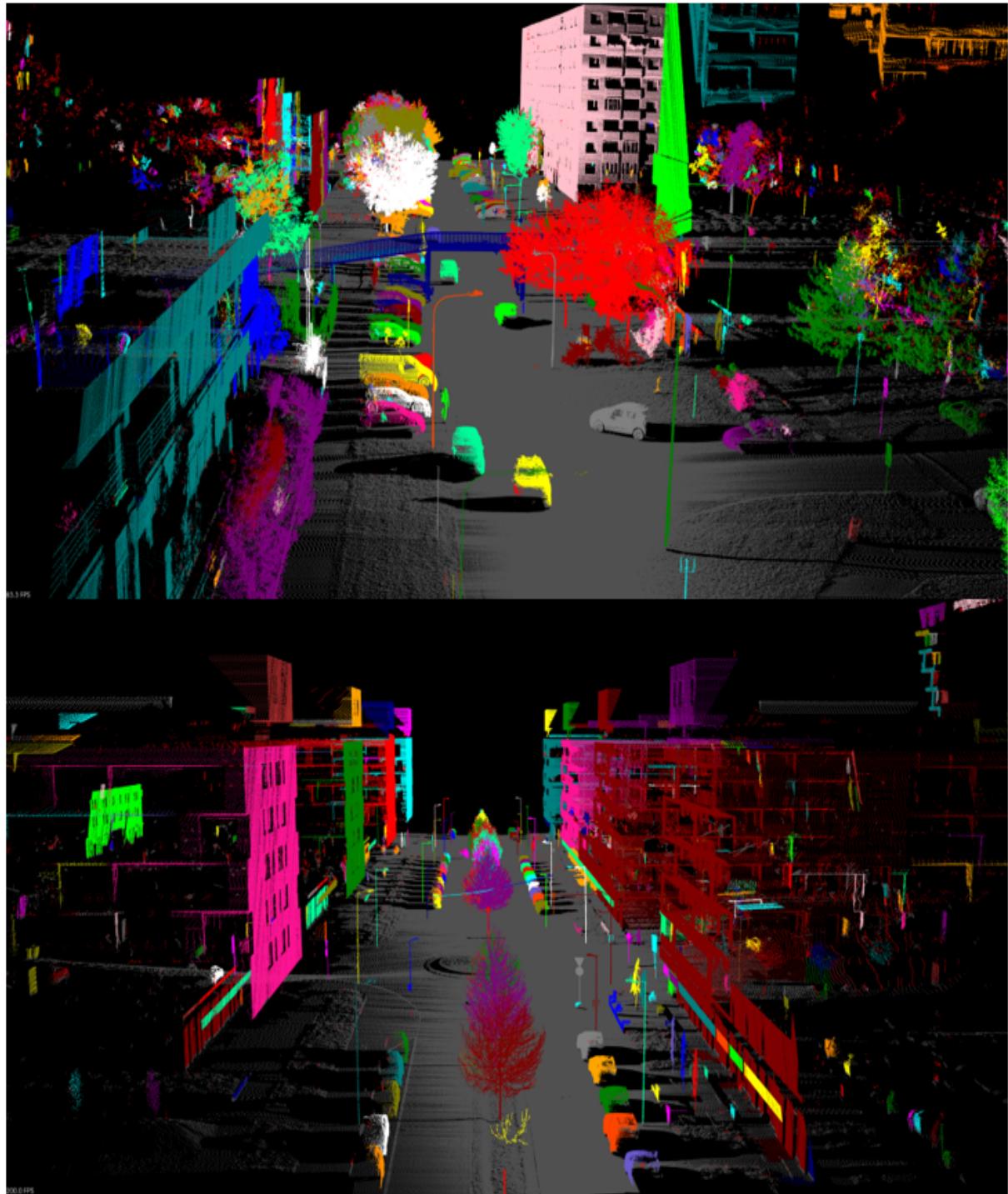
## 5.4 Segmentácia mračna bodov

Výsledkom predošlých krokov je samotná segmentácia mračna bodov objektov. Táto segmentácia nám rozdelí body mračna na jednotlivé objekty, čo následne umožní individuálny prístup pri ďalších krokoch, ako sú podvzorkovanie alebo rekonštrukcia povrchu. Vďaka segmentácii budeme mať väčšiu kontrolu a dosiahneme lepšie výsledky.

Pre segmentáciu mračna bodov využijeme PCL implementáciu CPC (Constrained Planar Cuts) metódy. Táto metóda využíva graf súsednosti vytvorený v predošom kroku pomocou VCCS, na to aby získala informáciu o lokálnych konkávnostiach/konvexnostiach, pomocou čoho vytvára EEC (Euclidean Edge Cloud). Toto mračno hrán reprezentuje každý bod, ako hranu grafu súsednosti, pričom uchováva veľkosť hrany a jej smer (konvexný alebo konkávny). Na základe tejto reprezentácie, je možné využiť geometricky obmedzený deliaci model, pre nájdenie možných rezov. Pre určenie roviny rezu je použitý vážený RANSAC, ktorý sa snaží maximalizovať hodnotiacu funkciu, pričom body s konkávnym smerom hrany sú uprednostňované pred tými so konvexnou. Mračno bodov je na základe tejto roviny rekurzívne rezané, až pokým nie sú dosiahnuté ukončovacie podmienky, čím získavame výsledné segmentované mračno bodov. [30]

Samotná PCL implementácia ponúka viacero nastaviteľných parametrov, pričom rôzne konfigurácie majú svoje výhody a nevýhody. Výsledná konfigurácia vyzerá následovne:

- Prah tolerancie konkávnosti (uhol v stupňoch) = 15,0
- Maximálny počet rezov = 20
- Minimálna veľkosť rezaného segmentu = 300
- Minimálna veľkosť segmentu (menšie budu zlúčené) = 150
- Počet RANSAC iterácií = 3000
- K-faktor (faktor pre rozšírenú kontrolu konvexnosti) = 0
- Minimálne rezné skóre (skóre na vykonanie rezu) = 0,3
- Prah hladkosti (kontrola hladkosti hrán) = 0,1
- Použitie kritéria rozumnosti (odstránenie oblastí s jedným spojením) = nie
- Použitie lokálnych obmedzení rezania = áno
- Použitie usmerneného rezania (kolmé rezanie na hranu) = nie
- Čisté rezanie (rezanie hrán so suprevoxelmi na opačných stranách roviny) = áno



Obr. 5.4: Mračná bodov segmentované na jednotlivé objekty, označené rôznou farbou (*SD1* hore, *SD2* dole)

## 5.5 Podvzorkovanie mračna bodov

Ďalším krokom pred samotnou rekonštrukciou povrchu je podvzorkovanie mračna bodov. Tento krok nám zníži celkový počet bodov, čo zabezpečí nie len zníženie pamäťovej náročnosti, ale aj zníženie výpočtového času pri rekonštrukcii povrchu. Pre tento účel budeme používať dve odlišné metódy, pričom budeme rozdielne pristupovať ku bodom zeme a ku bodom objektov.

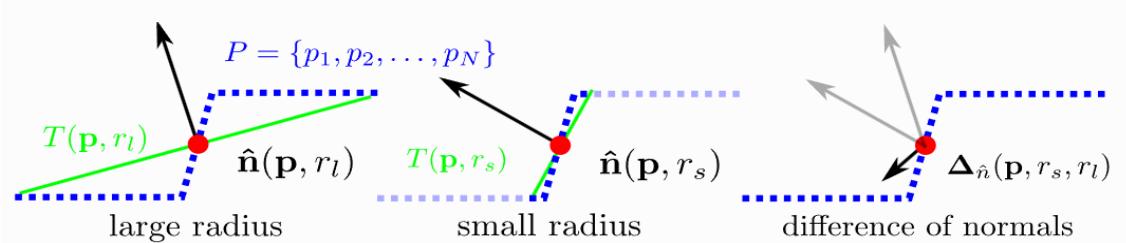
### 5.5.1 Podvzorkovanie bodov objektov

Ako prvými sa budeme zaoberať bodmi objektov, ktoré obsahujú pomerne veľa detailov, ktoré chceme zachovať, a preto zvolíme adaptívny prístup podvzorkovania.

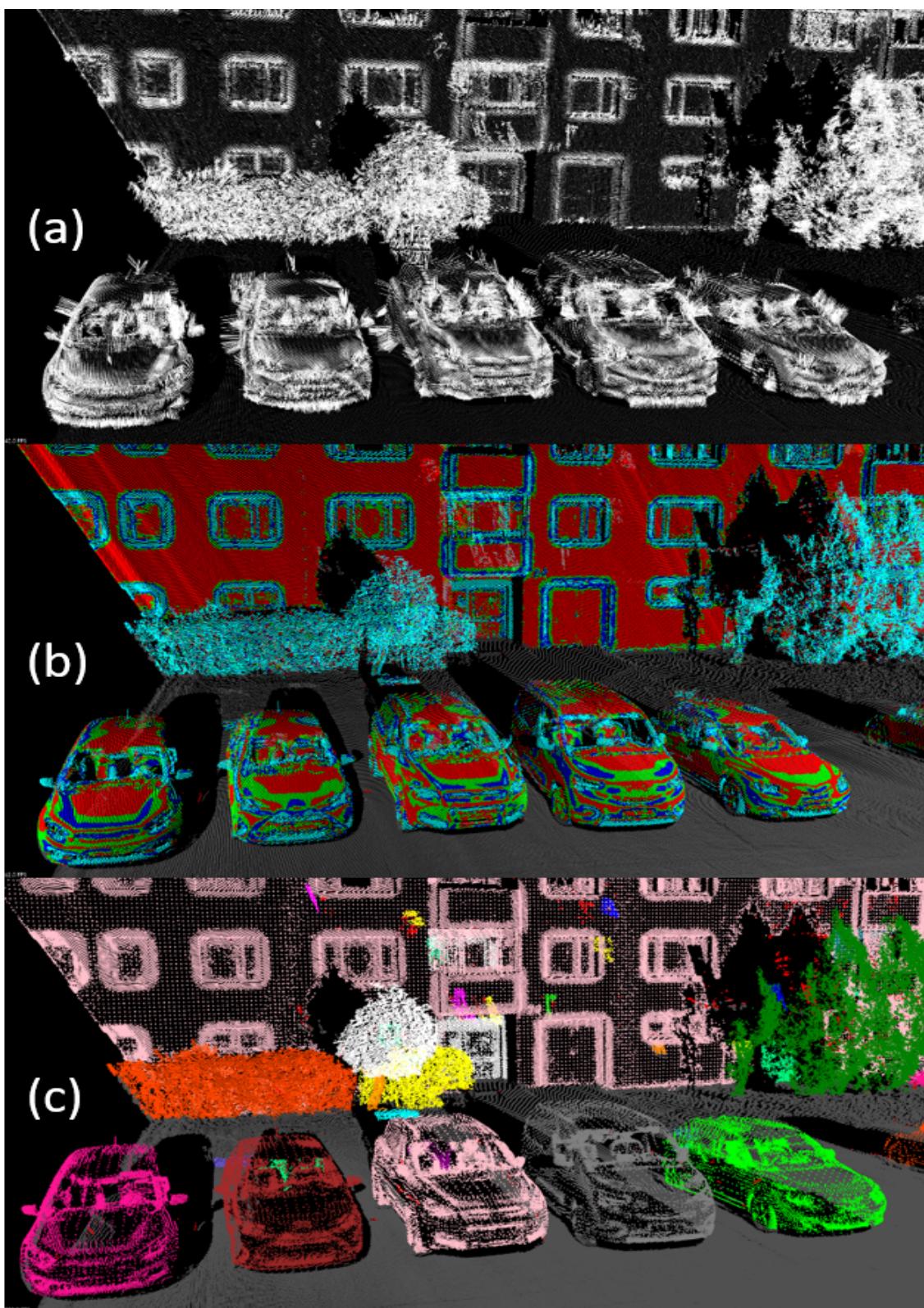
Pre tento účel využijeme DoN (Difference of Normals), pomocou ktorého klasifikujeme jednotlivé body objektu podľa ich dôležitosti. Táto metóda je založená na odhadе normál pomocou dvoch rôznych rádiusov, v ktorých vyhodnocuje body. Myšlienka je tá, že ak je odhad normál robený na rovnom povrchu, pre obidve rádiusy bude smer normály podobný, a tým pádom ich differencia bude minimálna. Naopak ak je tento odhad robený na zakrivenom povrchu ich smer bude rozdielny, a tým pádom aj ich differencia bude veľká (vid. Obr. 5.5). Na základe veľkosti novej normály (0 - 1), ktorá vznikla diferenciou, vieme určiť úroveň zakrivenia povrchu v danom bode, a teda môžeme dané body klasifikovať podľa ich úrovne dôležitosti, pričom viac zakrivený povrch je dôležitejší.

Pre samotné podvzorkovanie využijeme voxelovú mriežku, ktorá rozdelí priestor bodov na rovnomerné voxely, pričom vždy zachová iba bod najbližší ku tăžisku voxelu. Pre rôzne triedy dôležitosti sme využili rôzne rozlíšenie voxelovej mriežky, ktoré prináleží dôležitosti danej triedy. Objekty sme rozdelili do štyroch tried a podvzorkovali sme ich pri nasledovnej konfigurácii parametrov:

- Rádiusy pre odhad normál (malý, veľký) = 0,03-0,07 ; 0,40
- Veľkosť diferencie normál (trieda 1-4) = (0-0,06) ; (0,06-0,10) ; (0,10-0,13) ; (0,13-1,0)
- Rozlíšenie voxelovej mriežky (trieda 1-4) = 0,13 ; 0,05 ; 0,02 ; 0,00 (bez podvzorkovania)



Obr. 5.5: Ukážka princípu fungovania DoN na zakrivenom povrchu [31]



Obr. 5.6: (a) Mračno diferencií normál (b) Klasifikované mračno na základe dôležitosti bodov v vzostupnom poradí červené, zelené, modré a najdôležitejšie tyrkysové (c) Výsledné podvzorkované mračno bodov objektov ( $SD1$ )

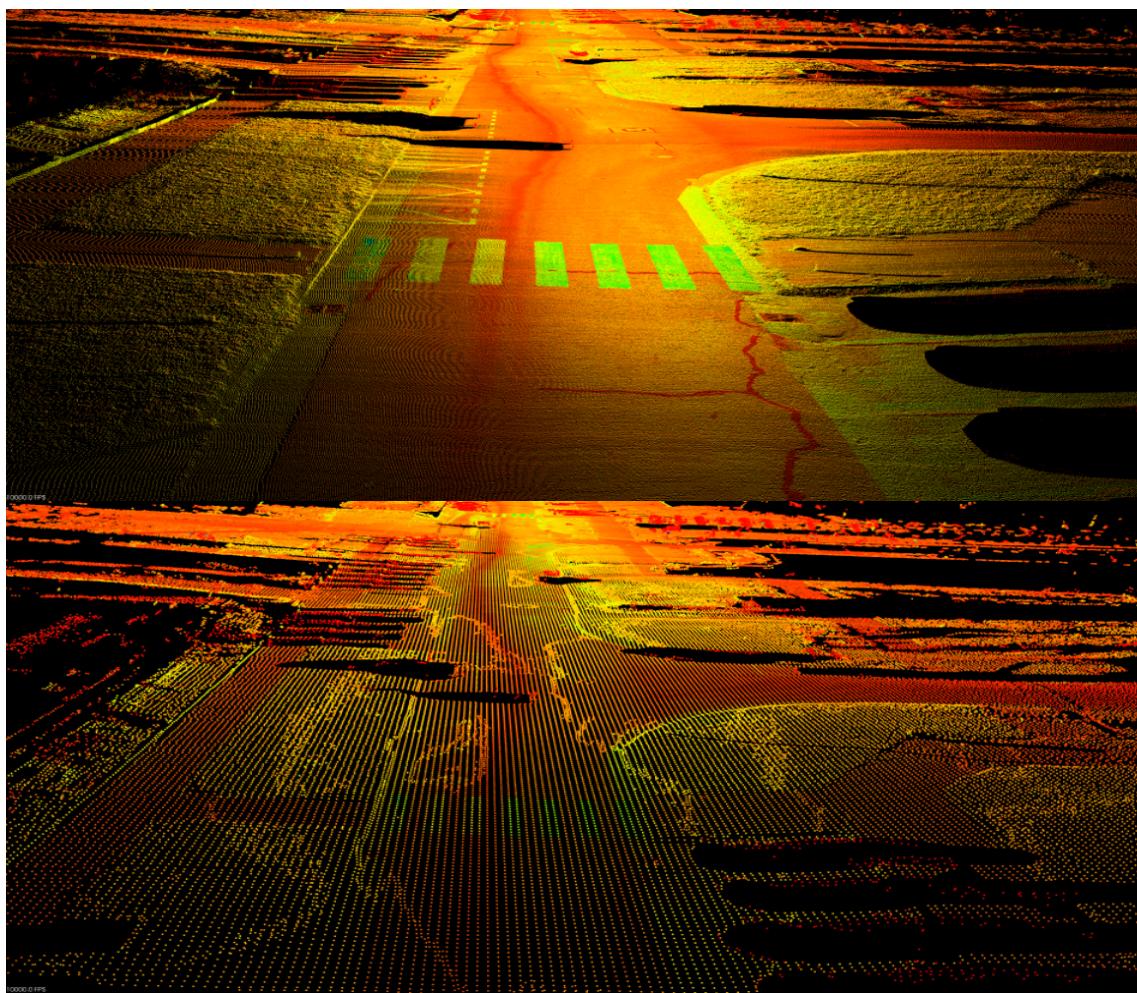
### 5.5.2 Podvzorkovanie bodov zeme

Po podvzorkovaní bodom objektov ostávajú iba body zeme. Tieto body v druhej väčšine neobsahujú toľko detailov ako body objektov, a preto na ich podvzorkovanie využijeme iba druhú časť predošlého postupu, a teda podvzorkujeme ich pomocou voxelovej mriežky s fixným rozlíšením, čo zabezpečí vysokú úroveň podvzorkovania.

Toto rozhodnutie sme urobili hlavne z dôvodu, že body zeme obsahujú hlavne rovinné plochy, ktoré majú medzi sebou pomerne hladký prechod. Ak by sme chceli využiť predošlý postup aj na body zeme, zbytočne by sme zvýšili ponechaný počet bodov, nakoľko oblasti s nízkou vegetáciou, ako je tráva, obsahujú pomerne nerovný povrch, pričom aj pri ponechaní viacerých bodov, sú zložité na dôveryhodnú rekonštrukciu povrchu.

Konfigurácia pre podvzorkovanie bodov zeme je nasledovná:

- Rozlíšenie voxelovej mriežky = 0,20



Obr. 5.7: Mračno bodov zeme pred (hore) a po (dole) podvzorkovaní. (SD1)

	<b>Nový počet bodov zeme</b>	<b>Nový počet bodov objektov</b>	<b>Zmenšenie počtu bodov zeme [%]</b>	<b>Zmenšenie počtu bodov objektov [%]</b>	<b>Celkové zmenšenie počtu bodov [%]</b>	<b>Celková potrebná pamäť [MB]</b>
<b>SD1</b>	244 262	4 229 627	97,11	39,77	71,09	64,60
<b>SD2</b>	192 495	1 297 412	93,74	32,01	68,91	23,10
<b>SD3</b>	237 870	1 468 073	95,47	60,89	79,46	29,00
<b>SD4</b>	63 678	1 204 318	98,08	37,67	75,21	19,00
<b>SD5</b>	582 313	3 197 181	81,83	21,31	47,90	54,70

Tabuľka 5.3: Porovnanie zmenšenia počtu bodov pred (vid'. Tab. 5.2) a po podvzorkovaní mračien bodov

## 5.6 Rekonštrukcia povrchu

Predošlým predspracovaním dostávame podvzorkované mračno bodov, očistené od odlahlých bodov, ktoré je zároveň segmentované na jednotlivé objekty, vďaka čomu môžeme prejsť na samotnú rekonštrukciu povrchu. Prvým krokom pri rekonštrukcií povrchu bude výber použitej metódy, ktorá dosahuje najlepšie výsledky.

### 5.6.1 Výber metódy

Pri výbere metódy na rekonštrukciu sa zameriame na metódy, ktoré ponúka PCL knižnica, a to sú Poissonová rekonštrukcia, Marching cubes (Hoppe) a GPT (Greedy projection triangulation). Principiálne fungovanie prvých dvoch bolo bližšie vysvetlené v kapitolách 2.1.2, 2.1.1 a GPT metódu si v krátkosti vysvetlíme teraz.

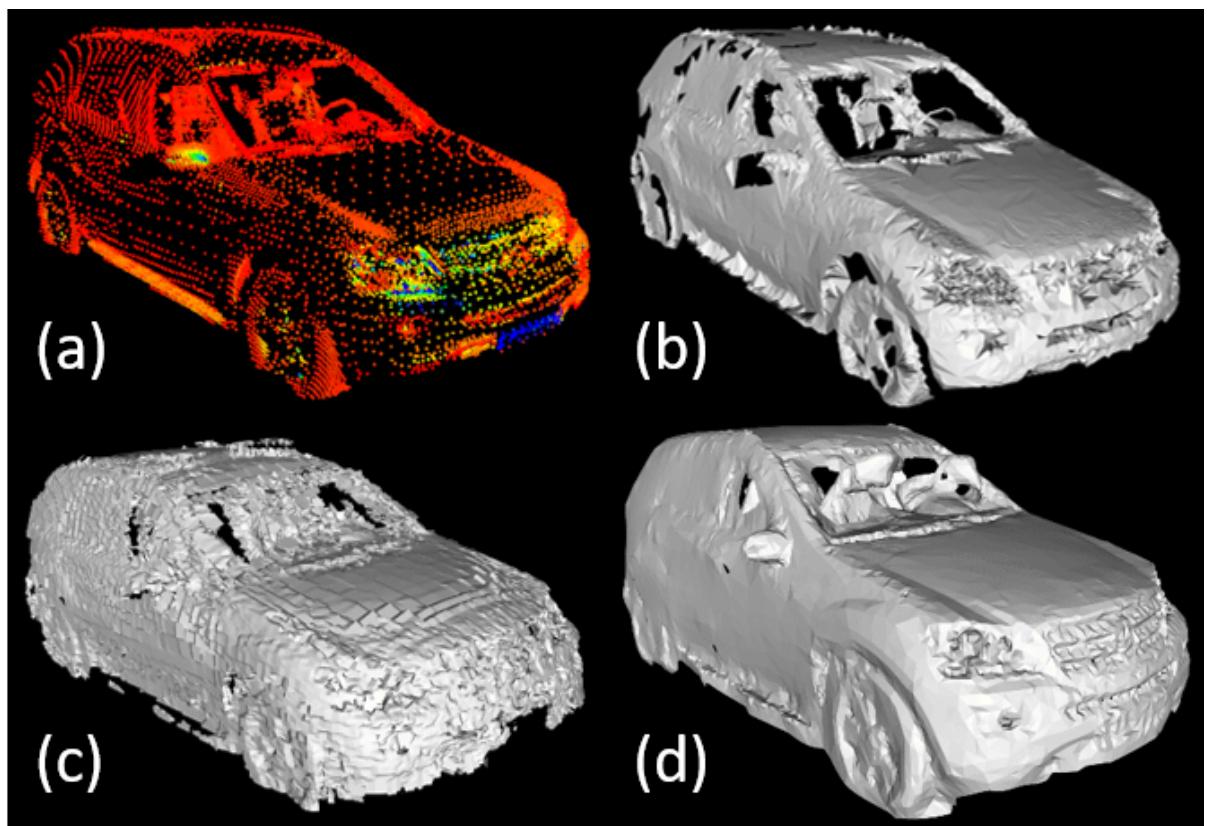
Fungovanie tejto metódy je pomerne jednoduché a vieme ju rozdeliť do štyroch základných krovok. Prvým krokom je nájdenie lokálnych susedstiev pre každý bod, ktoré je zložené z  $K$  najbližších susedov. Následne je vykonaný odhad normál každého referenčného bodu, pomocou čoho je vytvorená rovina, ktorá je kolmá na smer normály a všetky body zo susedstva sú projektované na túto rovinu. Na tejto rovine je použitý priestorovo rastúci algoritmus, ktorý sa z počiatočného trojuholníku rozrastá, až kým nie je skompletizovaný celý povrch, pričom dodržuje obmedzenia. Na záver sa projektovaná reprezentácia vráti do pôvodného 3D priestoru, čím dostávame výsledný povrch. [32]

Pri výbere používanej metódy sa zameriame na hodnotiace faktory, ako sú dôveryhodnosť rekonštruovaného povrchu, časová náročnosť a výpočtová náročnosť (obmedzenia hardvéru).

Na obrázku 5.8 môžeme vidieť dosiahnuté výsledky jednotlivých metód, ktoré boli upravené pre potreby porovnávania. Čo sa týka dôveryhodnosti najlepšie výsledky dosahuje Poissonová metóda, ktorá vytvára hladký vodotesný povrch, ktorý dobre zachytáva aj menšie detaily. Porovnatelne dobrá je aj GPT metóda, ktorá je ale parametricky závislá od veľkosti a hustoty objektu, čo vedie ku nežiadaným dieram v povrchu. Posledná je Marching cubes metóda, ktorá vytvára kockovaný povrch, ktorý nie je prirodzený a vedie ku skreslovaniu výzoru objektu.

V prípade časovej a výpočtovej náročnosti bola najlepšia GPT metóda, ktorá povrch ukážkového objektu zrekonštruovala najrýchlejšie (1,35s), pričom porovnatelne rýchla bola aj Poissonová rekonštrukcia (1,74s), ktorá ale následne potrebovala ďalšie úpravy. Najpomalšia bola Marching cubes (16,26s), ktorá ako jediná mála hardvérový problém a nebolo ju možné skúsiť na objektoch s väčším množstvom bodov.

S predošlých výsledkov bolo rozhodnuté, že sa budeme pokračovať s Poissonovou metódou rekonštrukcie povrchu, nakoľko vizuálne dosahovala najlepšie výsledky a nie je parametricky závislá od hustoty mračna bodov. Metóda má ale svoje nedostatky, ktoré budú prebrané v ďalšej podkapitole.

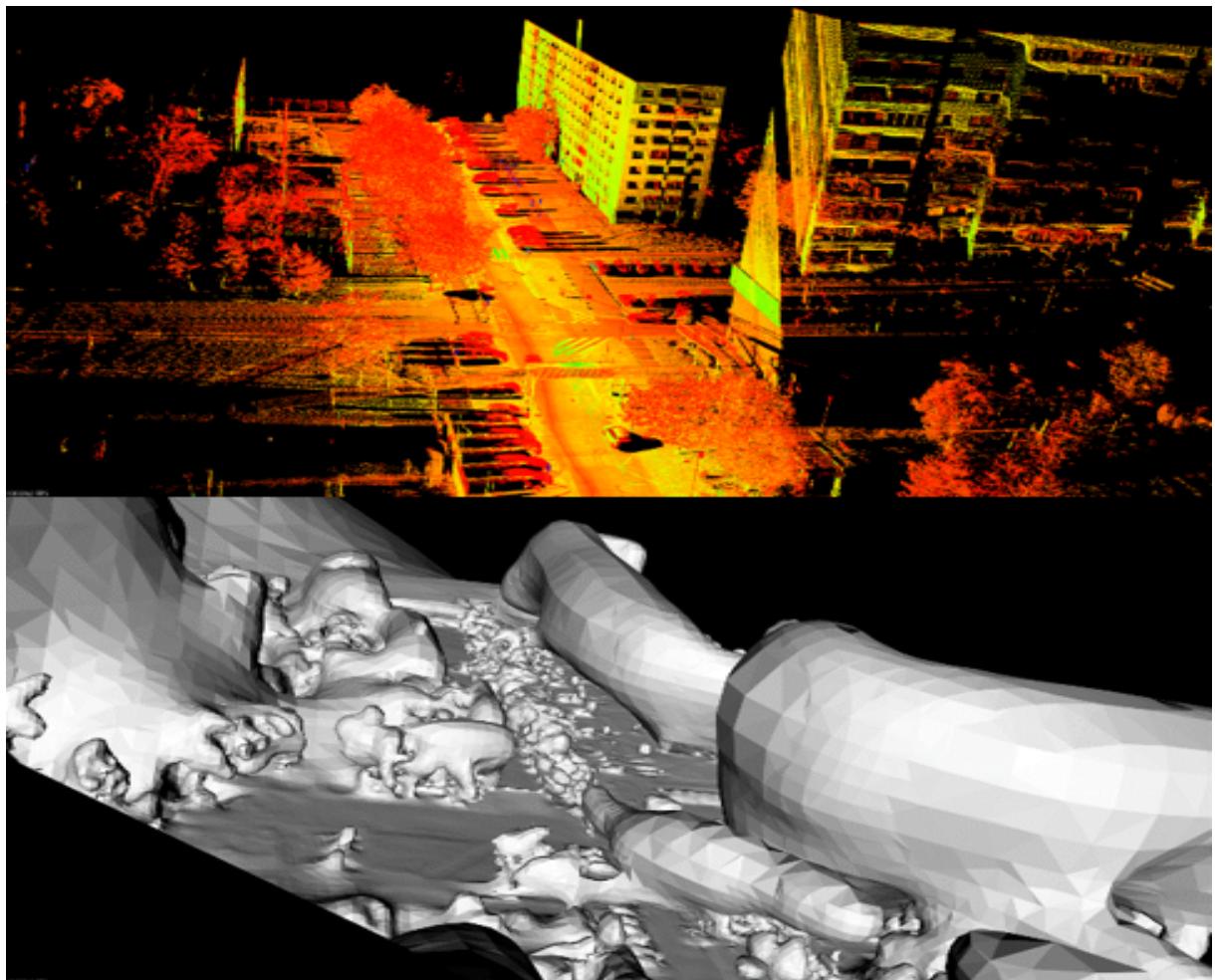


Obr. 5.8: Porovnanie rôznych metód rekonštrukcie povrchu na objekte auta - (a) Podvzorkované mračno bodov (b) GPT (c) Marching cubes (d) Poissonová rekonštrukcia (SD1)

## 5.6.2 Odstránenie nedostatkov

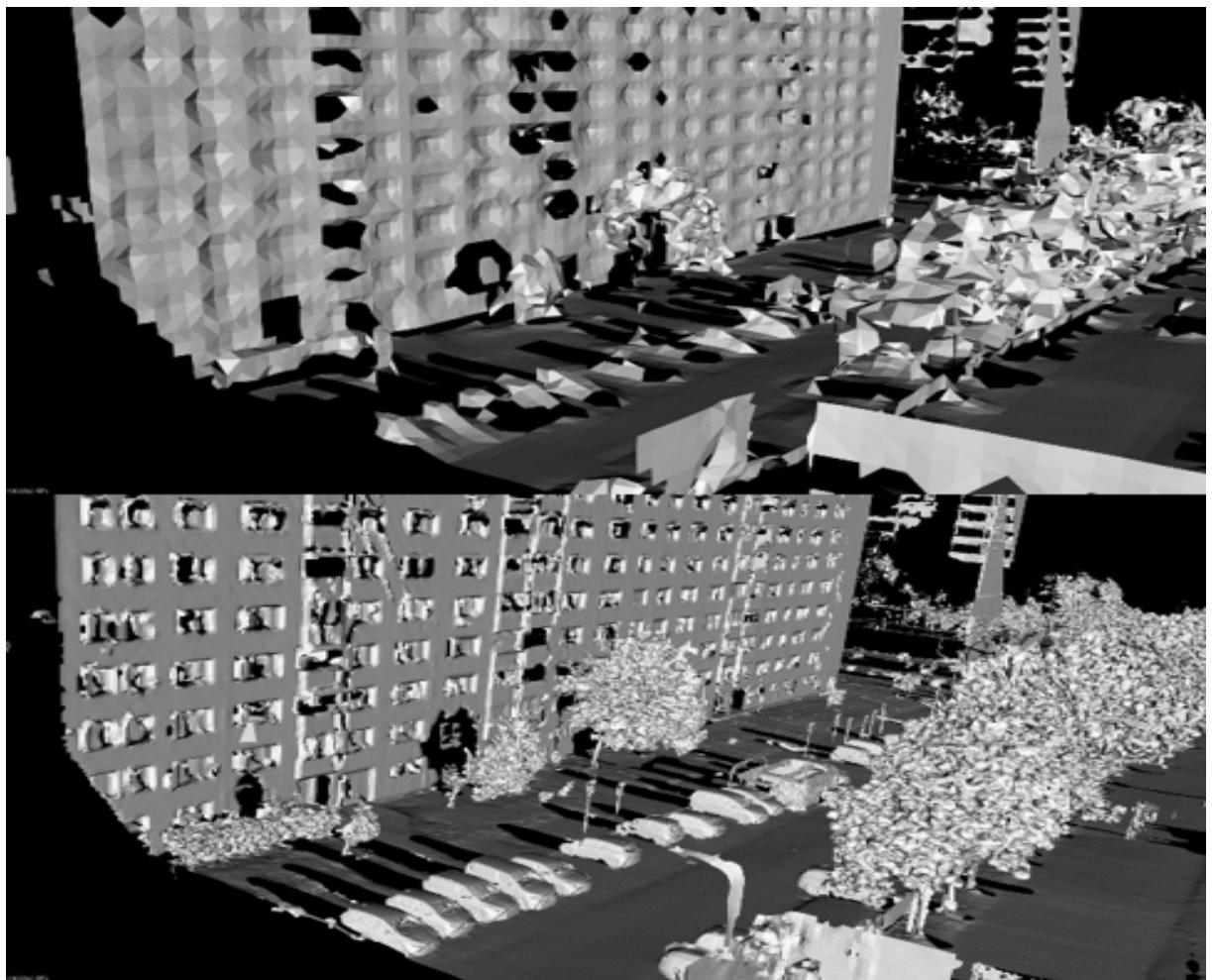
V predošej podkapitole bolo rozhodnuté, že rekonštrukcia povrchu bude vykonávaná pomocou Poissonovej rekonštrukčnej metódy, nakoľko dosahovala najlepšie výsledky. Tieto výsledky boli ale dosiahnuté, až po odstránení niekoľkých nedostatkov.

Prvým dôležitým nedostatkom bolo vytváranie nadbytočných polygónov (vid'. Obr. 5.9), ktoré vytvorili falošnú plochu, a tým pádom zničili dôveryhodnosť rekonštrukcie. Táto plocha je vytvorená na základe princípu fungovania Poissonovej metódy, pričom ukončenie jej vytvárania, je riadené pomocou hraničnej (z ang. boundary) podmienky, ktorá ale v PCL knižnici nie je nastaviteľná. Existuje viacero spôsobov, ako túto plochu odstrániť, pričom v našom prípade využijeme pôvodné mračno bodov a vyhľadávanie najbližšieho suseda pomocou kd-stromu. Pre každý vrchol polygónu nájdeme najbližšieho suseda v pôvodnom mračne bodov a na základe určeného prahu vzdialosti vyhodnotíme, či sa jedná o falošný polygón, ktorý následne odstráime.



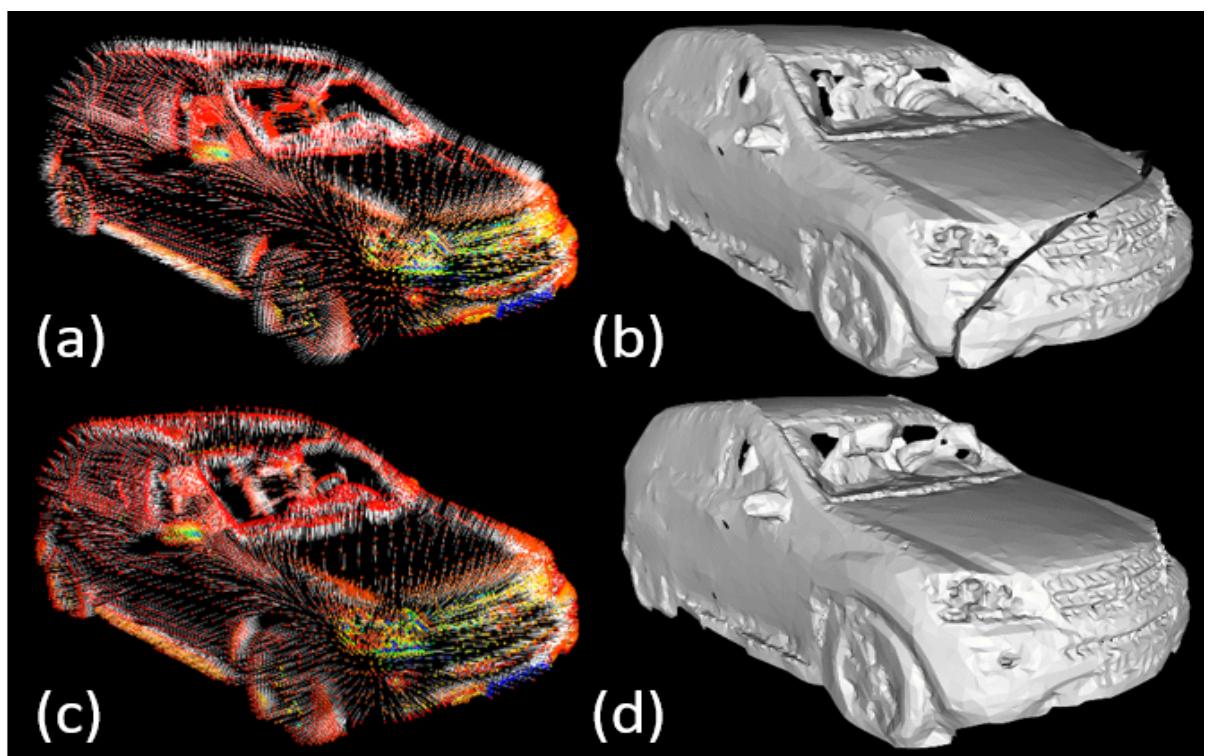
Obr. 5.9: Podvzorkované mračno bodov (hore), plocha s nadbytočnými polygónmi (dole) (SD1)

Po odstránení falošných polygónov, získavame očistený povrch (vid'. Obr. 5.10). Plocha v hornej časti bola vytvorená použitím Poissonovej metódy na celé mračno bodov naraz, čo malo za následok pomerne malé rozlíšenie povrchu a stratu potrebných detailov. Tento problém bol vyriešený pomocou využitia predošej segmentácie, čo nám umožnilo použiť rekonštrukciu na jednotlivé objekty osobitne a zabezpečilo zvýšenie zachovanej presnosti.



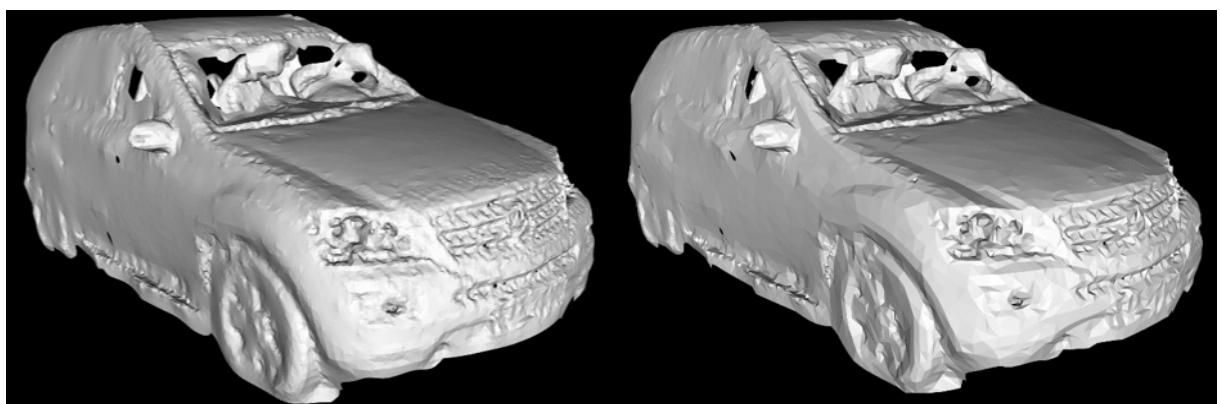
Obr. 5.10: Rekonštrukcia celého povrchu naraz (hore), rekonštrukcia individuálnych objektov (dole) (SD1)

Poissonová metóda je založená na správnom odhade normál, čo viedlo ku ďalšiemu problému (Obr. 5.11). Pri odhade normál nastáva nejasnosť, či majú byť normály otočené o 180 stupňov alebo nie, pričom o tom rozhoduje umiestnenie tzv. hľadiska. Použité sady dát boli získané s pohyblivého vozidla, pričom objekty nasnímali z viacerých uhlov, čo znamená, že každý objekt má niekoľko hľadísk. Nesprávne umiestnenie hľadiska má za následok vznik trhlín, čo sa ale dá vo veľkej miere vyriešiť, pomocou umiestnenia hľadiska do ľahška menších objektov alebo naopak ďaleko od väčších objektov.



Obr. 5.11: (a) Nesprávny odhad normál (b) Rekonštrukcia zo nesprávneho odhadu normál (c) Správny odhad normál (d) Rekonštrukcia zo správneho odhadu normál (*SD1*)

Posledný problém Poissonovej metódy, bolo vytvorenie príliš veľkého počtu polygónov, aj napriek predošlému podvzorkovaniu. Tento problém sme vyriešili pridaním kvadratickej decimácie vrcholov polygónov (vid' . kapitola 2.2.1), ktorá zabezpečila zníženie pamäťovej náročnosti povrchu a súčasné zachovanie detailov (vid' . Obr. 5.12), pričom predošlé podvzorkovanie zlepšilo jeho efektivitu.



Obr. 5.12: Porovnanie originálneho objektu (vľavo) so zdecimovaným (vpravo) pri znížení pamäťovej náročnosti o 70%

Po odstránení predošílých problémov, dosiahla rekonštrukcia vysoké zníženie pamäťovej náročnosti (vid' Tab. 5.4) a súčasné zachovania detailov. Celý postup rekonštrukcie bol vykonaný pre následujúcu konfiguráciu parametrov:

- Hĺbka oktálového stromu (objekty, zem) = 9 ; 13
- Hĺbka riešenia Laplacej rovnice = 10
- Hĺbka extrahovaného izo povrchu = 10
- Počet vzoriek na uzol (pre vytvorenie ďalšej úrovne v oktálovom strome) = 2
- Prah vzdialenosť falošného polygónu = 0,013
- Redukčný faktor decimácie = 0,70-0,95

	<b>Počet vrcholov</b>	<b>Počet polygónov</b>	<b>Potrebná pamäť [MB]</b>	<b>Zmenšenie potrebnej pamäti oproti mračnu bodov [%]</b>
<b>SD1</b>	2 382 215	4 099 270	78,00	64,81
<b>SD2</b>	998 040	1 683 075	32,20	81,72
<b>SD3</b>	1 454 153	2 639 484	49,30	64,83
<b>SD4</b>	741 312	1 327 302	24,90	67,41
<b>SD5</b>	2 261 276	4 095 192	76,60	26,83

Tabuľka 5.4: Výsledné hodnoty rekonštruovaných povrchov pre jednotlive súbory dát

## 5.7 Textúrovanie povrchu

V poslednej časti sa zameriame na textúrovanie povrchu pomocou dát z kamery. Vytvorený povrch obohatíme o vizuálny realizmus, takže mu pridáme informácie o farbe, drsnosti a ďalších rôznych atribútoch, pomocou čoho nadobudne hĺbku a zvýši tak predstavu o priestore.

Ako prvý krok textúrovania, je namapovanie jednotlivých vrcholov polygónov, na príslušné UV súradnice z kamerovej fotky. Pre tento účel potrebujeme transformovať vrcholy polygónov do súradnicového systému kamery (použitím poskytnutých transformácií), čím zabezpečíme, že kamera je v počiatku súradnicového systému a následná projekcia bude jednoduchšia. Na obrázku 5.14, môžeme vidieť, že ide o 360 stupňový ekvidistantný (z ang. equirectangular) typ fotky, čo znamená, že dané vrcholy polygónov projektujeme na jednotkovú guľu, pričom ich UV súradnice sú rovné normalizovaným polárnym súradniciam gule, a teda môžeme ich získať pomocou následujúceho vzťahu:

$$U = \frac{\operatorname{atan}2\left(\frac{pt_y}{\sqrt{pt_x^2 + pt_y^2 + pt_z^2}}, \frac{pt_x}{\sqrt{pt_x^2 + pt_y^2 + pt_z^2}}\right)}{2\pi} + 0,5 \quad (5.1)$$

$$V = \frac{\operatorname{asin}\left(\frac{pt_z}{\sqrt{pt_x^2 + pt_y^2 + pt_z^2}}\right)}{2\pi} + 0,5 \quad (5.2)$$

kde  $pt$  je vrchol polygónu a UV súradnice sú normalizované v rozsahu (0-1).

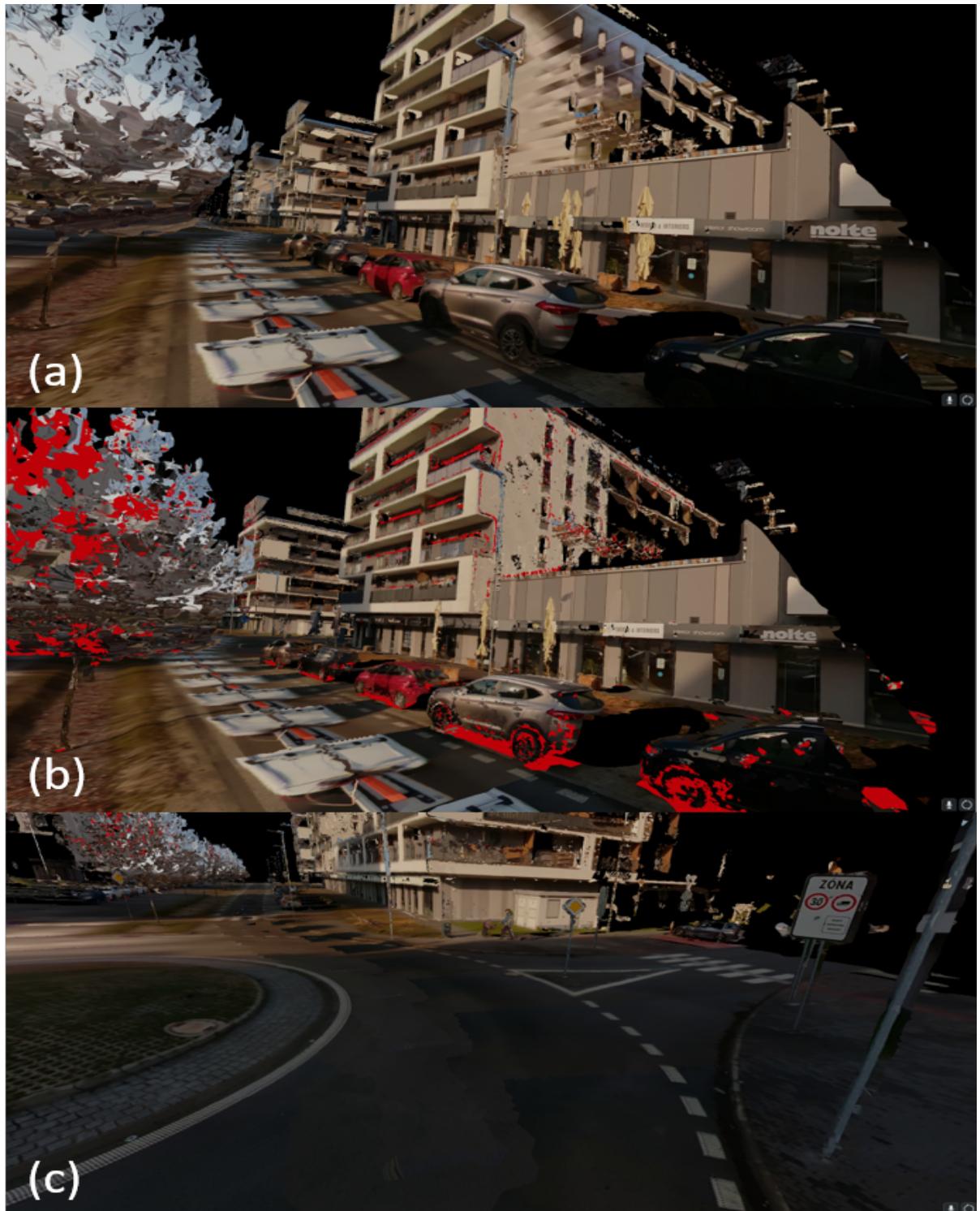
Ked' už vieme mapovať súradnice fotky na súradnice polygónov, musíme vyriešiť, ako textúrovať povrch z viacerých fotiek. Vo väčšine prípadov, objekty bližšie ku kamere bývajú ostrejšie, zatiaľ čo vzdialenejšie objekty, začínajú byť skreslené a rozmazané. Z tohto dôvodu si pre každý polygón ukladáme informáciu o najbližšej kamere, z ktorej ho následne otextujeme, čím zabezpečíme požadovaný výsledok (Obr. 5.13a).

Predošlým krokom sme zabezpečili pomerne dobrý výsledok, ale takýmto spôsobom dochádza ku textúrovaniu povrchov, ktoré v skutočnosti kamera nevidí. Využijeme preto PCL implementáciu sledovania lúčov (z ang. ray tracing), ktorá je založená na reprezentácii pomocou oktálového stromu. Metóda vysiela lúč z bodu kamery do jednotlivých vrcholov polygónov a zistuje, či dochádza ku kolízií s inými polygóňmi (voxelmi), ktoré ležia medzi nimi. Takto vieme zabezpečiť, aby bol texturovaný polygón skutočne viditeľný z danej kamery a predídeme tým falošnej projekcií textúry na nevidené povrhy (Obr. 5.13b).

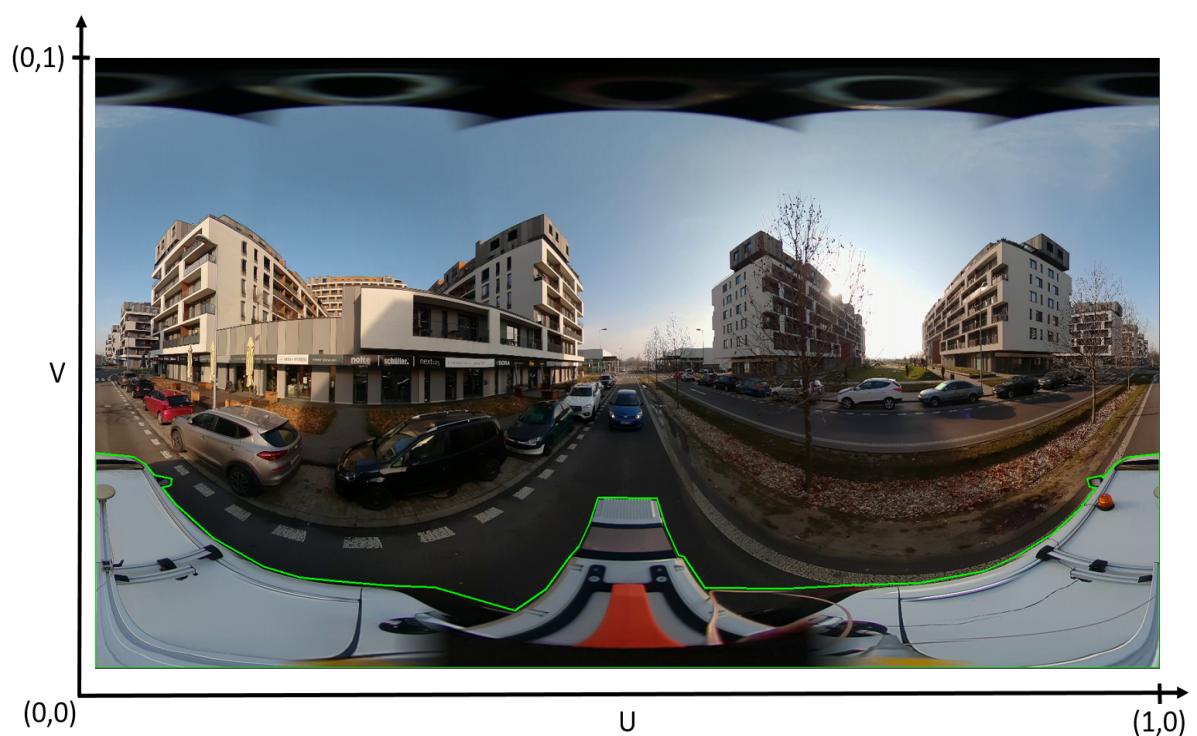
Posledný problém, ktorý sme riešili, bolo odstránenie statickej prekážky. Mohli sme si všimnúť, že na použitých fotkách sa nachádza samotné auto, z ktorého boli zbierané dátá a napokol'ko sa toto auto vždy nachádza najbližšie ku kamere, zanechávalo po sebe falošnú textúru na ceste. Tento problém sme vyriešili manuálnym zadefinovaním polygónu, ktorý obkresluje okraje auta (Obr. 5.14) a následnou detekciou, či daná UV koordinácia leží vo vnútri tohto polygónu, sme príslušné polygóny povrchu automaticky označili za nevidené, pričom neskôr boli otexturované z rozdielnej fotky (Obr. 5.13c).

Pre celý predošlý postup sme využívali PCL knižnicu, ktorá do istej miery ponúkala rozhranie pre mapovanie a textúrovanie povrchu, pričom výsledky boli dosiahnuté pri následujúcej konfigurácií parametrov:

- Hĺbka oktálového stromu = 3
- Rozlíšenie oktálového stromu = 0,05
- Počet použitých fotiek = 50 (20%)



Obr. 5.13: (a) Textúrovanie povrchu z najbližšej kamery (b) Textúrovanie z najbližšej kamery, iba ak kamera skutočne vidí povrch (červené polygóny nevidené žiadnou kamerou) (c) Odstránenie statickej prekážky auta na ceste (*SD2*)



Obr. 5.14: Ukážka kamerovej fotky s označeným autom, pomocou zeleného polygónu (*SD2*)

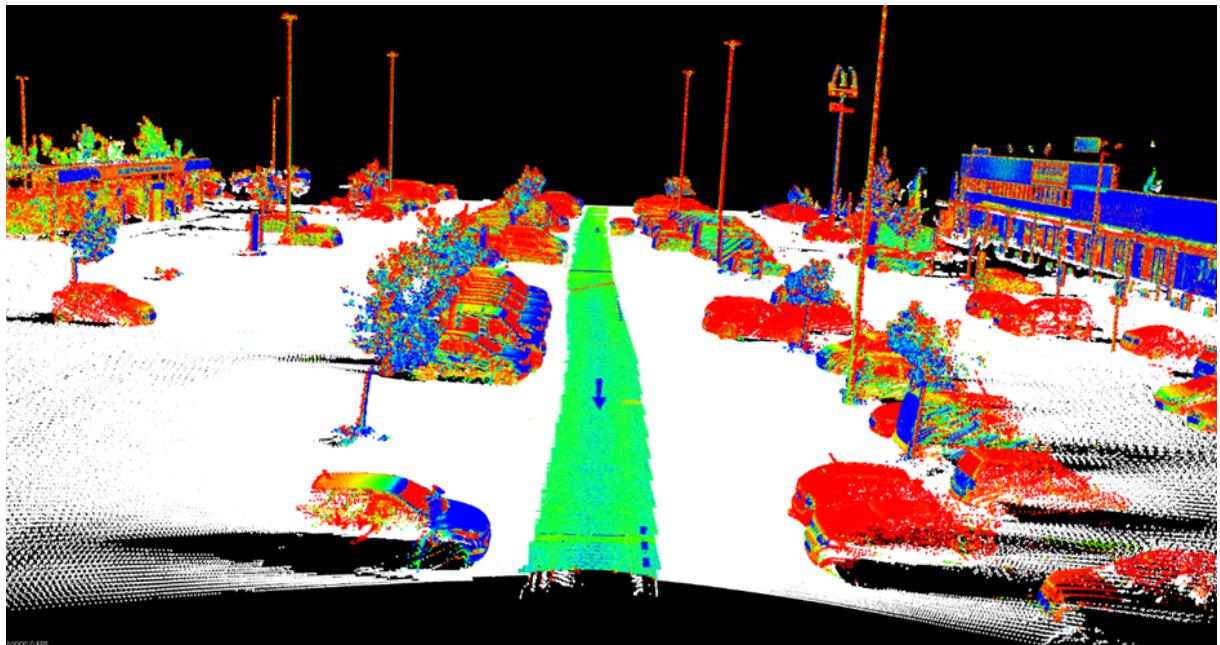
# 6 Overenie riešenia

V poslednej časti tejto práce overíme predošlé riešenie pomocou niekol'kých experimentov, pričom väčšina z nich bude vykonaná pomocou vizuálnej kontroly, nakoľko vo väčšine prípadov nemáme inú možnosť porovnania. Zameriame sa na jednotlivé kroky postupu, pričom ukážeme ich správnosť fungovania, vytkneme niektoré nedostatky a na záver ukážeme ich dôsledok na vytvorený povrch.

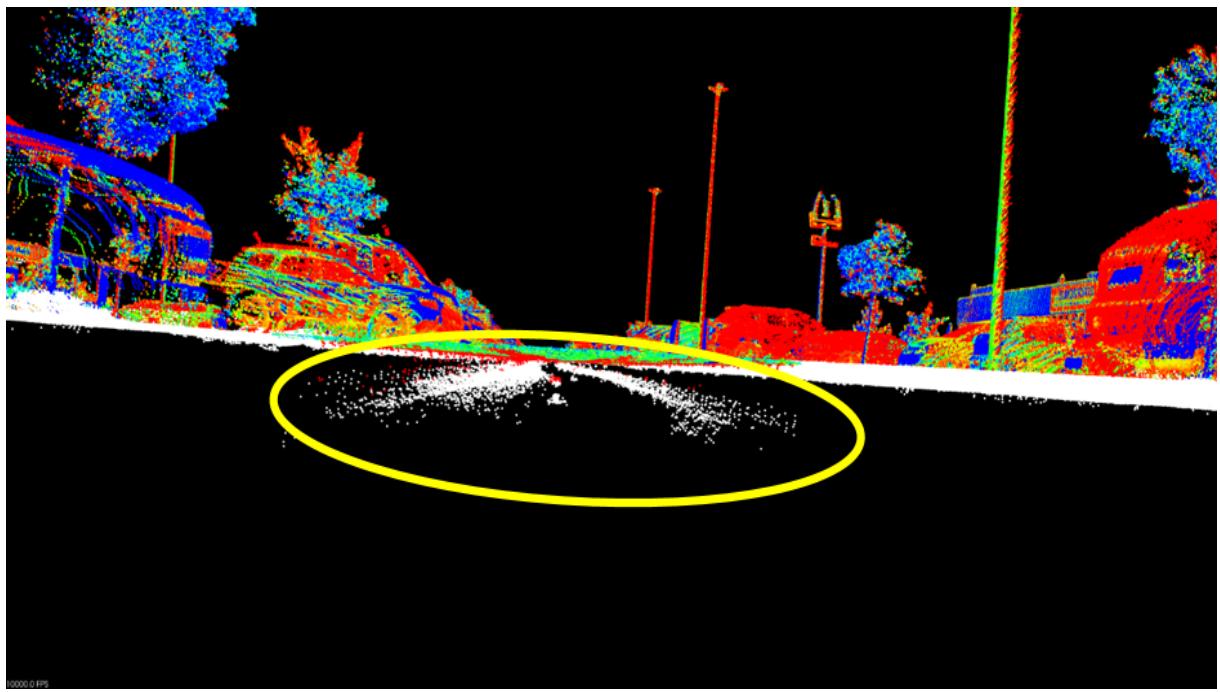
## 6.1 Overenie separácia bodov zeme

V tejto podkapitole sa zameriame na separácia bodov zeme, pričom pri tomto kroku hral dôležitú úlohu šum, a preto sa zameriame hlavne na neho. Pre tento účel sme využili sady dát *SD3* a *SD5*, nakoľko slúžia ako validačné súbory dát a navyše boli získané pomocou rozdielneho LIDAR-u, ktorý obsahuje vyššiu úroveň zle nasnímaných bodov, v podobe šumu.

Nedostatky progresívneho morfologického filtra môžeme vidieť v prípade, kedy auto našimalo falošné body na trase ktorou prešlo (Obr. 6.1 a 6.2). Môžme si všimnúť, že body boli husto zoskupené, a tým pádom neboli odstránené pomocou filtrovania odľahlých bodov, čo zapríčinilo nesprávnosť fungovania filtra v tejto oblasti. S princípu fungovania progresívneho morfologického filtra vychádza, že v danom "okne" vyhodnotí najnižšie body, ako body zeme, čo v tomto prípade spôsobilo, že falošné body boli vyhodnotené ako body zeme, namiesto skutočných nad nimi.

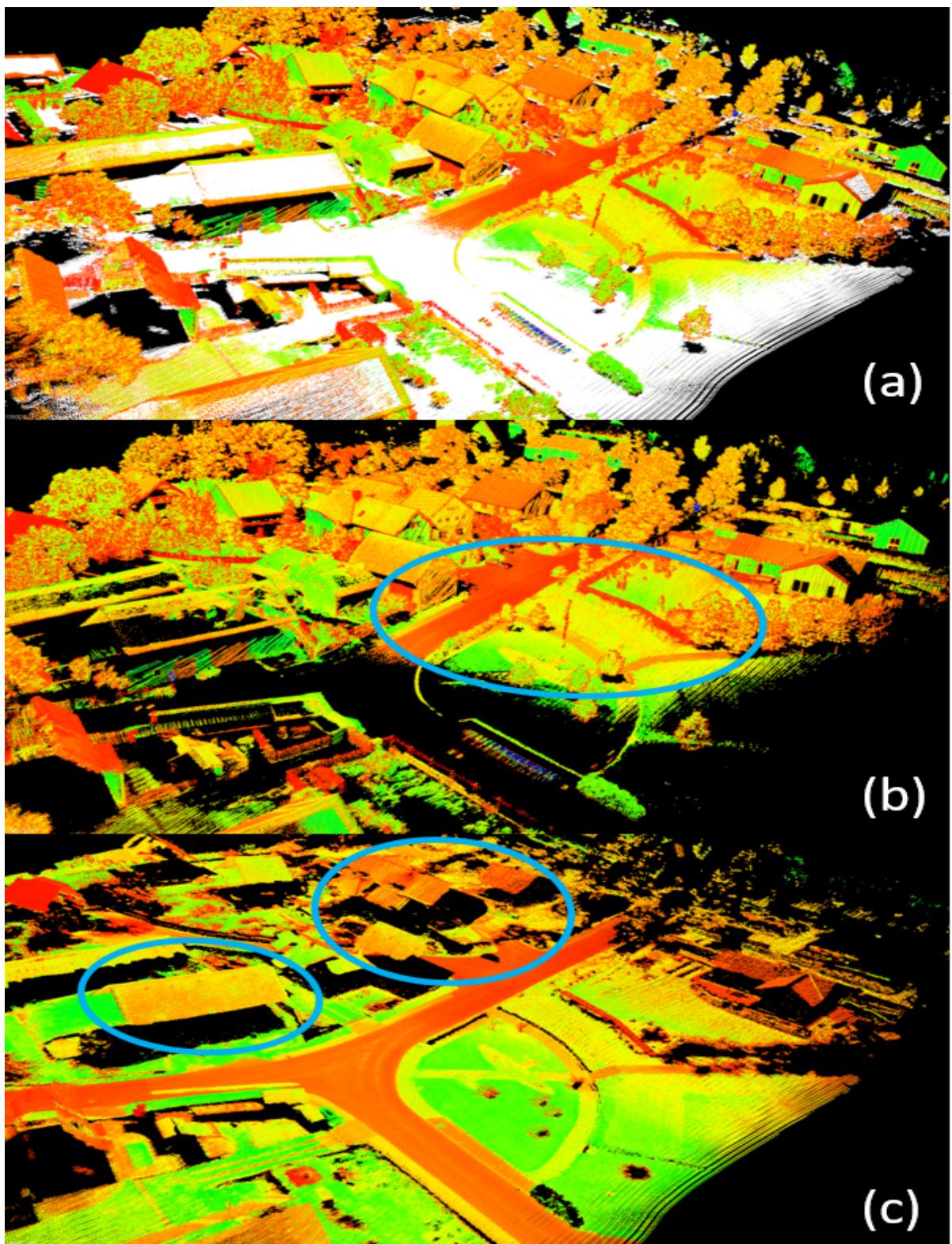


Obr. 6.1: Body cesty klasifikované ako body objektov (*SD3*)



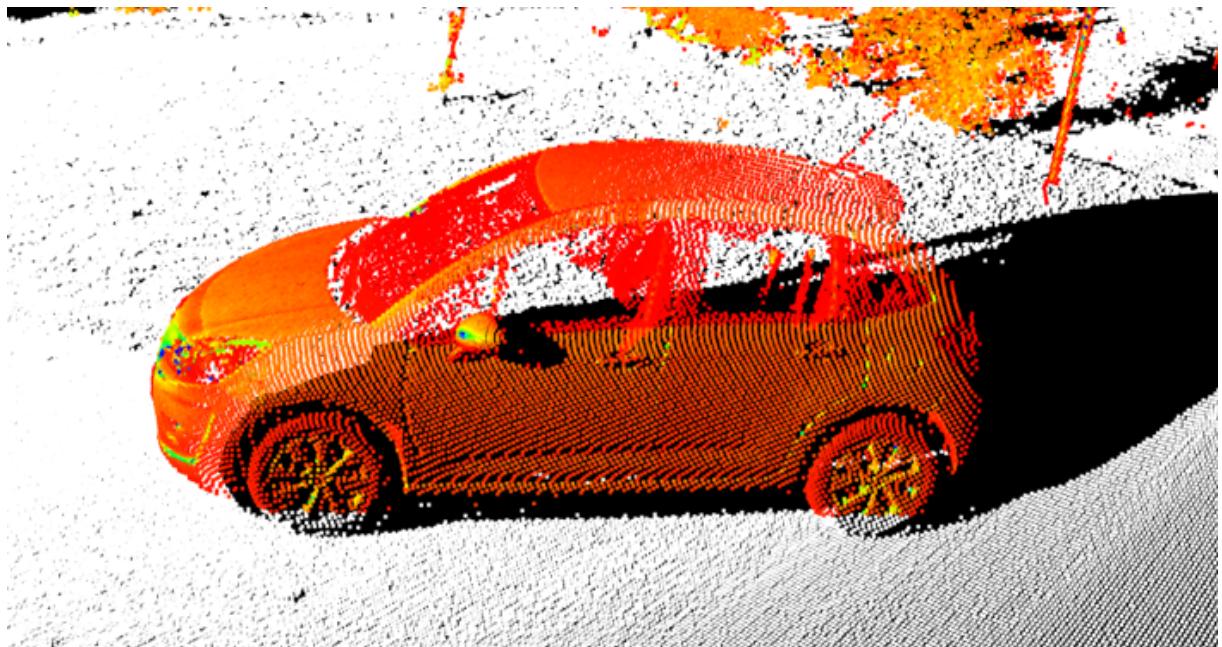
Obr. 6.2: Body cesty klasifikované ako body objektov (detail šumu) (*SD3*)

Následok šumu bolo ešte lepšie vidieť v prípade, kedy boli dáta získané z letiaceho dronu (Obr. 6.3), kde miera šumu bola omnoho väčšia a postihovala takmer celé mračno bodov. V tomto prípade dochádzalo ku nesprávnej filtrácii nie len pri bodoch zeme, ale navyše aj pri bodoch striech, ktoré boli nesprávne označené za body zeme. Tento výsledok bol zapríčinený šumom a taktiež nízkym nastavením maximálnej veľkosti “okna” pre tento súbor dát. Môžme teda vidieť, že v prípade vysokej úrovne šumu filter vo veľkej miere zlyháva, čo bude mať za následok nižšiu úroveň podvzorkovania.

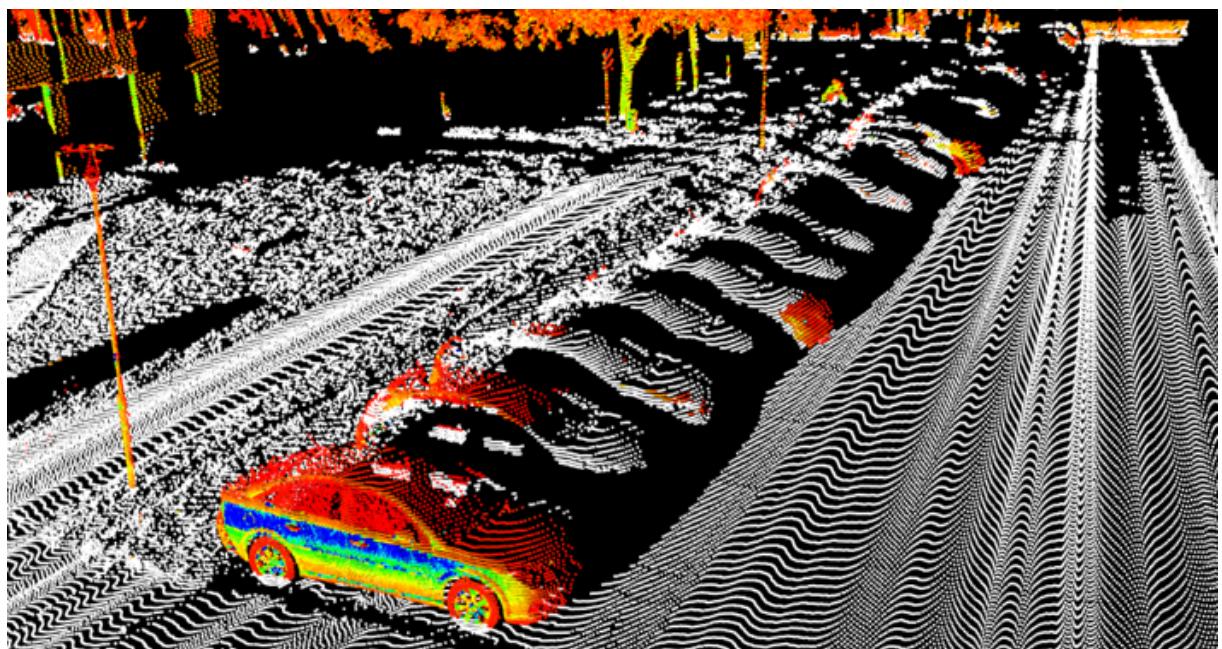


Obr. 6.3: Odolnosť morfologickej separácie bodov zeme na šum - (a) Mračno bodov objektov (pôvodná farba), body zeme (biela farba) (b) Body objektov obsahujú aj body zeme (c) Body zeme obsahujú strechy budov ( $SD5$ )

Medzi posledné nedostatky filtrovania zeme patrilo to, že progresívny morfologický filter berie za body zeme, aj body tesne nad jeho skutočným povrchom (Obr. 6.4) a taktiež má problém s málo hustými bodmi, ktoré sú v blízkej tesnosti zeme (Obr. 6.5).



Obr. 6.4: Body kolies auta vyhodnotená ako body zeme (*SD1*)

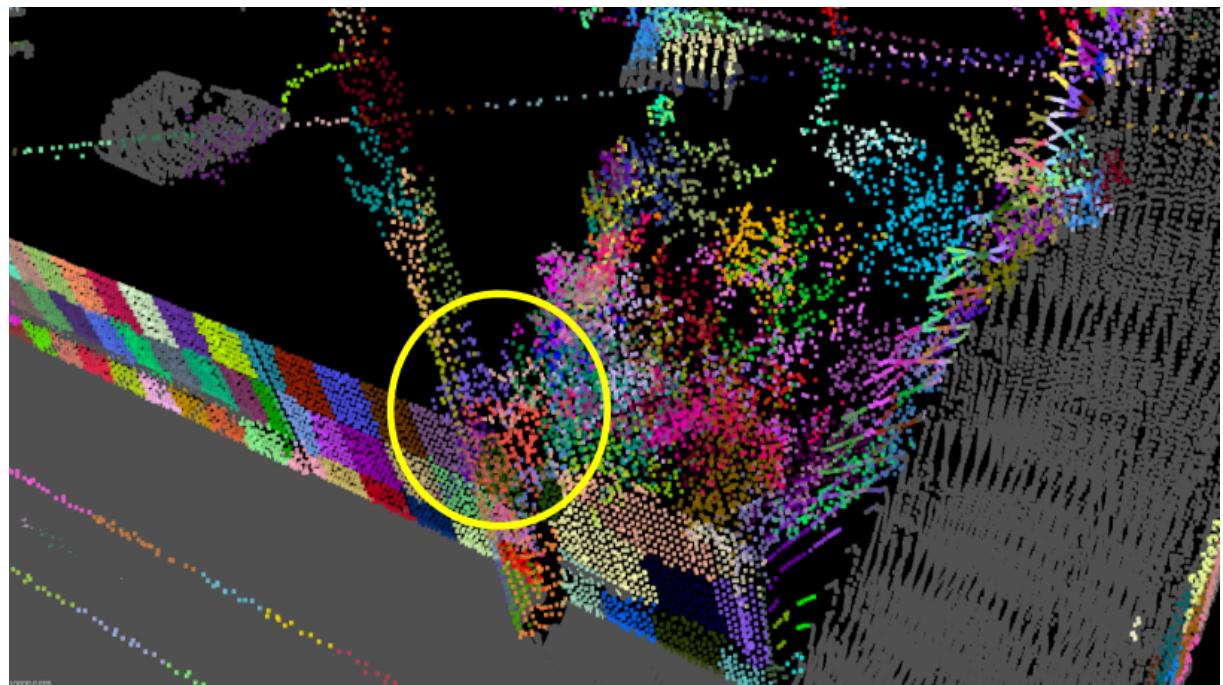


Obr. 6.5: Málo hustne body áut vyhodnotené ako body zeme (*SD1*)

## 6.2 Overenie generovania supervoxelov a segmentácie

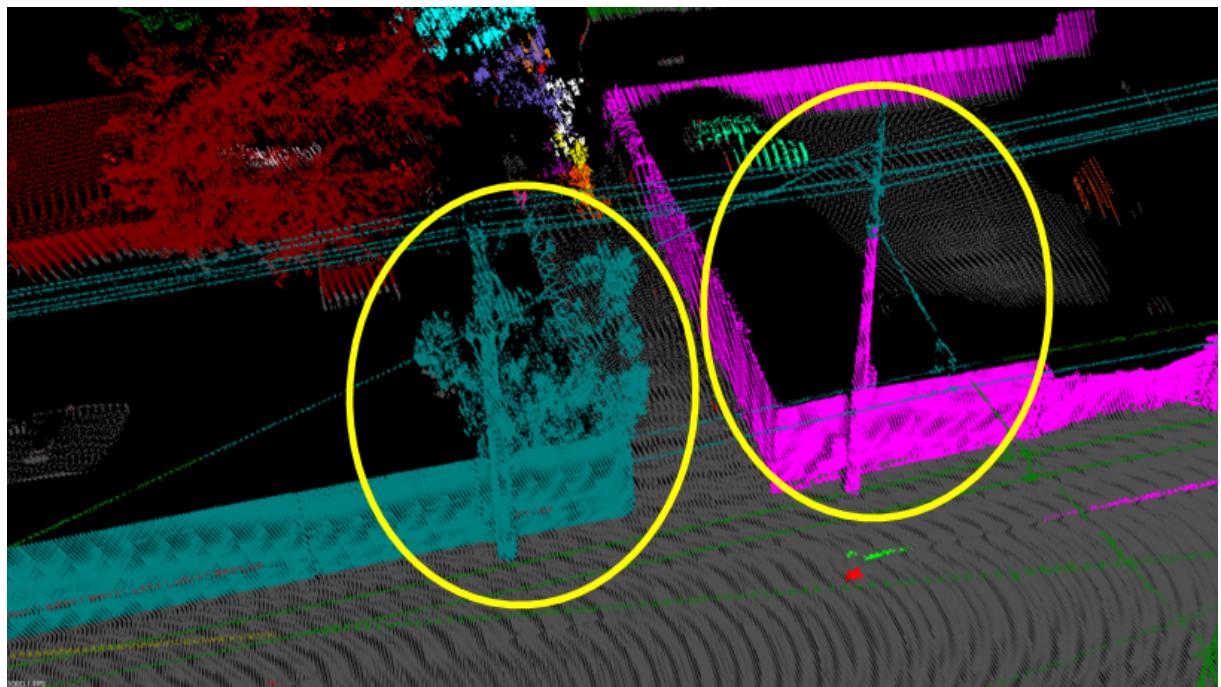
V tejto podkapitole sa zameriame na overenie zhľukovania bodov do supervoxelov a spojíme to so segmentáciou, nakoľko tieto kroky spolu blízko súvisia. Vizuálne overíme jednotlivé kroky, pričom si ukážeme ich nedostatky.

Zhlukovanie bodov do supervoxelov bolo z veľkej časti úspešné, ale vznikol problém v prípade, kedy sa pri sebe v malej vzdialosti nachádzalo viacero objektov. V takomto prípade niekedy supervoxely prechádzali cez hranice objektov, čo bolo obzvlášť pravdou v prípade ak sa jednalo o vegetáciu (Obr. 6.6).

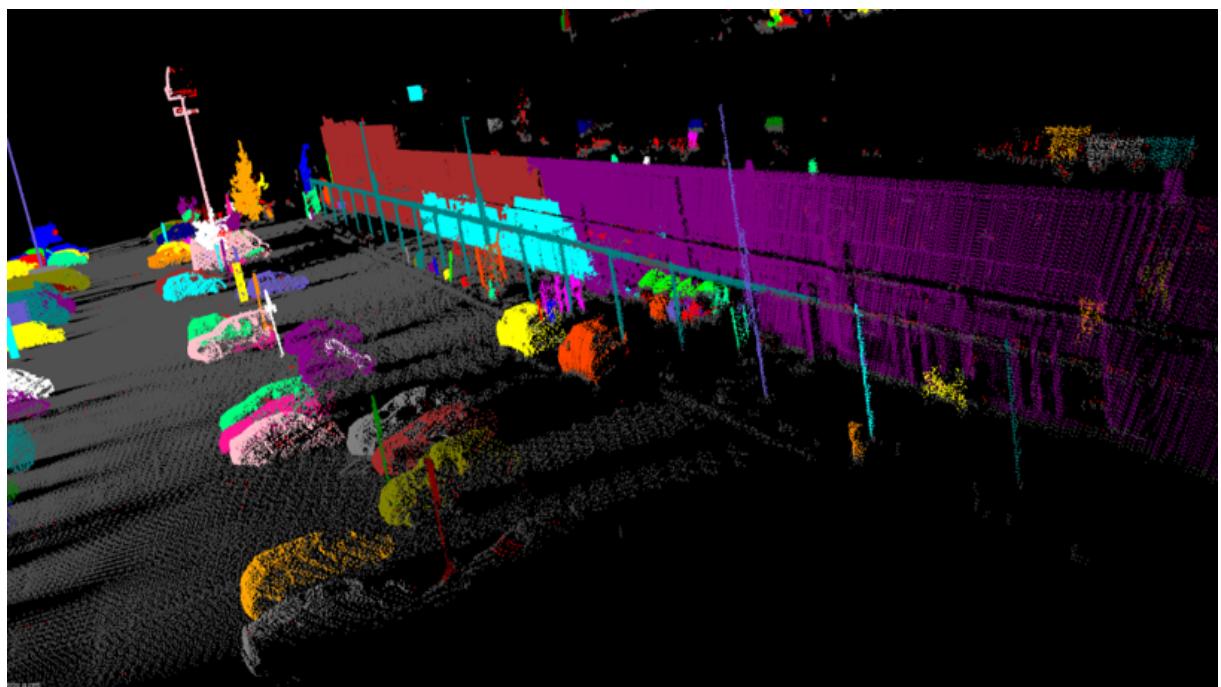


Obr. 6.6: Prechod supervoxelov medzi bodmi stĺpu a bodmi stromu (SD4)

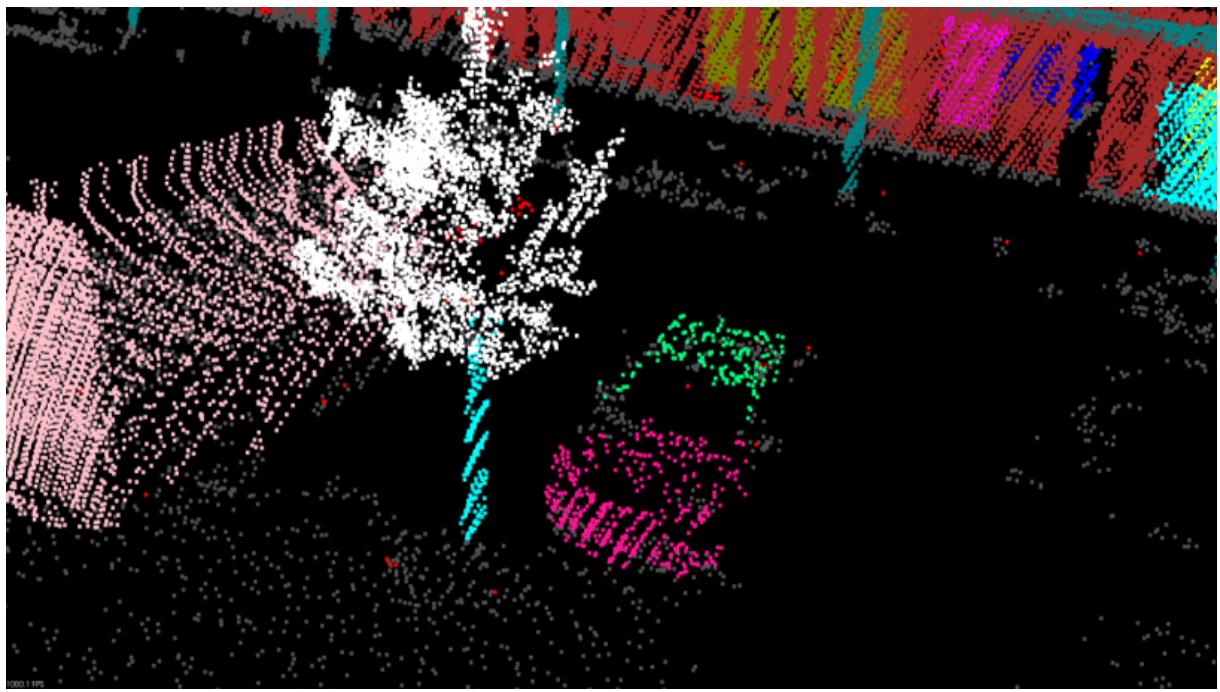
V prípade samotnej segmentácie sme sa stretli s podobným problémom, a teda ak bolo pri sebe viacero objektov v malej vzdialosti, tak v niektorých prípadoch nastalo ich spojenie (Obr. 6.7), čo čiastočne súviselo s predošlou chybou pri vytváraní supervoxelov. Navyše v prípade veľkých, ale aj menších horšie nasnímaných objektov dochádzalo ku rozdeleniu objektu na viacero menších objektov (Obr. 6.8 a 6.9), čo ale v niektorých prípadoch nemuselo byť nevyhnutnou chybou.



Obr. 6.7: Body viacerých objektov segmentované ako jeden objekt (SD4)



Obr. 6.8: Segmentovanie bodov veľkej budovy na viacero objektov (SD3)



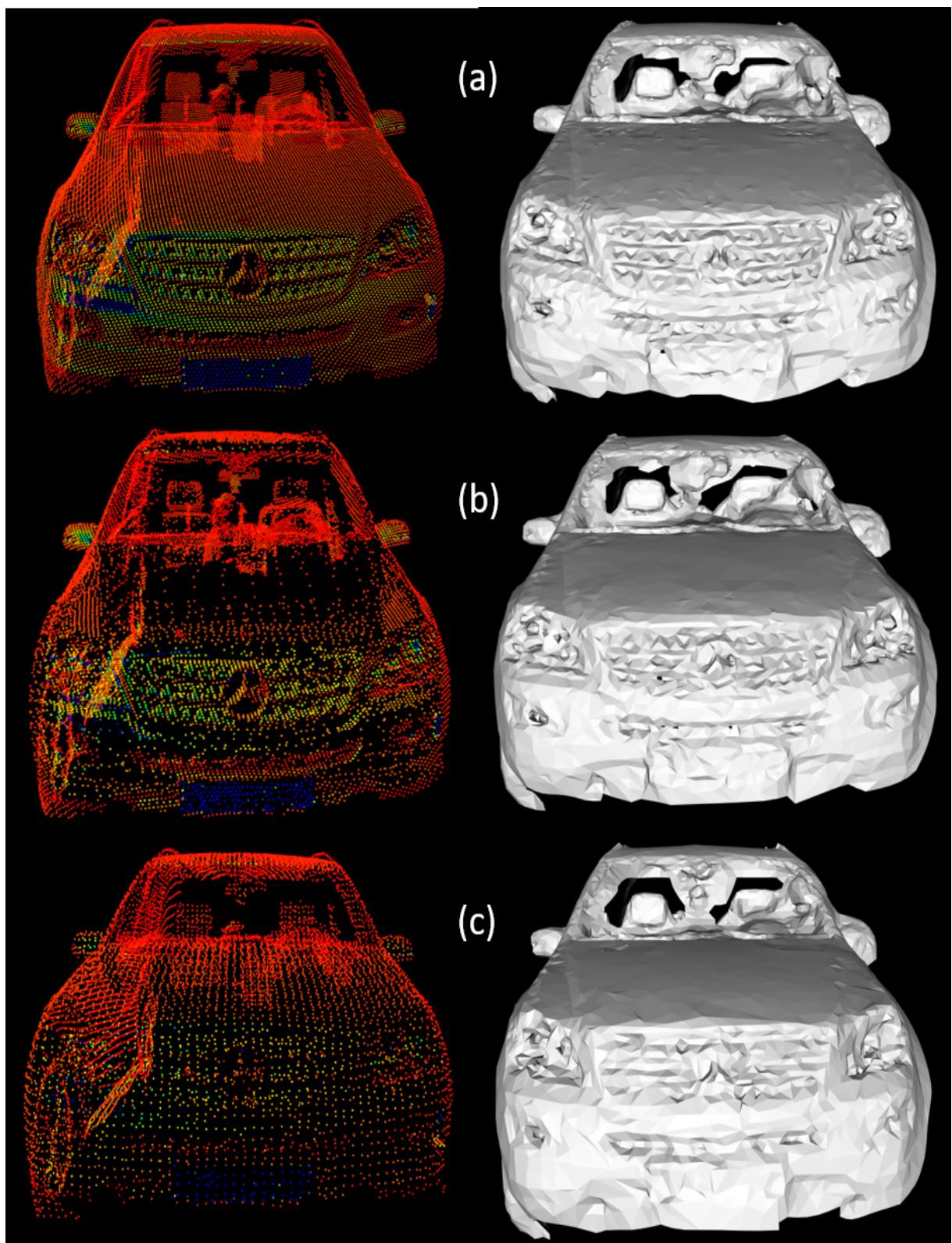
Obr. 6.9: Segmentovanie bodov stromu a auta na viacero objektov (SD3)

### 6.3 Overenie adaptívneho podvzorkovania

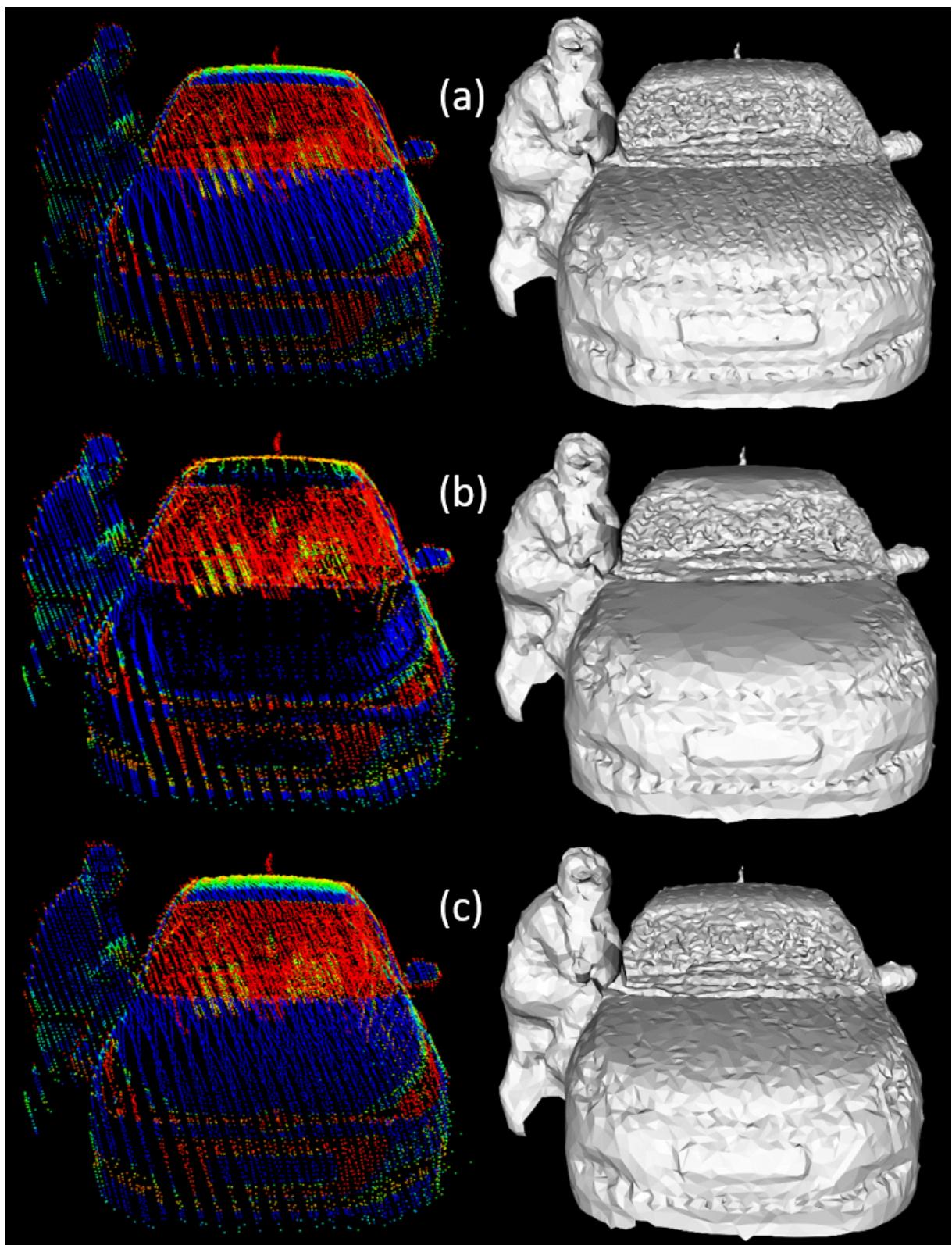
Pri adaptívnom podvzorkovaní sa zameriame hlavne na vizuálne porovnanie vytvorených povrchov pre nepodvzorkované, adaptívne podvzorkované a rovnomerne podvzorkované mračná bodov, keďže dosiahnuté úrovne podvzorkovania boli ukázané skôr (Tab. 5.3). Týmto porovnaním si ukážeme výhody adaptívneho prístupu oproti bežnému a zároveň si ukážeme niektoré nedostatky.

Dosiahnuté výsledky adaptívneho podvzorkovania môžeme dobre vidieť na obrázku 6.10, kde pre porovnanie bolo pridané rovnomerne podvzorkovanie pomocou voxelovej mriežky (neadaptívne), ktoré bolo nastavené, tak aby dosahovalo takmer rovnakú úroveň podvzorkovania (počtu bodov). Z vizuálneho hľadiska si môžeme všimnúť, že boli zachované ostrejšie detaily najmä v oblasti značky auta, pričom pri neadaptívnom prístupe, začali miznúť. Adaptívne podvzorkovanie znížilo výpočtovú náročnosť pri rekonštrukcií povrchu a zároveň pri porovnaní s nepodvzorkovaným mračnom bodov, zachovalo potrebné črty objektu.

Pre ďalšie porovnanie sme využili menej husté mračno bodov z rozdielneho súboru dát (Obr. 6.11). V tomto prípade z vizuálneho hľadiska nie je vidieť výrazný rozdiel, čo bolo najmä zapríčinené menšou hustotou bodov (nižšou prítomnosťou detailov). Ďalej si môžme všimnúť, že v tomto prípade pomáha adaptívne podvzorkovanie vyhľadeniu rovnejších povrchov (kapota auta), ale na druhú stranu zlyháva pri efektívnom znížení počtu bodov na zašumenom povrchu (čelné sklo).



Obr. 6.10: Mračná bodov a ich príslušný povrch - (a) Nepodvzorkované mračno bodov (b) Adaptívne podvzrokovane mračno bodov (c) Rovnomerne podvzorkované mračno bodov pomocou voxelovej mriežky ( $SD1$ )

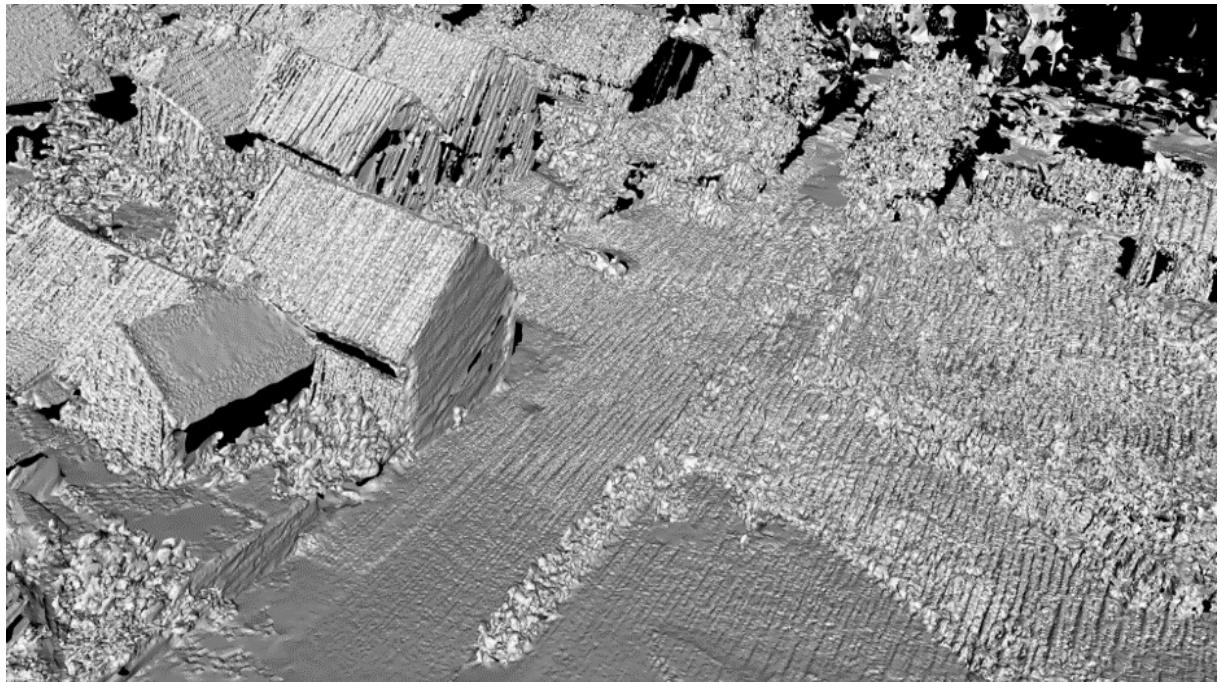


Obr. 6.11: Mračná bodov a ich príslušný povrch - (a) Nepodvzorkované mračno bodov (b) Adaptívne podvzorkované mračno bodov (c) Rovnomerne podvzorkované mračno bodov pomocou voxelovej mriežky ( $SD3$ )

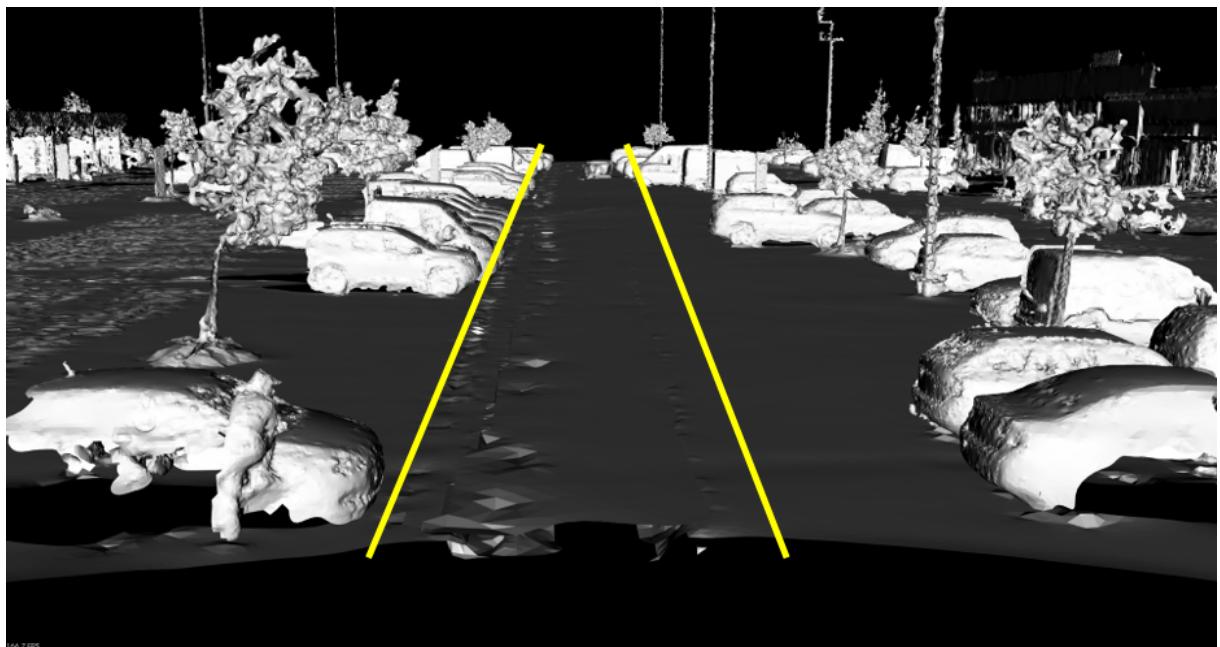
## 6.4 Overenie rekonštrukcie povrchu

Pri rekonštrukcií povrchu sme dosiahli pomerne vysoké zníženie pamäťovej náročnosti oproti pôvodnému mračnu bodov (vid'. Tab. 5.4). V tejto podkapitole sa preto budeme zaoberať vizuálnej kontrolou dosiahnutých výsledkov, pričom sa zameriame aj na nedostatky spôsobené predošlými krokmi.

Ako prvé sme overili rekonštrukciu povrchu v prítomnosti falošných bodov v podobe šumu (Obr. 6.12 a 6.13). V prvom prípade bola úroveň šumu príliš vysoká, na základe čoho zlyhala predošlá separácia bodov, adaptívne podvzorkovanie vyhodnotilo šum za dôležité body a viedlo to ku vzniku drsného povrchu, ktorý nie je prirodzený. V ďalšom prípade taktiež čiastočne zlyhala separácia bodov zeme, ale vďaka nižšej úrovni šumu, bol tento nedostatok zakrytý zemou (v podobe objektu), ktorá sa pomerne plynulo spojila so skutočnou zemou. Nevýhodou zlyhania predošlých krokov je nižšia úroveň podvzorkovania a vytvorenie väčšieho počtu polygónov, ktoré navyše opisujú falošné dátá.

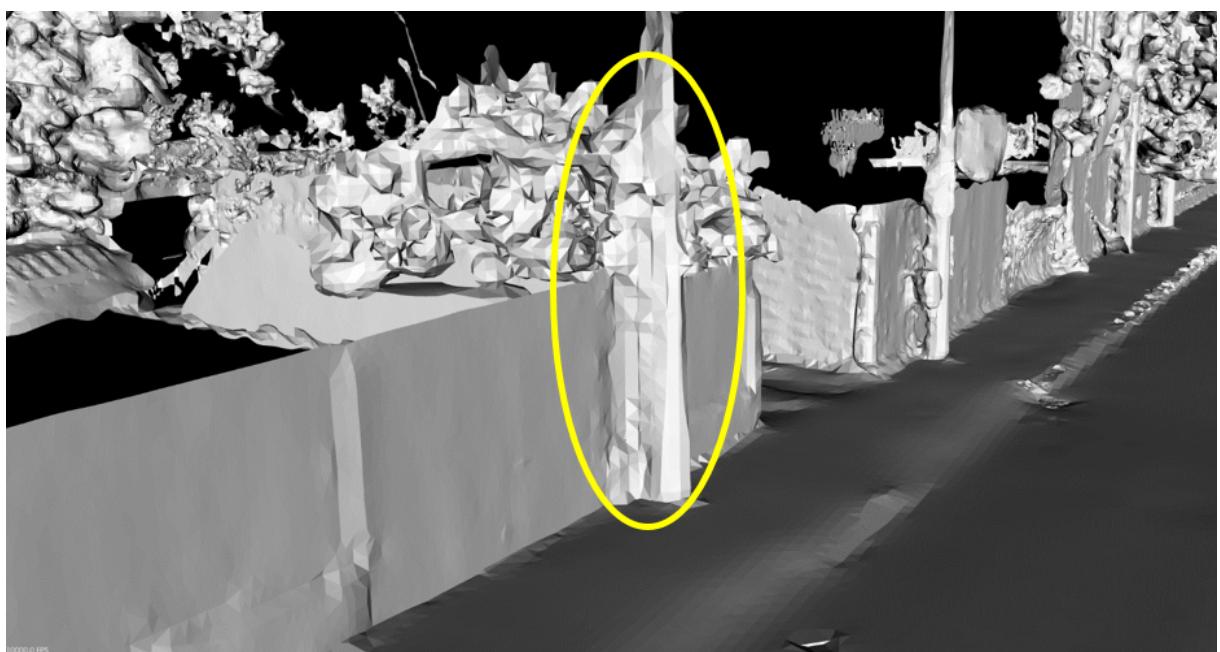


Obr. 6.12: Vznik drsného povrchu pri nesprávnej separácii bodov zeme (Obr. 6.3) a rekonštrukcií zo zašumených dát (*SD5*)

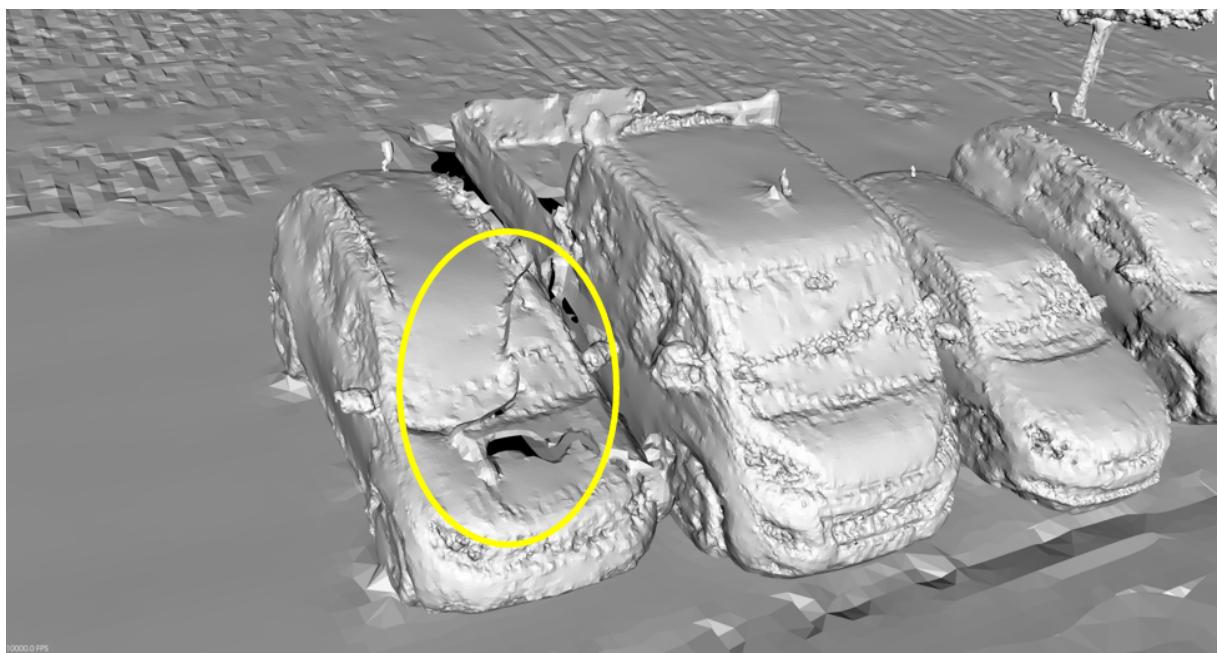


Obr. 6.13: Hladký prechod medzi viacerými povrchmi zeme, aj napriek nesprávnej separácii zeme (Obr. 6.1) (SD3)

Ďalším nedostatkom pri rekonštrukcií boli prípady, kedy zlyhala segmentácia, čo zapríčinilo spojenie povrchu viacerých objektov (Obr. 6.14) a odstránenie skutočnej medzery medzi nimi. Taktiež sa našli prípady, kedy aj napriek úpravám zlyhal odhad normál, čo zapríčinilo vytvorenie trhlín v povrchu (Obr. 6.15).

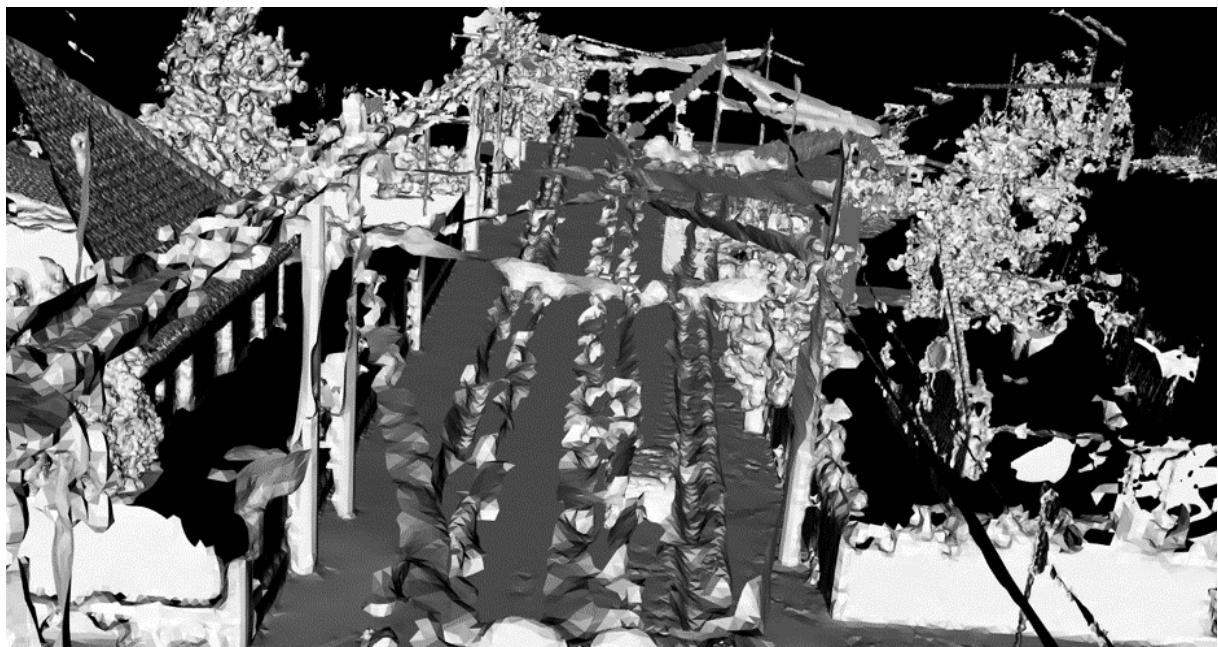


Obr. 6.14: Prepojenie viacerých objektov spolu, na základe zlej segmentácie (Obr. 6.7) (SD4)

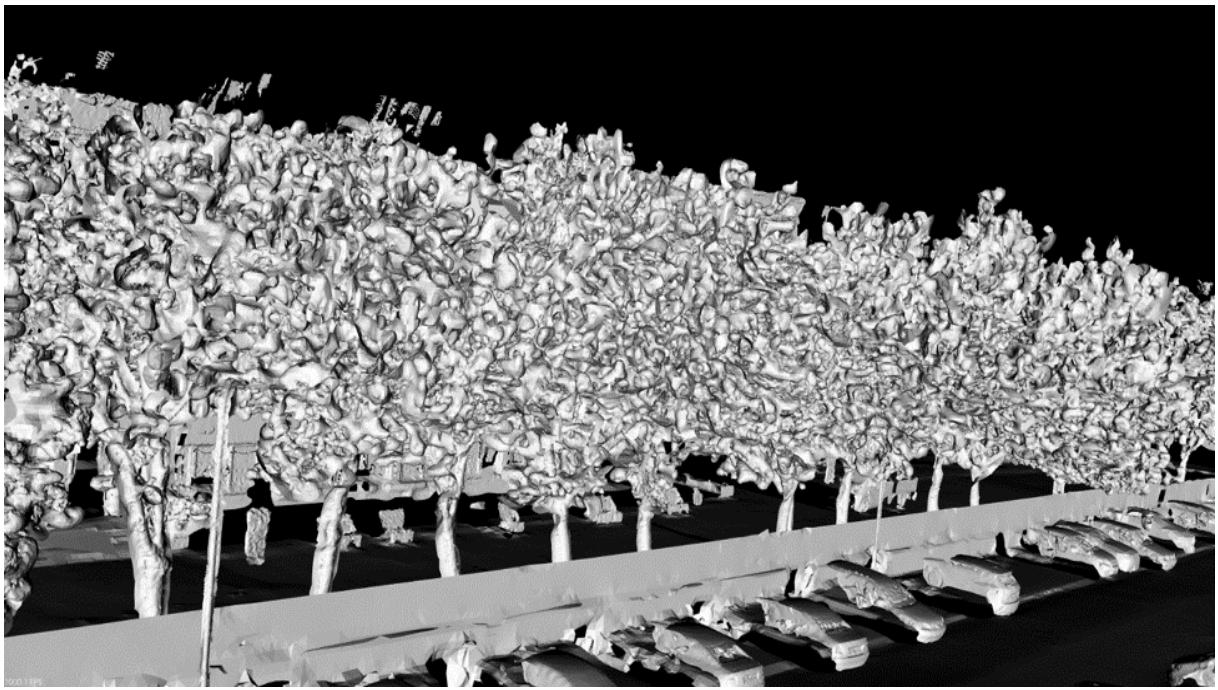


Obr. 6.15: Výskyt trhlín na základe zlého odhadu normál (*SD3*)

V poslednom rade sme sa zamerali na dôveryhodnosť povrchu, kde sme si všimli, že rekonštrukcia má problém s úzkymi objektami a vegetáciou (Obr. 6.16 a 6.17), pričom tento efekt bol posilnený neúplným odstránením nadbytočných polygónov na hranách objektu.



Obr. 6.16: Problém Poissonovej rekonštrukčnej metódy s rekonštrukciou úzkych objektov (trojbusové vedenie) (*SD4*)



Obr. 6.17: Problém Poissonovej rekonštrukčnej metódy s dôveryhodnou rekonštrukciou stromov (*SD1*)

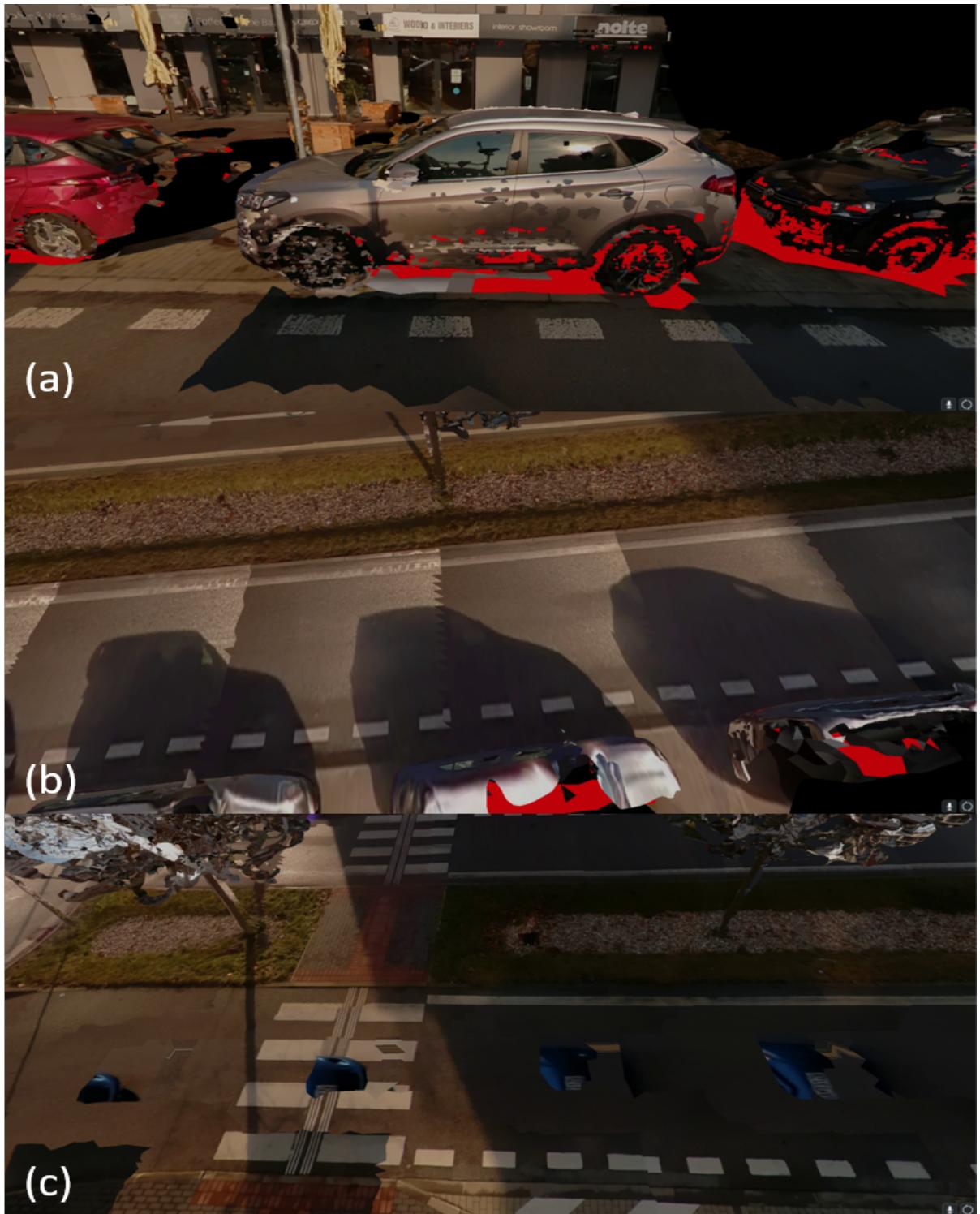
## 6.5 Overenie textúrovania povrchu

Textúrovanie povrchu dosiahlo pomerne dobré výsledky a obohatilo to povrch o dávku realizmu, avšak nachádzali sa tam nedostatky, ku ktorým sa teraz vyjadríme.

Ako prvý problém sa prejavilo využitie sledovania lúčov, ktoré bolo založené na využití oktálового stromu. Vo veľkej miere to síce odstránilo falošné textúrovanie, ale kvôli neúplnému zachyteniu geometrie polygónov pomocou voxelov, dochádzalo ku mozaikovému textúrovaniu a taktiež niektoré polygóny neboli otextúrované vôbec, aj napriek tomu, že boli v skutočnosti videné (Obr. 6.18a).

Ďalší nedostatok vznikol pri textúrovaní s viacerých fotiek. Na hranách medzi jednotlivými snímkami dochádzalo ku ostrému prechodu farby (Obr. 6.18b), čo vytváralo neprirozený vzhľad. Tento problém by sa dal adresovať, pomocou využitia Gaussového filtra na jednotlivých hranách, pre zjemnenie ich prechodu.

Posledný nedostatok sa týkal dynamických objektov, ktoré boli pri získavaní mračna bodov odstránené, ale ostali na fotkách z kamery. Tieto objekty spôsobujú falošné textúrovanie iných objektov (Obr. 6.18c) a pre ich odstránenie by bolo potrebné implementovať detekciu dynamických objektov na fotkách.



Obr. 6.18: (a) Mozaikové textúrovanie kvôli zlej reprezentácií polygónov pomocou oktálového stromu (sledovanie lúčov) (b) Ostrý prechod farby medzi viacerými snímkami (c) Neodstránenie dynamických objektov (auto na ceste) (SD2)

# Záver

V tejto diplomovej práci sme sa venovali návrhu riešenia na predspracovanie mračien bodov vysokej hustoty, pričom sme sa snažili znížiť ich pamäťovú náročnosť a zároveň zachovať dôležité body, ktoré opisovali detaľy. Zo spracovaného mračna bodov sme následne vytvorili tzv. mesh, ktorá opisovala body spojitým povrchom, ktorý bol na záver otextúrovaný.

V prvej časti práce sme sa venovali zhnutiu existujúcich metód pre spracovanie mračien bodov, ako aj pre rekonštrukciu povrchu. Venovali sme všeobecným prístupom, ale aj fungovaniu konkrétnych algoritmov, čo nám umožnilo hlbší náhľad pri ich využití.

V ďalšej časti práce sme sa zamerali na samotný návrh riešenia, pričom pridávanie jednotlivých krokov, bolo usmernené priebežnými výsledkami. Začali sme filtráciou mračna bodov od odľahlých bodov využitím SOR metódy, čo do istej úrovne zabezpečilo odstránenie falošných bodov, ako aj menej hustých oblastí, ktoré neboli podstatné pre konečný výsledok. Pokračovali sme využitím progresívneho morfologického filtra, ktorý nám oddelil body zeme a objektov. Z oddelených bodov objektov sme následne vytvorili tzv. supervoxely, ktoré spájali podobné body do väčších celkov a boli predpokladom pre využitie CPC segmentačnej metódy. Táto metóda nám umožnila prístup ku jednotlivým objektom, čo sme využili pri ďalších krokoch.

Dôležitým krokom bolo podvzorkovanie, ktoré výrazne znížilo celkový počet bodov. Pre body objektov sme využili adaptívne podvzorkovanie založené na metóde diferencií normál, ktorou sme klasifikovali dôležitosť jednotlivých bodov a následne sme využili príslušnú úroveň podvzorkovania. Využitím Poissonovej rekonštrukčnej metódy, sme vytvorili povrch, ktorý pri menšej pamäťovej náročnosti opisoval pôvodné mračno bodov. Na záver sme vytvorený povrch otextúrovali pomocou fotiek z kamery, čo obohatilo výsledok o realizmus.

Celé riešenie sme na overili pomocou vizuálnej kontroly a porovnaní, pričom sme odhalili niekoľko nedostatkov, ktoré ale vo väčšine prípadov nemali závažný dopad na výsledok. Z dosiahnutých výsledkov môžeme povedať, že uvedený postup funguje najlepšie v prípade štrukturovaných prostredí z nízkou úrovňou šumu, zatiaľ čo v opačnom prípade sa znižuje efektivita podvzorkovania a dôveryhodnosť rekonštruovaného povrchu.

Napriek dosiahnutým výsledkom má táto práca stále mnoho miesta pre vylepšenie. Mnoho využitých metód v tejto práci je založených na nastavení parametrov, ktoré môžu fungovať optimálne pre jeden súbor dát, ale nemusia pre druhý. Bolo by preto vhodné doplniť dynamické nastavovanie niektorých parametrov na základe vlastností konkrétneho súboru dát, a taktiež by bolo možné doplniť medzikrok, ktorý by zvýšil úroveň odolnosti na šum.

# Zoznam použitej literatúry

1. XU, Yusheng, TONG, Xiaohua and STILLA, Uwe. Voxel-based representation of 3D point clouds: Methods, applications, and its potential use in the construction industry. *Science Direct* [online]. 2021 [cit. 2023-12-24]. Dostupné z : <https://www.sciencedirect.com/science/article/abs/pii/S0926580521001266>.
2. NIKOLOV, Ivan. Neighborhood Analysis, KD-Trees, and Octrees for Meshes and Point Clouds in Python. *Towards Data Science* [online]. 2022 [cit. 2023-12-24]. Dostupné z : <https://towardsdatascience.com/neighborhood-analysis-kd-trees-and-octrees-for-meshes-and-point-clouds-in-python-19fa96527b77>.
3. CAI, Yixi, XU, Wei and ZHANG, Fu. ikd-Tree: An Incremental K-D Tree for Robotic Applications [online]. 2021 [cit. 2023-12-25]. Dostupné z : <https://arxiv.org/pdf/2102.10808.pdf>.
4. BENTLEY, Jon Louis. Multidimensional Binary Search Trees Used for Associative Searching [online]. 1975 [cit. 2023-12-25]. Dostupné z : <https://dl.acm.org/doi/pdf/10.1145/361002.361007>.
5. ANZOLA, John, TARAZONA, Giovanny, ESPADA, Jordán a GONZALEZ CRESPO, Ruben. A Clustering WSN Routing Protocol Based on k-d Tree Algorithm. *Sensors*. 2018, roč. 18, s. 1–25. Dostupné z DOI: 10.3390/s18092899.
6. ELSEBERG, Jan, BORRMANN, Dorit a NÜCHTER, Andreas. One billion points in the cloud – an octree for efficient processing of 3D laser scans. *ISPRS Journal of Photogrammetry and Remote Sensing*. 2013, roč. 76, s. 76–88. ISSN 0924-2716. Dostupné z DOI: <https://doi.org/10.1016/j.isprsjprs.2012.10.004>. Terrestrial 3D modelling.
7. *PCL/OpenNI tutorial 2: Cloud processing (basic)* [online]. [cit. 2023-12-26]. Dostupné z : <https://robotica.unileon.es/index.php?title=File:Octree.png>.
8. HAN, Xian-Feng, JIN, Jesse S., WANG, Ming-Jie, JIANG, Wei, GAO, Lei a XIAO, Li-ping. A review of algorithms for filtering the 3D point cloud. *Signal Processing: Image Communication*. 2017, roč. 57, s. 103–112. ISSN 0923-5965. Dostupné z DOI: <https://doi.org/10.1016/j.image.2017.05.009>.
9. CARRILHO, A. C., GALO, M. a SANTOS, R. C. dos. STATISTICAL OUTLIER DETECTION METHOD FOR AIRBORNE LIDAR DATA. 2018. Dostupné tiež z: <https://d-nb.info/1167928644/34>.

10. *Removing outliers using a StatisticalOutlierRemoval filter* [online]. [cit. 2023-12-27]. Dostupné z : [https://pcl.readthedocs.io/projects/tutorials/en/latest/statistical\\_outlier.html](https://pcl.readthedocs.io/projects/tutorials/en/latest/statistical_outlier.html).
11. POLŤÁK, Alexander. Detekcia známych objektov v mračne bodov a ich filtrácia : diplomová práca. Bratislava : STU. 2023.
12. LIN, Yun-Jou, BENZIGER, Ronald R a HABIB, Ayman. Planar-Based Adaptive Down-Sampling of Point Clouds. In: 2016. Dostupné z DOI: <https://doi.org/10.14358/PERS.82.12.955>.
13. NGUYEN, Anh a LE, Bac. 3D point cloud segmentation: A survey. In: *2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM)*. 2013, s. 225–230. Dostupné z DOI: [10.1109/RAM.2013.6758588](https://doi.org/10.1109/RAM.2013.6758588).
14. SCHNABEL, R., WAHL, R. a KLEIN, R. Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum*. 2007, roč. 26, č. 2, s. 214–226. Dostupné z DOI: <https://doi.org/10.1111/j.1467-8659.2007.01016.x>.
15. STROM, Johannes, RICHARDSON, Andrew a OLSON, Edwin. Graph-based segmentation for colored 3D laser point clouds. In: *2010 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2010, s. 2131–2136.
16. ESTER, Martin, KRIEGEL, Hans-Peter, SANDER, Jörg, XU, Xiaowei et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *kdd*. 1996, zv. 96, s. 226–231. Č. 34.
17. IKOTUN, Abiodun M., EZUGWU, Absalom E., ABUALIGAH, Laith, ABUHAIJA, Belal a HEMING, Jia. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*. 2023, roč. 622, s. 178–210. ISSN 0020-0255. Dostupné z DOI: <https://doi.org/10.1016/j.ins.2022.11.139>.
18. *Understanding DBSCAN Algorithm and Implementation from Scratch* [online]. [cit. 2023-12-31]. Dostupné z : <https://towardsdatascience.com/understanding-dbscan-algorithm-and-implementation-from-scratch-c256289479c5>.
19. FISCHLER, Martin A a BOLLES, Robert C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*. 1981, roč. 24, č. 6, s. 381–395.
20. KHATAMIAN, Alireza a ARABNIA, Hamid R. Survey on 3D surface reconstruction. *Journal of Information Processing Systems*. 2016, roč. 12, č. 3, s. 338–357.

21. LORENSEN, William E. a CLINE, Harvey E. Marching cubes: A high resolution 3D surface construction algorithm. *SIGGRAPH Comput. Graph.* 1987, roč. 21, č. 4, s. 163–169. ISSN 0097-8930. Dostupné z DOI: 10.1145/37402.37422.
22. *Marching Cubes, Explained Using Marching Squares*. Dostupné tiež z: <https://blog.42yeah.is/algoritm/2023/07/15/marching-cubes.html>.
23. KAZHDAN, Michael, BOLITHO, Matthew a HOPPE, Hugues. Poisson surface reconstruction. In: *Proceedings of the fourth Eurographics symposium on Geometry processing*. 2006, zv. 7. Č. 4.
24. BERNARDINI, F., MITTELMAN, J., RUSHMEIER, H., SILVA, C. a TAUBIN, G. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*. 1999, roč. 5, č. 4, s. 349–359. Dostupné z DOI: 10.1109/2945.817351.
25. BERGER, Matthew, TAGLIASACCHI, Andrea, SEVERSKY, Lee M., ALLIEZ, Pierre, GUENNEBAUD, Gaël, LEVINE, Joshua A., SHARF, Andrei a SILVA, Claudio T. A Survey of Surface Reconstruction from Point Clouds. *Computer Graphics Forum*. 2017, roč. 36, č. 1, s. 301–329. Dostupné z DOI: <https://doi.org/10.1111/cgf.12802>.
26. OHTAKE, Yutaka, BELYAEV, Alexander, ALEXA, Marc, TURK, Greg a SEIDEL, Hans-Peter. Multi-level partition of unity implicits. In: *ACM SIGGRAPH 2005 Courses*. Los Angeles, California: Association for Computing Machinery, 2005, 173–es. SIGGRAPH '05. ISBN 9781450378338. Dostupné z DOI: 10.1145/1198555.1198649.
27. TALTON, Jerry. A short survey of mesh simplification algorithms. 2004.
28. ZHANG, Keqi, CHEN, Shu-Ching, WHITMAN, D., SHYU, Mei-Ling, YAN, Jianhua a ZHANG, Chengcui. A progressive morphological filter for removing nonground measurements from airborne LIDAR data. *IEEE Transactions on Geoscience and Remote Sensing*. 2003, roč. 41, č. 4, s. 872–882. Dostupné z DOI: 10.1109/TGRS.2003.810682.
29. *Clustering of Pointclouds into Supervoxels*. Dostupné tiež z: [https://pcl.readthedocs.io/projects/tutorials/en/latest/supervoxel\\_clustering.html](https://pcl.readthedocs.io/projects/tutorials/en/latest/supervoxel_clustering.html).
30. SCHOELER, Markus, PAPON, Jeremie a WORGOTTER, Florentin. Constrained planar cuts-object partitioning for point clouds. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, s. 5207–5215.
31. *Difference of Normals Based Segmentation*. Dostupné tiež z: [https://pcl.readthedocs.io/projects/tutorials/en/latest/don\\_segmentation.html](https://pcl.readthedocs.io/projects/tutorials/en/latest/don_segmentation.html).

32. GUO, Ning, ZHANG, Baohua, ZHOU, Jun, ZHAN, Ketian a LAI, Shuang. Pose estimation and adaptable grasp configuration with point cloud registration and geometry understanding for fruit grasp planning. *Computers and Electronics in Agriculture*. 2020, roč. 179, s. 105818. Dostupné z DOI: 10.1016/j.compag.2020.105818.

# **Prílohy**

A	Technická dokumentácia . . . . .	69
B	Odkaz na zdrojový kód . . . . .	70
C	Ukážka rekonštruovaného povrchu . . . . .	71

# A Technická dokumentácia

Projekt použitý pre vytvorenie tejto práce (odkaz v prílohe B), bol vypracovaný v C++ jazyku v prostredí Visual Studio 2022. Pre úspešné spojazdnenie a ovládanie projektu, je potrebné postupovať podľa následujúcich krokov:

1. **Použité knižnice** - Ako prvým krokom je spojazdnenie knižníc na používanom zariadení.  
V tejto práci sme použili knižnicu *PCL* – 1.13.1, ktorá je jadrom celého projektu a je ju možné nainštalovať pomocou “All in one installer-u” dostupnom na internete. Následne je voliteľne nainštalovať *OpenCv* knižnicu, ktorá bola použitá pre zobrazenie fotiek v kapitole o textúrovaní povrchu a je taktiež dostupná na internete.
2. **Rozloženie projektu** - Projekt sa skladá z dvoch hlavných súborov:  
*cloud\_handler.cpp* - V tomto súbore sa nachádzajú konkrétnie implementácie jednotlivých algoritmov, ktoré boli použité v práci. Algoritmy patria hlavnej triede *CloudHandler*, ktorá slúži ako rozhranie a ponúka jednoduchú prácu s mračnami bodov, ako aj rekonštrukciou povrchu.  
*main.cpp* - V tomto súbore sa nachádzajú pomocné funkcie, ktoré slúžili na testovanie jednotlivých algoritmov, ako aj na vykreslovanie potrebných prípadov. Nachádza sa tu aj postupné volanie konkrétnych algoritmov, ktoré zabezpečí vytvorenie rekonštruovaného povrchu.
3. **Spustenie projektu** - Pre spustenie postupu uvedenom v práci, je v hlavnom adresári projektu potrebné vytvoriť adresár (napr. zdroje). Cestu ku tomuto adresáru je potrebné vložiť do premennej *resource\_path\_* a následne sa do neho môže vytvoriť ďalší podadresár, ktorý bude obsahovať všetky medzikroky postupu. Do podadresáru je potrebné vložiť mračno bodov vo formáte .ply a následne zavolať funkciu *calculate\_all*, v ktorej treba nastaviť premennú *base\_path* s príslušným menom podadresáru. Po vykonaní predošlých krokov, by mal zbehnúť celý postup a vygenerovať výsledný povrch.

## B Odkaz na zdrojový kód

V prílohe sa nachádza textový súbor s odkazom na *GitHub* repozitár.

## C Ukážka rekonštruovaného povrchu

V prílohe sa nachádza model zrekonštruovaného povrchu (SD1).