



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

DETEKCE CESTY PRO AUTONOMNÍ VOZIDLO

ROAD DETECTION FOR AUTONOMOUS CAR

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MATÚŠ KOMORA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MICHAL ŠPANĚL, Ph.D.

BRNO 2016

Zadání diplomové práce

Řešitel: **Komora Matúš, Bc.**

Obor: Počítačová grafika a multimédia

Téma: **Detekce cesty pro autonomní vozidlo**
Road Detection for Autonomous Car

Kategorie: Zpracování obrazu

Pokyny:

1. Seznamte se s problematikou senzorů používaných v autonomních vozidlech (kamera, Velodyne LiDAR) a základními principy zpracování získaných dat. Zaměřte se zejména na zpracování tzv. mračen bodů ze senzoru Velodyne LiDAR.
2. Zorientujte se v současných metodách detekce cesty pro autonomní vozidla s využitím senzoru Velodyne LiDAR a kamer.
3. Vyberte vhodné metody a senzory a navrhnete rychlý detektor cesty v okolí vozidla.
4. Experimentujte s vaší implementací a případně navrhnete vlastní modifikace metod.
5. Porovnejte a diskutujte dosažené výsledky a zvažte možnosti budoucího vývoje.
6. Vytvořte stručný plakát nebo video prezentující vaši práci, její cíle a výsledky.

Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Bez požadavků.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Španěl Michal, Ing., Ph.D., UPGM FIT VUT**

Datum zadání: 1. listopadu 2015

Datum odevzdání: 29. července 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Táto diplomová práca sa venuje problematike detekcie cesty v okolí autonómneho vozidla. Cesta sa vyhodnocuje na základe dát z laserového radaru Velodyne LiDAR. V práci je použité už existujúce riešenie, ktoré je rozšírené o strojové učenie SVM s postupným učeníím. Práca porovnáva staré a nové riešenie na datase KITTI. Úspešnosť odhadu cesty je vypočítaná podľa ukazateľa F-measure.

Abstract

This thesis deals with detection of the road adjacent to an autonomous vehicle. The road is recognition is based on the Velodyne LiDAR laser radar data. An existing solution is used and extended by machine learning - a Support Vector Machine with online learning. The thesis evaluates the existing solution and the new one using a KITTI dataset. The reliability of the road recognition is then computed using F-measure.

Kľúčové slová

detekcia cesty, SVM, Lidar, strojové učenie, ROS, KITTI

Keywords

road detection, SVM, Lidar, machine learning, ROS, KITTI

Citácia

KOMORA, Matúš. *Detekce cesty pro autonomní vozidlo*. Brno, 2016. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Španěl Michal.

Detekce cesty pro autonomní vozidlo

Prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením pána Ing. Michala Španěla Ph.D.. Uviedol som všetky literárne pramene a publikácie s ktorých som čerpal.

.....
Matúš Komora
29. júla 2016

Podakovanie

Chcel by som poďakovať vedúcemu práce za podporu a tolerantnosť počas celej spolupráce. V neposlednej rade by som chcel poďakovať aj najbližšiemu okoliu, ktoré ma po celý čas podporovalo, či už vedomostne, alebo psychicky.

© Matúš Komora, 2016.

Táto práca vznikla ako školské dielo na FIT VUT v Brně. Práca je chránená autorským zákonom a jej využitie bez poskytnutia oprávnenia autorom je nezákonné, s výnimkou zákonne definovaných prípadov.

Obsah

1	Úvod	3
2	Súčasn\acute{e} met\acute{o}dy pre detekciu cesty pre auton\acute{o}mne vozidl\acute{a}	4
2.1	Road Terrain Detection: Avoiding Common Obstacle Detection Assumptions Using Sensor Fusion	4
2.2	Road Detection and Corner Extraction Using High Definition Lidar	5
2.3	Road boundary detection and tracking for structured and unstructured roads using a 2D LiDAR sensor	5
2.4	RoadCompass: following rural roads with vision + ladar using vanishing point tracking	6
3	Detekcia cesty pre robota Toad	9
3.1	Robot Toad	9
3.2	Ground map	10
3.3	Velodyne LiDAR HDL-32E	13
3.4	ROS – robotic operating system	14
4	Klasifik\acute{a}cia s postupn\acute{y}m u\acute{c}en\acute{i}m	16
4.1	Support Vector Machine	16
4.2	SVM s postupn \acute{y} m u \acute{c} en \acute{i} m	17
4.3	Pegasonov algoritmus	17
4.4	Metriky pre vyhodnotenie \acute{u} spešnosti klasifik \acute{a} cie	18
5	N\acute{a}vrh detekcie cesty s využit\acute{i}m Velodyne LiDAR a SVM	19
6	Implement\acute{a}cia riešenia	22
7	Experimenty a v\acute{y}sledky	24
7.1	Dataset vytvoren \acute{y} robotom Toad	24
7.2	KITTI dataset	25
7.3	Anotované datasety a ich chyby	26
7.4	Testovanie	30
7.5	V \acute{y} ber zauj \acute{i} mav \acute{y} ch sn \acute{i} mkov	30
7.6	Experiment 1 – porovnanie \acute{u} spešnosti z \acute{a} kkladn \acute{e} ho riešenia a SVM s postupn \acute{y} m u \acute{c} en \acute{i} m bez predtr \acute{e} novania	33
7.7	Experiment 2 – SVM s postupn \acute{y} m u \acute{c} en \acute{i} m s predtr \acute{e} novan \acute{i} m na anotovan \acute{y} ch datasetoch	35
7.8	Vyhodnotenie a n \acute{a} vrh pokračovania pr \acute{a} ce	39

8 Záver	42
Literatúra	43

Kapitola 1

Úvod

Hľadanie cesty pre autonómne vozidlo je veľká téma posledných rokov. Automobilky sa predhávajú v tom, ktorá rýchlejšie a lepšie dokáže upozorniť šoféra vozidla pred kolíziou. Ale nie je to len o bezpečnosti, ale aj o možnosti bezpečného pohybu autonómnych vozidiel medzi nami.

Táto práca sa venuje hľadaniu cesty v okolí autonómneho vozidla. Hľadanie cesty je odhadnuté z mračna bodov z laserového radaru Velodyne LiDAR.

Riešenie diplomovej práce vylepšuje už existujúci algoritmus, kde nahradzuje prahovanie príznakov za strojové učenie support vector machine. Veľká časť práce je zameraná na testovanie a porovnávanie starého a nového riešenia.

V 2. kapitole o súčasných metódach je priblížených pár rozličných postupov, ako danú problematiku riešiť. Analýzou polárnych histogramov? Použitím fuzzy clusterov? Detekovaním a trakovaním okrajov cesty? Rýchlym prahovaním základných vlastností z priestorových dát? Alebo použitie fúzie viacerých senzorov na riadenie autonómneho vozidla v púšti? Každý z prístupov má svoje silné a slabé stránky.

V 3. kapitole je objasnené už existujúce riešenie pre autonómneho robota Toad. Kapitola načrtáva jednu z možných platforiem a aplikácií analýzy okolia nielen pre autonómne vozidlá. Približuje aké má Toad možnosti pri detekcii okolia. Táto kapitola tu je hlavne preto, lebo riešenie z ktorého práca vychádza, implementuje detektor pre túto platformu. V tejto kapitole je pôvodné riešenie analyzované a vyhodnotené.

Teoretické znalosti sú priblížené v 4. kapitole o klasifikácii s postupným učením. Presnejšie stručný prehľad základného SVM algoritmu a základy verzie s postupným učením. Poslednou témou tejto kapitoly sú teoretické znalosti k vyhodnoteniu snímok.

V kapitole o návrhu riešenia a kapitole o implementácii je priblížený postup úpravy pôvodného riešenia na nové s SVM. Ako sa hľadali parametre SVM a ako sa trénuje. Pri implementácii bolo treba riešiť je fakt, že pôvodná aplikácia očakáva dáta v inom formáte, ako sú v anotovanom datasete KITTI.

V kapitole o vyhodnotení sú priblížené jednotlivé použité datasety, ich možnosti a chyby. Porovnanie priemernej úspešnosti na malom počte snímok, ako aj na celých datasetoch. Pre jednotlivé snímky je priemerná úspešnosť pri starom riešení 58% a pri novom 71%. Pri vyhodnocovaní celých datasetov boli úspešnosti nižšie. Najhoršia hodnota pre pôvodné riešenie je 38% a 63% pre najlepšie. Pre algoritmus s SVM bol najhorší výsledok 48% a najlepší 72%.

Kapitola 2

Súčasné metódy pre detekciu cesty pre autonómne vozidlá

V tejto kapitole je priblížený prístup niektorých metód. Každá z nich používa iný prístup k preskúmaniu okolia cesty. Ako senzor je vždy zastupený laserový radar, či už v 2D alebo 3D verzií. Každá z nasledujúcich kapitol približuje inú metódu.

2.1 Road Terrain Detection: Avoiding Common Obstacle Detection Assumptions Using Sensor Fusion

Robustná metóda, ktorá spája viaceré vstupné senzory [17]. Metóda zvláda detekciu prekážok vo veľkom množstve rozličných scenárov s použitím minimálneho počtu parametrov. Prístup je založený na analýze priestorových vzťahov získaných z jednej kamery a 3D LiDAR senzoru. Testovanie prebiehalo v rozličných podmienkach z datasetu ROAD-KITTI s uspokojujúcim výsledkami (viac ako 80%).

Metóda využíva spojenie riedkeho a neštrukturovaného 3D mračna bodov a obrázku (z kamery). Vyžaduje kalibračný krok k odhadu vnútorných a vonkajších parametrov. Z týchto je možné spočítať transformáciu medzi 3D mračnom bodov z reálneho sveta a zobrazíť do 2D súradného systému obrázku. Hlavnou myšlienkou je využiť priestorové vzťahy v obrázku skombinované s 3D dátami k rozhodnutiu, či bod zodpovedá prekážke, alebo nie. Následne je vytvorený polárny histogram, ktorý je použitý na zostrojenie tzv. mapy dôvery. Táto mapa reprezentuje cestu v obrázku.

Celý systém pozostáva z postupného vykonávania piatich krokov:

- V prvom kroku sa tzv. spájajú dva vstupné senzory, čiže 3D mračno bodov sa premieta do obrazu.
- Druhý krok vytvára graf Delaunayovou trianguláciou¹, ktorý uchováva lokálne priestorové vzťahy medzi všetkými bodmi.
- V treťom kroku sa klasifikujú body, či sú alebo nie sú prekážkou. Jediným ukazovateľom je porovnanie výšky susedných bodov. (Obrázok 2.1)
- Štvrtý krok vytvára niekoľko polárnych histogramov, ktoré odhadujú voľné plochy.

¹https://en.wikipedia.org/wiki/Delaunay_triangulation

- V poslednom – piatom kroku je vytvorená tzv. mapa dôvery z kombinácie všetkých voľných plôch z predošlého kroku.



Obr. 2.1: Výsledok detekcie prekážok. Zelené body reprezentujú tie prechody kde nie sú prekážky, modré body prechod medzi prekážkou a cestou, červené body prechod v prekážke.

2.2 Road Detection and Corner Extraction Using High Definition Lidar

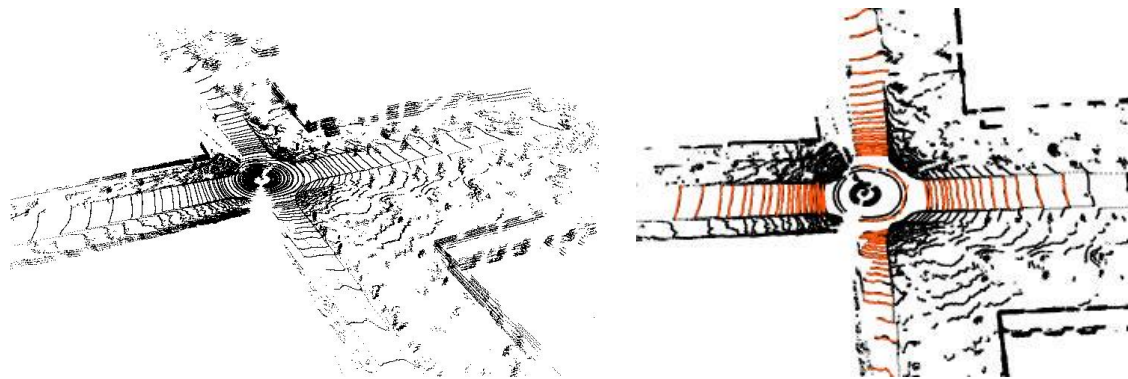
Algoritmus na detekovanie a hľadanie okrajov vozovky v 3D mračne bodov [21]. Analýza mračna bodov ako spôsob na rozpoznávanie cesty, alebo iného objektu, je časovo náročná úloha. Predstavovaný algoritmus je prispôbený pre rýchle predspracovanie a jeho parametre nepotrebujú postupné učenie. Je to dosiahnuté použitím Fuzzy cluster metódy, založenej na rozdeľovaní bodov podľa maximálnej entropie. Na rozdiel od niektorých metód filtre sú navrhnuté tak, aby pracovali s povrchom cesty a nie s jej okrajmi. Potom lineárne filtre viacnásobne vyvažované metódou najmenších štvorcov sú použité na rozlíšenie medzi lineárnym a nelineárnym segmentom rozložených bodov. Metóda na hľadanie rohov určí pozíciu budov na základe rozdielného rozloženia bodov. Priestorové závislosti medzi rozdielnymi laserovými detektormi sú zohľadnené vo výsledných vlastnostiach.

Keď robot naviguje v metskem prostredí automaticky, musí nájsť cestu a jej ohraničenie. Ako senzor bol zvolený Velodyne LiDAR HDL-64E. Získavanie príznakov prebieha pre každú skenovanú líniu zvlášť, následne sa vyváži naprieč 64 laserovým skenom v jednej línii. Týmto spôsobom algoritmus funguje rýchlejšie, než keby sa klasifikovalo celé mračno bodov.

2.3 Road boundary detection and tracking for structured and unstructured roads using a 2D LiDAR sensor

Algoritmus na detekovanie a následné sledovanie okrajov cesty pre spevnené aj nespevnené cesty [13]. Algoritmus získa príznaky cesty ako úsečky v polárnom súradnom systéme. Tieto príznaky sa sledujú v závislosti od súradného systému vozidla s použitím tzv. "nearest neighbor" filtra. Tento algoritmus presne detekuje okraje cesty bez ohľadu na typ cesty.

Systém sa skladá z týchto troch častí. Laserový senzor LiDAR vytvára dáta v polárnych súradniciach. Z týchto dát sú získané úsečky. Tieto úsečky sú korigované voči výške a natočeniu senzora.



Obr. 2.2: Detekcia cesty. Ľavý obrázok reprezentuje výstup z LiDAR HDL-64E. Pravý obrázok výsledok: oranžové línie predstavujú cestu [21].

Každá z priamok ohraničuje ľavý a pravý okraj. Ľavý a pravý okraj je meraným výstupom na sledovanie ľavého a pravého okraja cesty. Každý z okrajov používa na určenie pozície algoritmus Nearest Neighbor Filter.

2D lidar dáva dáta v polárnych súradniciach, tieto dáta sa ďalej nekonvertujú, ale používajú sa nezmenené v metóde na hľadanie rovných úsekov – priamok. Táto metóda prihliada aj na výšky, uhol náklonu a uhol stúpania terénu. Metóda na vyhľadávanie priamok je založená na Iterative-End-Point-Fit (IEPF) algoritme². Body, ktoré nie sú na pomyselných úsečkách, sa využívajú v algoritme na určenie hľadaných sekvencií. Následne sú získané úsečky v každej postupnosti dát z LiDARu.

Na hľadanie ohraňovania sa používa adaptívny algoritmus, jeho úlohou je nájsť nelinearitu vo vzdialenostných dátach ρ . Ak je rozdiel ρ_i a ρ_{i-1} väčší než adaptívny prah. Tento prah je odvodený z ρ_{i-1} s korekciou na presnosť senzora, šum a najhorší možný uhol odvodený z pre ρ_i .

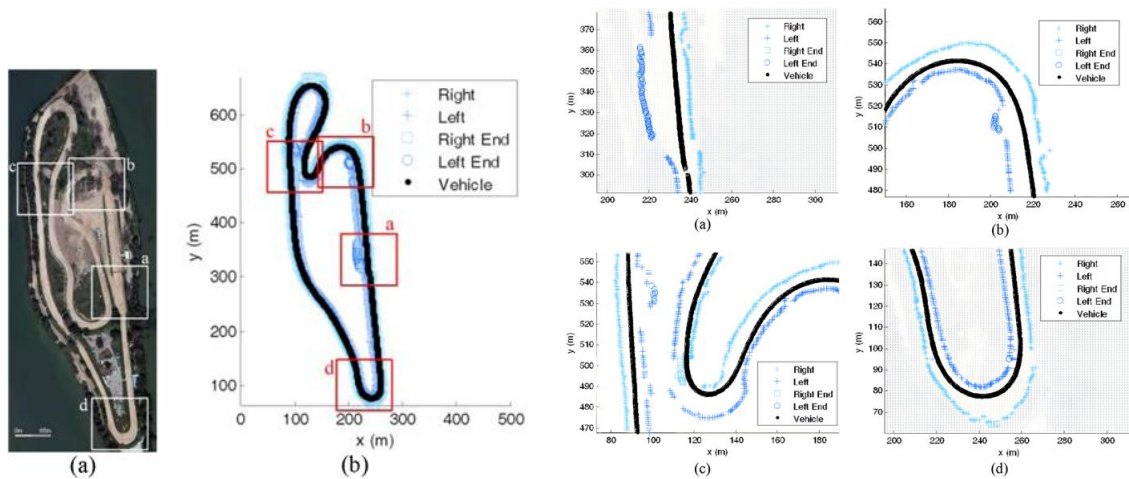
Výsledky pre nespevnené cesty ukázali, že napriek rýchlosti vozidla, ktorá bola počas testu od 30 km/h do 50 km/h a nerovnému povrchu, po ktorom vozidlo šlo, bola účinnosť metódy nad 92,0% pri detekovaní cesty s menej než 0,8% nesprávnou pozitívnou detekciou. Výsledok tohto experimentu je možné vidieť na obrázku 2.3.

Spevnené cesty sa rozdeľovali na dva typy – tie ktoré boli ohraňované obrubníkmi a tie ktoré nemali výrazný rozdiel vo výške ohraňovania cesty. Cesty s obrubníkmi v prvom testovacom scenári dosiahli aspoň 81,4% úspešnosť detekcie s menej než 2,7% nesprávnou pozitívnou detekciou. Druhý scenár dosiahol úspešnosť aspoň 85,8% s menej než 1,1% nesprávnou pozitívnou detekciou. Spevnené cesty s obrubníkmi dosiahli úspešnosť aspoň 95,7% s menej než 4,5% nesprávnou pozitívnou detekciou.

2.4 RoadCompass: following rural roads with vision + lidar using vanishing point tracking

Prístup k riešeniu autonómneho vozidla na rustikálnych cestách a na púšti, ktoré sú riadené na základe analýzy dát z LiDARu a kamery, boli rozsiahle testovania na uzavretých systémoch [11]. Analýza z kamery je založená na Gáborových vlnkových filtroch. Tieto fil-

²Borges and Aldon, 2004; Duda and Hart, 1973

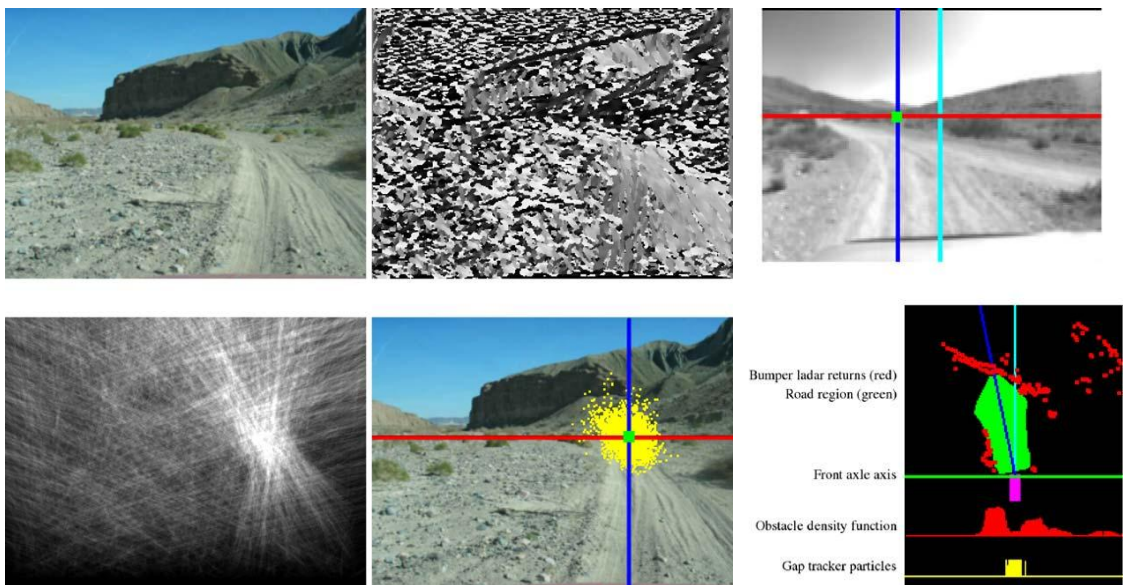


Obr. 2.3: Výsledok algoritmu pre nespevnené cesty. Ľavý obrázok ukazuje tréningový okruh. Pravý obrázok sporné detaily, kde chyba bola najväčšia [13].

tre vyhodnocujú textúru tak, aby našli koľaje, alebo stopy po vozidle, ktoré už tadiaľto prechádzalo. Z výstupu tohto filtra sa dá odvodiť obraz úbežného bodu (vanishing point) a to z výsledku Houghofho hlasovania sa získa odhad smeru pre ovládanie riadenia. Analýza dát z LiDARu slúži k detekcii prekážok, ktoré sú v bezprostrednej blízkosti ku smeru jazdy vozidla a následnej korekcií tak, aby sa vozidlo pohybovalo po ideálnej stope. Tento proces sa snaží priblížiť obrázok 2.4.

Metóda obsahuje tri významné kroky k odhadu tvaru cesty.

- Analýza textúry je vykonaná pomocou výpočtu dominantnej orientácie textúry v aktuálnej snímke.
- Lineárny odhad smeru cesty je zvolený na základe hlasovania všetkých orientovaných textúr pre najlepší ubežník.
- Určenie bočnej odchýlky dráhy od stredu cesty a šírka cesty sú vypočítané z LiDAR dát.



Obr. 2.4: Postup algoritmu pri určovaní cesty. Prvé štyri obrázky vľavo znázorňujú postup na nájdenie najlepšieho úbežníka a to od obrázku z kamery, cez zobrazenie textúry dominantných hrán. Tretí obrázok ukazuje výstup z Gáborových vlkových filtrov. Posledný zo štyroch obrázkov vľavo ukazuje výsledok hlasovania orientovaných textúr. Ostatné dva obrázky vpravo ukazujú spojenie výstupu z analýzy obrázka a laserového senzoru [11].

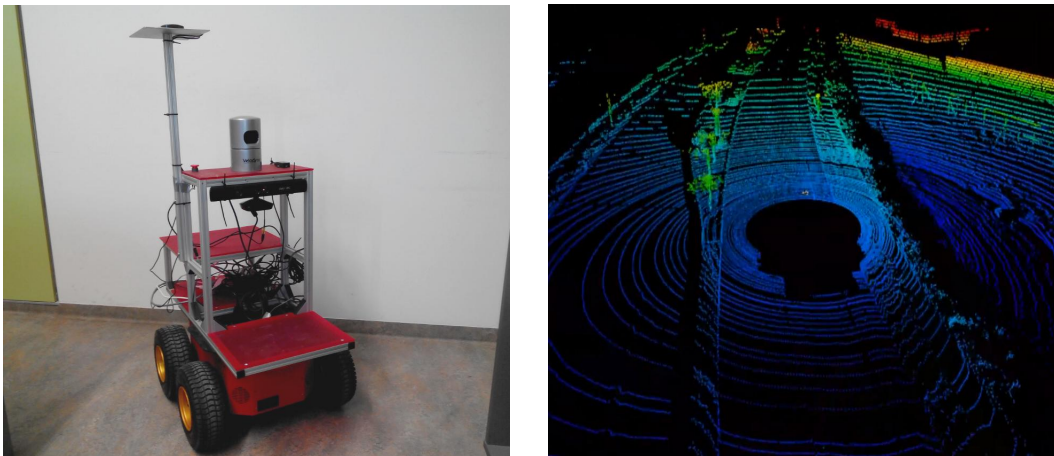
Kapitola 3

Detekcia cesty pre robota Toad

Táto práca sa venuje rozpoznávaniu cesty pre robota, ktorý má osadený laserový radar Velodyne LiDAR. Na záznam dát a riadenie používa robotický framework ROS. Pre analýzu dát z radu je možné a vhodné použiť niektorý z už existujúcich frameworkov (napr. Point Cloud Library).

3.1 Robot Toad

Toad je robotická platforma výskumnej skupiny Robo@FIT. Tento robot je zameraný na vývoj systémov na navigáciu, prehľadávanie, lokalizáciu a mapovanie v reálnom čase (simultaneous localization and mapping – SLAM). Toad (obrázok 3.1) disponuje senzorom Kinect ©. Tento senzor okrem RGB obrázku poskytuje aj hĺbkovú mapu. Ďalším senzorom je laserový radar Velodyne LiDAR, presnejšie model HDL-32. Na určenie polohy používa miniature GPS-Aided Inertial Navigation System. Všetky senzory sú v pohybe vďaka podvozku Pioneer 3-AT. Na jednotné riadenie sa používa pseudo-operačný systém ROS bežiaci vo verzii Hydro [15, 7].



Obr. 3.1: Ľavý obrázok je robot Toad. Pravý obrázok je príklad výstupného mračna bodov zo senzoru LiDAR [7].

3.2 Ground map

Celá kapitola napísaná podľa zdrojového kódu z [1].

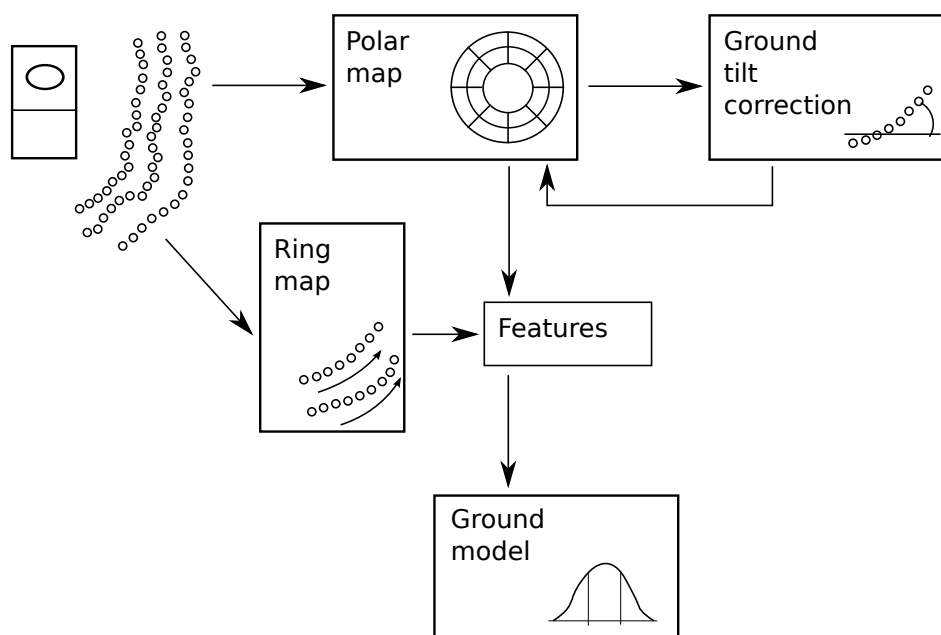
Ďalším existujúcim riešením je skupina aplikácií pre spracovanie dát z laserového radaru Velodyne Lidar. Toto riešenie bude brané ako základ pre ďalší vývoj aplikácie pre túto diplomovú prácu.

Projek založený vedeckou skupinou Robo@FIT implementuje tri aplikácie Laser Scan, Cloud Assembler a Ground Map. Každá z nich je nezávisle spustiteľná a použiteľná. Celý balík aplikácií sa volá `but_velodyne_proc`.

Cloud Assembler spája viaceré snímky radaru do jedného obrázku. Týmto dosahuje hustejší model. Aplikácia Laser Scan simuluje konvečný 2D skener.

Posledná zo spomínaných je bližšie približná v nasledujúcich riadkoch.

Ground map je aplikácia, ktorá na vstupe očakáva Velodyne 3D sken a výstupom je mapa okolia robota s bezpečnou cestou. Algoritmus na zistenie bezpečnej cesty používa dva prístupy. Body z velodyne v okolí robota sú rozdelené do buniek. Na bunku sa dá pozerat ako na najmenší prvok z ktorého sa odvodzuje cesta. Z týchto buniek sa spočítajú príznaky. Na základe vyhodnotenia príznakov je vyhodnotené, či bunka je, alebo nie je cesta. Blokovú schému približuje obrázok 3.2.



Obr. 3.2: Bloková schéma metódy [16].

Ako vstup do algoritmu je štruktúra `velodyne_pointcloud::PointXYZIR`. Každý bod je reprezentovaný troma súradnicami, intenzitou a určením čísla prstenca. Číslo prstenca sa používa pri vyhodnocovaní príznaku hranatosti.

Ako prvý krok treba odhadnúť zem. Tento odhad základovej dosky je vykonaný za pomoci point cloud library a RANSAC algoritmu. Od zeme sa odvodzuje výška bodov (súradnica z).

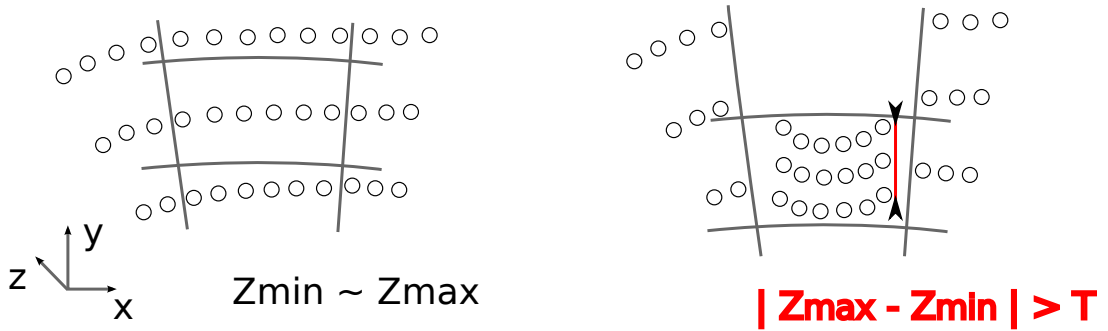
Algoritmus prechádza bunky v dvoch smeroch. Po prstencoch a polárne. Po prstencoch zisťuje príznak hranatost (obrázok 3.3), polárne zase príznak drsnost (obrázok 3.4),

posledným použitým príznakom je odrazivosť.

Model cesty je popísaný troma gausovými funkciami. Každá z funkcií je odvodená z buniek, ktoré by mohli byť cestou. Tento odhad je založený na minimálnom počte vzoriek v bunke a parametre najväčšieho rozdielu výšky medzi bodmi v bunke.

Príznačky použité pre výpočet modelu cesty:

- Drsnosť (roughness)
- Intenzita povrchovej odrazivosti (reflectivity)
- Hranatosť (edginess)



Obr. 3.3: Príznačky získané z analýzy polárnej mapy. Obrázky ukazujú príznak drsnosť. [16].

Príznak drsnosť G je pre bunku charakterizovaný ako rozptyl výšky od zeme:

$$G_{var} = \frac{1}{N} \left(\sum_{i=1}^N (p_z^i)^2 - \frac{1}{N} \left(\sum_{i=1}^N p_z^i \right)^2 \right) \quad (3.1)$$

kde N je počet bodov v bunke, p_z je súradnica z bodu p .

Príznak odrazivosť R je pre bunku charakterizovaný ako priemerná výška od zeme:

$$R_{sum} = \frac{1}{N} \sum_{i=1}^N p_{intensity}^i \quad (3.2)$$

kde N je počet bodov v bunke, p_z je súradnica z bodu p .

Pre príznak hranatosť E sa body podľa informácie z mračna bodov rozdelia do prstencov. Pre rozdelenie sa použije informácia o indexe prstenca. Prstence sa ďalej delia do buniek, toto delenie je už na základe zvolených uhlov.

Najprv je treba vypočítať pozíciu bunky voči radaru (vzťah 3.3).

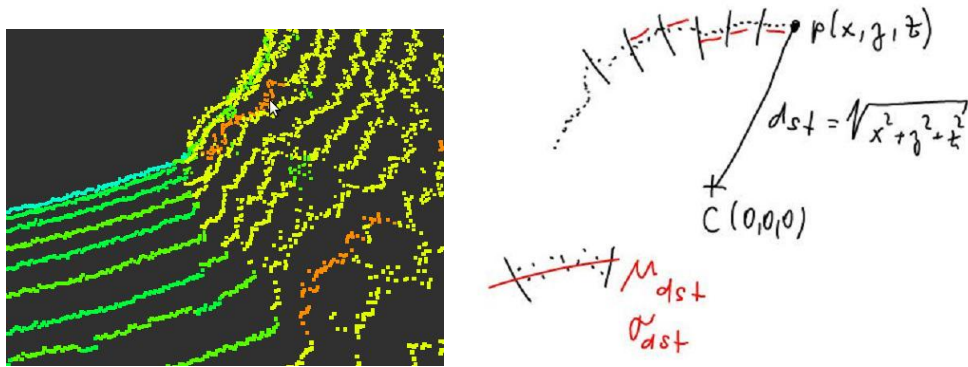
$$\text{bin}_{\text{pos}} = \min \left(\sqrt{(p_x^i)^2 + (p_y^i)^2 + (p_z^i)^2} \right) \quad \left| \quad i \in 1 \dots N \right. \quad (3.3)$$

kde N je počet bodov v bunke, p_x je súradnica x bodu p .

Hranatosť bunky je potom porovnanie dvoch susedných bodov:

$$E = \left| \text{bin}_{\text{pos}}^{\text{current}} - \text{bin}_{\text{pos}}^{\text{next}} \right| \quad (3.4)$$

kde current je aktuálna bunka, next je susedná bunka na rovnakom prstenci.



Obr. 3.4: Na ľavom obrázku je vidieť povrchovú odrazivosť (zmena farby bodu), tieto dáta sú k dispozícii priamo od Lidaru. Na pravom obrázku príznaky získané z analýzy hranatosti [16].

Gausové funkcie popisujúce cestu z vyššie spomenutých príznakov:

$$\mu_R = \frac{1}{K} \sum_{j=1}^K R_{var}^j \quad (3.5)$$

$$\sigma_R = \sqrt{\frac{1}{K} \left(\sum_{j=1}^K (R_{var}^j)^2 \right) - \frac{1}{K} \left(\sum_{j=1}^K R_{var}^j \right)^2} \quad (3.6)$$

kde K je počet buniek, z ktorých sa počítali príznaky.

Vzorce pre gausové funkcie ostatných príznakov sú takmer identické.

Po určení modelov sa vyhodnotí pravdepodobnosť, či bunka je cesta. Výsledná pravdepodobnosť je len súčinom pravdepodobností modelov príznakov.

Prahovanie modelov je založené na dvoch faktoroch. Výška bodov v bunke a pravdepodobnosť, či bunka patrí do modelu. Výška je vzdialenosť od zeme, ktorá sa porovnáva s prahom a vstupným parametrom vyjadrujúcou maximálnu nepravidelnosť cesty. Príslušnosť je súčin výsledkov Gausových funkcií modelu, ktorá musí byť väčšia než zvolený prah.

Výsledná mapa úspešne detekuje hrany prekážok, ale výsledok pre väčší počet buniek býva vyhodnotený ako neznámy.

Zhodnotenie detekcie

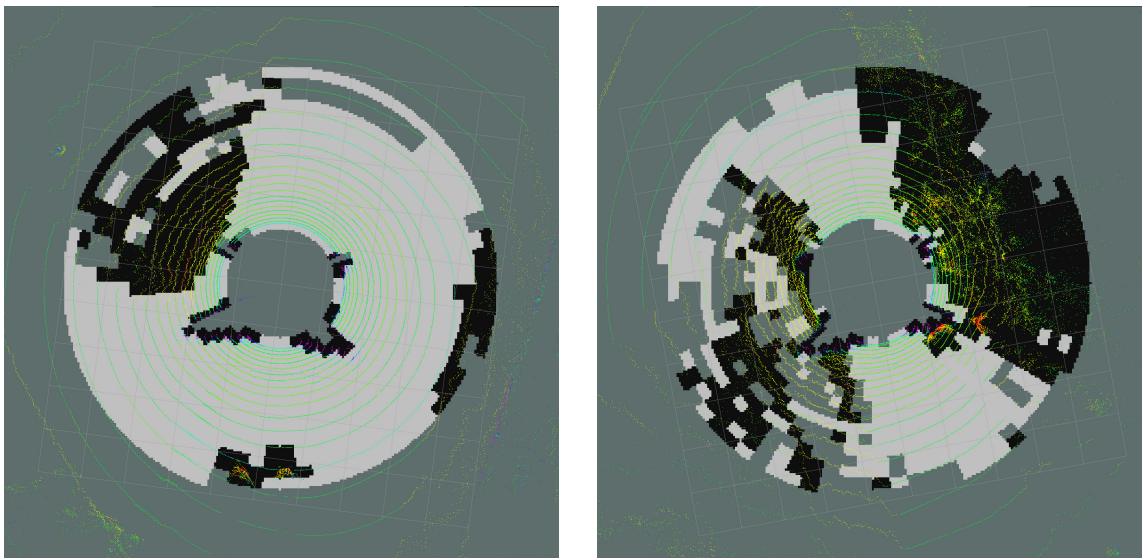
Ako je vidieť na obrázkoch 3.5, detekcia cesty funguje rozumne, ak je cesta v okolí robota rovná, bez väčších prekážok a s odrazivosťou prostredia prakticky nemennou. Obrázok 3.6 zase ukazuje nevydarenú detekciu okolia robota.

Na obrázkoch 3.5 a 3.6 je vidieť ortogonálny pohľad na robota, ktorý je otočený zospodu navrch. Uprostred obrázka je radar. Najbližšie úseky sú zatienené ostatnými senzormi a konštrukciou robota.

Scéna (obrázok 3.5 vľavo), v ktorej sa robot nachádza, sa dá popísať ako 'T' križovanie väčšieho a menšieho chodníka. Robot sa nachádza v tesnej blízkosti trávnatého povrchu, ktorý sa nachádza po jeho ľavici. Tento trávnatý úsek je detekovaný správne. Je vidieť, že body v bunkách nie sú v takmer dokonalých kruhoch a majú inú svetelnú odrazivosť

(iná farba bodu). Za robotom sa nachádzajú osoby. Napravo od robota na samom okraji vyhodnocovanej oblasti je správne nájdený okraj cesty.

Na obrázku 3.5 vpravo sa robot nachádza na tenkom chodníku. Po pravici je správne detekovaný vysoký krík a dve osoby. Na ľavej strane je vidieť obrys trávinatej plochy, ale detekcia prekážky nie je celistvá. Najbližšie bunky pri robotovi sú detekované ako prekážka, ale veľké množstvo buniek je neznámych.



Obr. 3.5: Na obrázkoch je vidieť výstup z pôvodného algoritmu. Oba obrázky ukazujú dobrý odhad cesty [1].

Toto riešenie sprevádza niekoľko väčších, či menších problémov. Na obrázku 3.6 vľavo sa robot nachádza na úzkom chodníku, ktorý je obkolesený trávnatou plochou. Na tejto snímke je vidieť, že pre veľmi veľké množstvo buniek nebolo možné detekovať, či je prekážkou alebo cestou. Tieto bunky vznikajú keď počet bodov bunke je menší než tri, alebo nebolo možné odhadnúť pravdepodobnosť bunky.

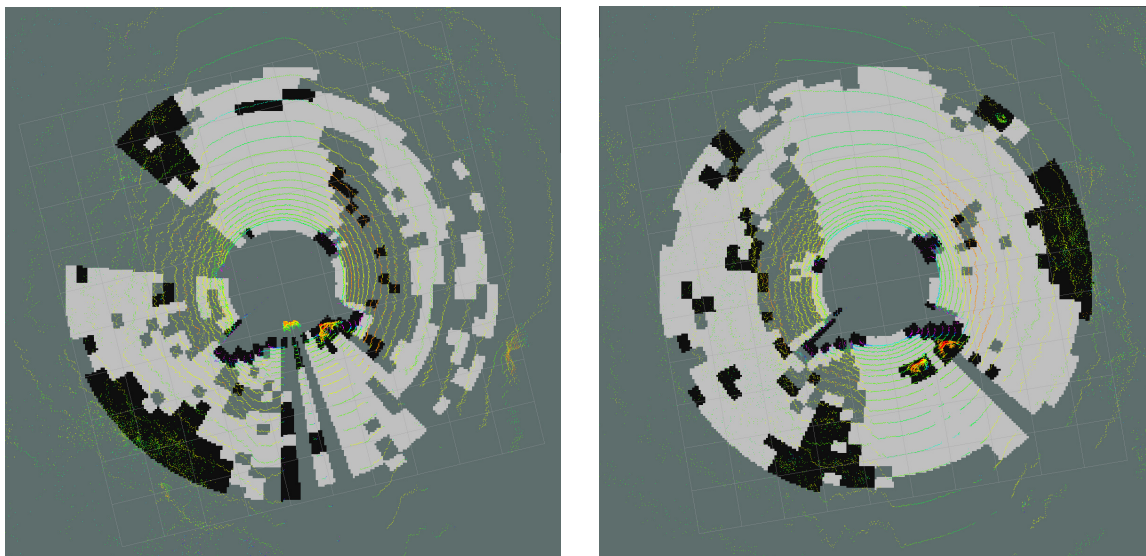
Na obrázku 3.6 vpravo sa robot nachádza v rovnakej scéne ako na obrázku vľavo. Napravo od robota je vidieť trávnatá plocha, ktorá je vyhodnotená ako cesta. Tento problém je spôsobený zle nastavenými prahmi. Naľavo od robota už nie je trávnatá plocha cestou, ale vo väčšej vzdialenosti správne detekovaný krík.

Algoritmus má problém, ak sa robot nachádza na naklonenej rovine. Ak napríklad stúpa do kopca, tak sa zem nedá jednoznačne odhadnúť. Keďže príznak drsnot je definovaný ako rozptyl od zeme, tak aj bunka, ktorá má body v rovine, môže byť detekovaná ako prekážka.

Slabinou tohto prístupu sú vopred určené prahy, ktoré nedovoľujú prispôbenie sa zmene okolia. Čiže napr. ak by robot mal prejsť po prašnej ceste, ale prahy by boli nastavené na rovný asfaltový povrch. Výhodou tohto riešenia je rýchlosť spracovania snímkov.

3.3 Velodyne LiDAR HDL-32E

Velodyne je americká spoločnosť, ktorá sa okrem iného venuje aj vývoju laserových senzorov LiDAR. Jedným zo senzorov je práve HDL-32E.



Obr. 3.6: Na obrázkoch je vidieť výstup z pôvodného algoritmu. Obrázok vľavo ukazuje veľké množstvo neindikovatelných buniek (zfarbené do šeda). Obrázok vpravo ukazuje, zľú detekciu prekážok vpravo od robota [1].

HDL-32E využíva 32 laserov zarovnaných medzi $+10^\circ$ a -30° , tak, aby poskytol kolmé zorné pole. Senzor používa rotojúcu hlavu na frekvencii 10 Hz tak, aby v reálnom čase pokryla 360° horizontálneho zorného poľa. Výstup z radaru je kontinuálne mračno bodov o 700 000 bodov za sekundu. Dosah s presnosťou do 2 cm je 100 m [2].

3.4 ROS – robotic operating system

Operačný systém pre robotov (ROS) je flexibilný framework na písanie robotického softwaru. Obsahuje zbierku nástrojov, knižníc a konvencií, ktorých cieľom je zjednodušiť vývoj komplexných a robustných popisov správania robota naprieč širokou rozmanitosťou robotických platforiem [4].

ROS je open source pseudo-operačný systém, ktorý ponúka abstrakciu nad hardwarom, low-level prístup k riadeniu zariadení, implementáciu bežne používanej funkcionality a balíčkový systém. Komunikácia medzi procesmi je v ROS implementovaná rôznymi spôsobmi, ako synchronizovaný RPC cez služby (services), asynchrónny prúd dát cez tzv. témy (topics), alebo dáta uložené v tzv. parametrovom serveri [5].

Projekt Ground map používa na komunikáciu témy. Témy sú pomenované zbernice, cez ktoré si uzly vymieňajú správy. Témy majú anonymnú producent/konzument sémantiku, ktorá oddeľuje informácie o producentovi od jeho konzumenta. Čo znamená, uzly nepoznajú s kým komunikujú. Namiesto toho uzly, ktoré sa zaujímajú o dáta, sa prihlasujú ako konzumenti k téme, alebo ak sú uzly producentmi, tak sa prihlásia k téme. Je podporované viacnásobne prihlásenie ako producent, či konzument k jednej téme [6].

Súčasťou ROS je aj nezávislá C++ knižnica na spracovanie 3D mračna bodov Point Cloud Library.

Framework Point Cloud Library (PCL) je projekt na prácu s mračnom bodov a na spracovanie 2D/3D obrázkov. V PCL je možné nájsť najvýkonnejšie algoritmy na filtrovanie,

odhad príznakov, odhad povrchu, segmentácia Tieto algoritmy sa dajú využiť na napr. na filtrovanie extrémov zo zašumených dát, spájanie 3D mračien bodov dokopy, rozdelenie scény na súvisiace časti, vytváranie povrchov, vizualizácia,

PCL je open source software, ktorý je možné použiť na rôznych platformách ako Linux, MacOS, Windows a Android / iOS [3].

Pre potreby ROS sú v PCL implementované nové dátové typy, možnosť ako tieto typy prevádzať a metódy na publikovanie alebo odber správ. Táto funkcionálnosť je využitá v aplikáciách ground map.

Kapitola 4

Klasifikácia s postupným učením

4.1 Support Vector Machine

Support Vector Machine (SVM) je jeden z modelov strojového učenia, ktorý sa zväčša trénuje tzv. s dohľadom (supervised learning) [9]. Najčastejšie využitie nájde v klasifikácii a regresnej analýze. Vstupom je trénovacia sada, kde pre každú vzorku je vopred známa príslušnosť do jednej z dvoch kategórií. SVM potom vytvorí model, ktorý je schopný priradovať nové prvky do jednej z dvoch kategórií. Preto sa táto metóda radí medzi nepravdepodobnostné binárne lineárne klasifikátory. Ako rozšírenie je tiež možné použiť nelineárnu klasifikáciu pomocou využitia jadra (kernel).

Lineárna metóda, alebo Lineárne SVM je metóda, ktorá rozdeľuje triedy pomocou priamky 4.1 [10, 9].

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \quad (4.1)$$

Kde \mathbf{w} je kolmý vektor k priamke, \mathbf{x} je vektor reálnych čísel. Hľadanie priamky prebieha pomocou nájdenia maximálneho okraja. Okraj lineárneho klasifikátoru (obrázok 4.1) je definovaný ako šírka hranice, ktorá môže byť rozširovaná pokiaľ neprekrie definovaný bod.

Okraj je popísaný dvoma rovinami:

$$\mathbf{w} \cdot \mathbf{x} + b = 1 \quad (4.2)$$

$$\mathbf{w} \cdot \mathbf{x} + b = -1 \quad (4.3)$$

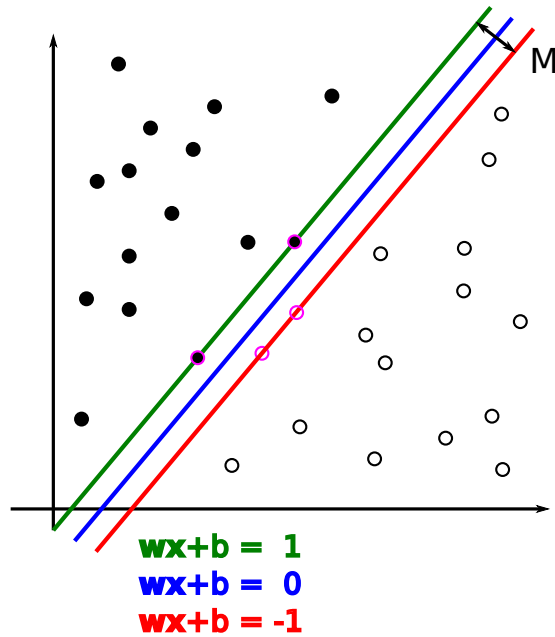
Body, ktoré sa nachádzajú na okrajových priamkach sa nazývajú pomocné vektory (support vectors).

Šírka okraja M je potom definovaná vzorcom 4.4.

$$M = \frac{2}{\sqrt{\mathbf{w} \cdot \mathbf{w}}} \quad (4.4)$$

Priamka, ktorá najlepšie rozdeľuje množiny bodov je tá, ktorá má najväčší okraj.

Základná verzia metódy pracuje s vopred známou množinou trénovacích dát. Po spracovaní celej trénovacej množiny je možné použiť model ku klasifikácii. Tento prístup sa volá offline. Existuje ale aj postupné učenie.



Obr. 4.1: Okrajové funkcie pre SVM tréovanie so vzorkami z dvoch tried sú zelené pre jednu triedu a červené pre druhú. Deliaca priamka je modrou farbou. Fialový okraj majú body nazývané support vectors, tieto body určujú maximálne M [10].

4.2 SVM s postupným učením

Postupné učenie je výhodné, ak sa musíme vysporiadať s veľmi veľkými alebo nestálymi (non-stationary) dátami. V prípade nestálych dát by učenie na celej konečnej množine obecnne zlyhalo, ak by sa napr. rozdelenie menilo v čase. Na veľa zaujímavých problémov sa v strojovom učení môže hľadať ako na vhodné pre postupné učenie [18]. Praktická výhoda tohto prístupu je, že umožňuje zahrnúť ďalšie tréovacie dáta vtedy, keď sú dostupné a to bez toho, aby bolo potrebné opätovne celú sieť pretréovať. Je dané, že tréovanie je zvyčajne výpočetne najnáročnejšie a nikoho neprekvapí, že prístupnosť algoritmov, ktoré sa dokážu postupne učiť sú vhodné pre prácu s veľkým množstvom dát. Ďalším dobrým príkladom je odhad v reálnom čase pre nepretržitý prúd dát, ako napr. dolovanie dát z web-u alebo rozhranie medzi mozgom a počítačom.

4.3 Pegasonov algoritmus

Jednoduchý a efektívny algoritmus tréovania SVM, ktorý pracuje na princípe stochastického sub-gradient descent pre riešenie optimalizačných problémov nachádzajúcich sa v SVM [20]. Je dokázané, že počet iterácií potrebných k získaniu výsledku s presnosťou ϵ je $\tilde{O}(1/\epsilon)$, kde každá iterácia pracuje s jednoduchým tréovacím príkladom. Na rozdiel od predchádzajúcej analýzy stochastických gradient descent metód pre SVM vyžaduje $\omega(1/\epsilon^2)$ iterácií. V predchádzajúcom výskume dokázali, že počet iterácií rastie lineárne s $1/\lambda$, kde λ je regularizačný parameter SVM. Pre lineárne jadro je celkový čas behu v tomto algoritme určený $\tilde{O}(d/(\lambda\epsilon))$, kde d je počet nenulových príznačov v každom z príkladov. Pretože čas behu algoritmu priamo nezávisí na veľkosti tréovacej sady, tak je vhodný na použitie pre veľké

datasets. Tento algoritmus umožňuje použitie nelineárnych jadier, ale v tomto prípade čas behu algoritmu lineárne závisí od veľkosti trénovacej sady.

Po analýze efektívnosti algoritmu sa zistilo, že pre lineárne jadrá Pegasos dosahuje výrazne lepšie výsledky ako konkurenčné algoritmy. Pri použití komplikovanejších jadier, algoritmus je stále porovnateľný so zložitejšími prístupmi.

4.4 Metriky pre vyhodnotenie úspešnosti klasifikácie

Na vyhodnotenie úspešnosti klasifikácie algoritmu sú použité binárne štatistické ukazovatele presnosť a citlivosť (*Precision & Recall*) [8].

Tieto ukazovatele zavádzajú pojmy ako:

- pravdivo pozitívne (*true positives*)
- pravdivo negatívne (*true negatives*)
- nepravdivo pozitívne (*false positives*)
- nepravdivo negatívne (*false negatives*)

Ktoré vyjadrujú vzťah medzi odhadom a skutočnosťou. Pravdivo pozitívne a pravdivo negatívne sú snímky, ktoré boli správne určené, v tomto prípade ak cesta bola vyhodnotená ako cesta a prekážka ako prekážka. Nepravdivo pozitívne sú zase tie body, ktoré sú prekážkou, ale boli vyhodnotené ako cesta. Nepravdivo negatívne sú body, ktoré boli vyhodnotené ako prekážka, ale pritom sú cestou.

Presnosť (4.5) vyjadruje vzťah medzi bodmi, ktoré boli správne vyhodnotené ako cesta a všetkými vybranými bodmi. Citlivosť (4.6) určuje koľko plochy cesty bolo správne určených.

Ďalší ukazovateľ (4.7) určuje mieru, s akou presnosťou boli určené prekážky – *true negative rate*. Je to rovnaký ukazovateľ ako citlivosť, len pre negatívne vzorky. Ukazovateľ *accuracy* (4.8) ukazuje, koľko z celého počtu snímkov bolo správne klasifikovaných.

F-measure (4.9) je harmonickým priemerom ukazovateľov presnosť a citlivosť.

$$\text{Precision} = \frac{tp}{tp + fp} \quad (4.5)$$

$$\text{Recall} = \frac{tp}{tp + fn} \quad (4.6)$$

$$\text{True negative rate} = \frac{tn}{tn + fp} \quad (4.7)$$

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn} \quad (4.8)$$

$$\text{F-measure} = 2 \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4.9)$$

Kde tp sú snímky, ktoré boli správne vyhodnotené ako cesta. tn snímky, ktoré boli správne vyhodnotené ako prekážka. fp snímky, ktoré boli nesprávne vyhodnotené ako cesta. fn snímky, ktoré boli nesprávne vyhodnotené ako prekážka.

Kapitola 5

Návrh detekcie cesty s využitím Velodyne LiDAR a SVM

Táto práca nadväzuje na už existujúce riešenie Ground map (kapitola 3.2) a rozširuje ho.

Aplikácia Ground map funguje rozumne, ak je splnených niekoľko podmienok. Prvou z nich je, že rozdiel medzi cestou a prekážkou je výrazný. Napríklad, ak ide robot po chodníku obkolesenom trávnatou plochou, tak detekcia cesty funguje dobre.

Druhou dôležitou podmienkou je, že robot sa pohybuje po takmer rovnej ploche. Toto obmedzenie vychádza zo zvoleného príznaku drsnosť. Tento príznak je odvodený od vzdialenosti od zeme, tá je ale odhadnutá zo scény. Čiže sa môže stať, že aj bunka, ktorá je rovná môže byť označená ako prekážka a to kvôli priemernej vzdialenosti od nepresne odhadnutej zeme.

Výsledná mapa často pôsobí zašumene a to hlavne kvôli veľkému množstvu buniek, ktoré sú označené ako neznáme. Tento stav môže nastať, ak bunka obsahuje menej než tri body. Častejšie sa stáva, že výsledná pravdepodobnosť (súčin príznakov drsnosť, odrazivosť a hranatosť) je menšia než zvolený prah.

Cielom rozšírenia tejto práce by malo pri zachovaní rovnakých príznakov (aj s ich chybami) zlepšiť odhad cesty. K dosiahnutiu tohoto cieľu je použité strojové učenie. Strojové učenie môže nájsť v zdanlivo nesúvisiacich príznakoch závislosť, ktorá by sa len ťažko analyticky hľadala.

Možností ako strojovo učiť je mnoho. Pre tento typ príkladu som sa rozhodol použiť SVM a to hlavne z toho dôvodu, že sa ponúka možnosť využiť už navrhnuté príznaky. Príznaky sa dajú bez úprav použiť ako vstup do SVM. Základná verzia SVM podporuje binárnu klasifikáciu, čo v prípade detekcie cesty je dostačujúce.

Pre porovnanie s SVM by sa dali vybrať neuronové siete. Neurónové siete síce používajú iný prístup k trénovaniu, ale je to jedna z možných alternatív, ktorým by sa úloha detekcie cesty dala riešiť. Za výhodu by sa dalo považovať nepotrebnosť hľadať správne príznaky. Nevýhodou je, že pre natrénovanie neuronovej siete v rozumnom čase, je potrebný veľký výpočetný výkon.

Typ SVM som zvolil s postupným učením a v implementácii Pegasos (kapitola 4.3). Pri tomto kroku som chcel zistiť nakoľko sa SVM dokáže s danými príznakmi prispôbiť meniacemu sa okoliu.

Pri riešení netreba zabudnúť na chyby spôsobené nesprávnymi anotáciami v datasetoch. Tieto chyby ovplyvňujú učenie s učiteľom. Aký vzťah bude mať učenie s učiteľom, ale s nedokonalými anotáciami a učenie bez učiteľa. Ak by sme brali v potaz neuronové

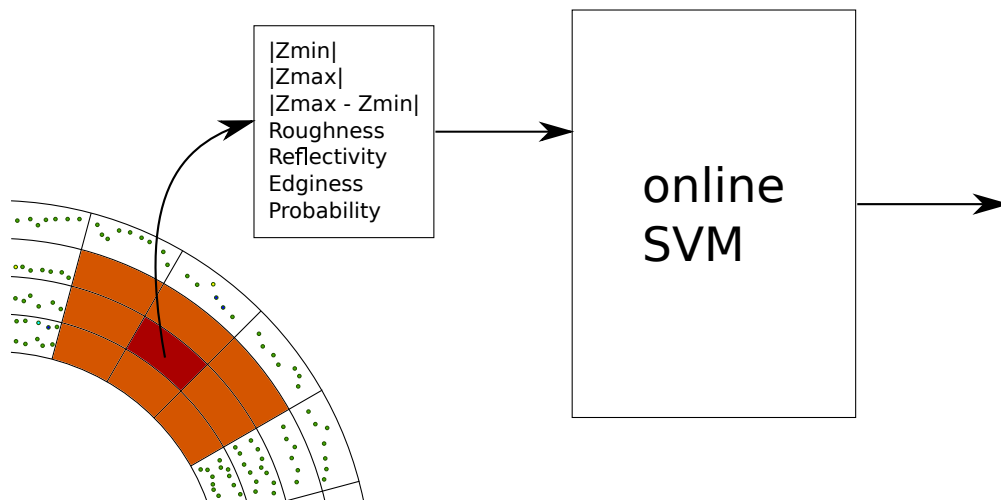
siete, tak je otázne, či sú vhodné tréovanie s učiteľom na dátach, ktoré môžu obsahovať nezanedbateľné chyby.

Návrh SVM s postupným učením bol prevažne výsledok experimentálneho bádania medzi rôznymi parametrami. Tieto parametre boli: zvolenie jadra kernelu a jeho parametre, vybrať príznaky, ktoré sa bude SVM učiť a nakoniec vybrať vhodné bunky pre učenie. Pri rozhodovaní sa zväčša postupovalo experimentálne, tento postup by sa dal prirovnať k optimalizačnej metóde hill-climbing, ale vykonávanej človekom.

SVM

Kernel, ktorý je použitý v SVM je radial bases kernel¹. Pri návrhu sa experimentovalo aj s inými kernelmi a ich parametrami. Príklady kernelov sú sigmoid, polynomial a linear.

SVM s postupným učením je pridaný na úplný záver vyhodnocovacieho algoritmu, čiže po vyhodnotení všetkých príznakov sa spracované bunky prechádzajú postupne. Pri analyzovaní každej bunky je potrebné vyhodnotiť, či je bunka vhodná pre učenie SVM. Výber vhodných buniek, je navrhnuté tak, aby vyberalo bunky, ktoré sú extrémne. Čiže len tie, ktoré by sa dali jednoducho opísať ako najrovnejšie, alebo najčlenejšie, čiže práve tie, ktoré sa najväčšou pravdepodobnosťou boli, alebo neboli cestou.



Obr. 5.1: Schéma zapojenia online SVM.

Extrémne bunky sa dajú popísať príznakmi cesty, ako drsnosť (vzťah 3.1), hranatosť (vzťah 3.5) a odrazivosť (vzťah 3.2). Tie bunky, ktoré majú výslednú pravdepodobnosť, čiže súčin príznakov väčší než zvolená minimálna pravdepodobnosť pre cestu sú tréované ako cesta. Naopak tie bunky, ktoré majú pravdepodobnosť menšiu než zvolenú maximálnu pravdepodobnosť pre prekážku sú tréované ako prekážka. Ak je bunka vybraná pre učenie, berie sa jej celé okolie bez toho, aby sa určovalo, či sú extrémne alebo nie. Pravdepodobnosti sa dajú určiť ako vstupné parametre.

Dôležitým faktorom pre učenie je vyvažovanie pozitívnych a negatívnych vzorkov. Vyvažovanie znamená, že počet tréovaných pozitívnych buniek nie je výrazne vyšší ako počet negatívnych. Táto optimalizácia bola pridaná až po tom, čo detekcia bola výrazne ovplyvnená prvými snímkami datasetov.

¹https://en.wikipedia.org/wiki/Radial_basis_function

Ako je na obrázku 5.1 vidieť, vstupom do SVM nie je nikdy len práve analyzovaný bod, ale aj jeho okolie. Okolie je vypočítané z polárnych súradníc a sú vybrané susedné bunky. Čiže 7 príznakov pre každú z 9 buniek. Týmto dosiahneme, že SVM dostane väčšie povedomie o okolí a môže sa naučiť, aj na prvý pohľad ťažko určiteľné vzťahy.

Príznačky boli vybrané z viacerých kombinácií. Pri výbere príznakov bol experimentoval s rôznymi kombináciami. Prekvapujúce bolo, že príznaky ako priemerná odrazivosť bunky, alebo rozptyl odrazivosti pre bunku, neprispievali k lepšiemu odhadu.

Príznačky, ktoré sa používajú pre tréning online SVM sú:

- $|z_{min}|$
- $|z_{max}|$
- $|z_{max} - z_{min}|$
- roughness
- reflectivity
- edginess
- probability (roughness · reflectivity · edginess)

Príznačky z_{min} a z_{max} sú minimálna a maximálna absolútna vzdialenosť bunky od odhadu zeme v ose z . Príznaček *probability* je súčinom *roughness*, *reflectivity* a *edginess*, pričom tieto príznaky sú bližšie vysvetlené v kapitole 3.2. Pre každú bunku je vyhodnotený SVM a určené či je, alebo nie je cesta.

Normalizácia snímok

Pred učením je potrebné príznaky normalizovať. Normalizácia snímku je vyjadrená vzťahom 5.3:

$$\bar{p} = \frac{1}{N} \sum_{i=1}^N p^i \quad (5.1)$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (p^i - \bar{p})^2} \quad (5.2)$$

$$p_n^i = \frac{p^i - \bar{p}}{\sigma} \quad i \in 1 \dots N \quad (5.3)$$

Kde N je počet vzorkov. Vzťah 5.1 je priemer \bar{p} príznakov p . Vzorec 5.2 je smerodajná odchýlka σ .

Kapitola 6

Implementácia riešenia

Implementácia vychádza a rozširuje už existujúce riešenie. Aplikácia je navrhnutá pre pseudo-operáčny systém ROS vo verzii Groovy, čiže je samostatne spustiteľným ROS uzlom. Uzol komunikuje cez jednu vstupnú a jednu výstupnú tému. Pričom aplikácia čaká na nadviazanie oboch prúdov komunikácie a až potom začne spracovávať vstup a poskytovať výstup. V aplikácii je možné zapnúť `#define PHOTOS` na ukladanie výsledných máp ako obrázky v odtieni šedej do súboru.

Keďže vývoj aplikácie bol vykonaný na dátach nahraných Toadom (kapitola 3.1), bolo treba aplikáciu upraviť tak, aby bolo možné načítať dáta v inom formáte. Nový formát vyžaduje vypočítať index prstenca. Index prstenca sa počíta pre každý bod zvlášť.

Ako druhá zmena bolo treba zmeniť scénu, čiže veľkosť analyzovanej časti. Zvolené parametre sú v tabuľke 6.1.

angular_res	3
radial_res	0.3
min_range	3.7 m
max_range	20 m

Tabuľka 6.1: Parametre aplikácie pre veľkosť a rozlíšenie analyzovanej scény.

Parametre `angular_res` a `radial_res` určujú rozlíšenie polárnej mapy. Zato parametre `min_range` a `max_range` určujú minimálnu a maximálnu analyzovanú vzdialenosť od radaru.

Tieto parametre majú veľký vplyv na celkovú rýchlosť aplikácie, keďže určujú koľko buniek bude potreba pre každú snímku vyhodnotiť.

Pre možnosti rôzneho testovania je možné pomocou `#define SUPERVISED` zvoliť, či sa má SVM trénovať bez učiteľa, alebo najprv predtrénovať na anotovaných dátach a až potom dotrénovávať bez učiteľa.

Vstup a výstup aplikácie

Vstupom do aplikácie je mračno bodov typu `sensor_msgs::PointCloud2`. Vstup má tému s názvom `cloud_pcd`.

Výstupom je mapa dátového typu `nav_msgs::OccupancyGrid`. Výstup má tému s názvom `map2d_out`.

Keďže vstupné dáta sú vytvorené z RAW dát, čiže neobsahujú informáciu o synchronizácii, tak spracovanie vstupného mračna bodov je nútené prebiehať bez `tf` synchronizácie.

Spracovanie vstupného mračna bodov po jednotlivých snímkoch zabezpečuje callback funkcia `GroundMap::process`. Funkcia spracováva vstup nezávisle (až na stav SVM) od predchádzajúcich snímkov.

Pôvodný algoritmus musel byť upravený tak, aby pracoval so vstupnou štruktúrou XYZIA namiesto XYZIR. Pre každý bod štruktúra XYZIA obsahuje súradnice x , y , z , následuje intenzita a anotácia. Narozdiel od štruktúry XYZIR, ktorá neobsahuje anotáciu, ale číslo prstenca. Keďže body z KITTI datasetov neobsahovali informáciu o čísle prstenca, tak túto informáciu bolo treba dopočítať zo samostatných súradníc bodov.

Index prstenca nie je zložité zistiť, keďže z dokumentácie k LiDAR-u je možné vyčítať počet indexov, maximálny a minimálny uhol pod ktorým sú lúče vysielané. Z toho sa dajú jednoducho určiť intervaly uhlov a ich odpovedajúci index. Výpočet uhlu τ vyslaného lúču je pre každý bod $\tau = \tan \frac{z}{\sqrt{x^2+y^2}}$.

Dlib

Ako implementáciu Pegasonovho algoritmu som zvolil dlib [14]. Dlib je open-source knižnica, zameraná ako na inžinierov tak aj na vedcov, ktorý chcú bohaté prostredie. Knižnica umožňuje jednoduchý vývoj algoritmov postavených na strojovom učení v programovacom jazyku C++. K splneniu tohto cieľu dlib obsahuje rozšíriteľnú lineárnu algebru s podporou BLAS¹. Implementuje algoritmi na prácu s Bayesianovými sieťami a metódmi pracujúcimi s jadrami (kernel-based methods) na rozpoznávanie, regresiu, zhukovanie a detekciu anomálii. Knižnica má kompletnú dokumentáciu a nástroje pre debugovanie.

Z knižnice Dlib som využil práve implementáciu SVM `pegasos`.

SVM s postupným učením

Na online SVM som použil implementáciu `dlib::svm_pegasos` z knižnice `dlib/svm.h`.

Experimentálne som zistil, že pri zvolených príznakoch mi najlepšie výsledky dáva SVM kernel: `dlib::radial_basis_kernel`. Pre tento kernel som zvolil parametre $\gamma = 0.005$ a $\lambda = 0.00001$. Pre zrýchlenie výpočtu som musel zvoliť nižší počet pomocných vektorov na 20. Základná nastavenie SVM v dlib pracuje s 40 pomocnými vektormi.

Normalizovanie vzorkov pomocou `dlib::vector_normalizer`. V prvom prechode bunkami hľadám ich okolie a ukladám príznaky do vektoru. Zozbierané príznaky uložené vo vektore normalizujem.

Pri učení vyvažujem vzorky podľa počtu naučených buniek cesty a buniek prekážky. Presnejšie ak je počet buniek naučených väčší ako naučených buniek prekážky $\frac{bin_{road}}{8} \geq bin_{obstacle}$. Tak je možné učiť.

¹Basic Linear Algebra Subprograms

Kapitola 7

Experimenty a výsledky

Vyhodnotenie úspešnosti rozšírenej implementácie je vykonané na anotovanom datasete. Čiže na datasete, ktorý obsahuje snímky mračna bodov a informáciu o tom, či daný bod je zem alebo prekážka. Pre porovnanie potenciálneho zlepšenia je zameraný aj pôvodný algoritmus. Algoritmus bol ale napísaný pre robota Toad.

7.1 Dataset vytvorený robotom Toad

Dataset je nahravaný pomocou robota Toad za pomoci pseudo-operačného systému ROS. Robot používa laserový radar Velodyne Lidar HDL-32E, čiže body v mračne bodov okrem súradnice obsahujú intenzitu a index prstenca. Nahrávky obsahujú asi 60 minút dát.

Pôvodná aplikácia bola napísaná a odladená pre dataset, ktorý bol vytvorený robotom Toad. Dataset vznikol v parku za areálom fakulty informatiky na Božetechovej ulici. Park sa nachádza na rovine. Cesty, či skôr chodníky sú zväčša nespevnené a ohraničené trávou.

Záznamy z datasetu prechádzajú po chodníkoch v tesnej blízkosti trávnatého povrchu. Rozdiel medzi chodníkom, ktorý je takmer rovný a členitým trávnatým povrchom, je výrazný.

Ako je vidieť na obrázkoch 7.1, tak výstupná mapa už neobsahuje neznáme bunky (bunky bez podfarbenia) a počas priebehu je vidieť ako algoritmus konverguje k jednoznačnému oddeleniu prekážky od cesty.

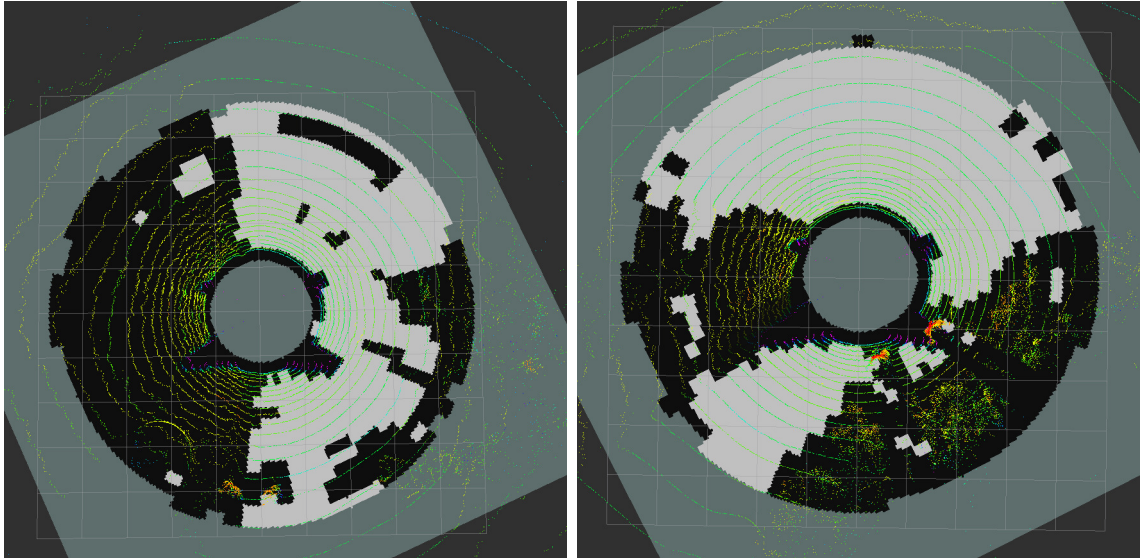
Na obrázku 7.1 vľavo je vidieť ako robot prechádza v tesnej blízkosti zatravnenej plochy. Na druhej strane chodníka je vidieť ako krík zasahuje do cesty. Pred robotom je vidieť pár buniek, ktoré nie sú neobsahujú body, ale sú označené ako prekážka. Tento jav je spôsobený zlým rozhodovaním pri detekcii, tento jav sa mi podarilo odstrániť až v neskorších fázach vývoja.

Na obrázku 7.1 vpravo je vidieť ako robot prišiel z úzkeho chodníka. Na ľavej strane od robota je vidieť úzky pás zatravnenej plochy. Za robotom sa nachádzajú operátori a za nimi vysoké kríky.

Pri testovaní na datasete sa pekné výsledky dostavili asi po 40 sekundách.

Nevýhodou tohto datasetu je fakt, že nie je anotovaný. Čiže neexistuje tzv. ground truth pre body, ktoré sú cestou. Preto bolo treba zvoliť dataset, ktorý obsahuje rozdelenie do bodov.

Dataset, ktorý ponúka ako záznam z LiDAR-u, tak aj anotácie k mračnu bodov, sa volá KITTI.



Obr. 7.1: Ukážka výsledku detekcie cesty pomocou Online SVM na datasete vytvorenom robotom Toad.

7.2 KITTI dataset

KITTI je komplexný dataset, ktorý je používaný aj k benchmarkom [19]. Hlavným zameraním KITTI data setov je poskytnutie voľných dát na vývoj v odvetviach počítačového videnia ako stereo kamera, optický tok, vizuálnu odometriu, detekcia 3D objektov a 3D sledovanie objektov. Tieto datasey obsahujú pomerne širokú škálu záznamov z rôznych prostredí a to od mestských ulíc (obrázok 7.2) a križovanie, vidieckych ciest až k záznamom z diaľnice. Dataset je zaznamenaný z bežného auta pomocou štyroch kamier (dve nahrávajúce záznam v odtieňoch sivej, dve farebné) so záznamom vo vysokom rozlíšení, jeden laserový radar Velodyne Lidar vo verzí HDL-64E a inerciálny systém (GPS/IMU). Radar je osadený vo výške 1,73 m.



Obr. 7.2: Príklad dát z KITTI datasetu.

KITTI ponúka [12] všetky záznamy s týmito informáciami:

- dáta ako v RAW formáte (nesynchronizované a neopravené), tak aj ich synchronizo-

vanej a opravenej verzii.

- Kalibrácia medzi senzormi (kamera, kamera - GPS, kamera - Velodyne).
- Označené 3D objekty (autá, nakladné autá, električky, chodci, cyklisti).

7.3 Anotované datasety a ich chyby

Na webovej stránke KITTI v sekcii sémantika je možné nájsť aj anotáciu pre Velodyne dáta¹[19]. Celkovo je anotovaných 8 datasetov, kde sú objekty rozdelené do 11 kategórií: 1 – budovy, 2 – obloha, 3 – cesta, 4 – vegetácia, 5 – chodník, 6 – auto, 7 – chodec, 8 – cyklista, 9 – značky, 10 – plot a 0 – neanotované.

Nie všetky kategórie sú použité pre tie body, na ktoré nie je vidieť z pohľadu kamier. Pre testovanie úspešnosti detekcie cesty som použil body kategórie 3. Zvyšné kategórie boli označené ako prekážka.

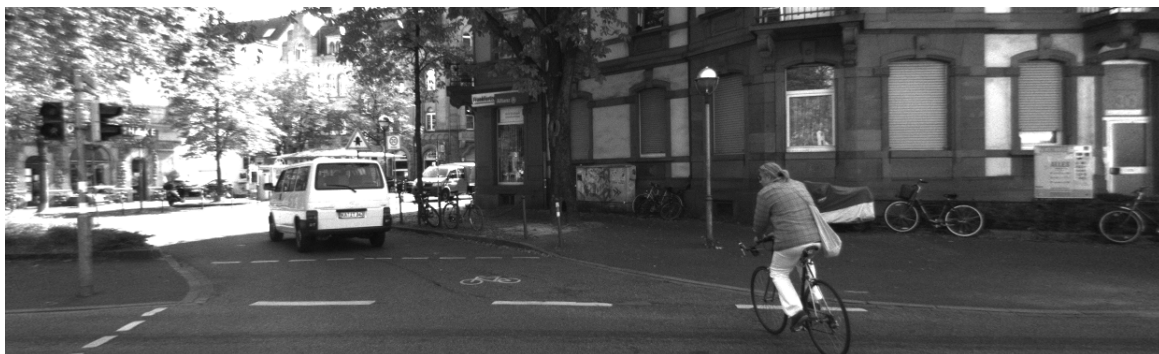
Všetky anotované datasety sú z urbanistického prostredia. Na záznamoch je vidieť prechod križovatkou, štvorprúdová cesta rozdelená plotom, prechod po ceste popri zaparkovaných autách. Všetky tieto snímky ponúkajú zaujímavé scény pre vyhodnotenie.

K vyhodnoteniu som použil snímky zo štyroch datasetov. Dva sú z mestského prostredia, a po jednom z cesty a obytného prostredia.

Metské prostredie je zastúpené dvoma datasetmi:

- 2011_09_26_drive_0005 (obrázok 7.3)
- 2011_09_26_drive_0014 (obrázok 7.4)

Prvý spomínaný záznam má 16 sekúnd a obsahuje 16 anotovaných snímkov z celkového počtu 154 snímkov. Na zázname je vidieť deväť áut, tri dodávky, dvoch chodcov a jedného cyklistu. Na zázname je vidieť prechod metskými uličkami a troma križovatkami.



Obr. 7.3: Dataset 0005 prechádza metskými uličkami s nízkymi obrubníkmi.

Záznam 0014 má 32 sekúnd. Anotovaných je 31 snímkov z celkového počtu 318 snímkov. Na zázname je vidieť dvadsaťšesť áut, štyri dodávky, jedno nákladné vozidlo, päť chodcov, štyroch cyklistov a jednu električku. Záznam ukazuje širšie a rovné cesty.

Obytné prostredie (obrázok 7.5) je zachytené na anotovanom datasete 2011_09_26_drive_0036. Dĺžka tohto záznamu je 80 sekúnd. Z celkového počtu 803 snímkov je anotovaných 41. Na

¹http://www.cvlibs.net/datasets/kitti/eval_semantics.php



Obr. 7.4: Dataset 0014 je nahraný na ceste lemujucej električkovú trať. Druhá strana vozovky je buďto ohraničená chodníkom alebo pripájaním ciest.

zázname je vidieť osemdesiat áut, sedem dodávok, jedno nákladné vozidlo a jedného chodcu. Na zázname je vidieť prechod z obytného prostredia s veľkým množstvom zaparkovaných áut až k ceste lemujucej železničnú trať.



Obr. 7.5: V prvej polovici datasetu 0036 prechádza vozidlo úzkymi uličkami malého mestečka lemovanými zaparkovanými autami. Druhá polovica je rovná cesta ohraničená vysokými obrubníkmi.

Anotovaný cestný záznam 2011_09_26_drive_0015 má 30 sekúnd, 297 radarových snímkov a 30 anotovaných. Na zázname je vidieť tridsaťtri áut, jednu dodávku, jedno nákladné vozidlo a jedného cyklistu. Záznam obsahuje rovnú cestu s jedným nespevneným okrajom vozovky (obrázok 7.6).

Ďalšie anotované datasety sú z prostredia nevhodného pre rozpoznávanie len pomocou laserového radaru. Sú vyhotovené na okraji peších zón, kde sú síce chodníky, ale nie je ich možné rozlíšiť len na základe výšky.

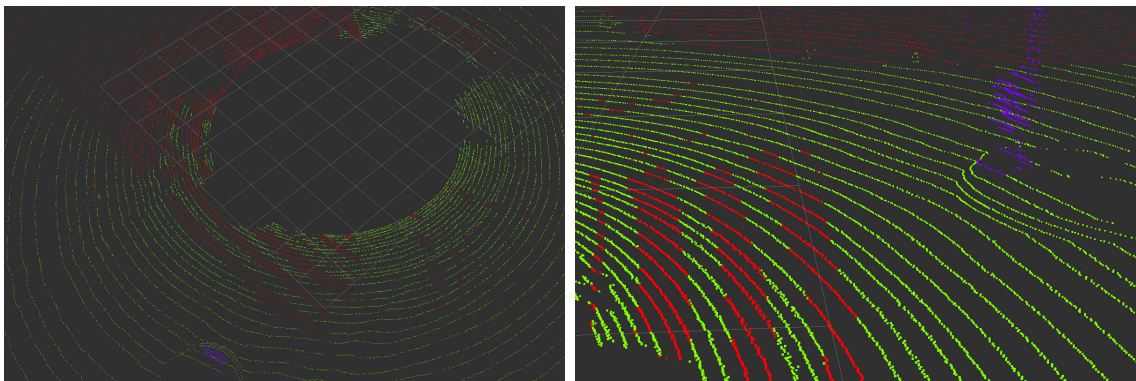
Pri bližšom preskúmaní anotácií som zistil, že nie všetky snímky sú úplne dotiahnuté do dokonalosti. Hlavnými nedostatkami sú:

- Pásky zle anotovaných bodov v niektorých po sebe idúcich snímkach (obrázok 7.7 vľavo).
- Chybná anotácia menších prekážok napr. ostrovček okolo značky (obrázok 7.7 vpravo).
- Autá do určitej výšky boli označené ako cesta (obrázok 7.8).



Obr. 7.6: Dataset 0015 je najjednoduchší, snímky sú nahrané na rovnej ceste kde vozovka je z jednej strany ohraničená trávou a z druhej strany vysokým obrubníkom.

- Chodník, pás zelene pri zvodidlách je označený ako cesta (obrázok 7.9).
- Na okrajoch zatienených miest autom, alebo senzormy sú dáta veľmi nepresné.

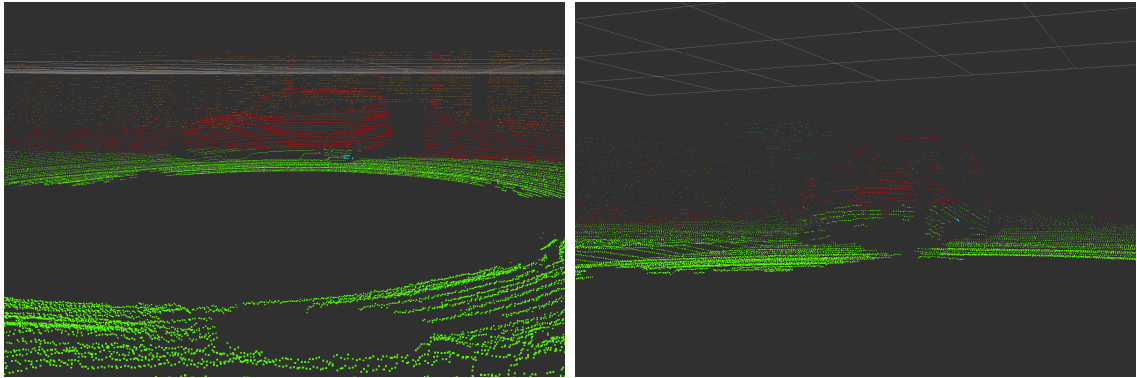


Obr. 7.7: Chyby v anotovanom datasete – zle anotované body.

Obrázok vľavo ukazuje zle anotované časti pred prednou maskou automobilu a po pravej strane. Obrázok vpravo sa pozerá na rovnakú scénu, ale z iného uhla. Ďalšou zaujímavosťou okrem zle anotovaných pásov je ostrovček so značkou. Značka je síce anotovaná správne, ale ostrovček už nie.

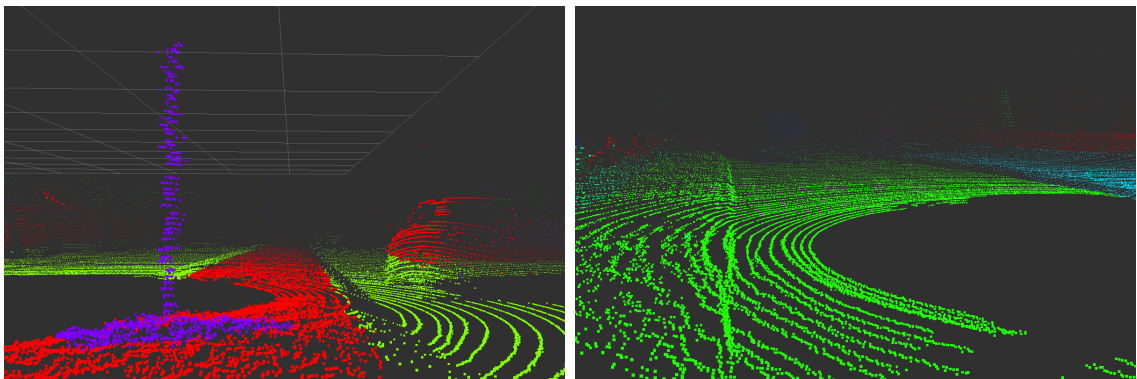
Tieto chyby sťažujú učenie s učiteľom, keďže do SVM vstupujú aj zle anotované vzorky. Čiže aj vzorky, ktoré obsahujú obrubník, ale sú anotované ako cesta, alebo naklonený členitý povrch.

Vyhodnotenie nie je o nič lepšie, keďže vzorky, ktoré nie sú cestou sú označené ako cesta. Príkladom môže byť obrubník oddeľujúci chodník od cesty. Ak je tento obrubník označený ako cesta, tak algoritmus vyhodnotí obrubník ako prekážku, čo je správne, ale anotácia hovorí, že toto je stále cesta. Druhým častým problémom je, že chodník je niekedy rovnejší ako cesta. Tento problém by sa dal vyriešiť prísnejším prahom na výšku od zeme. Keďže ale pracujeme s reálnymi dátami, tak problém by nastal pri stúpaní väčšom ako daný prah.



Obr. 7.8: Chyby v anotovanom datasete.

Obrázky ukazujú zle anotované auto. Horná časť je braná ako prekážka, ale spodná časť už je opäť cesta.



Obr. 7.9: Chyby v anotovanom datasete – chodník označený ako cesta.

Ľavý obrázok ukazuje síce dobre anotovaný vyvýšený pás medzi pruhmi, ale nábežná hrana je ešte cesta. Obrázok vpravo ukazuje obrubník a časť chodníka, ktoré sú zle anotované.

Príprava dát a programu

Pri prechode zo záznamov z Toad-a 3.1 som musel vyriešiť niekoľko väčších či menších prekážok. Prvou bolo pripravenie datasetu z RAW dát a spojiť ich s anotovanými dátami. RAW dáta boli uložené v binárnom súbore. Anotácie zase v matlabovom súbore.

Velodyne dáta z RAW datasetov obsahovali tri súradnice bodu a jeho intenzitu/odrazivosť. Anotované dáta v tomto prípade predstavujú jednu novú celočíselnú hodnotu. Táto hodnota určuje do ktorej z desiatich kategórií bod patrí. Nie je ale anotovaný každá snímka v zázname, ale každá desiata.

Pre uloženie a ďalšie spracovanie som použil textový formát PCD². Spojenie a uloženie je naimplementované v jazyku GNU Octave. Z tohto formátu už nebol problém pomocou ROS balíčku *pcl_ros* vytvoriť BAG³.

Korekcia výstupného obrázku o tieň automobilu a inými senzormi (kamery, GPS systém, samotný automobil), pre presnejšie meranie bola vytvorená mapa tieňov.

Pre testovanie bolo potrebné previesť výslednú mapu na obrázok. Na vytvorenie obrázku som použil knižnicu OpenCV. Na získanie obrázkov z anotovaných dát som upravil pôvodný program tak, aby načítal mračno bodov z bag súboru, vytvoril mapu podľa anotácií a túto mapu následne uložil ako obrázok.

7.4 Testovanie

V tejto časti sú priblížené vyhodnocované mračná bodov – snímky, ukazovatele, podľa ktorých sú vyhodnotené jednotlivé merania aj s výsledkami.

Kvôli chybám v anotovanom datasete nie je možné overiť celú dĺžku záznamu. Z tohto dôvodu je vyhodnotených len pár zaujímavých snímok. Tieto snímky sú podrobne popísané v ďalšej podkapitole.

Pri vyhodnovení jednotlivých snímok sa nemenili vstupné prahy algoritmu, čiže všetky snímky z merania majú rovnaké vstupné prahy.

Pri vyhodnovení úspešnosti sú mapy zeme porovnávané pixel po pixeli. Porovnávanie spočíva v odčítaní intenzity testovaného pixelu od intenzity anotovaného pixelu. Keďže pri vytváraní boli použité známe intenzity, tak je možné vytvoriť kategórie (true positive, false positive, atď.) na základe numerickej hodnoty rozdielu. Potom vyhodnotenie spočíva v zaradení skúmaného pixelu do kategórie na základe hodnoty výslednej intezity. Tiene auta a kruh v strede nie je zarátaný do úspešnosti detekcie. Toto porovnávanie je implementované ako jednoduchý skript napísany v programovacom jazyku Python a za pomoci knižnice OpenCV.

7.5 Výber zaujímavých snímok

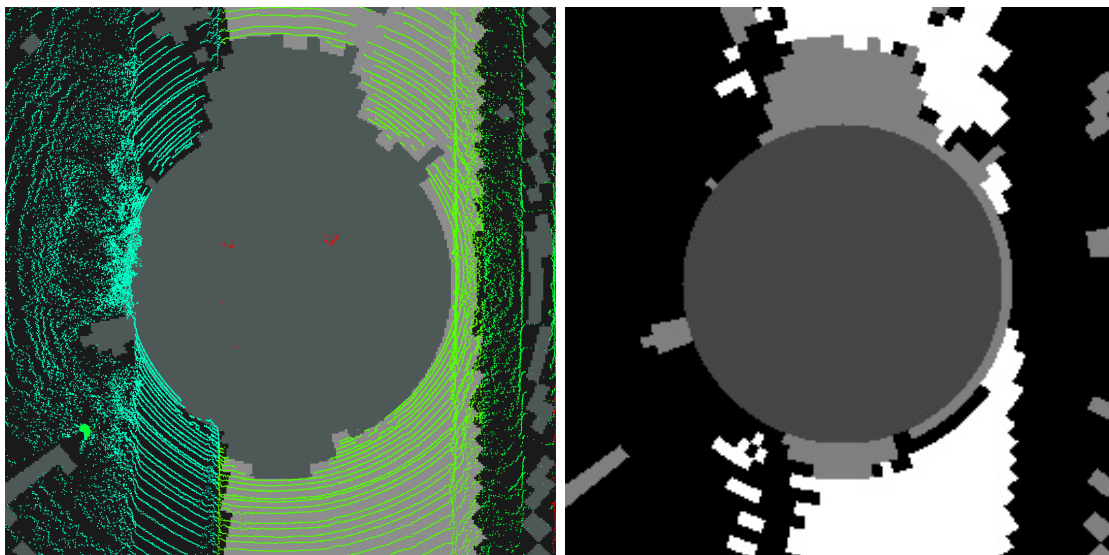
Keďže datasety obsahovali nemalé množstvo chýb v anotáciach, tak som sa rozhodol nevyhodnotiť len celkovú úspešnosť v datasete, ale aj ukázať detekciu na vybraných snímkoch z rôznych datasetoch, ktoré sú dobre anotované (snímka 7.16 má menšiu chybu v anotácii) a niečím zaujímavé.

Hlavným dôvodom pre výber snímok sú práve chyby v datasete. Dobrým príkladom je obrázok 7.10. Na snímke je vidieť takmer totožný obrázok. Ľavý je mračno bodov, ktoré je

²Point Cloud Data http://pointclouds.org/documentation/tutorials/pcd_file_format.php

³<http://wiki.ros.org/Bags/Format>

podfarbené anotovanou mapou zeme. Pravý obrázok je výsledok SVM s postupným učením. Úspešnosť detekcie cesty (biela farba) je len 55%. Najväčšou chybou (okrem buniek v tesnej blízkosti automobilu) je zle anotovaná časť chodníka. Už takáto malá chyba dokáže znížiť úspešnosť k skoro 50%.



Obr. 7.10: Porovnanie chyby anotácie snímku a výsledku online SVM. Úspešnosť detekcie cesty (biele bunky) je 55%.

Na obrázkoch (napr. 7.11 vľavo) je vidieť pohľad zvrchu (*topdown ortho*) na analyzovaný záber, ktorý je podfarbený anotovanou mapou. V strede obrázku je laserový senzor LiDAR. Svetlo-zelené body sú cesta. Šedý kruh v strede ukazuje už neanalyzovanú časť pod minimálnou vzdialenosťou. Šedé pásy mimo stredový kruh sú tieň prednej kapoty (dole), kufru (hore) a iné senzory na streche auta. Všetky ostatné farby sú prekážka.

KITTI datasey obsahujú ako záznamy z laserového radaru LiDAR, tak aj záznam z predných kamier. Na obrázkoch (napr. 7.11 vpravo) je vidieť práve tieto fotky z prednej kamery. Každá z fotiek sa pozerá na jednu z vyhodnocovacích scén, ktoré môžeme vidieť z pohľadu radaru na obrázkoch.

Prvá snímka (obrázok 7.11) ukazuje dvojprúdovú cestu. Na jednom okraji tráva, na druhom obrubník a tesne za ním múrik. Zaujímavé na tejto snímke je, že trávnik je približne v rovnakej výške ako vozovka.

Na druhej snímke (obrázok 7.12) je vidieť priechod vozidla cez svetelnú križovatku s cyklistickým chodníkom. Aj keď obe strany priechodu sú v rovnakej výške, tak ten bližšie k železničnej trati je označený ako prekážka, pričom ten na druhej strane zase ako voľná cesta. Na tesnom okraji analyzovaného mračna bodov je vidieť protiídúce auto. Zaujímave na tejto snímke je kolmá hrana vozidla.

Tretia snímka (obrázok 7.13) ukazuje prejazd tenkou uličkou. Z jednej strany ohraničený dvoma stojacimi autami, zo strany druhej zase chodník s nízkym obrubníkom a stĺpiky. Strana so stĺpikmi je križovatka bočnej cesty.

Štvrtý snímok (obrázok 7.14) ukazuje rovnú trojprúdovú cestu ohraničenú vysokými obrubníkmi.

Na piatom snímku (obrázok 7.15) je vidieť opäť križovatku. Zaujímavá je veľmi nízkym obrubníkom v tesnej blízkosti auta a uhlom, pod ktorým je druhá strana vozovky ohraničená



Obr. 7.11: Testovacia snímka 1. Obrázok naľavo ukazuje scénu, na ktorej sa nachádza výrazne ohraničená dvojprúdová cesta. Obrázok vpravo ukazuje tú istú scénu, ale ako ortogonálny pohľad na mračno bodov.



Obr. 7.12: Testovacia snímka 2. Obrázok naľavo ukazuje scénu, na ktorej je prechod vozidla cez krížovku. Ohraničenie na ľavej strane je veľmi nízke. Obrázok vpravo ukazuje tú istú scénu, ale ako ortogonálny pohľad na mračno bodov.



Obr. 7.13: Testovacia snímka 3. Na ľavom obrázku je vidieť scénu, ktorá je z jednej strany ohraničená dvoma zaparkovanými autami a na druhej strane nízkym obrubníkom. Obrázok vpravo ukazuje tú istú scénu ale z ortogonálneho pohľadu na mračno bodov.

vysokým obrubníkom, na anotovnom snímku je ale vidieť, že vysoký obrubník nie je po celej dĺžke správne označený.

Šiesty snímok (obrázok 7.16) je dvojprúdová cesta ohraničená na jednej strane obrubníkom a chodníkom. Na druhej strane zle anotovaným ostrovčekom oddelujúcim vozovku od železničnej trate.



Obr. 7.14: Testovacia snímka 4. Obrázok naľavo ukazuje skúmanú scénu, kde sa vozidlo nachádza trojprúdovej ceste, ktorá je ohraničená vysokými obrubníkmi. Na obrázku v pravej je ortogonálny pohľad na mračno bodov tej istej scény.



Obr. 7.15: Testovacia snímka 5. Na ľavom obrázku je vidieť scénu, kde vozidlo prechádza križovatku. Križovatka je z jednej strany ohraničená pomerne nízkym obrubníkom a z druhej strany výrazným. Obrázok vpravo ukazuje tú istú scénu, ale ako ortogonálny pohľad na mračno bodov.



Obr. 7.16: Testovacia snímka 6. Scéna na ľavom obrázku ukazuje rovnú cestu po oboch stranách ohraničenú výraznými obrubníkmi. Obrázok vpravo ukazuje tú istú scénu, ale ako ortogonálny pohľad na mračno bodov. Na tejto snímke je vidieť chybnú anotáciu ľavého obrubníka.

7.6 Experiment 1 – porovnanie úspešnosti základného riešenia a SVM s postupným učením bez predtrénovania

Ako prvý experiment som zvolil vyhodnotenie, či použitie SVM s postupným učením zlepšilo detekciu cesty oproti pôvodnému riešeniu, ktoré detekuje cestu pomocou prahov. Pre vyhodnotenie som použil vyššie spomínané snímky (kapitola 7.5).

Metodika merania

Pri vyhodnocovaní pôvodného riešenia *Ground map* som najprv experimentálne našiel parametre (tabuľka 7.1), ktoré ukazovali najlepšie hodnoty.

Maximálna nepravidelnosť cesty	0.04 m
Maximálny rozdiel výšky	0.025 m
Pravdepodobnosť cesty	0.9
Pravdepodobnosť prekážky	0.5

Tabuľka 7.1: Parametre aplikácie pre vyhodnotenie pôvodnej implementácie *Ground map*

Pre vyhodnotenie SVM s postupným učením som najprv redukoval SVM na tréningových snímkoch bez použitia anotácií a potom nechal vyhodnotiť na testovacom datasete. Keďže anotovaných datasetov nie je veľa, tak tréningová množina s testovacou množinou nie je disjunktná. Čiže jeden testovací dataset môže byť použitý ako tréningový pre iný dataset.

Parametre pre SVM som zistil na základe experimentovania a vyhodnocovania na rôznych datasetoch a sú zapísané v tabuľke 7.4.

Maximálna nepravidelnosť cesty	0.04 m
Maximálny rozdiel výšky	0.025 m
Pravdepodobnosť cesty	0.95
Pravdepodobnosť prekážky	0.2

Tabuľka 7.2: Parametre aplikácie pre vyhodnotenie implementácie *Online SVM*

Zhodnotenie

Numerické výsledky merania sú v tabuľke 7.4. Riadky tabuľky sú jednotlivé snímky (obrázky 7.11 až 7.16), stĺpec ukazovateľ špecifikuje jednotlivé metriky z kapitoly 4.4. Stĺpec *Ground map* ukazuje výsledky pre pôvodné riešenie a stĺpec *Online SVM* ukazuje výsledky pre riešenie s SVM s postupným učením.

Grafické výsledky merania sú v tabuľke 7.5. Tak ako v tabuľke s numerickými výsledkami, tak aj tu sú riadky vybrané snímky. Stĺpec *Anotácia* ukazuje anotované dáta, čiže referenčný obrázok, voči ktorému boli výsledné obrázky porovnávané. Ďalšie stĺpce už obsahujú výsledky, ako pôvodného riešenia *ground map* a implementácie s SVM.

Na všetkých výsledných obrázkoch je vidieť, že body v najbližšom okolí auta sú veľmi často brané ako prekážka. Ako je vidieť aj na obrázkoch 7.4, tieto body sú nepravidelné ako v smere prstencov, tak aj vo výške nad zemou. Tieto chyby sa prejavujú aj kvôli príznakom, ktoré sú vo veľkej väčšine odvodené od vzdialenosti od zeme. Ďalšou častou chybou je odhad obrubníka (obrázok 7.16) ako cesta. Toto je spôsobené tým, že častokrát je chodník rovnejší ako vozovka a obrubník nie je výrazne vzdialený od zeme.

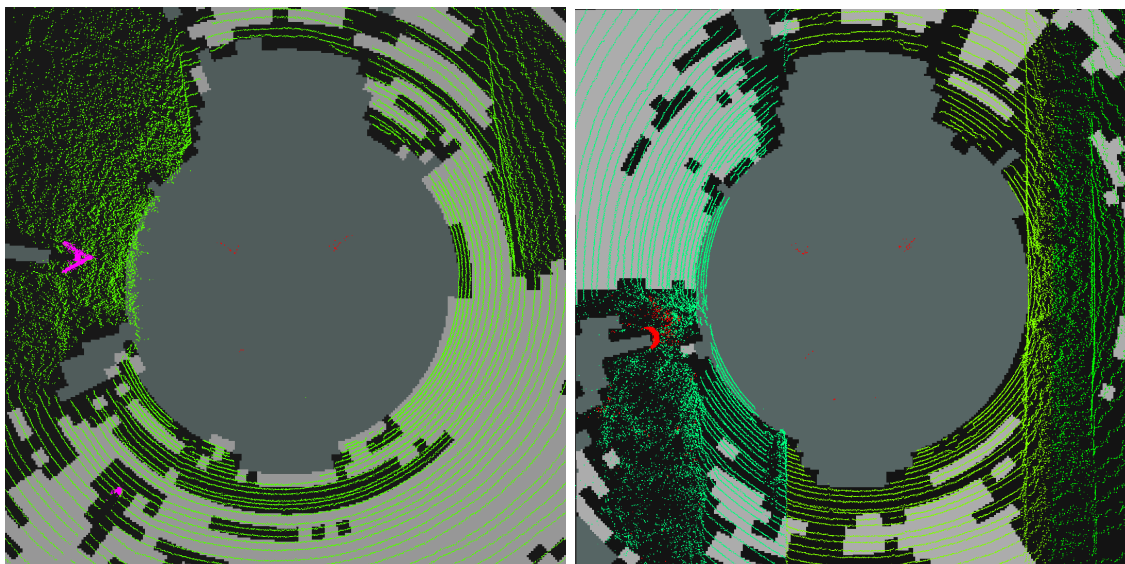
Pre úplnosť je zameraná aj priemerná úspešnosť ukazovateľa F-measure vo všetkých meraných datasetoch (tabuľka 7.3).

Príklad snímku s najmenšou úspešnosťou a to kvôli slabším anotáciám v datasete a zlou detekciou pre slabé stránky implementácie je vidieť na obrázku 7.17.

Na obrázku 7.17 vľavo je vozidlo na križovatke. Po pravej i ľavej strane má vozidlo trávnatý povrch a značku. Problém ale je, že anotácia hovorí, že celé okolie až na značky je cesta. Úspešnosť detekcie prekážky je pre túto snímku len 0.1049.

Tabuľka 7.3: Priemerná úspešnosť Ground Map a Online SVM v datasetoch.

Dataset	Ground Map	Online SVM
0015	0.6337	0.7230
0005	0.3833	0.4742
0014	0.4771	0.5492
0036	0.4095	0.4834



Obr. 7.17: Snímky s nízkou úspešnosťou detekcie

Obrázok vľavo ukazuje zle anotovaný snímok – takmer celá cesta je označená ako cesta. Obrázok vpravo je príklad snímky z datasetu 0036, ktorá bola zle detekovaná *Online SVM* algoritmom.

Na obrázku 7.17 vpravo je vozidlo prechádzajúce po ceste ohraničenej chodníkom na strane pravej a zeleňou na strane ľavej. Detekcia zelene ako prekážky je úspešná. Detekcia chodníka je odhadnutá nesprávne. Chodník je natoľko rovný a nízky, že je detekovaný ako cesta. Čo sa nedá ale povedať o ceste. Tá je vyhodnotená ako prekážka a to z dôvodu nerovností na vozovke. Úspešnosť detekcie cesty je pre túto snímku len 0.1869.

7.7 Experiment 2 – SVM s postupným učením s predtrénovaním na anotovaných datasetoch

V druhom experimente je SVM s postupným učením najprv predtrénované na anotovaných dátach. Pre porovnanie boli vybrané dve sady parametrov. Jeden z nich je rovnaký ako je použitý pre Online SVM v kapitole 7.6.

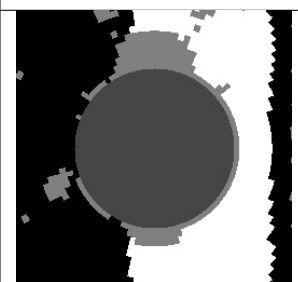
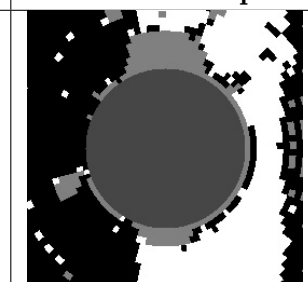
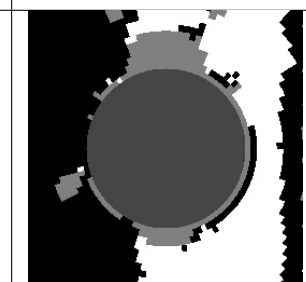
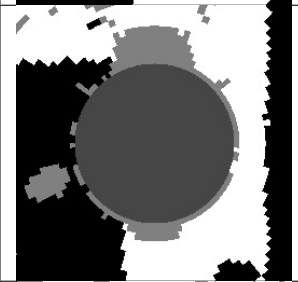
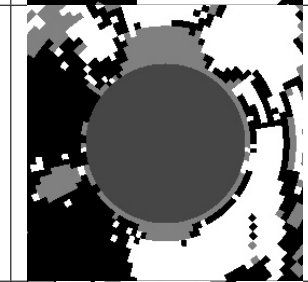
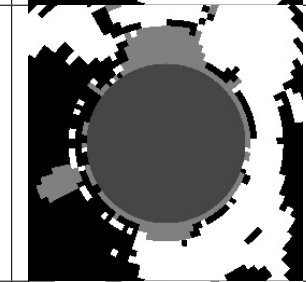
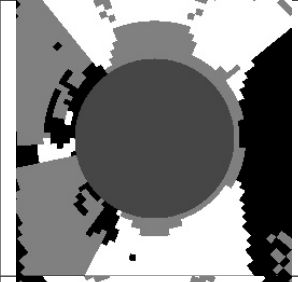
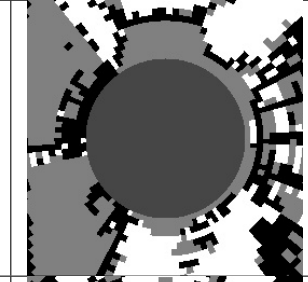
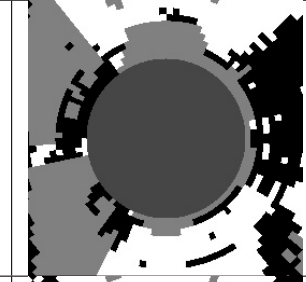
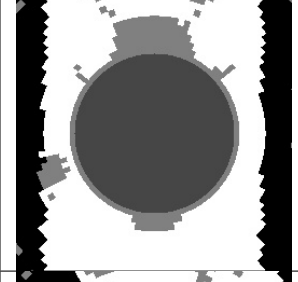
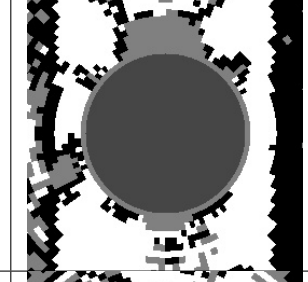
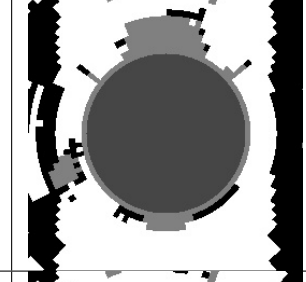
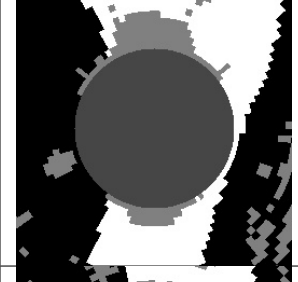
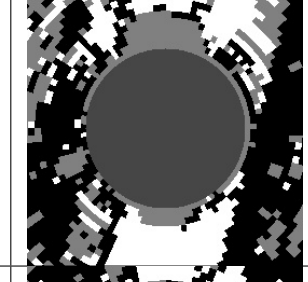
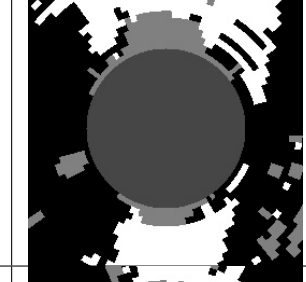
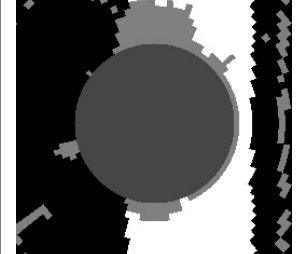
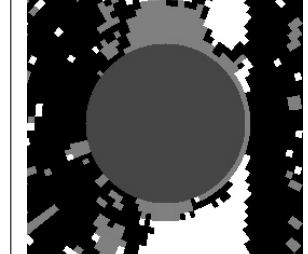
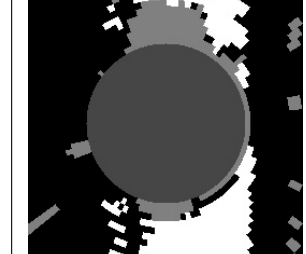
Metodika merania

Pri vyhodnocovaní je ukázaný vplyv vstupných parametrov, čiže výber extrémnych buniek, na úspešnosť detekcie cesty.

Tabuľka 7.4: Porovnanie výsledkov Ground map a Online SVM

Snímka	Ukazovateľ	Ground map	Online SVM
1	Precision	0.7561	0.8728
	Recall	0.7692	0.8328
	True negative rate	0.8166	0.9120
	Accuracy	0.7965	0.8787
	F-measure	0.7626	0.8524
2	Precision	0.6124	0.6934
	Recall	0.6148	0.7521
	True negative rate	0.6283	0.6845
	Accuracy	0.6217	0.7175
	F-measure	0.6136	0.7216
3	Precision	0.5638	0.7262
	Recall	0.5197	0.6623
	True negative rate	0.3966	0.6533
	Accuracy	0.4705	0.6585
	F-measure	0.5409	0.6928
4	Precision	0.7975	0.8837
	Recall	0.6771	0.8586
	True negative rate	0.6359	0.7744
	Accuracy	0.6639	0.8305
	F-measure	0.7324	0.8710
5	Precision	0.3807	0.6153
	Recall	0.4033	0.5152
	True negative rate	0.5488	0.7974
	Accuracy	0.4896	0.6885
	F-measure	0.3917	0.5608
6	Precision	0.5327	0.7362
	Recall	0.3937	0.4451
	True negative rate	0.7753	0.9025
	Accuracy	0.6250	0.7290
	F-measure	0.4528	0.5548

Tabuľka 7.5: Porovnanie anotovanej mapy s výsledkami Ground map a Online SVM

Snímka	Anotácia	Ground map	Online SVM
1			
2			
3			
4			
5			
6			

Trénovanie SVM s postupným učením prebiehalo najprv na anotovaných datasetoch. Pre trénovanie boli vždy použité dva rôzne tréningové datasety v jednom behu. Následne bol spustený a vyhodnotený testovací dataset.

Ako prvé zvolené parametre (tabuľka 7.6) som použil tie isté ako v prvom experimente (kapitola 7.6). Rozhodol som sa tak preto, aby bolo možné porovnať hodnoty, ktoré vznikli bez tréningu na anotovaných dátach a hodnoty natrénované s učiteľom.

Maximálna nepravidelnosť cesty	0.04 m
Maximálny rozdiel výšky	0.025 m
Pravdepodobnosť cesty	0.95
Pravdepodobnosť prekážky	0.2

Tabuľka 7.6: Parametre aplikácie pre vyhodnotenie implementácie *Param1*

Pre druhé meranie som parametre zvolil tak, aby pri tréningu boli použité väčšie množstvo buniek reprezentujúcich prekážku. Parametre sú zapísané v tabuľke 7.7.

Maximálna nepravidelnosť cesty	0.04 m
Maximálny rozdiel výšky	0.025 m
Pravdepodobnosť cesty	0.95
Pravdepodobnosť prekážky	0.5

Tabuľka 7.7: Parametre aplikácie pre vyhodnotenie implementácie *Param2*

Trénovanie online SVM v oboch prípadoch prebiehalo totožne, čiže trénovanie bolo vykonané na dvoch anotovaných datasetoch. Testovanie prebehlo s online učením SVM.

Zhodnotenie

Numerické výsledky merania sú v tabuľke 7.9. Riadky tabuľky sú jednotlivé snímky (obrázky 7.11 až 7.16), stĺpec ukazovateľ špecifikuje jednotlivé metriky z kapitoly 4.4. Stĺpec *Param1* ukazuje výsledky pre rovnaké parametre ako boli použité v predošlom experimente a stĺpec *Param2* ukazuje výsledky kde pre dotrénovanie bolo použité aj tie bunky prekážok, ktoré nie sú extrémne.

Grafické výsledky merania sú v tabuľke 7.10. Tak ako v tabuľke s numerickými výsledkami, tak aj tu sú riadky vybrané snímky. Stĺpec *Anotácia* ukazuje anotované dáta, čiže referenčný obrázok, voči ktorému boli výsledné obrázky porovnávané. Ďalšie stĺpce už obsahujú výsledky, ako pre parametre *Param1* a tak pre parametre *Param2*.

Pre tento experiment asi najzaujímavejším výsledkom je porovnanie celkovej úspešnosti na celých datasetoch. Tieto výsledky sú uložené v tabuľke 7.8.

Tabuľka 7.8: Priemerná úspešnosť *Param1* a *Param2* s Online SVM natrénovaným na anotovaných dátach

Dataset	Param1	Param2
0015	0.7240	0.6936
0005	0.4711	0.4583
0014	0.5601	0.5555
0036	0.5081	0.4945

Výsledkom tohto experimentu je, že úspešnosť detekcie cesty nie je veľmi závislá na parametre *pravdepodobnosť prekážky*. Priemerné hodnoty z tabuľky 7.3 sa maximálne líšia o 0.0304.

V predošlom experimente v *Online SVM* boli použité rovnaké parametre ako parametre *Param1*. Rozdiel týchto dvoch meraní je v spôsobe ako bol SVM natrénovaný. V predošlom experimente neboli pri online tréningu používané anotované dáta. V tomto experimente sa v tréningu práve na anotáciách. Pri porovnaní úspešnosti datasetov, čiže tabuľky 7.3 a 7.8, sú výsledky takmer totožné. Maximálna odchýlka je 0.0247.

7.8 Vyhodnotenie a návrh pokračovania práce

Dataset KITTI som vybral kvôli jeho rozšírenosti.

Počas testovania a hlavne na výslednej snímke (tabuľka 7.5 snímka 2) je vidieť, že v tesnom okolí automobilu a senzorov sa často vyskytujú bunky, ktoré sú označené ako prekážka. Tieto bunky sú označené svojim spôsobom správne, keďže dáta sú veľmi blízko tienov automobilu vo všetkých osiach nerovné.

Algoritmus nebol navrhnutý tak, aby detekoval cestu, čiže zem s nejakou toleranciou pre náklon. Algoritmus bol navrhnutý tak, aby hľadal rovné plochy, po ktorých by bolo možné ísť, aj keby to mal byť chodník. Čo často viedlo k tomu, že obrubník bol vyhodnotený ako prekážka, ale chodník už ako cesta. Ak by sa ale podarilo rozpoznať obrubník, bolo by možné nezaraďovať tieto bunky do učenia. Hľadanie obrubníkov je nejednoznačná úloha, keďže hlavným faktorom určovania prekážok je výška nad zemou. Prvým problémom je samotný náklon cesty, druhým a rovnako zložitým je nedokonalosť cesty. Riešenie tohto problému by mohla vyriešiť fúzia dát s iným sensorom – napr. kamerou.

Na základe výsledkov merania sa Pegasonov algoritmus s príznakmi odvodenými od výšky bodu od zeme nehodí na jednoznačné určenie cesty v urbanistickom prostredí.

Rozšírením práce by mohlo byť zrýchlenie spracovania dát pre SVM. Pri učení SVM bunku po bunke je potrebné nájsť a prístup k celému jej okoliu. Toto by sa dalo jednoducho zoptimalizovať. Optimalizácia by spočívala len v predávaní už načítaných a stále potrebných buniek medzi iteráciami, čím by sa dosiahol menší počet čítaní z pamäte.

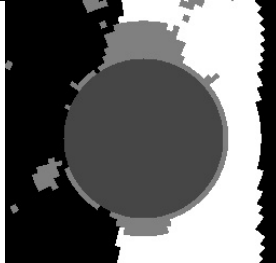

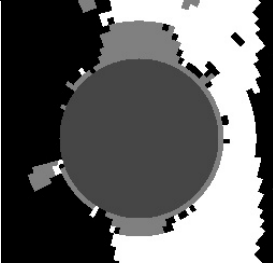
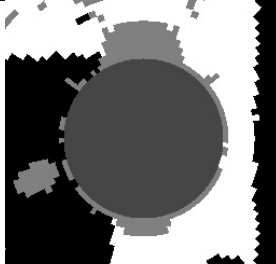

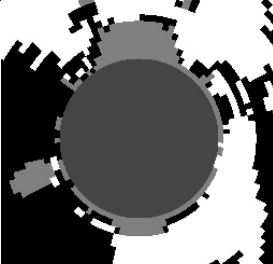
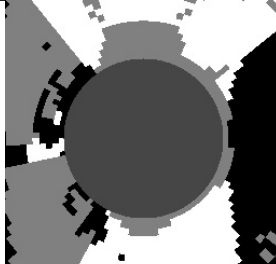
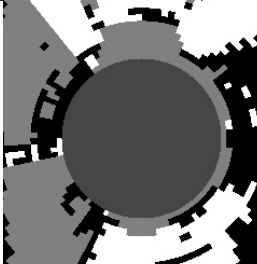
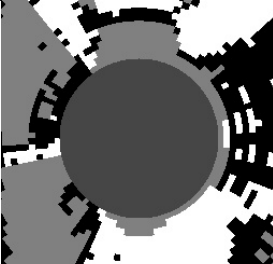
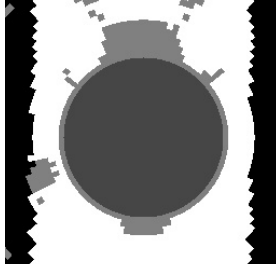
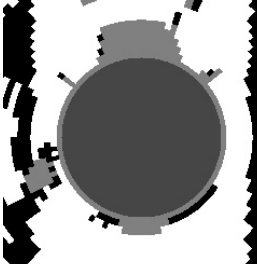
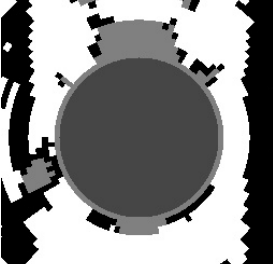
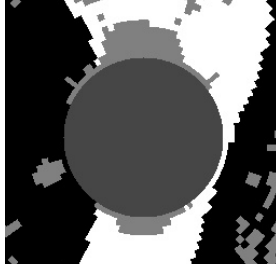
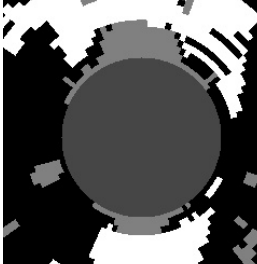
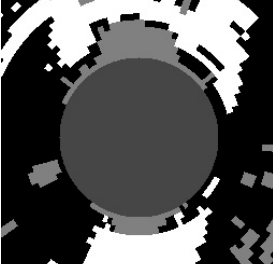
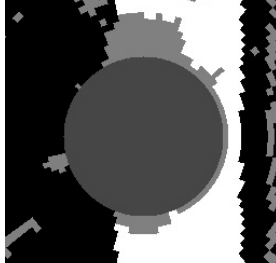
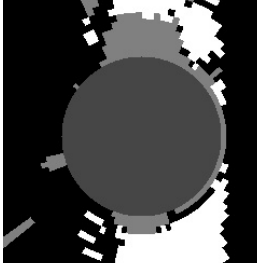
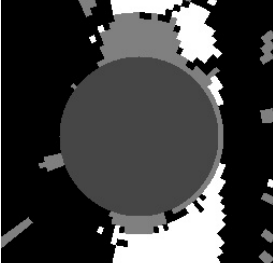
Iným spôsobom ako roziříť túto prácu, by mohlo byť automatické hľadanie prahov a parametrov pre SVM. Keďže v implementácii sú použité štyri prahy pre hľadanie extrémnych snímok a SVM, kde sú použité dva parametre, tak by za zváženie stálo použiť openMPI.

Ako bolo už spomínané vyššie, v urbanistickom prostredí by bolo vhodné použiť kameru ako ďalší zdroj informácie. Prínosom kamery by bol napríklad v detekcii obrubníkov.

Tabuľka 7.9: Porovnanie výsledkov online SVM učenia na anotovaných dátach s rôznymi parametrami

Snímka	Ukazovateľ	Param1	Param2
1	Precision	0.8619	0.8498
	Recall	0.9061	0.8477
	True negative rate	0.8962	0.8918
	Accuracy	0.9003	0.8733
	F-measure	0.8834	0.8487
2	Precision	0.7146	0.6952
	Recall	0.7875	0.7253
	True negative rate	0.7036	0.6965
	Accuracy	0.7443	0.7106
	F-measure	0.7493	0.7099
3	Precision	0.6818	0.7019
	Recall	0.5832	0.6607
	True negative rate	0.6200	0.5999
	Accuracy	0.5986	0.6357
	F-measure	0.6286	0.6807
4	Precision	0.8726	0.8578
	Recall	0.8734	0.7951
	True negative rate	0.7444	0.7348
	Accuracy	0.8305	0.7751
	F-measure	0.8730	0.8253
5	Precision	0.5608	0.5714
	Recall	0.4858	0.4991
	True negative rate	0.7578	0.7610
	Accuracy	0.6519	0.6589
	F-measure	0.5206	0.5328
6	Precision	0.7361	0.7994
	Recall	0.4144	0.4413
	True negative rate	0.9039	0.9365
	Accuracy	0.7292	0.7386
	F-measure	0.5518	0.5459

Tabuľka 7.10: Porovnanie anotovanej mapy s predučeným online SVM s rôznymi parametrami

Snímka	Anotácia	Param1	Param2
1			
2			
3			
4			
5			
6			

Kapitola 8

Záver

Cieľom tejto práce bolo získať povedomie o súčasných metódach ako vyhodnotiť okolie autonómneho vozidla za pomoci laserového radaru Velodyne LiDAR a kamery. Vybral som si jedno z existujúcich riešení a rozšíril som ho. Pôvodné riešenie používalo k rozhodnutiu či je cesta v okolí prahy. Tieto prahy som nahradil za strojové učenie SVM s postupným učením.

Porovnanie týchto prístupov som spravil nad datasetom KITTI, ktorý obsahuje aj anotované dáta z Velodyne LiDAR. Pre porovnanie som použil ukazovateľ F-measure.

Anotácie v datasete KITTI neboli dotiahnuté do úplnej dokonalosti, tak som vyhodnotil nielen celé datasety, ale aj zvolené snímky. Snažil som sa vybrať snímky, ktoré sú dobre anotované a ktoré su aj niečím zaujímavé.

Pôvodný algoritmus dosahuje na vybraných snímkoch priemernú presnosť odhadu cesty 58%. Moja implementácia 71%. Pri porovnávaní celých datasetov je to od 38% do 63% pre pôvodné riešenie a od 47% do 72% pre novú implementáciu. Hlavným nedostatkom pôvodného riešenia je veľký počet neodhadnutých buniek. Nevýhodou nového riešenia je rýchlosť.

Nový algoritmus bol navrhnutý tak, aby hľadal rovné plochy, po ktorých by bolo možné ísť, aj keby to mal byť chodník. Čo často viedlo k tomu, že obrubník bol vyhodnotený ako prekážka, ale chodník už ako cesta. Pokračovaním tejto práce by mohlo byť fúzia mračna bodov z Velodyne LiDAR s kamerou. Kamera by mohla hľadať hrany obrubníkov a tým by zamedzila odhadu rovných plôch za prekážkou. Ďalším vylepšením by mohla byť optimalizácia opätovného prechodu buniek.

Literatúra

- [1] but_velodyne_proc. [online], 2016.
URL http://wiki.ros.org/but_velodyne_proc
- [2] HDL-32E. [online], 2016.
URL http://velodynelidar.com/docs/datasheet/97-0038_Rev%20E_%20HDL-32E_Datasheet_Web.pdf
- [3] Point Cloud Library. [online], 2016.
URL <http://pointclouds.org/>
- [4] ROS. [online], 2016.
URL <http://www.ros.org/about-ros/>
- [5] ROS. [online], 2016.
URL <http://wiki.ros.org/ROS/Introduction>
- [6] ROS. [online], 2016.
URL <http://wiki.ros.org/Topics>
- [7] Výzkumná skupina robotiky Robo@FIT. [online], 2016.
URL <http://www.fit.vutbr.cz/research/groups/robo/eqp.php.cs>
- [8] Wikipedia: Precision and recall. [online], 2016.
URL https://en.wikipedia.org/wiki/Precision_and_recall
- [9] Wikipedia: Support vector machine. [online], 2016.
URL https://en.wikipedia.org/wiki/Support_vector_machine
- [10] Andrew W. Moore: Support Vector Machines. 2001.
URL <http://www.autonlab.org/tutorials/svm.html>
- [11] Christopher Rasmussen: RoadCompass: following rural roads with vision + lidar using vanishing point tracking. *Auton Robot*, ročník 25, 2008: s. 205–229.
- [12] Geiger, A.; Lenz, P.; Stiller, C.; aj.: Vision meets Robotics: The KITTI Dataset. *International Journal of Robotics Research (IJRR)*, 2013.
- [13] J. Han, D. Kim, M. Lee and M. Sunwoo: Road boundary detection and tracking for structured and unstructured roads using a 2D Lidar sensor. *International Journal of Automotive Technology*, ročník 15, 2014: s. 611–623.
- [14] King, D. E.: Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research*, ročník 10, 2009: s. 1755–1758.

- [15] Martin Velas, Michal Spanel, Zdenek Materna, Adam Herout: Calibration of RGB Camera With Velodyne LiDAR. *WSCG 2014 Communication Papers Proceedings*, ročník 1, 2014: s. 135–144.
- [16] Michal Spanel: Road Detection from Velodyne Lidar Data. 2013.
- [17] Patric Y. Shinzato, Denis F. Wolf, Christoph Stiller: Road Terrain Detection: Avoiding Common Obstacle Detection Assumptions Using Sensor Fusion. *IEEE Intelligent Vehicles Symposium*, ročník 4, 2014: s. 687–692.
- [18] Pavel Laskov, Christian Gehl, Stefan Kruger, Klaus-Robert Muller: Incremental Support Vector Learning: Analysis, Implementation and Applications. *Journal of Machine Learning Research*, ročník 7, 2006: s. 1909–1936.
- [19] Richard Zhang, Stefan A. Candra, Kai Vetter, Avideh Zakhor: Sensor fusion for Semantic Segmentation of Urban Scenes. *ICRA*, 2015.
- [20] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, Andrew Cotter: Pegasos: primal estimated sub-gradient solver for SVM. *Math. Program., Ser. B*, ročník 25, 2010: s. 3–30.
- [21] Yuan Xia, Zhao Chun-Xia, Zhang Hao-Feng: Road Detection and Corner Extraction Using High Definition Lidar. *Information Technology Journal*, ročník 9, 2010: s. 1022–1030.