Bakalárska práca

České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra kybernetiky

Lokálna vonkajšia navigácia mobilného robota pomocou mapovania priechodnosti terénu

Michal Kasarda

Školiteľ: Ing. Jan Chudoba
Január 2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Kasarda** Jméno: **Michal** Osobní číslo: **492050**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra kybernetiky**

Studijní program: **Kybernetika a robotika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Lokální venkovní navigace mobilního robotu s mapováním průchodnosti terénu

Název bakalářské práce anglicky:

Local Outdoor Navigation of a Mobile Robot by Terrain Traversability Mapping

Pokyny pro vypracování:

Cílem práce je návrh a implementace metody pro navigaci mobilního robotu v nerovném terénu. Vstupem metody je cílová pozice, které chceme dosáhnout. Metoda má zmapovat okolní prostředí robotu, vyhodnotit průjezdnost prostředí, zvolit vhodnou cestu s ohledem na průjezdnost a generovat řídicí příkazy které robota provedou po naplánované cestě.

- Seznamte se s metodami pro mapování prostředí a senzory používanými pro tento účel. Vypracujte rešerši.
- Navrhněte metodu navigace do zadaného cílového bodu. Vhodné senzorické vybavení konzultujte s vedoucím práce.
- Implementujte navrženou metodu a otestujte její funkčnost v simulátoru a/nebo (podle reálných možností) na laboratorním robotu.
- Proveďte vyhodnocení testů, diskutujte možnosti a případné omezení vytvořené metody.

Seznam doporučené literatury:

[1] Shan, T., Wang, J., Englot, B. & Doherty, K.. (2018). Bayesian Generalized Kernel Inference for Terrain Traversability Mapping. Proceedings of The 2nd Conference on Robot Learning, in Proceedings of Machine Learning Research 87:829-838

[2] Gerald Cook. Mobile Robots: Navigation, Control and Remote Sensing. Wiley-IEEE Press, 2011

[3] P. E. Hart, N. J. Nilsson and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," in IEEE Transactions on Systems Science and Cybernetics, vol. 4, no. 2, pp. 100-107, July 1968

[4] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," in IEEE Robotics & Automation Magazine, vol. 13, no. 2, pp. 99-110, June 2006

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Jan Chudoba inteligentní a mobilní robotika CIIRC

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **14.09.2022** Termín odevzdání bakalářské práce: **10.01.2023**

Platnost zadání bakalářské práce: **19.02.2024**

Ing. Jan Chudoba podpis vedoucí(ho) práce	prof. Ing. Tomáš Svoboda, Ph.D. podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání	Podpis studenta

© ČVUT v Praze, Design: ČVUT v Praze, VIC

Poďakovanie

Chcel by som sa poďakovať Ing. Janovi Chudobovi za trzpezlivosť, rady a odborné vedenie tejto bakalárskej práce. Taktiež dakujem svojej rodine za všetkú poskytnútu pomoc.

Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval samostatne a že som uviedol všetky použité informačné zdroje, v súlade s pravidlami Metodického pokynu o dodržovaní etických princípov pri príprave záverečných prac.

V Prahe, 5. januára 2023

.....
Michal Kasarda

Abstrakt

Táto práca sa zaoberá autonómnou navigáciou robota v neznámom prostredí. V rámci práce je vypracovaný prehľad techník a senzorov používaných pri simultánnej lokalizácii a mapovaní (SLAM). Následne je predstavený algoritmus iterative closest point (ICP), na ktorom je založená metóda ICP SLAM. Pri mapovaní je potrebné vyhodnotiť priechodnosť prostredia s ohľadom na kinematické a dynamické vlastnosti robota. V práci sú predstavené metódy analýzy priechodnosti prostredia. Zároveň je navrhnutý spôsob analýzy priechodnosti, založený na geometrických vlastnostiach prostredia. Na navigáciu robota je vybraná ICP SLAM metóda a implementovaná analýza priechodnosti spolu s hľadaním cesty v mape a regulátorom na sledovanie nájdenej cesty. Implementácia je otestovaná v rámci simulátoru Gazebo a testov v reálnom mestskom prostredí.

Kľúčové slová: simultánna lokalizácia a mapovanie (SLAM), iterative closet point (ICP), analýza priechodnosti prostredia

Školiteľ: Ing. Jan Chudoba
CIIRC B-323,
Jugoslávských partyzánů 3,
16000 Praha 6

Abstract

This thesis deals with autonomous navigation of mobile robot in unknown enviroment. This works contains overview of methods and techniques used for simultaneous localization and mapping (SLAM). The iterative closest point (ICP) alghorithm is introduced, on which the ICP SLAM method is based. It is necesery to evaluate traversability of enviroment while mapping. This evaluation is based on kinematic and dynamic properties of robot used. This work contains overview of traversability analysis methods. A method for traversability evaluation is proposed, based on geometric features of enviroment. For navigation, ICP SLAM method is chosen and a method for traversability analysis, pathfinding and regulator for path following is implemented. The implementation is then tested in Gazebo simulator and in real city enviroment.

Keywords: simultaneous localisation and mapping (SLAM), iterative closet point (ICP), traversability analysis of enviroment

Title translation: Local Outdoor Navigation of a Mobile Robot by Terrain Traversability Mapping

Obsah

1 Úvod 1
1.1 Ciele práce 1
2 Simultánna lokalizácia a mapovanie 3
2.1 Formulácia a ²truktúra úlohy SLAM 3
2.2 Lokalizácia 4
2.3 Mapovanie 4
2.3.1 Mrieºka obsadenosti 5
2.3.2 2.5D mapa 5
2.3.3 Bodová mapa 6
2.4 SLAM metódy 6
2.4.1 EKF SLAM 6
2.4.2 SLAM s pouºitím £asticového ltra 7
2.4.3 Grafový SLAM 7
2.4.4 Scan-matching SLAM 7
3 Senzory 9
3.1 Odometria 9
3.1.1 Inerciálna meracia jednotka - IMU 9
3.2 Globálny druºicový polohový systém - GNSS 9
3.3 Sonar 10
3.4 Kamery 10
3.5 LIDAR 11
4 ICP SLAM 13
4.1 ICP algoritmus 13
4.2 Kniºnica Libpointmatcher 15
4.3 ICP SLAM 16
4.4 Filtre a funkcie modulov ICP . . . 19
4.4.1 DataPointsFilter 19
4.4.2 Matcher 21
4.4.3 Outlier ltre 22
4.4.4 ErrorMinimizer 22
4.4.5 TransformationChecker 23
4.5 Filtrácia dynamických prekáºok v mape 23
5 Mapovanie priechodnosti terénu 25
5.1 Proprioceptívná analýza priechodnosti 25
5.2 Extroceptívna analýza priechodnosti 25
5.2.1 Geometricka analýza priechodnosti 26
5.2.2 Vzh©adová analýza priechodnosti 27
5.3 Hybrídna analýza priechodnosti 27
5.4 Navrhovaná analýza priechodnosti 27
6 Implementácia 31
6.1 Robot operating system 31
6.1.1 Architektúra ROS 31
6.1.2 Balí£ky 31
6.1.3 Simulátor Gazebo 32
6.2 Kon gurácia robota HUSKY A200 32
6.2.1 HUSKY A200 32
6.2.2 VLP-16 33
6.2.3 GNSS Tribmle BX982 33
6.3 Pouºívané ROS uzly 34
6.3.1 Uzol SLAM 35
6.3.2 Uzol MAPPER 36
6.3.3 Uzol PATH_FOLLOWER . . 44
7 Experimentálne výsledky 47
7.1 Rozbor výsledkov analýzy priechodnosti 57
8 Záver 61
Bibliogra a 63

Obrázky

2.1 Vý²ková mapa [4] 5
2.2 Mrieºka obsadenosti v 2D [8] 5
2.3 Bodová mapa simula£ného prostredia 6

4.1 Bloková schéma ICP algoritmu. . 15
4.2 Blokové schéma ICP SLAM. . . . 17
4.4 Vyzualizácia krokov metódy ICP SLAM - na obrázku 4.3a sa robot nachádza v prostredí. ƒierne £iary popisujú kontúry zmapovaného prostredia. Súradnicové systémymap a odommajú v tomto momente rovnaký po£iatok. Na obrázku 4.3b robot vykoná pohyb. Fialové £iary popisujú meranie zo senzora a £iernou £iarou je popísané odometrické meranie, medzi súradnicovým systémomodoma base_link . Na obrázku 4.3c je meranie zarovnané domapya robot je v mapelokalizovaný. Keªºe bolo odometrické meranie nepresné, súradnicové systémymapa odomuº nemajú rovnaký po£iatok. Posun súradnicového systémuodomod súradnicového systémumap kompenzuje chybu odometrického merania. 18
4.5 Aplikovanie ltra MaxDensityDataPointsFilter - pri ltrovaní bola stanovená maximálna hustotou na hodnotu 50000 bodov na m^3. ƒervenou farbou je ozna£ený pôvodný bodový mrak a bielou farbou je ozna£ený bodový mrak po ltrovaní [20]. 20
4.6 Aplikovanie ltra DistanceLimitDataPointsFilter - z bodového mraku boli odstránené v²etky body vzdialené od po£iatku na 1 m. Bielou farbou je ozna£ený bodový mrak po ltrovaní [20]. . . . 21

5.1 Teoretická analýza priechodnosti - na obrázkoch je moºné pozorova´ rozdielne hodnoty priemernej vý²ky a rozptylu, pre dva druhy prekáºok. V tomto prípade je na základe rozptylu moºné rozhodnú´ o priechodnosti prekáºok rozdielne, keªºe rozptyl prekáºky na obrázku 5.1 je takmer o polovicu men²í oproti prekáºke na obrázku 5.1a. Sivé vertikálne £iary predstavujú hranice jednotlivých buniek. 29

6.1 Robot HUSKY A200 v základnej kon gurácií [38]. 33
6.2 Senzor VLP-16 [39]. 33
6.3 Kon gurácia HUSKY A200 - na robotovi je umiestnený senzor VLP-16 na platforme a GNSS Trimble BX982 v prednej £asti vozidla. 34
6.4 Senzor Trimble BX982 [40]. 34
6.5 Diagram uzlov v ROS - farebné boxy reprezentujú uzly a £iarami sú reprezentované komunika£né kanály. Zelenou farbou sú ozna£ené uzly ktoré som implementoval. 34
6.6 Funkcia ltra bodového mraku - limit $h_{max\,1}$ je vypo£ítaný na základe hodnoty parametrov $h_r = 1{:}0$ m a $'_{max} = 23$. Bielou farbou sú ozna£ené od ltrované body a £ervenou farbou sú ozna£ené body po ltrovaní. 37

6.7 Filtrovanie dynamických bodov a mapa priechodnosti - na obrázku je moºné vidie´ mapu priechodnosti, reprezentovanú bielymi a £ervenými bunkami. Biele bunky popisujú priechodný priestor a £ervené bunky naopak popisujú priestor nepriechodný. Fialovými kockami sú ozna£ené body od ltrované pomocou ltra, zaloºeného na najbliº²ích susedoch. V strede obrázku je vidie´ od ltrované body, ktoré boli nasnímané na osobe prechádzajúcej parkom. Zvy²né body reprezentujú bodový mrak mapy. 38
6.8 Body zasadené do buniek mrieºky - bielou farbou sú ozna£ené body zasadené do 2D mrieºky a £ervenou farbou sú ozna£ené od ltrované body. V pravo hore je na obrázku vidie´ zhluk bodov, ktoré boli zasadené do 2D mrieºky napriek tomu, ºe sa nachádzajú vysoko nad povrchom chodníka. Tieto body sa z 2D mrieºky vy ltrujú v momente, kedy bude nasnímaný povrch pod nimi, teda povrch chodníka. 38
6.9 Nafukovanie buniek - nafukovaná bunka je vyzna£ená modrou hranou. ƒervenou farbou sú zobrazené bunky, ktoré zmenia svoj stav priechodnosti na stav nafukovanej bunky. ƒísla v ostatných bunkách reprezentujú ve©kos´ penalizácie pri h©adaní cesty cez dané bunky. Po£et £ervených buniek je daný parametrom obstacle_distance, ktorý ozna£uje vzdialenos´ na ktorú sa daná bunka nafukuje. 39
6.11 Popis moºných situácií pri h©adaní cesty - na obrázkoch sú bielou farbou vyzna£ené priechodné bunky, £ervenou farbou nepriechodné bunky a modrou farbou bunky o ktorých nebolo doposia© rozhodnuté. Zelenou farbou sú vyzna£ené bunky, ktoré tvoria £as´ cesty po priechodných bunkách a rúºovou farbou je vyzna£ený potenciálny koniec cesty. ƒiernou ²ípkou je ozna£ená poloha robota a modrozelenou gu©ou je ozna£ený cie©. 43
6.12 Vizualizácia výpo£tu odchýlky y a polomeru R [44]. 45

7.1 Test klasi kácie obrubníkov - cie©om tohto testu bolo správne klasi kova´ nepriechodné bunky ktoré leºia na schodoch. Na ©avej £asti obrázku sa nachádzajú schody o vý²ke 10 cm a ved©a nich je ²ikmá plocha. Tieto schody sú pre robota nepriechodné, ke¤ºe majú ve©ký vý²kový rozdiel. ’ikmá plocha je správne klasi kovaná ako priechodná. Na pravej £asti obrázku je moºné vidie´ podobný prípad. Oproti prvému schodu je rozdielna vý²ka druhého a tretieho schodu. Tieto schody majú 5 cm a boli správne vyhodnotené ako prejazdné. 47
7.2 Test klasi kácie ²ikmých plôch - na tomto teste bola otestovaná funkcia klasi kácie na ²ikmých plochách. Algoritmus tieto plochy vyhodnotil správne ako nepriechodné, ke¤ºe sú pre robota príli² strmé. 48
7.3 Klasi kácia obrubníka v reálnych podmienkach - na obrázku je moºné pozorova´ správne vyhodnotenie priechodnosti obrubníka a jeho okolia. V strede a ©avej strane mapy priechodnosti sa nachádza ²um zanesený prechádzajúcou osobou. . 49

7.4 Mapa priechodnosti parku - na obrázku 7.5a je moºné pozorova´ správne vyhodnotenie nepriechodných prekáºok a obrubníka v ©avej £asti obrázku. ƒiernou farbou je vyzna£ená trajektória robota, ktorú robot prejazdil pri zadaní viacerých cie©ových polôh. Taktieº je moºné pozorova´ viacero miest, ktoré sú za²umené vplyvom dynamických objektov. V pravej £asti sa nachádza nesprávne vyhodnotená oblas´ na okraji bodového mraku. 50
7.5 Prípad nesprávnej klasi kácie malého obrubníka - na bodovej mape je moºné pozorova´ vyhladenie obrubníka ICP SLAM algoritmom. V bodovom mraku je daný obrubník takmer nepozorovate©ný a skôr sa javí ako mierny sklon terénu. Tento obrubník je v mape vyhodnotený ako nepriechodný. Na ©avo od obrubníka sa nachádza ²um spôsobený prechádzajúcim men²ím psom. 51
7.6 Porovnanie odstra¬ovania dynamických bodov - pred prekáºkami som sa po£as jazdy robota po laboratóriu pre²iel, aby som simuloval dynamický objekt. V modrom kruhu je zobrazená prekáºka na ktorej je vidie´ efekt odstra¬ovania dynamických bodov na bodovú mapu. Prekáºka je na obrázku 7.6d vyhodnotená ako prejazdná. Filtrovanie dynamických bodov ovplyv¬uje v²etky body v kuºeli vyºarovaného laserového lú£u. Ak je teda snímaný bod tesne za prekáºkou, je moºné ºe niektoré body na hranách danej prekáºky budú z bodového mraku odstránené. 52
7.7 Porovnanie satelitnej mapy z mapou priechodnosti terénu - táto ve©ká mapa priechodnosti vznikla priebeºným zlu£ovaním máp, ktoré vznikali pohybom robota. Zelenou farbou je ozna£ená trajektória, po ktorej sa robot pohyboval. V mape priechodnosti je moºné pozorova´ kontúry chodníka a kontúry mierne kopcovitého terénu na spodnej £asti obrázku 7.8b. Mapa taktieº obsahuje mnoºstvo nesprávne klasi kovaných buniek na miestach kde sa pohybovali dynamické objekty. 53
7.8 Porovnanie satelitnej mapy s mapou priechodnosti terénu - táto ve©ká mapa priechodnosti vznikla princípom popísaným v predo²lom obrázku. Zelenou farbou je ozna£ená trajektória, po ktorej sa robot pohyboval. Na obrázku 7.8b je moºné pozorova´ zvä£²a správne vyhodnotené obrubníky, schody a ve©ké prekáºky. Mapa obsahuje ve©a ²umu, spôsobeného osobami a prechádzajúcim automobilom. . . . 54
7.9 Mapa priechodnosti parku pri budove NTK - na obrázku je moºné pozorova´ správne vyhodnotenie ve©kých prekáºok. Ako malé kop£eky boli vyhodnotené oblasti, kde sa nachádza nahrabané lístie. V spodnej £asti obrázku sa nachádza ²um spôsobený dynamickými objektami. 55
7.10 Mapa priechodnosti cesty pri budove NTK - na obrázkoch je moºné pozorova´ správne vyhodnotené oblasti obrubníka a automobilov. . 55
7.11 Mapa priechodnosti cesty pri budove NTK - na obrázku je moºné pozorova´ správne vyhodnotenie oblasti obrubníka a schodov. V mape je taktieº vidie´ neod ltrované body, nasnímané na osobe, pohybujúcej sa pred robotom. 56

Tabu©ky

7.12 Mapa priechodnosti mierne zvlneného terénu - na obrázku je moºné pozorova´ pozvo©ný prechod priechodných a nepriechodných buniek v oblasti narastajúceho sklonu trávnatej plochy. 56
7.13 Príklad neodstránených bodov tesne nad plochou - na obrázku je moºné pozorova´ nesprávnu klasi káciu priechodnej oblasti. Modrou farbou sú ozna£ené body zasadené do mrieºky. V strede obrázku je moºné pozorova´ nevy ltrované dynamické body, nachádzajúce sa tesne nad plochou chodníka. 58
7.14 Príklad nesprávne pridaných bodov do bodovej mapy - nesprávne pridané body zapríi£¬ujú nesprávne vyhodnotenie oblasti cesty v strede obrázku 7.14a. 58
7.15 Test propriocep£nej analýzy na základe polohy robota - alovou farbou sú ozna£ené nepriechodné bunky, ktoré boli vyhodnotené na základe polohy robota. Zelenou farbou je ozna£ená trajektória po ktorej sa robot pohyboval pri h©adaní cesty k cie©u, ktorý je ozna£ený svetlo modrou farbou. 59

6.1 Vlastnosti senzora VLP-16 [39]. . 33
6.2 Dôleºité parametre uzla SLAM. Popis parametrov $_{max}$, e_d, e_a, , a $_{max}$ sa nachádza na konci podkapitoly 4.5. 36
6.3 Dôleºité parametre uzla Mapper. 44

7.1 Kon gurácia ICP SLAM. V²etky pouºité moduly, ltre a ich parametre sú popísané v podkapitole 4.4. 48
7.2 Parametre uzla SLAM. 49
7.3 Parametre uzla MAPPER. 49
7.4 Frekvencie s ktorými boli preposielané správy medzi uzlami. 60

Kapitola 1

Úvod

Autonómni roboti sú dnes pouºivaní v ²irokej ²kále aplikácií, cez roboty pouºivane v domácnostiach [1], v armáde [2] alebo ako autonómne vozidlá [3]. Títo roboti musia by´ schopní navigácie v neznámom prostredí, k £omu patrí schopnos´ lokalizácie a mapovania. Problémom simultánnej lokalizácie a mapovania (SLAM) sa vedecká komunita zaoberá posledných 30 rokov. Za tento £as boli navrhnuté a úspe²ne otestované rôzne metódy SLAM. Roboti sa pohybujú rôznymi spôsobmi, sú stavané v rôznych ve©kostiach a ²peci kované na rôzne úlohy. V zmapovanom prostredí je potrebné pre kaºdého robota vymedzi´ priestor, v ktorom je pohyb pre daného robota bezpe£ný. Touto problematikou sa zaoberá analýza priechodnosti prostredia. V tejto práci sa bliº²ie pozrieme na vyhodnotenie priechodnosti terénu pre robota HUSKY A200, pomocou LIDAR senzora.

1.1 Ciele práce

Hlavným cie©om tejto práce je vytvori´ systém, ktorý bude schopný autonómnej navigácie robota HUSKY A200 vo vonkaj²om teréne. Prácu je moºné rozdeli´ do nasledujúcich podúloh:

1. Na²tudovanie a výber vhodných senzorov, metód SLAM a metód analýzy priechodnosti prostredia.
2. Návrh a implementácia softvéru, ktorý zabezpe£í analýzu priechodnosti terénu, h©adanie cesty a sledovanie danej cesty.

Kapitola 2

Simultánna lokalizácia a mapovanie

2.1 Formulácia a ²truktúra úlohy SLAM

Simultánnu lokalizáciu a mapovanie (SLAM) je mo°no formálne popísa´ z pravdepodobnostnej perspektívy [4]. Znakom t je ozna£ený £as a znak x_t popisuje pozíciu robota. Trajektória robota je daná ako

$$X_T = \{x_0, x_1, x_2, \ldots, x_T\}, \tag{2.1}$$

kde poloha robota v £ase t je ozna£ená ako x_t. Znakom T je ozna£ený £as, v ktorom metóda kon£í. Po£iato£ná poloha x_0 je známa, ostatné pozície sa nedajú presne pozorova´.

Odometria poskytuje relatívne informácie o pohybe medzi po sebe nasledujúcimi polohami. Informácia o pohybe vykonanom medzi £asom $t-1$ a £asom t je ozna£ená ako u_t. Sekvencia

$$U_T = \{u_0, u_1, u_2, \ldots, u_T\} \tag{2.2}$$

reprezentuje relatívne pohyby vykonané robotom. Pre merania bez nepresností by U_t bolo dosta£ujúce na získanie X_t pomocou po£iato£nej polohy x_0. Av²ak, odometrické merania sú nepresné a £asom akumulujú chybu od skuto£nej polohy.

Robot je schopný pomocou senzorov získava´ informácie o prostredí, v ktorom sa nachádza. Mapam reprezentuje prostredie skuto£ne bez ²umu a chýb. Senzorické merania poskytujú informáciu medzi rysmi mapy m a polohou robota x_t. Ak predpokladáme, °e robot vykoná jedno meranie v ka°dom £asovom okamihu, sekvencia meraní je daná ako

$$Z_T = \{z_0, z_1, z_2, \ldots, z_T\}. \tag{2.3}$$

Problém SLAM rie²i získanie mapy m a získanie sekvencie polôh robota X_T z odometrie a senzorických meraní. Problém SLAM vieme rozdeli´ na dve hlavné formy, full SLAM a online SLAM.

Full SLAM odhaduje podmienenú pravdepodobnos´ v²etkých polôh v trajektórií X_T spolu s mapoum:

$$p(X_T, m \mid Z_T, U_T). \tag{2.4}$$

Online SLAM odhaduje podmienenú pravdepodobnos´ polohy x_t spolu s mapou m:

$$p\ x_t; m\ Z_T; U_T\ : \tag{2.5}$$

Algoritmy online SLAM metód sú zvy£ajne inkrementálne a sú schopné spracováva´ merania osobitne. Naopak full SLAM metódy spracovávajú v²etky merania naraz, ke¤ºe odhad prebieha naprie£ v²etkými polohami X_T. Online SLAM metódy sú oproti full SLAM metódam výpo£tovo menej náro£né, av²ak full SLAM metódy majú vä£²iu ²ancu zotavi´ sa z chybných meraní [4].

2.2 Lokalizácia

Lokalizáciou je myslený proces odhadu aktuálnych súradníc robota na základe informácií z dostupných senzorov. Úloha lokalizácie sa dá rozdeli´ na problémy sledovania polohy a globálnej lokalizácie. Sledovanie polohy znamená ur£ova´ polohu robota vzh©adom na nejakú po£iato£nú pozíciu. V tomto prístupe sú dôleºité relatívne súradnice vzh©adom k po£iato£nej polohe, preto nie je nutné pozna´ model prostredia. V prípade globálnej lokalizácie sa ur£uje poloha vzh©adom k predom známemu modelu prostredia. Obecne sa súradnice robota skladajú z polohy a orientácie. Poloha je popísaná v troch karteziánskych súradniciach (x,y,z) a orientácia v troch smerových uhloch ozna£ovaných ako roll/pitch/yaw", t.j. rotácie vo£i osiam súradnicového systému x, y, z. Lokalizáciu je moºné zjednodu²i´, pokia© sa robot pohybuje po rovine. V takom prípade je poloha popísaná v dvoch karteziánskych súradniciach (x,y) a orientácia pomocou uhla , ktorý reprezentuje rotáciu vo£i ose kolmej na roviny x, y [5].

2.3 Mapovanie

Mapovanie je proces zostavenia modelu okolitého prostredia. Mapovanie, z poh©adu robotiky, prevádza fyzické rysy sveta na vhodnú dátovu reprezentáciu. Na základe mapy je robot schopný vytvára´ rozhodnutia z oh©adom na teraj²í stav okolitého prostredia. Pri tvorbe mapy [4] popisuje tri základne problémy.

1. Mapy musia by´ dostato£ne kompaktné nato, aby s nimi bolo moºné pracova´ v ostatných £astiach systému ako napríklad h©adanie trasy.
2. Mapy musia by´ prispôsobené rie²enému problému a charakteru mapovaného prostredia.
3. Mapa musí bra´ v úvahu nepresnosti, spojené so senzormi a odhadom stavu v ktorom sa robot nachádza.

Obecne neexistuje univerzálna mapa, aplikovate©ná k rie²eniu kaºdého problému. V nadchádzajúcom preh©ade sú popísané niektoré z druhov máp, vhodných k rie²eniu problému SLAM vo vonkaj²om teréne [4].

2.3.1 Mrieºka obsadenosti

Mrieºka obsadenosti obr. 2.2 rozde©uje priestor na bunky, kde kaºdá bunka obsahuje pravdepodobnostný odhad jej stavu. Bunky sú de nované predvolenou d¨ºkou hrany a majú presne ur£enú polohu. Bunka môºe by´ prázdna, obsadená, poprípade nie de novaná, v prípade, ºe nebola doposia© zmapovaná. [6],[7]. Výpo£et hodnoty obsadenosti sa skladá z dvoch krokov. Najprv je vytvorený pravdepodobnostný model senzoru, ktorý umoº¬uje ur£í´ hodnotu obsadenosti pre bunky, obsiahnuté v jednotlivých senzorických meraniach. V druhom kroku sa hodnoty obsadenosti zlu£ujú do kone£nej mrieºky obsadenosti, na základe Bayesovej vety alebo iných pravdepodobnostných prístupov. Nevýhodou mrieºky obsadenosti je kvadratický, poprípade kubický rast pamä´ovej náro£nosti. Výhodou je zahrnutie nepresnosti senzorov do mapy a de novanie prázdneho priestoru, £o je vhodné pre algoritmy na plánovanie ciest [5].

2.3.2 2.5D mapa

Táto forma mapy sa skladá z 2D mrieºky buniek, ktoré obsahujú pridanú informáciu. Jedným s najbeºnej²ích pouºití je priradenie nadmorskej vý²ky do kaºdej bunky, £ím dostávame takzvanú vý²kovú mapu. Vý²ková mapa obr. 2.1 je oproti bodovej mape menej náro£nej²ou dátovou reprezentáciou a je obzvlá²´ vhodná pre aplikácie, kde sa robot pohybuje vo vonkaj²om teréne, ktorý neobsahuje rôzne previsy, tunely a mosty. Ako pridanú informáciu je taktieº moºné pouºí´ informáciu o priechodnosti danej bunky, v takomto prípade hovoríme o mape priechodnosti [4].

Obrázok 2.1: Vý²ková mapa [4]

Obrázok 2.2: Mrieºka obsadenosti v 2D [8]

2.3.3 Bodová mapa

Bodovú mapu obr. 2.3 tvorí mnoºina bodov, ktorá reprezentuje diskretizovanú geometriu okolitého prostredia. Bodová mapa je vhodná na lokalizáciu a mapovanie pomocou scan-matching metód [5]. Bodová mapa je schopná uchova´ senzorické merania v ich originálnom rozloºení. Jej nevýhodou je pomerne rýchlo narastajúci po£et celkových bodov [4]. Výpo£tovú náro£nos´ operácií, aplikovaných na bodovú mapu, je moºné zmen²i´ pouºitím vhodnej reprezentácie bodovej mapy. Jednou z takýchto ²truktúr je K-D strom, popísaný v kapitole 4.4.

Obrázok 2.3: Bodová mapa simula£ného prostredia

2.4 SLAM metódy

Vä£²ina SLAM metód spadá do nasledujúcich kategórií:

- SLAM s pouºitím roz²íreného kalmanového ltra (EKF SLAM)
- SLAM s pouºitím £asticového ltra
- grafový SLAM

V tejto kapitole je popísaný základný princíp daných metód spolu s metódou scan-matching SLAM, ktorá je pouºitá v praktickej £asti tejto práce.

2.4.1 EKF SLAM

EKF SLAM pouºíva stavový vektor na odhad pozícii robota a pozícií rysov prostredia, spolu s kovarian£nou maticou, ktorá reprezentuje chyby a neistotu týchto odhadov. Kovarian£ná matica taktieº obsahuje korelácie medzi odhadmi stavov robota a rysov prostredia. Robot pri pohybe prostredím získava senzorické merania, pod©a ktorých sa aktualizuje stavový vektor a kovarian£ná matica, pouºitím roz²íreného kalmanového ltra. Pozorovaním nových rysov prostredia sú do stavového vektoru pridávané nové stavy. Tieto metódy pracujú s predpokladom toho, ºe polohu robota a polohy rysov prostredia môºeme reprezentova´ normálnym rozdelením [4], [9].

2.4.2 SLAM s pouºitím £asticového ltra

SLAM s pouºitím £asticového ltra je metóda, ktorá pouºíva £astice na reprezentáciu polohy robota a mapy prostredia. Kaºdá £astica obsahuje vlastnú polohu robota, mapu prostredia a takzvaný importance faktor. Importance faktor vyjadruje pravdepodobnos´ toho, ºe sa robot v danej polohe nachádza. Metóda je inicializovaná £asticami s rovnakým importance faktorom a polohou, naj£astej²ie v po£iatku súradnicového systému mapy prostredia. S pohybom robota sú získavané odometrické a senzorické merania. Nasleduje rekurzívne opakovanie takzvanej predikcie a korekcie. Pri predikcii je poloha £astíc posunutá na základe odometrického merania. V korekcií je pre kaºdú £asticu vypo£ítaný importance factor, na základe zhody mapy prostredia a senzorického merania. Na záver sú £astice s malým importance factorom zahodené a nahradené novými £asticami. Na tomto princípe funguje prelomový FastSLAM algoritmus [4].

2.4.3 Grafový SLAM

Grafový SLAM reprezentuje polohy robota, rysy prostredia a vz´ahy medzi nimi my²lienkou grafu. Graf je zloºený z uzlov a hrán. Uzly môºu popisova´ bu¤ polohu robota v £ase, alebo pozorované rysy. Hrany reprezentujú obmedzenia medzi uzlami. Medzi dvoma uzlami polohy je obmedzením odhad vzdialenosti medzi danými polohami, naj£astej²ie formou odometrie. Obmedzením medzi uzlom polohy a uzlom pozorovaného rysu je vzdialenos´ pozorovaného rysu od danej polohy [4].
Takýto graf si je moºné predstavi´ ako systém hmotných bodov spojených pruºinami [10]. Výpo£et SLAM rie²enia je analógiou výpo£tu stavu tohoto systému s minimálnou energiou. Na tento výpo£et sú aplikovate©né optimaliza£né metódy, £o nám umoº¬uje rie²i´ SLAM problém ako optimaliza£nú úlohu. Vzh©adom nato, ºe optimalizácia prebieha naprie£ celým grafom, grafový SLAM patrí full SLAM metódam [4].

2.4.4 Scan-matching SLAM

Scan-matching metódy h©adajú transformáciu, ktorá minimalizuje vzdialenos´ medzi meraním a referenciou. Základným predpokladom tejto metódy je to, ºe dané merania pochádzajú z rovnakého prostredia. Merania môºu by´ reprezentované ako body, alebo ako ²peci cké rysy prostredia, napríklad detekované objekty. Scan-matching metódy je moºné deli´ na dva druhy, a to scan-to-scan metódy a scan-to-map metódy. Výpo£et transformácie medzi po sebe idúcimi meraniami rie²ia scan-to-scan metódy. V prípade h©adania transformácie senzorického merania vo£i mape, hovoríme o scan-to-map metóde. Polohu robota je moºné ur£i´ na základe scan-to-map postupu, ktorého výsledkom je transformácia medzi senzorickým meraním a mapou. Senzorické meranie je moºné pod©a nájdenej transformácie zakomponova´ do mapy[11], [12]. Na tomto princípe funguje metóda ICP SLAM 4.

Kapitola 3

Senzory

V tejto kapitole sú stru£ne popísane princípy fungovania be^{o}ných senzorov, pou^{o}ívaných v SLAM metódach. Konkrétne senzory, pou^{o}ité v praktickej £asti, sú podrobnej^{2}ie popísané v kapitole 6.2.

3.1 Odometria

Odometria je základný senzorický systém pre ur£ovanie polohy robota. Na ur£enie polohy sú pou^{o}ívané informácie z ak£ných £lenov robota alebo zo senzorov, ktoré robot obsahuje. V prípade robota, ktorý má za ak£ný £len koleso, je mo^{o}né na odometriu pou^{o}í´ senzorické merania z inkrementálnych rota£ných senzorov [5]. Pre v^{2}eobecný odhad odometrie je taktie^{o} mo^{o}no pou^{o}i´ inerciálnu meraciu jednotku (IMU).

3.1.1 Inerciálna meracia jednotka - IMU

Inerciálna meracia jednotka je zlo^{o}ená z trojice gyroskopov a trojice akcelerometrov. Tieto senzory sú montované do troch navzájom kolmých osí. V takomto rozlo^{o}ení sú schopné mera´ nato£enie a zrýchlenie v 3D priestore. Akcelerometry slú^{o}ia na meranie zrýchlenia a gyroskopy na meranie uhlovej rýchlosti rotácie. Na základe nameraného rýchlostného vektora a nato£enia vo£i osiam, je mo^{o}né získa´ relatívnu polohu, orientáciu a zrýchlenie senzoru. Inerciálne senzory majú rýchlu odozvu. Sú v^{2}ak za´a^{o}ené chybou odhahu rýchlosti a polohy, ktorá bez korekcie £asom kumulatívne narastá [5].

3.2 Globálny dru^{o}icový polohový systém - GNSS

Globálny dru^{o}icový polohový systém (GNSS) je jedným s najpou^{o}ívanej^{2}ích systémov na lokalizáciu polohy. Systém je zalo^{o}ený na prijímaní informácií od dru^{o}íc obiehajúcich okolo zeme. Dru^{o}ice kontinuálne vysielajú signály, ktoré obsahujú informáciu o ich polohe a trajektórii v danom £ase. Z prijatých informácii od viacerých dru^{o}íc je následne mo^{o}né dopo£íta´ aktuálnu polohu prijíma£u. Daná poloha je vyjadrená v horizontálnej rovine zeme [4]. Základná presnos´ ur£enia polohy obsahuje chybu v jednotkách metrov. Chyba závisí

hlavne na podmienkach príjmu signálov z druºíc a taktieº na po£te druºíc, od ktorých bol signál prijatý. Presnos´ GNSS lokalizácie je moºné vylep²i´ príjmom a spracovaním diferen£ných korek£ných signálov, ktoré vysielajú pozemné alebo druºicové vysiela£e. Nevýhodou GNSS je absencia signálov z druºíc v budovách a pod zemou. V takýchto prípadoch nie je moºné GNSS pouºi´ [5].

3.3 Sonar

Sonar je jednoduchý senzor, ktorý pouºíva akustické pulzy na detekovanie vzdialenosti k objektu. Ke¤ºe rýchlos´ ²írenia zvuku je zvy£ajne známa, vzdialenos´ detekovaného objektu je priamo úmerná £asu, v ktorom zvuková vlna letí od senzoru k prekáºke a naspä´. Medzi hlavnú nevýhodu patrí presnos´ meraní, obzvlá²´ vo vonkaj²om prostredí. Presnos´ meraní závisí na správnom ur£ení rýchlosti vzduchu, ktorú ovplyv¬uje teplota prostredia a vzdu²né prúdenie. Kvalitu taktieº ovplyv¬uje materiál prekáºok. Materiál môºe zvukovú vlnu pohlti´, poprípade odrazi´. Pri úplnom pohltení zvukovej vlny prekáºka nie je detekovaná. Pri odrazení zvukovej vlny môºe dôjs´ k ur£eniu nesprávnej vzdialenosti prekáºky od senzoru. Výhodou sonaru je jeho relatívne nízka cena a ve©kos´ prostredia do ktorého vyºaruje zvukové vlny[5].

3.4 Kamery

Beºné monokulárne kamery je moºné pouºi´ na získanie obrazu prostredia. Výhodou týchto kamier je ich dostupnos´ a cena. Nevýhodou je absencia h¨bkovej informácie pixelov. H¨bkovú informáciu je moºné získa´ na základe závislosti detekovaných rysov v kamerových záberoch. Ak poznáme polohu kamery v kaºdom zábere, je moºné danú kameru v prostredí lokalizova´ a dané prostredie zmapova´. Týmto problémom sa zaoberajú Monocular SLAM metódy. Tieto metódy sú podobné structure-from-motion metódam z oblasti po£íta£ového videnia [13].

RGB-D senzor je kombináciou monokulárnej kamery a infra£erveného mera£a vzdialenosti. To umoº¬uje získava´ obraz prostredia spolu s informáciou o h¨bke pixelov. Oproti monokulárnym kamerám je výhodou pridaná informácia o h¨bke pixelov v obraze. RGB-D senzory pouºívajú metódy RGB-D SLAM [14].

Stereo kamery sú in²pirované ©udským zrakom. Tieto kamery obsahujú dvojicu objektívov, ktorá sníma tú istú £as´ prostredia. Na základe obrazu z kamier a známej vzdialenosti a orientácií objektívov, je moºné vypo£íta´ h¨bkovú informáciu snímaných objektov [14].

3.5 LIDAR

Light Detection And Ranging (LIDAR) je metóda merania vzdialenosti medzi senzorom a prostredím pomocou laserového lú£u. Vzdialenos´ je mo°né vypo£íta´ na základe doby od vy°iarenia laserového lú£u po £as, kedy je odrazený lú£ opä´ zaznamenaný senzorom alebo na základe fázového posunu prijatého laserového lú£u. Vzorec na výpo£et vzdialenosti D je de novaný v nasledujúcej rovnici [15].

$$D = c \frac{T}{2}; \tag{3.1}$$

kde c ozna£uje rýchlos´ svetla a T je £as od vy°iarenia po príjem laserového lú£u.

Laserový lú£ mô°e by´ statický, alebo rozpestrený po prostredí pomocou rotujúcich zrkadiel. Pre ú£ely mobilnej robotiky sa pou°íva lú£ druhého typu. V jednoduch²ích aplikáciách, kde je predpokladom pohyb po rovine, je mo°né pou°i´ lacnej²ie senzory. Tieto senzory poskytujú 2D horizontálny rez prostredím, ktorý je reprezentovaný ako mno°ina bodov v polárnych súradniciach s po£iatkom v polohe senzoru. Naopak výstupom 3D LIDAR senzorov je mno°ina bodov v sférických súradniciach, taktie° s po£iatkom v polohe senzoru [5].

Výhodou týchto senzorov je rýchla odozva v jednotkách hertzov a presnos´ merania vzdialenosti s chybou v jednotkách centimetrov. Niektoré LIDAR senzory doká°u sníma´ státisíce bodov za sekundu. Kvôli týmto vlastnostiam sú LIDAR senzory pou°ívané v oblasti mobilnej robotiky, ke¤°e umo°¬ujú vytvára´ model prostredia robota v reálnom £ase [5].

Kapitola 4

ICP SLAM

4.1 ICP algoritmus

Cie©om iterative-closest-point (ICP) algoritmu je nájdenie transformácie medzi meraníma referenciou , ktorá dané bodové mraky (z anglického point cloud) najlep²ie zosadí [16]. Bodové mraky popisujú neusporiadanú mno°inu bodov v priestore. Body majú svoje súradnice v súradnicovom systéme daného bodového mraku. Formálne [16] popisuje tento algoritmus nasledovne.

Nech tvar ${}^{A}P$ reprezentuje meranie v súradnicovom systéme A a tvar ${}^{B}Q$ reprezentuje referenciu v súradnicovom systéme B. Cie©om ICP je odhad transformácie ${}^{B}_{A}T$ minimalizáciou chybovej funkcie $\text{error}(P^{0}, Q)$:

$$ {}^{B}_{A}\hat{T} = \underset{T}{\operatorname{argmin}}\,(\text{error}(T({}^{A}P); {}^{B}Q)); \tag{4.1} $$

kde $T({}^{A}P)$ je aplikovanie transformácie T na meranie ${}^{A}P$.

Chybová funkcia je po£ítaná medzi pármi bodov, ktoré boli asociované medzi bodovými mrakmi. Základná asociácia je vykonaná nájdením najbli°²ieho bodu merania pre ka°dý bod referencie . V ideálnom prípade zdru°enie nastáva pre pár bodov, ktoré majú po zarovnaní merania a referencie najmen²iu vzdialenos´.

Nech $M = \text{match}(P; Q) = \{(p; q) : p \in P; q \in Q\}$ je mno°ina asociovaných párov bodov medzi P a Q. Potom mô°e by´ chybová funkcia vyjadrená ako

$$ \text{error}(P; Q) = \sum_{(p;q)\in M} d(p; q); \tag{4.2} $$

kde $d(p; q)$ ozna£uje funkciu popisujúcu vzdialenos´ bodov p, q.

Mno°ina M mô°e obsahova´ od©ahlé páry. Tieto páry sú identi kované a vymazané zo skupiny M. Dodato£ne sú zavedené váhy $W = \text{outlier}(M) = \{w(p; q) : \forall (p; q) \in M\}$, ktoré mô°u by´ pridelene párom zo skupiny M, aby zvý²ili alebo zní°ili ich dopad v chybovej funkcii

$$ \text{error}(P; Q) = \sum_{(p;q)\in M} w(p; q)\, d(p; q). \tag{4.3} $$

V prípade ideálnej asociácie bodov do mno°iny M, minimalizácia chybovej funkcie produkuje najlep²í odhad transformácie ${}^{B}_{A}T$ 4.1. Obecne problém

asociácie bodov nie je moºné presne vyrie²i´. Hlavnou my²lienkou ICP algoritmu je to, ºe aj s nepresnou asociáciou bodov do skupiny M, minimalizácia chybovej funkcie produkuje lep²í odhad po£iato£nej transformácie ${}^{B}_{A}T$. To znamená, ºe nepresnou asociáciou bodov v iterácii $i-1$, je ICP algoritmus schopný vyprodukova´ presnej²iu asociáciu bodov v iterácii i. Cie©om je teda vybudova´ postupnos´ transformácií ${}_{i}{}^{i}_{1}T$, ktoré sú aplikované na meranie P. V kaºdej iterácii sú body asociované do skupiny M, na základe predo²lého odhadu transformácie ${}_{i}{}^{i}_{1}T$. Potom, na základe aktualizovanej asociácie bodov, je vypo£ítaná nová transformácia ${}^{i+1}_{i}T$ minimalizáciou chybovej funkcie

$$ {}^{i+1}_{i}T = \underset{T}{\operatorname{argmin}} \left(\operatorname{error}\left(T\left({}^{i}P^{0} \right); Q \right) \right) : \tag{4.4} $$

Po ukon£ení behu algoritmu je výsledná transformácia zloºená z transformácií získaných v iteráciách algoritmu

$$ {}^{B}_{A}\hat{T} = {}_{i}{}^{i}_{1}T \cdot \ldots \cdot {}^{3}_{2}T \cdot {}^{2}_{1}T \cdot T_{init} ; \tag{4.5} $$

kde T_{init} je po£iato£ný odhad medzi meraním a referenciou a i ozna£uje po£et iterácií algoritmu. ICP algoritmus môºe by´ zastavený na základe konvergen£ných kritérií alebo ur£ením maximálneho po£tu iterácií. Konvergen£ným kritériom môºe by´ napríklad minimálna ve©kos´ vzdialenosti a rotácie transforma£nej matice ${}_{i}{}^{i}_{1}T$, teda matice získanej v jednej iterácií algoritmu. Vylep²i´ robustnos´ ICP algoritmu je moºné pred filtrovaním bodových mrakov merania a referencie. Na výber je taktieº viacero chybových a outlier funkcií. Filtre a funkcie pouºité na konfiguráciu ICP SLAM sú popísané v 4.4. Tento postup je zhrnutý formou pseudokódu v Algoritme 1.

Algorithm 1 ICP

```
Require: ^A P                                          . meranie
Require: ^B Q                                          . referencia
Require: T_init                                        . po£iato£ná transformácia
  ^A P^0 ← datafilter(^A P)                            . Aplikovanie filtrov bodových mrakov
  ^B Q^0 ← datafilter(^B Q)                            . Aplikovanie filtrov bodových mrakov
  _i^i_1T ← T_init
  while not convergence do
      ^Bi P^0 ← _i^i_1T(^i_1P^0)                       . Aktualizácia merania
      M_i ← match(^i P^0, Q^0)                         . Asociácia bodov
      W_i ← outlier(M_i)                               . Filtrovanie outlierov
      ^{i+1}_i T ← argmin_T (error(T(^i P^0); Q))
  end while
  ^B_A T̂ ← _i^i_1T · ... · ^3_2T · ^2_1T · T_init
```

4.2 Kniºnica Libpointmatcher

Libpointmatcher [1] je vo©ne dostupná kniºnica, slúºiaca na registráciu mrakov bodov [17]. ICP algoritmus je rozdelený na modulárne £asti, ktoré sú optimalizované, parametrizovate©né a jednoducho kon gurovate©né pomocou YAML [2] kon gura£ných súborov. Libpointmatcher ponúka nasledujúce typy modulov:

- DataPointsFilter je modul, ktorý slúºi na ltráciu mrakov bodov. Niektoré z týchto ltrov taktieº pridávajú informácie k bodom, napríklad normálu bodu vo£i povrchu.
- Modul Matcher slúºi na asociáciu bodov z merania k referencii.
- Modul OutlierFilter obsahuje funkcie na vymazávanie, alebo váºenie spojitostí bodov v meraní s ich asociovanými bodmi vreferencii .
- ErrorMinimizer je modul, ktorý sa pouºíva na výpo£et transforma£nej matice $^{B}_{A}T$.
- Modul TransformationChecker zastavuje ICP algoritmus na základe zvoleného kritéria.

Blokové schéma ICP algoritmu z kniºnice libpointmatcher je zobrazené na obrázku 4.1. Na za£iatku algoritmu sú nameranie a referenciu aplikované ltre bodových mrakov. Po ltrácií za£ína cyklus ICP. Na meranie sa aplikuje po£iato£ná transformácia, poprípade nájdená transformácia z predo²lého cyklu. V ¤al²om kroku prebieha asociáciamerania vo£i referencii . Outlier ltre bu¤ vymaºú, alebo zváºia spojitosti medzi bodmi z merania a referencie . Následne chybová funkcia vykoná optimaliza£ný krok a vypo£íta novú transforma£nú maticu. V poslednom kroku sa skontrolujú podmienky ukon£enia ICP algoritmu a pod©a nich algoritmus kon£í, alebo sa celý cyklus opakuje znova.

Obrázok 4.1: Bloková schéma ICP algoritmu.

[1] https://github.com/ethz-asl/libpointmatcher
[2] YAML je pre £loveka jednoducho £itate©ný formát, slú¹iaci na serializáciu dát.

4.3 ICP SLAM

Základný princíp metódy ICP SLAM bol popísaný v podkapitole 2.4.4. V tejto podkapitole je popísaná implementácia v rámci balí£ku norlab_icp_mapper [3]. Tento balí£ek importuje ICP algoritmus z kniºnice libpointmatcher . Norlab_icp_mapper poskytuje moºnosť SLAM v reálnom £ase s nastaviteľnými parametrami. Tento balí£ek je moºné zakomponovať v robot operating system (ROS), v rámci balí£ka norlab_icp_mapper_ros [4], ktorý slúºi na interpretovanie ROS správ z a do balí£ka norlab_icp_mapper .

Algoritmus ICP SLAM má na vstupe odhad transformácie a bodový mrak reprezentujúci meranie. V prípade tejto práce sú odhadom transformácie odometrické dáta a bodový mrak z LIDAR senzora. Výstupom je naopak mapa a aktuálna poloha robota v danej mape. Pre presný popis je dôleºité zadefinovať si nasledujúce súradnicové systémy a transformácie medzi nimi:

1. Popis súradnicových systémov:
   - Súradnicový systém map reprezentuje súradnice, v ktorých je uloºená výsledná mapa
   - V súradnicovom systéme odom sú prijímané odometrické merania. V prípade tejto práce je po£iato£ná poloha po£iatku súradnicových systémov mapa a odom rovnaká. ƒasom ICP SLAM opravuje chybu odometrie a vzájomná poloha týchto systémov sa mení.
   - Súradnicový systém base_link má po£iatok zakotvený niekde na robotickej platforme. Ten slúºi na transformovanie bodov zo senzora do súradnicového systému odom. Senzor má svoj vlastný súradnicový systém lidar .
2. Popis transformácií medzi súradnicovými systémami:
   - Transformácia ${}^{odom}_{map}T$ popisuje vzájomnú polohu systémov mapa a odom.
   - Algoritmus prijíma odmetrické dáta ako transformáciu ${}^{base_link}_{odom}T$ popisujúcu vzájomnú polohu systémov odom a base_link .
   - Statická transformácia ${}^{lidar}_{base_link}T$ medzi systémom base_link a lidar popisuje vzájomnú polohu senzoru vo£i robotickej platforme.

Na za£iatku algoritmu sú na meranie ${}^{lidar}P$ aplikované ltre bodových mrakov. Referencia je v tomto prípade mapa ${}^{map}M$ prostredia. Mapa sa v prvej iterácií inicializuje ako bodový mrak merania a výsledkom prvej lokalizácie je odometrické meranie. V prípade inicializovanej mapy je meranie transformované zo systému lidar do systému map pomocou transformácie

$$ {}^{lidar}_{map}T = {}^{lidar}_{base_link}T \; {}^{base_link}_{odom}T \; {}^{odom}_{map}T : \tag{4.6} $$

[3] https://github.com/norlab-ulaval/norlab_icp_mapper
[4] https://github.com/norlab-ulaval/norlab_icp_mapper_ros

Po£iato£ný odhad medzimeraníma mapouje zakomponovaný v transformovaní bodového mraku merania do systému ^{map}. Toto meranie je spolu s mapou následne pou^{o}ité v ICP algoritme za ú£elom získania korek£nej transformácie T_c z rovnice 4.1. Kombináciou korek£nej transformácie T_c a transformácie $^{lidar}_{map}T$ získame lokalizovanú polohu senzora v systéme ^{map} formou transformácie

$$^{lidar}_{map}p = T_c \; ^{lidar}_{map}T : \tag{4.7}$$

Pod©a tejto transformácie je meranie pridané do mapy a na mapu sú aplikované filtre. Následne je upravená transformácia

$$^{odom}_{map}T = {}^{lidar}_{map}p \; {}^{lidar}_{base_link}T^{-1} \; {}^{base_link}_{odom}T^{-1} : \tag{4.8}$$

Reálna poloha robota v priestore je popísaná transformáciou

$$^{base_link}_{map}T = {}^{base_link}_{odom}T \; {}^{odom}_{map}T : \tag{4.9}$$

Úlohou transformácie $^{odom}_{map}T$ je korekcia chyby v odometrickom meraní. Proces algoritmu je priblí^{o}ený na obrázkoch 6.11 a popísaný formou pseudokódu v algoritme 2. Zre´azenie operácií je vizualizované v diagrame 4.2.

Algorithm 2 Iterácia ICP SLAM v reálnom £ase

Require: $^{lidar}P$. meranie zo senzora
Require: $^{base_link}_{odom}T$. odometria
$^{lidar}P^0 \leftarrow \text{datafilter}(^{lidar}P)$. Filtrovanie.
$^{lidar}_{odom}T \leftarrow {}^{lidar}_{base_link}T \; {}^{base_link}_{odom}T$
$^{map}P^0 \leftarrow {}^{lidar}P^0 \; {}^{lidar}_{odom}T \; {}^{odom}_{map}T_{i-1}$. Transformácia merania.
$T_{ci} \leftarrow \text{icp}(^{map}P^0, {}^{map}M^0_{i-1})$
$^{map}M_i \leftarrow \text{updateMap}(T_{ci}, {}^{map}P^0, {}^{map}M^0_{i-1})$. Aktualizácia mapy
$^{map}M^0_i \leftarrow \text{datafilter}(^{map}M_i)$
$^{lidar}_{map}T_i \leftarrow T_{ci} \; {}^{lidar}_{map}T_{i-1}$
$^{odom}_{map}T_i \leftarrow {}^{lidar}_{map}T_i \; {}^{lidar}_{odom}T^{-1}$
$^{base_link}_{map}T_i \leftarrow {}^{base_link}_{odom}T \; {}^{odom}_{map}T$. Lokalizácia polohy
return $(^{map}M^0_i, {}^{base_link}_{map}T_i)$

Obrázok 4.2: Blokové schéma ICP SLAM.

Balí£ek norlab_icp_mapper filtruje bodový mrak mapy v súradnicovom systéme senzora. Kvôli tomu je mo^{o}né zachova´ si lokálnu mapu prostredia v okolí robota, ktorý sa pohybuje.

(a) : Robot v prostredí.

(b) : Robot po vykonaní pohybu.

(c) : Zarovnanie merania a mapy pomocou ICP.

Obrázok 4.4: Vyzualizácia krokov metódy ICP SLAM - na obrázku 4.3a sa robot nachádza v prostredí. ƒierne £iary popisujú kontúry zmapovaného prostredia. Súradnicové systémymapa odommajú v tomto momente rovnaký po£iatok. Na obrázku 4.3b robot vykoná pohyb. Fialové £iary popisujúmeranie zo senzora a £iernou £iarou je popísané odometrické meranie, medzi súradnicovým systémom odoma base_link . Na obrázku 4.3c jemeranie zarovnané domapya robot je v mapelokalizovaný. Ke¤º e bolo odometrické meranie nepresné, súradnicové systémy mapa odomuº nemajú rovnaký po£iatok. Posun súradnicového systému odomod súradnicového systémumapkompenzuje chybu odometrického merania.

4.4 Filtre a funkcie modulov ICP

V tejto kapitole budú popísané ltre bodových mrakov, chybové funkcie a outlier funkcie, ktoré boli pouºité v kon gurácií ICP SLAM v rámci praktickej £asti. Filtrovanie bodových mrakov je dôleºité, keªºe zníºenie po£tu bodov v bodových mrakoch výrazne napomáha rýchlosti ICP algoritmu [18].

K-D Tree ²truktúra

K-D strom je dátová ²truktúra slúºiaca na ukladanie viacrozmerných bodových mrakov v priestore. Vstupom je bodový mrak s k rozmermi. K-D strom je zloºený z uzlov, ktoré je moºné rozdeli´ na ©avý a pravý podstrom. Tieto podstromy môºu by´ rekurzívne delené rovnakým spôsobom. Kaºdý uzol v K-D strome popisuje k-rozmerný obd¨ºnik. Pre K-D strom a jeho podstromy, platia nasledujúce podmienky [19]:

- Ak ©avý podstrom nie je prázdny, tak hodnota v²etkých uzlov v ©avom podstrome nie je va£²ia ako hodnota rodi£ovského uzla.
- Ak pravý podstrom nie je prázdny, tak hodnota v²etkých uzlov v pravom podstrome je va£²ia ako hodnota rodi£ovského uzla.
- ‰avý a pravý podstrom je taktieº K-D strom.

K-D strom umoº¬uje rýchle h©adanie najbliº²ích susedných bodov, preto je táto dátová ²truktúra k©ú£ová v implementácií ICP a ltrov bodových mrakov.

4.4.1 DataPointsFilter

Vä£²ina ltrov slúºi na odstránenie neºiadúcich bodov z bodového mraku. Niektoré ltre pridávajú bodom dodato£nú informáciu. Dotato£nou informáciou môºu by´ vlastosti bodu v bodovom mraku nazývané deskriptory. Príkladom deskriptora môºe by´ odhadovaná hustota bodového mraku v okolí ur£itého bodu.

FixStepSamplingDataPointsFilter

Tento lter ponecháva vo výstupnom bodovom mraku kaºdý n_f -tý bod, kde n_f je nastavite©ný parameter. Tento lter je vhodný na ltráciu bodového mraku zo senzorov, ktorých meranie má ur£itý geometrický tvar.

BoundingBoxDataPointsFilter

V priestore je parametrami $b = [x_{min}; x_{max}; y_{min}; y_{max}; z_{min}; z_{max}]$ de novaný kváder. V závislosti na kon gurácií sú body vymazané, ak sa nachádzajú v danom kvádri alebo mimo neho. Tento lter je pouºívaný najmä na odstránenie bodov, ktoré vznikli nasnímaním kon²trukcie robota. Parametrami sú

tri dvojice súradníc pre kaºdú os v priestore, kde kaºdá dvojica vy£le¬uje minimálnu a maximálnu hodnotu súradnice v danej ose. Posledný parameter kon guruje vymazávanie bodov z kvádra, alebo z jeho vonkaj²ku.

RandomSamplingDataPointsFilter

Tento lter priradí kaºdému bodu parametrom p_r de novanú pravdepodobnos´ vymazania bodu z bodového mraku. Následne sú tieto body s danou pravdepodobnos´ou vymazané z bodového mraku.

MaxDensityDataPointsFilter

Na funkciu tohto ltra je potrebné, aby body obsahovali desktriptor hustoty. Vstupným parametrom je maximálna povolená hodnota danej hustoty. Ak je hustota v okolí bodu vä£²ia ako tento parameter, bod je z bodového mraku odstránený. Oblasti bodového mraku z vysokou hustotou obsahujú £asto prebyto£nú informáciu o prostredí. Funkcia ltra je zobrazená na obrázku 4.6.

Obrázok 4.5: Aplikovanie ltra MaxDensityDataPointsFilter - pri ltrovaní bola stanovená maximálna hustotou na hodnotu 50000 bodov na m^3. ƒervenou farbou je ozna£ený pôvodný bodový mrak a bielou farbou je ozna£ený bodový mrak po ltrovaní [20].

SurfaceNormalDataPointsFilter

Hlavnou funkciou tohto ltra je výpo£et normály vo£i povrchu. Parametrom knn je moºné ur£i´ po£et susedných bodov pouºitých k danému výpo£tu. Parametrom keep_densities je taktieº moºné aktivova´ výpo£et hustoty v okolí kaºdého bodu v bodovom mraku.

OrientNormalsDataPointsFilter

Susedné normály vo£i povrchu nemusia ma´ nutne rovnakú orientáciu. Parametrom towardCenter mô°eme ur£i´ obmedzujúcu podmienku, pod©a ktorej sú normály preorientované vo£i po£iatku súradnicového systému alebo opa£ne. Týmto je mo°né dosiahnu´ konzistentné normály na rovnakom povrchu.

DistanceLimitDataPointsFilter

Parametrom dist je mo°ne ur£i´ maximálnu vzdialenos´ bodov od po£iatku v bodovom mraku. Taktie° je mo°né ²peci kova´ ltrovanie v osiach súradnicového systému. Ak má bod vzdialenos´ od po£iatku vä£²iu ako maximálnu povolenú, je z bodového mraku odstránený. Filter je taktie° mo°né parametrizova´ na odstránenie bodov, ktoré maximálnu povolenú vzdialenos´ neprekra£ujú. Funkcia ltra je zobrazená na obrázku [20].

Obrázok 4.6: Aplikovanie ltra DistanceLimitDataPointsFilter - z bodového mraku boli odstránené v²etky body vzdialené od po£iatku na 1m. Bielou farbou je ozna£ený bodový mrak po ltrovaní [20].

CutAtDescriptorThresholdDataPointsFilter

V tomto ltri je mo°né odstráni´ body z bodového mraku, na základe hodnoty desktriptorov. V parametroch sa ²peci kuje meno desktriptora descriptor a maximálna $d_{t_{max}}$ alebo minimálna $d_{t_{min}}$ hodnota daného deskriptora. Ak je hranica prekro£ená, bod je z bodového mraku odstránený.

4.4.2 Matcher

Tento modul obsahuje funkciu KDTreeMatcher. Táto funkcia spracováva bodový mrak do dátovej ²truktúry KD-tree. Parametrami sa ²peci kuje po£et n_m najbli°²ích susedov a ich maximálna vzdialenos´ d_{max} od bodu, z ktorého sú daní susedia h©adaní. Funkcia poskytuje zoznam najbli°²ích susedov pre ka°dý bod z bodového mraku.

4.4.3 Outlier ltre

Outlier ltre sú funkcie, ktoré na základe vzdialeností spárovaných bodov z mnoºiny M produkujú váhy, ktoré zvy²ujú alebo zniºujú ich dopad v chybovej funkcii 4.3. Ke¤ºe ICP pouºíva metódu najmen²ích ²tvorcov, vplyv jednotlivých párov v chybovej funkcií narastá z ich vzdialenos´ou. Skupina outlierov s dostato£ne ve©kou vzdialenos´ou môºe preváºi´ vplyv správne asociovaných bodov v ICP algoritme [21].

RobustOutlierFilter

V oblasti ²tatistiky boli vytvorené metódy na minimalizovanie vplyvu outlierov na ICP algoritmus. M-estimator je estimátor zaloºený na metóde maximálnej vierohodnosti. M-estimátor je beºne pouºívaný na ltrovanie outlierov pri pouºívaní metódy najmen²ích ²tvorcov [22]. Jedným z týchto estimátorov je Chauchyho M-estimator. Cauchyho estimátor pridáva váhu w kaºdému páru (p; q) z mnoºiny M pod©a nasledujúceho vzorca

$$w = \frac{1}{1 + \frac{\text{error}(p;q)^2}{k^2}}; \tag{4.10}$$

kde k je parameter, ktorý slúºi na ladenie tejto funkcie.

SurfraceNormalOutlierFilter

Tento lter odstra¬uje vplyv bodov z bodového mraku merania, pri ktorých uhol medzi normálovým vektorom daného bodu a normálovým vektorom spárovaného bodu z referencie prekra£uje parametrom $_{sn}$ stanovenú hodnotu. Ke¤ºe outlier ltre stanovujú váhy daných dvojích bodov (p,q), pod odstranením vplyvu bodov je myslené stanovenie hodnoty váhy na nulu.

4.4.4 ErrorMinimizer

Funkcie v module ErrorMinimizer stanovujú tvar chybovej funkcie error (P; Q). Cie©om minimalizácie chybovej funkcie je vyrie²enie rovnice 4.4.

Chybová funkcia point-to-plane

Ve©kos´ chyby point-to-plane chybovej funkcii naprie£ celou mnoºinou M je de novaný ako

$$\text{error}(P;Q) = \sum_{k=1}^{K} w(p_k; q_k)\, k(p_k - q_k) \cdot n_k k; \tag{4.11}$$

kde K popisuje po£et asociovaných dvojíc v mnoºine M, n_k je normálový vektor v bode q_k a k:k ozna£uje L2 normu [23]. Chybovú funkciu je moºné normalizova´ pomocou estimátorov ve©kosti. Jednou z moºností je medián absolútnych odchyliek vzdialeností dvojíc v mnoºine M [16].

4.4.5 TransformationChecker

Ako u° bolo spomenuté, tento modul obsahuje funkcie na zastavenie ICP na základe zvoleného kritéria. FunkciaCounterTransformationChecker zastaví ICP algoritmus po prekro£ení parametrom i_{max} nastavite©ného maximálneho po£tu iterácií. DifferentialTransformationChecker zastaví algoritmus, ak rota£ný a transla£ný komponent transforma£nej matice ${}^{i+1}_{i}T$ nedosiahne parametrami ε_{min}, $\varepsilon_{t_{min}}$ stanovene minimá. Parametrami vieme nastaví´ po°adovanú presnos´ zosadenia merania a referencie . BoundTransformationChecker naopak zastaví ICP v prípade, °e rota£ný alebo transla£ný komponent transforma£nej matice ${}^{i+1}_{i}T$ presiahne parametrami ε_{max}, $\varepsilon_{t_{max}}$ stanovené maximum.

4.5 Filtrácia dynamických preká°ok v mape

Predpokladom práce bolo statické vonkaj²ie prostredie. Av²ak skú²obne jazdy boli vykonané v mestských parkoch, kde sa pohybovali ©udia, autá a iné dynamické objekty. Bez ltrovania by sa do mapy permanentne dostával ²um od pohybujúcich sa objektov. Balí£ek norlab_icp_mapper obsahuje funkcionalitu, ktorá ka°dému bodu v mape pridá desktriptor probDynamic, ktorý popisuje pravdepodobnos´ toho, °e daný bod je dynamický. Výpo£et tejto pravdepodobnosti je implementovaný pod©a metódy predstavenej v £lanku [24]. Na výpo£et tejto pravdepodobnosti je pou°itý princíp ray-tracingu. Ak je z perspektívy senzora bod merania identi kovaný za bodom mapy pravdepodobnos´ toho, °e bod je dynamický, sa zvý²i. Následne je mo°né odstráni´ body, pre ktoré táto pravdepodobnos´ prekro£ila stanovenú hranicu. Pre popísanie výpo£tu je potrebné de nova´ si nasledujúce premenné:

- p: bod z merania v súradnicovom systéme senzora,
- q: bod z referencie asociovaný k bodu p v súradnicovom systéme senzora,
- Dyn: binárna premenná, ktorá ozna£uje dynamický alebo statický stav bodu q,
- Odyn: binárna premenná, ktorá ozna£uje dynamický alebo statický stav bodu q v minulosti,
- U: binárna premenná, ozna£ujúca potrebu zmeny stavu bodu,
- : uhol medzi bodmi p, q de novaný ako $\arccos(\frac{p\,q}{\|p\|\,\|q\|})$,
- : uhol dopadu na q de novaný ako $\arccos(n\ \frac{q}{\|q\|})$, kde n je normála vo£i ploche v bode q,
- : vzdialenos´ medzi bodmi p q de novaná ako $\|p\ \ q\|$.

Mô°eme zapísa´:

$$P(Dyn\,j\;;\;;\;\;)\;/\;;\;\sum_{U\,Odyn}\frac{P(Odyn)P(Uj\;;\;\;)}{P(Dyn\,jOdyn)P(\;\;jDyn)} \tag{4.12}$$

Kde:

- $P(Odyn)$,: je bu¤ po£iato£ná pravdepodobnos´ alebo výsledok predo²lého výpo£tu,
- $$P(Uj\;;\;\;) = \begin{cases} \frac{\;}{\;_{max}}(1 - \frac{\;^2}{\;}) & ak\ kpk\ \ k\ \ qk \\ 0 & inak \end{cases} \tag{4.13}$$

  je pravdepodobnos´ zmeny stavu zalo^{o}ená na uhloch a ,
- $$P(Dyn\,jOdyn) = \begin{bmatrix} \;\; & \;\; \\ 1 & 1 \end{bmatrix} \tag{4.14}$$

  je matica, ktorá umo^o¬uje zmenu stavu bodu,
- $$P(\;\;jDyn) = \begin{cases} \max(O;\min(1;\;\;_d + kpk\;_a\;\;\;)) & ak\ je\ Dyn\ \ nepravdivé \\ 1 \quad P(\;\;jDyn = nepravdivé) & ak\ je\ Dyn\ \ pravdivé \end{cases} \tag{4.15}$$

  je pravdepodobnostné rozdelenie pozorovania.

Tento výpo£et je mo^{o}né parametrizova´. Parameter $_{max}$ ur£uje uhol, pod©a ktorého je vymedzený priestor. Body referencie z tohto priestoru sú následne asociované k bodu p. Tento uhol je zalo^{o}ený na rozptyle laserového lú£u. H¨bkový ²um merania bodu p je mo^{o}né zoh©adni´ parametrom e_d a parametrom e_a je popísaný proporcionálny h¨bkový ²um merania bodu p. Parameter popisuje pravdepodobnos´ toho, ^{o}e statický bod ostane statickým. Naopak parameter popisuje pravdepodobnos´ toho, ^{o}e dynamický bod ostane dynamickým. Posledným parametrom $_{max}$ je ozna£ená hranica pravdepodobnosti, po ktorej je bod permanentne pova^{o}ovaný za dynamický. Tento model pri výpo£te zoh©ad¬uje viacero meraní.

Kapitola 5

Mapovanie priechodnosti terénu

Aby sa mohla konkrétna robotická platforma pohybova´ v prostredí, je nutné analyzova´ priechodnos´ prostredia s oh©adom na konkrétnu robotickú platformu. Do úvahy je potrebné zakomponova´ dynamické a kinematické vlastnosti danej platformy. Najbeºnej²ím spôsobom je implementácia 2.5D mapy priechodnosti, kde kaºdá bunka 2D mrieºky obsahuje informáciu o priechodnosti danej bunky, poprípade cenu alebo pravdepodobnos´ priechodnosti danej bunky. Predtým ako môºe by´ priechodnos´ terénu vyhodnotená, musí by´ daný terén interpretovaný pomocou senzorov. Interpretácia terénu môºe by´ rôzna z oh©adom na zameranie rie²eného problému [25].

5.1 Proprioceptívná analýza priechodnosti

Analýza priechodnosti zaloºená na princípe proprioceptívnych senzorov, je uºito£ná v u£ení modelu, ktorý obsahuje zachytenú náro£nos´ pri pohybe terénom. Náro£nos´ je analyzovaná na základe senzorických meraní z proprioceptívnych senzorov ako napríklad vibrácie, nárazy, zmeny rýchlosti robota a podobne [26]. Problémom tejto analýzy je to, ºe robot musí vykona´ jazdu terénom, aby mohol posúdi´ jeho priechodnos´. To vyºaduje mimoriadne robustné roboty, ktoré sú schopné odola´ opotrebovaniu a po²kodeniu, ktoré je spôsobene zlyhaniami pri jazde ako napríklad nárazy robota. Z tohto dôvodu táto analýza nie je ve©mi beºná a uprednost¬uje sa analýza pomocou extroceptívnych senzorov. Extroceptívnu analýzu je moºné skombinova´ na pred£asný odhad priechodnosti, ktorý môºe by´ následne pri prejazde terénom vylep²ený proprioceptívnou analýzou.

5.2 Extroceptívna analýza priechodnosti

Extroceptívna analýza pouºíva extroceptívne senzory na mapovanie priechodnosti terénu. Spôsoby interpretácie senzorických meraní je moºné rozdeli´ na postupy zaloºené na interpretácií geometrických charakteristikách terénu alebo postupy ktoré analyzujú priechodnos´ na základe rysov a vzh©adu prostredia (v angli£tine Appearance-based) [27].

5.2.1 Geometricka analýza priechodnosti

Geometrická analýza priechodnosti posudzuje priechodnos´ terénu na základe extrahovaných geometrických vlastností z prostredia. Pod©a autorov £lánku [27] základné my²lienky geometrickej analýzy dobre popisujú nasledujúce práce. Autori v £lánku [28] vytvorili 2D mrieºkové mapy, kde kaºdá bunka obsahovala informáciu o maxime, minime, vý²ke, rozptylu a sklonu prostredia v tejto oblasti. Tieto vlastnosti boli vypo£ítane z bodového mraku rozdeleného do mrieºkovej mapy. Hodnoty týchto vlastnosti boli následné porovnávane so stanovenými limitami a ak boli limity prekro£ené, dané bunky boli vyhodnotené ako nepriechodné. V £lánku [29] bola priechodnos´ vyhodnocovaná na základe váhovej funckie, ktorá brala do úvahy vý²ku, sklon, drsnos´ a presnos´ mapy.

V prostredí je taktieº moºné identi kova´ základne tvary ako hrany, plochy a zhluky bodov. Na základe detekcií týchto tvarov je následne vyuºi´ algoritmy, ktoré odhadnú priechodnos´ prostredia na základe polohy daných tvarov a charakteristík robota. V práci [30] autori klasi kujú bodový mrak z LIDAR senzora na triedu rozptylu, ktorou sú reprezentované objekty ako tráva a vegetácia, na lineárnu triedu, ktorá popisuje objekty ako káble, alebo konáre stromov a na triedu plochy, ktorou sú popísané povrchy ako zem, kamene a podobne [27].

ƒastou komplikáciou je analýza priechodnosti v oblastiach, v ktorých nie sme schopný zachyti´ senzorické merania. Príkladom týchto oblastí sú rôzne svahy, medzery a klesajúce schody. V práci [31] bol tento problém rie²ený detekciou prekáºok rozdelených do nasledujúcich skupín: Pozitívne prekáºky boli detekované v prípade, ºe rozptyl bodov v danej ploche prekro£il stanovené limity. Ostré hrany boli detekované na základe porovnávaní vý²ky medzi susednými bunkami. Následne plochy dostato£nej ve©kosti, ktoré neobsahovali ºiadne body, boli vyhodnocované ako potenciálne negatívne prekáºky. Filtrovacím procesom bolo rozhodnuté o tom, £i sa jedná o negatívnu prekáºku alebo o tienenie pozitívnej prekáºky [27].

„al²ím spôsobom analýzy je porovnávanie 3D modelu prostredia s 3D modelom robota, ktorý sa má po danom prostredí pohybova´. V £lánku [32] autori rozdelili bodovú mapu na uzly a hrany. Robota modelovali ako kváder a jeho kolesá ako válce. Následne detekovali v²etky uzly v mape, v ktorých by mohol by´ robot poloºený bez kolízie. Tieto uzly boli vyhodnotené ako prejazdné a h©adanie cesty spo£ívalo v jazde po hranách, ktoré spájali prejazdné uzly.

5.2.2 Vzh©adová analýza priechodnosti

Vzh©adová analýza priechodnosti spadá pod oblas´ po£íta£ového videnia. Roboti ktorý pouºívajú túto analýzu zvy£ajne klasi kujú zvolené triedy terénu. Autori v £lánkoch [33] a [34] analyzovali priechodnos´ prostredia meraním drsnosti, sklonu, tvrdosti a medzier v danom prostredí. Tieto vlastnosti boli extrahované z obrázkov. Sklon prostredia odhadovala natrénovaná neuronová sie´, ktorá bola nau£ená reagova´ na pixely okolo horizontov. Medzery slúºili na detekciu útesov a roklín, na základe vzdialenosti medzi detekovanými hranami. Tvrdos´ bola pouºitá na klasi kovanie rôznych materiálov (piesok, ²trk a podobne) a podie©ala sa na analýze priechodnosti ako vlastnos´, ktorá popisuje potenciálne ²myky kolies. Na základe týchto vlastností bola nasledovne vyhodnotená priechodnos´ daného prostredia [27].

5.3 Hybrídna analýza priechodnosti

Hybridná analýza spo£íva v kombinácií metód z proprioceptívnej a extroceptívnej analýzy. Autori v £lánku [35] kombinujú merania z kamery a LIDAR senzora. Z obrázkov z kamery je vytvorená prvá mapa, pomocou extrahovania rysov prostredia. Prostredie je klasi kované na hladké prostredie, drsné prostredie, preká ºku a pozadie. Následne je analyzovaný bodový mrak mapy z LIDAR senzora. Z bodového mraku je vytvorená druhá mapa s informáciou o sklone, vý²ke a drsnosti prostredia. Tieto mapy sú následne zlú£ené do nálnej mapy priechodnosti, pomocou pouºitia Bayesového ltra.

V £lánku [36] autori skombinovali vzh©adovú a proprioceptívnu analýzu. Najprv sú pomocou RGB-D kamery extrahované geometrické vlastnosti prostredia (sklon, drsnos´) a dané prostredie je taktieº klasi kované do tried (vegetácia, vodné plochy). Na základe týchto vlastnosti je vytvorená mapa priechodnosti. Následne pomocou sondy, obsahujúcej senzor sily, je odhadovaná poddajnos´ preká ºok v prostredí. Ke¤ºe sondovanie je £asovo náro£né, prebieha len v predom identi kovaných oblastiach. Týmto spôsobom je moºné odhali´ ²um spôsobený vegetáciou a taktieº overi´ h¨bku vodných povrchov.

5.4 Navrhovaná analýza priechodnosti

Analýzu priechodnosti prostredia v tejto práci budem vykonáva´ na základe bodového mraku mapy z ICP SLAM, pomocou geometrickej analýzy. Tento bodový mrak bude najprv pred ltrovaný a rozdelený do buniek 2D mrieºky. V prvom kroku budú identi kované potenciálne priechodné bunky. Pre kaºdú bunku, ktorá obsahuje nejaké bodyp = [x; y; z] 2 C, bude vypo£ítaná priemerná vý²ka bodov h a ich rozptyl ^{2} od vý²ky h pod©a nasledujúcich vzorcov:

$$h = \frac{1}{K} \sum_{k=1}^{K} z_{p_k}; \tag{5.1}$$

$$^{2} = \frac{1}{K} \sum_{k=1}^{K} (z_{p_k} \quad h)^2; \tag{5.2}$$

kde K je po£et bodov v mno°ine C a z_{p_k} reprezentuje vý²ku bodu k z mno°iny C. Bunky, ktorých rozptyl ^{2} presiahol stanovený limit rozptylu $^{2}_{max}$, sú klasi kované ako nepriechodné. Pre bunky, ktorých rozptyl ^{2} je v rámci stanoveného limitu $^{2}_{max}$, bude vykonaný druhý krok. V tomto kroku bude porovnávaný vý²kový rozdiel h pre ka°dú susednú bunku vypo£ítaný ako

$$h \quad = j h_c \quad h_n j; \tag{5.3}$$

kde h_c je priemerná vý²ka práve vyhodnocovanej bunky a h_n je vý²ka susednej bunky. V prípade, °e hodnota h je vä£²ia ako stanovený limit h_{max}, bunka je vyhodnotená ako nepriechodná. Vý²kový limit h_{max} je ur£ený nasledujúcim vz´ahom

$$h_{max} = l \quad \tan(\quad) \tag{5.4}$$

kde l je d¨°ka hrany bunky a je maximálny uhol, pod ktorým mô°e robot jazdi´. Pseudokód popisujúci vyhodnotenie buniek v mape priechodnosti je popísaný v algoritme 3.

Algorithm 3 Analýza priechodnosti jednotlivých buniek

```
Require: map                                  . mrie°ka obsahujúca bunky
Require: h_max                                . vý²kový limit
Require: ²                                    . limit rozptylu
  for each cell in map do
      cell_h = 1/K Σ_{k=1}^{K} z_{p_k}
      cell_ ² = 1/K Σ_{k=1}^{K} (z_{p_k} h)²
      if cell_ ² > ²_max then
          assume cell is traversable
          for each neighbour around cell do
              h = j h_c h_n j                 . Výpo£et vý²kového rozdielu
              if h > h_max then
                  cell not traversable
                  break
              end if
          end for
      else
          cell not traversable
      end if
  end for
```

(a) : Bodový mrak reprezentujúci obrubník.

(b) : Bodový mrak reprezentujúci ²ikmú plochu.

Obrázok 5.1: Teoretická analýza priechodnosti - na obrázkoch je moºné pozorova´ rozdielne hodnoty priemernej vý²ky a rozptylu, pre dva druhy prekáºok. V tomto prípade je na základe rozptylu moºné rozhodnú´ o priechodnosti prekáºok rozdielne, ke¤ºe rozptyl prekáºky na obrázku 5.1 je takmer o polovicu men²í oproti prekáºke na obrázku 5.1a. Sivé vertikálne £iary predstavujú hranice jednotlivých buniek.

Porovnávaním vý²kových rozdielov susedných buniek sú v podstate vypo£ítané gradienty medzi bunkami. Vý²ky jednotlivých buniek sú vypo£ítané z bodov v oblasti ur£enej ve©kos´ou jednotlivých buniek v mrieºke. Týmto procesom sú z bodovej mapy extrahované informácie, ktoré sú priemerom hodnôt vý²ky bodov v ur£itej oblasti. To znamená, ºe dve bunky s rovnakou priemernou vý²kou, môºu ma´ v realite úplne iný geometrický tvar. Zavedením rozptylu je naopak extrahovaná informácia o podobnosti meraní v spomínanej oblasti. V ideálnom prípade by bol rozptyl bodov vypo£ítaný vo£i ploche tvorenej bodmi v ur£itom okolí bunky. V tejto práci som sa tento výpo£et rozhodol zjednodu²i´. Rozptyl je po£ítaný vo£i horizontálnej ploche, ke¤ºe s robotom neplánujem jazdi´ pod ve©kým uhlom. Kombináciou gradientov a rozptylov buniek je moºné vytvori´ lep²í odhad priechodnosti terénu. V prípade, ºe gradienty medzi susednými bunkami sú v rámci limitu h_{max} ale rozptyl jednotlivých buniek je pomerne ve©ký, je moºné usúdi´, ºe pri výpo£te priemernej vý²ky jednotlivých buniek do²lo k vyhladeniu terénu z bodovej mapy. Tento prípad môºe nasta´ na ostrých prekáºkach, ako sú napríklad obrubníky, v prípade, kedy hrana obrubníku prechádza stredom bunky. Tento prípad je teoreticky vizualizovaný na obrázku 5.1. Naopak, na základe príli² ve©kých gradientov medzi susednými bunkami, ktorých rozptyl je pod limitom ^{2}, je taktieº moºné rozhodnú´ o nepriechodnosti bunky. Príkladom by bol obrubník ktorého hrana by sa nachádzala na hrane dvoch susedných buniek.

Nevýhodou výpo£tu priemernej vý²ky buniek je strata podrobnej²ej informácie o geometrií terénu v danej oblasti. Výhodou výpo£tu priemernej vý²ky je potlá£anie ²umu v bodovom mraku mapy. Naopak, výpo£et rozptylu je na tento ²um citlivý. Predpokladom pre pouºitie rozptylu na analýzu priechodnosti je dostato£né súvislá bodová mapa, ktorá reprezentuje plochy v teréne z dostato£nou presnos´ou.

Ako senzor taktieº pouºijem polohu robota v bodovej mape. Pokia© sa bude bunka bez bodov nachádza´ v ur£itej blízkosti robota, bude vyhodnotená ako nepriechodná. My²lienkou je takto vyradi´ rôzne medzery a diery v prostredí, ktoré LIDAR senzor nedokázal zachyti´ pri jazde k danej prekáºke. Priechodnos´ týchto buniek sa opä´ vyhodnotí v prípade pridania bodov do danej bunky.

Metóda je navrhovaná pre robota HUSKY A200. Tento robot sa po prostredí pohybuje pomocou kolies. K tejto vlastnosti sú prispôsobené hranice rozptylu ^{2} a vý²kového limitu h_{max}. S ur£itým ladením hraníc by bolo moºné pouºi´ túto metódu pre pásové roboty. Obecne sú pásové roboty schopné prekona´ náro£nej²í terén. Metóda nebude vhodná pre roboty, ktoré sa prostredím pohybujú pomocou ramien. Klasi kácia nepriechodných buniek, kon gurovaná pre robota HUSKY A200, by pre pásové a chodiace roboty produkovala príli² ve©a falo²ných negatívov, teda nepriechodných buniek ktoré sú v realite pre tieto roboty prejazdné.

Kapitola 6

Implementácia

6.1 Robot operating system

Robot operating system (ROS) je súbor nástrojov a kniºníc, pouºívaných vo vývoji robotiky. ROS je vo©ne dostupný a vznikol na Standfordskej univerzite v roku 2007. Pre rôzne robotické aplikácie je potreba zabezpe£i´ funkciu senzorov a komunikáciu medzi senzormi, aktuátormi a rôzným softvérom, ktorý je na danú aplikáciu vyvinutý. Vo výskume teda £asto nastávalo nezávisle implementovanie algoritmov s rovnakou funkciou, napríklad algoritmus na komunikáciu z ur£itým senzorom. Hlavnou my²lienkou ROS je poskytnutie prostredia, v ktorom je moºné dané algoritmy zdie©a´ formou balí£kov. Tým pádom je moºné venova´ viac £asu na budovanie inteligentných robotických aplikácií.

6.1.1 Architektúra ROS

Implementácia algoritmov v ROS sa snaºí o £o najvä£²iu modularitu. Základným stavebným prvkom je takzvaný uzol. Uzol je program, ktorý bol navrhnutý na ur£itú úlohu, napríklad na komunikácia so senzorom. ROS ponúka ve©a moºností komunikácie medzi uzlami, no pre túto prácu sú dôleºité komunika£né kanály a správy, posielané po daných kanáloch. Kaºdý kanál má ur£ený formát správy, ktorý môºe by´ komunikovaný v rámci daného kanálu. Uzly sa vedia na kanál pripoji´ pomocou tried publisher a subscriber. Trieda publisher zabezpe£uje posielanie informácií daným kanálom a trieda subscriber naopak odoberá informácie, posielané publsiherom z daného kanála. Pomocou komunika£ných kanálov je teda moºné jednoducho spojazdni´ komunikáciu medzi uzlami.

6.1.2 Balí£ky

V rámci ROS sú implementované algoritmy zorganizované do balí£kov. Obsahom balí£kov môºe by´ implementácia uzlov, kniºníc a podobne. Výhodou tohto systému je jednoduchá distribúcia základných algoritmov. Pomocou balí£kov je jednoduché zdie©a´ prácu v rámci vedeckých projektov.

6.1.3 Simulátor Gazebo

V rámci ROS je mo^{o}ná simulácia robotov pomocou simulátora Gazebo. Gazebo je schopné simulova´ interakciu objektov na základe fyzikálnych zákonov. V tomto simulátore je taktie^{o} mo^{o}né simulova´ rôzne senzory s nastavite©nými parametrami, ako je napríklad ²um. Taktie^{o} je mo^{o}né vytvori´ rôzne vonkaj²ie alebo vnútorne prostredia, v ktorých je mo^{o}né testova´ algoritmy spojené s robotmi.

6.2 Kon gurácia robota HUSKY A200

V tejto práci bol pou^{o}itý robot HUSKY A200 v kombinácií s LIDAR senzorom Velodyne VLP-16 a GNSS senzorom Trimble BX982. Robot a senzory boli simulované v simulátore Gazebo a uvedené do prevádzky v reálnom prostredí.

6.2.1 HUSKY A200

Husky A200 je modi kovate©né bezpilotné pozemné vozidlo (z anglického UGV). Toto vozidlo bolo dizajnované na pou^{o}itie v náro£nom vonkaj²om teréne. V základnej kon gurácií, zobrazenej na obrázku 6.1, je vozidlo vybavené po£íta£om, batériou a ²tvorkolesovým pohonom. S vozidlom je taktie^{o} mo^{o}né komunikova´ v rámci uzlov implementovaných v ROS. Pre túto prácu sú dôle^{o}ité uzly /husky/husky_kinematic a /basic_basic_odom . Prvý uzol poskytuje rozhranie medzi ROS a po£íta£om vozidla Husky. Taktie^{o} slú^{o}i na príjem správ na kanále /cmd_vel , pod©a ktorých je ovládaný pohon vozidla. Obsahom týchto správ sú po^{o}adované lineárne a uhlové rýchlosti vozidla. V rámci vozidla je implementovaný regulátor, ktorý tieto správy interpretuje ako ak£né zásahy na pohony kolies. Uzol /basic_basic_odom poskytuje informáciu o odometrii, vypo£ítanú z meraní rota£ných enkodérov. V rámci mojej kon gurácie bola odometria po£ítaná pomocou diferenciálneho modelu vozidla [37]. Lineárnu rýchlos´ v a uhlovú rýchlos´ ! je mo^{o}né získa´ pomocou nasledujúcich vz´ahov

$$v = \frac{v_r + v_l}{2}; \qquad (6.1) \qquad\qquad ! = \frac{v_r \quad v_l}{w}; \qquad (6.2)$$

kde v_r a v_l sú rýchlosti kolies na pravej a ©avej strane vozidla a w je rozchod kolies [38]. Následne je aktuálna poloha v iterácii i vypo£ítaná pomocou nasledujúcich vz´ahov

$$x_i = x_{i \quad 1} + T_s \quad v \quad \cos(! \; {}_{i \quad 1}); \qquad (6.3)$$

$$y_i = y_{i \quad 1} + T_s \quad v \quad \sin(! \; {}_{i \quad 1}); \qquad (6.4)$$

$$! \; {}_i = ! \; {}_{i \quad 1} + T_s \quad !; \qquad (6.5)$$

kde vektor $p = [x; y; ! \;]$ popisuje polohu a orientáciu vozidla a znakom T_s je ozna£ená doba od poslednej aktualizácie polohy.