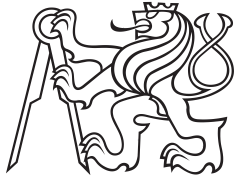


Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra kybernetiky

Lokalizace v prostředí stožáru vysokého napětí

Jan Mandík

Vedoucí: RNDr. Petr Štěpán, Ph.D.

Studijní program: Otevřená informatika

Specializace: Základy umělé inteligence a počítačových věd

Květen 2022

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Mandík** Jméno: **Jan** Osobní číslo: **491961**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra kybernetiky**
Studijní program: **Otevřená informatika**
Specializace: **Základy umělé inteligence a počítačových věd**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Lokalizace v prostředí stožáru vysokého napětí

Název bakalářské práce anglicky:

Localization near High Voltage Pylons

Pokyny pro vypracování:

- 1) Seznamte se s lidary Ouster s 16 a více rovinami snímání.
- 2) Nasnímejte a analyzujte data z okolí stožáru vysokého napětí.
- 3) Vytvořte algoritmus, který v datech ze senzoru bude hledat rovné konstrukce stožáru.
- 4) Vytvořený algoritmus zakomponujte do programu, který na základě dat z lidarů bude počítat 2D polohu dronu v okolí stožáru vysokého napětí. Výsledný program podrobně analyzujte z hlediska výpočetního času, přesnosti lokalizace v závislosti na vzdálenosti dronu od stožáru.

Seznam doporučené literatury:

- [1] Besl, Paul & McKay, H.D.. (1992). A method for registration of 3-D shapes. IEEE Trans Pattern Anal Mach Intell. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 14. 239-256. 10.1109/34.121791.
- [2] P. Li, R. Wang, Y. Wang and W. Tao, "Evaluation of the ICP Algorithm in 3D Point Cloud Registration," in IEEE Access, vol. 8, pp. 68030-68048, 2020, doi:10.1109/ACCESS.2020.2986470.
- [3] A. Diosi and L. Kleeman, "Laser scan matching in polar coordinates with application to SLAM," 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005, pp. 3317-3322, doi: 10.1109/IROS.2005.1545181.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

RNDr. Petr Štěpán, Ph.D. Multirobotické systémy FEL

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

Datum zadání bakalářské práce: **17.01.2022**

Termín odevzdání bakalářské práce: **20.05.2022**

Platnost zadání bakalářské práce: **30.09.2023**

RNDr. Petr Štěpán, Ph.D.
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

Poděkování

Tímto bych chtěl poděkovat vedoucímu mé bakalářské práce RNDr. Petru Štěpánovi, Ph.D. za cenné rady a náměty a za všechny čas, který mi věnoval. Dále bych chtěl poděkovat své rodině za podporu během celého studia a při tvorbě této práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, dne 20. května 2022

Abstrakt

Použití dronů se jeví jako vhodný nástroj pro zrychlení, zlevnění a zautomatizování pravidelných inspekcí stožárů vysokého napětí. Správné fungování lokalizačního systému GPS je však narušeno v blízkosti vedení vysokého napětí. Tato bakalářská práce se zabývá využitím LiDARu, jakožto nástroje, jehož fungování není v okolí vedení vysokého napětí nijak narušeno, pro lokalizaci v prostředí stožáru vysokého napětí. V práci je představen algoritmus, který pomocí algoritmu RANSAC detekuje konstrukci stožáru vysokého napětí v LiDARem nasnímaném mračnu bodů. Práce dále popisuje zakomponování tohoto algoritmu do programu, provádějícího lokalizaci ve 2D pomocí algoritmu ICP na základě detekované konstrukce stožáru. Měření, provedená na datech, nasnímaných pomocí LiDARu typu OS0 od společnosti Ouster, ukazují, že lokalizace pomocí uvedených algoritmů je možná, a to až do vzdálenosti okolo dvaceti metrů od stožáru vysokého napětí.

Klíčová slova: lokalizace, mračno bodů, LiDAR, RANSAC, ICP, stožár vysokého napětí, ROS

Vedoucí: RNDr. Petr Štěpán, Ph.D.
Multirobotické systémy FEL,
Katedra kybernetiky,
Resslova 307/9,
Praha

Abstract

The use of unmanned aerial vehicles appears to be a suitable solution to speed up, cheapen and automate regular inspections of high voltage pylons. However, the proper functioning of the GPS system is impaired near high voltage power lines. This bachelor thesis deals with the use of LiDAR, as a tool whose functioning is not disturbed near high voltage power lines, for localization in the high voltage pylon environment. In this thesis, an algorithm is presented that uses RANSAC algorithm to detect the structure of the high voltage pylon in point cloud, obtained by the LiDAR. This thesis further describes the incorporation of this algorithm into a program, that performs 2D localization, based on the detected high voltage pylon structure using the ICP algorithm. The measurements, performed on data obtained by Ouster's OS0 type LiDAR, show that localization with use of presented algorithms is possible, up to a distance of about 20 meters from the high voltage pylon.

Keywords: localization, point cloud, LiDAR, RANSAC, ICP, high voltage pylon, ROS

Title translation: Localization near High Voltage Pylons

Obsah

1 Úvod	1		
1.1 Projekt AERIAL-CORE	3		
1.2 Použití GPS v prostředí vedení vysokého napětí	4		
1.3 Cíle a struktura práce	6		
2 LiDAR	7		
2.1 Ouster LiDAR	9		
2.2 Parametry a nastavení použitého zařízení	11		
3 Ukázka a analýza nasnímaných dat	13		
4 Identifikace stožáru vysokého napětí	17		
4.1 Spuštění programu	17		
4.2 Algoritmus RANSAC	20		
4.3 Využití algoritmu pro identifikaci stožáru vysokého napětí	21		
4.3.1 Převod dat do vhodného formátu a filtrování bodů na základě jejich polohy	22		
4.3.2 Detekce krajiny pod stožárem	24		
4.3.3 Detekce svislých částí stožáru	26		
4.3.4 Úprava nalezených bodů pro další část programu	29		
5 Lokalizace na základě zpracovaných dat ze senzoru	33		
5.1 Spuštění programu	33		
5.2 Algoritmus ICP	34		
5.2.1 Varianty algoritmu ICP	36		
5.2.2 Faktory ovlivňující činnost algoritmu	36		
5.3 Lokalizace pomocí ICP algoritmu	37		
6 Analýza funkčnosti programu	43		
6.1 Rekonstrukce trasy postupného vzdalování od stožáru vysokého napětí	45		
7 Závěr	47		
Literatura	49		
A Struktura přiloženého programu	53		

Obrázky

1.1 Celosvětová spotřeba elektrické energie mezi lety 1980 a 2019 [2]	1	4.3 Data před a po použití detekce krajiny	25
1.2 Ilustrace cílů a použitých konceptů projektu AERIAL-CORE [3]	4	4.4 Ukázka dat po tomto kroku filtrace	26
2.1 Ilustrace způsobu fungování LiDARu [31]	8	4.5 Ukázka detekce stožárů vysokého napětí	28
2.2 Ukázka senzorů ze stránek společnosti Ouster [24]	9	4.6 Označení kvadrantů	30
2.3 Sférické souřadnice [4]	10	4.7 Ukázka dat před převedením do svislé polohy (červeně) a po něm (zeleně)	30
3.1 Ukázka dat s přiblíženými stožáry	13	5.1 Vizualizace ICP ve 2D [7]	35
3.2 Ukázka dat s celým záběrem LiDARu	14	5.2 Vizualizace ICP ve 3D [33]	35
3.3 Přiblížený záběr na stožár	15	5.3 Vizualizace určené polohy při pohybu okolo stožáru vysokého napětí	40
3.4 Přiblížený záběr na stožár z úhlu, ve kterém vidíme jen jeho přední stranu	15	6.1 Vizualizace rekonstruované trasy pohybu	46
4.1 Ukázka fungování RANSAC na příkladu proložení přímky ve 2D. [22]	21	6.2 Vizualizace rekonstruované trasy pohybu včetně špatných určení polohy ve větší vzdálenosti	46
4.2 Data před a po použití tohoto kroku filtrace	23		

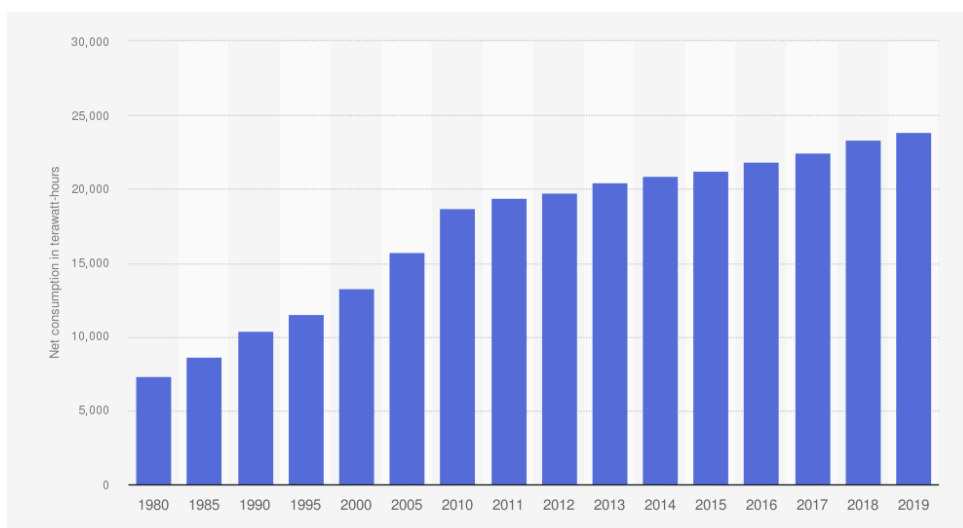
Tabulky

6.1 Statistiky fungování identifikace v různých vzdálenostech	44
6.2 Statistiky fungování lokalizace v různých vzdálenostech	45

Kapitola 1

Úvod

Spotřeba elektrické energie dlouhodobě celosvětově stoupá, a to především vlivem nárůstu populace, ekonomického růstu a v neposlední řadě také vlivem klimatických změn. Očekává se, že právě vlivem ekonomického růstu se do roku 2050 zvýší celosvětová spotřeba energie 1.5 až 2.8 krát. Díky dalším faktorům, jakými jsou například výše zmiňované klimatické změny, které způsobují větší spotřebu elektrické energie, a to jak v extrémně teplých, tak v extrémně chladných obdobích, se pak očekává další nárůst spotřeby energie až o dalších 25 - 58 %. [29] Následující graf ukazuje celosvětový nárůst spotřeby elektrické energie mezi lety 1980 a 2019. [2]



Obrázek 1.1: Celosvětová spotřeba elektrické energie mezi lety 1980 a 2019 [2]

Nárůst spotřeby energie tak vytváří stále větší nároky na vedení vysokého napětí. Společnosti, spravující elektrické distribuční sítě, se proto snaží investovat peníze do různých způsobů, jak v co největší míře předejít problémům s vedením vysokého napětí, způsobenými jak přírodními vlivy tak také neefektivní, špatnou nebo nedůslednou údržbou. U problémů, způsobených přírodními vlivy, jde například o výsadbu zeleně a jiných přírodních bariér za účelem snížení síly větru, která působí přímo na vedení elektrického proudu. [5] Větší pozornost i investice jsou však směřovány ke druhé zmíněné příčině poškození, tedy neefektivní, špatné nebo nedůsledné údržbě.

Nejčastější strategií při předcházení tomuto problému jsou letecké kontroly stožárů elektrického vedení, prováděné v pravidelných intervalech. V současnosti jsou k těmto kontrolám využívány helikoptéry s lidskou posádkou. Stroj je pro provedení kontroly a získání dat vybaven velkým množstvím senzorů a různých kamer. Celou operaci pak musí provádět zkušená a proškolená posádka, jejímž úkolem je pomocí senzorů nasbírat potřebná data. Nasbíraná data jsou následně postupně po jednotlivých kontrolovaných komponentech vedení ručně prohlížena s cílem najít poruchy a místa potenciálního vzniku problémů, jakými je například poškozený vodič či jeho uchycení nebo poškození samotné konstrukce stožáru. Tato strategie má však četné nevýhody, mezi které patří velká časová náročnost, možnost lidské chyby nebo cena, která se pohybuje okolo \$1500 tedy přibližně 30 000Kč za jednu hodinu letu.[26] Jde také o velmi nebezpečnou práci, neboť se helikoptéra musí pohybovat v bezprostřední blízkosti vedení vysokého napětí a k tomu, aby došlo k nehodě, tak stačí jen drobné zaváhání.

Není tedy divu, že se objevily výzkumy a projekty, jejichž cílem je pravidelné inspekce vedení vysokého napětí zrychlit, zlevnit a především zautomatizovat. Jako vhodná řešení pro tento problém se ukázaly především dva různé způsoby. Tím prvním je použití takzvaných ROW robotů (z anglického Rolling On Wire), tedy robotů, kteří se pohybují přímo po drátu vysokého napětí, odkud také provádějí snímání a kontrolu vedení. Druhým slibným způsobem je pak použití bezpilotních letadel neboli UAV (z anglického Unmanned Aerial Vehicle), známých také jako drony. Bepilotní letadlo je letadlo bez posádky, které může být řízeno na dálku, nebo létat samostatně za pomoci předprogramovaných letových plánů či pomocí autonomních systémů. Bepilotní letadla jsou schopna vykonávat inspekce vedení vysokého napětí různými způsoby a na různých úrovních, podle typu křídel a způsobu pohybu. Letadla s pevnými křídly, která se musí pro udržení ve vzduchu neustále pohybovat, se hodí pro kontrolu vedení ve větším měřítku a na delší vzdálenost. Bepilotní letadla s vrtulemi, která jsou schopna ve vzduchu setrvat na jednom místě, se naopak hodí pro detailní inspekci stožárů vysokého napětí a jejich komponentů. [26][27] Právě využitím obou typů bezpilotních letadel pro kontrolu a údržbu lineárních struktur, jakou je i vedení vysokého napětí, se zabývá také evropský projekt AERIAL-CORE. [3]

1.1 Projekt AERIAL-CORE

Hlavním cílem projektu je vývoj základních technologických modulů a integrovaného kognitivního leteckého robotického systému s doposud nevídaným rozsahem schopností a s důrazem na bezpečnost při interakci, a to jak s lidmi tak s dalšími bezpilotními letadly (Aerial Co-Workers). Hlavní aplikací systému pak bude právě kontrola a údržba velkých lineárních infrastruktur, jakými je vedení vysokého napětí. Systém plánuje integrovat různé bezpilotní letecké roboty s rozdílným zaměřením a tomu odpovídajícími vlastnostmi. Výsledný systém by pak měl splňovat následující vytyčené cílové požadavky:

- Schopnost práce na dlouhých vzdálenostech se zachováním velmi přesné a důkladné kontroly po celé délce
- Schopnost přímo provádět údržbové činnosti, a to i ty, které vyžadují, aby bezpilotní letadla přímo manipulovala s nástroji a používala sílu
- Schopnost bezpečně a efektivně spolupracovat s lidmi při práci na kontrole a údržbě

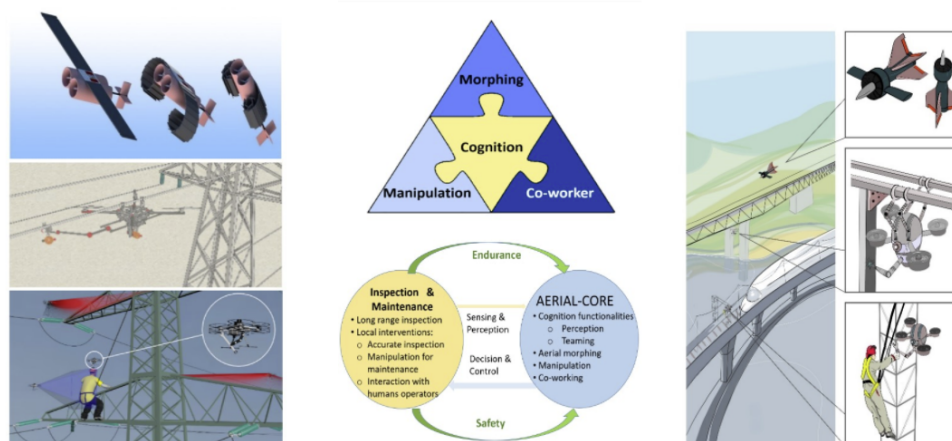
Moduly AERIAL-CORE budou implementovat kognitivní schopnosti, díky kterým budou bezpilotní letadla schopna vnímání okolí, týmové práce a morfování (přeměňování) přímo za letu ve vzduchu, které umožní lepší výdrž na dlouhé vzdálenosti, přesnou navigaci pro možnost uchycení na objektu při detailnějším zkoumání jednoho místa, manipulaci s objekty vyžadující sílu a při spolupráci s lidskými pracovníky.[3]

Na základě požadavků byly vytyčeny následující konkrétní technické cíle projektu:

1. Vytvoření kognitivních funkcí pro bezpilotní letecké roboty, včetně funkcí pro rychlou reaktivitu, on-line plánování, týmovou spolupráci, možnosti učení a funkcí pro pokročilé vnímání okolí, založených na propojování vstupů z různých moderních senzorů jako je kamera pro záznam událostí (z anglického event camera, tedy kamera, která zaznamenává změny v intenzitě světla na jednotlivých pixelech).
2. Vytvoření vzdušných platforem se schopnostmi přeměn, jako jsou například přeměny mezi různými letovými módy, či přeměn mezi létáním a pohybem po zemi. Účelem těchto možných přeměn je ušetření energie a možnost detailnějších a přesnějších kontrol jednotlivých komponentů.

3. Schopnost manipulovat s předměty, a to včetně manipulace přímo za letu či při případném uchycení na objektu, za účelem větší přesnosti či za účelem vyvinutí větší síly.
4. Vytvoření bezpilotních leteckých spolupracovníků se schopností fyzicky interagovat s lidskými pracovníky.
5. Vytvoření uceleného systému bezpilotních leteckých robotů schopných spolehlivé kontroly a údržby velkých infrastruktur.

Pro lepší představu o projektu se můžeme podívat na následující ilustraci přímo ze stránek projektu:



Obrázek 1.2: Ilustrace cílů a použitých konceptů projektu AERIAL-CORE [3]

Počítá se, že skupina bezpilotních helikoptér, vybavených všemi potřebnými senzory pro vlastní provoz i pro nasbírání vizuálních dat, bude operovat ve známém prostředí reprezentovaném mapou nebo modelem. Nasbíraná data pak mohou být odesílána a vzdáleně vyhodnocována za účelem rychlejšího nalezení míst s poruchou, vadou či poškozením.

1.2 Použití GPS v prostředí vedení vysokého napětí

Jedním z nejrozšířenějších způsobů lokalizace, tedy určení umístění objektu v prostoru, je v dnešní době systém GPS. GPS (z anglického Global Positioning System, tedy globální polohový systém) je satelitní radionavigační systém, vyvinutý ministerstvem obrany Spojených států amerických za účelem možné navigace a určení přesné polohy po celém světě. Primárně byl systém určen jen pro armádu USA, ale vzhledem k tomu, že pro použití systému je nutný jen

přijímač schopný jednoduchých výpočtů, což je v dnešní době cenově poměrně dostupné zařízení, si systém rychle našel velké množství uživatelů a dávno již nejde jen o armádní projekt. Funkci systému zajišťuje 28 satelitů umístěných na oběžné dráze. Tyto satelity jsou vybaveny velmi přesnými atomovými hodinami s přesností okolo 3 ns. Přesná poloha přijímače je pak určena pomocí času, který zabere rádiové vlně cesta od vysílače, umístěného v satelitu na oběžné dráze, k přijímači na zemském povrchu. K přesnému určení polohy je pak potřeba přijímat takový signál minimálně od tří satelitů najednou. V praxi se při určování polohy přijímače používá ještě signál od čtvrtého satelitu, který slouží k určení odchylky času přijímače oproti atomovým hodinám v satelitech. [28]

Činnost navigačního systému však může být ovlivněna elektromagnetickým polem vznikajícím v okolí vedení vysokého napětí. Elektromagnetické pole je generováno vlivem elektrického proudu ve vodičích vysokého napětí. Jeho elektrická složka je generována přítomností elektricky nabitých částic, jeho síla závisí na napětí v konkrétní rozvodové síti a na vzdálenosti vodiče od místa pozorování. Magnetické pole je generováno vlivem proudu protékajícího vodičem. Jeho intenzita tak závisí na proudu v síti a na vzdálenosti vodiče od místa pozorování. Obecně platí, že intenzita elektromagnetického pole se s přibývajícím vzdáleností mezi vodičem a místem pozorování snižuje. Podle výzkumů může dojít k problematické interferenci, a tedy možnému porušení signálu mezi elektromagnetickým polem a signálem GPS při naplnění následujících podmínek [13]:

1. Napětí ve vodiči je vyšší než 230 kV
2. Používaná frekvence je nižší než 30 MHz
3. Vzdálenost mezi vodičem a přijímačem je velmi malá (méně než 100 metrů)

Přestože se výzkumy v závěrech ohledně možného narušení signálu GPS v okolí vedení vysokého napětí liší, vyplývá z nich, že vliv vedení vysokého napětí se často liší podle frekvence užívané pro komunikaci. Výzkumy také jako bod nejbližší vedení vysokého napětí uvažují umístění přijímače na zem pod sloupy vysokého napětí. V našem případě, kdy předpokládáme pohyb bezpilotních helikoptér v bezprostřední blízkosti vodičů, bude intenzita elektromagnetického pole daleko vyšší a může tak mít na komunikaci daleko větší vliv. Navíc při pohybu v bezprostřední blízkosti vedení a jeho komponentů stačí jen drobná výchylna v signálu a může dojít ke kolizi, jejíž následky mohou být fatální.

Dalším nástrojem, důležitým pro správný let dronu, je kompas. Ten slouží pro správné určení orientace dronu v prostoru. V normálním prostředí se střelka kompasu natočí ve směru magnetických indukčních čar zemského magnetického pole. Na základě tohoto směru se pak odvodí natočení dronu. V okolí vodiče, kterým prochází elektrický proud, je však magnetické pole, které ovlivní výchylku střelky magnetu a odvozené natočení dronu tak není správné.[19]

1.3 Cíle a struktura práce

Z výše uvedených poznatků vyplývá, že pro přesnou lokalizaci v okolí vedení vysokého napětí je potřeba zvolit jiný systém, který dokáže určit polohu v prostoru pouze na základě pozorování okolí, bez nutnosti komunikace s jiným zařízením. Postačí nám určení polohy ve 2D, neboť zařízení, používané pro určení výšky, ve které se dron nachází nad povrchem, je založeno na principu laserového měření vzdálenosti [18] a není tedy v blízkosti vedení vysokého napětí nijak ovlivněno. Jediným výrazným objektem v okolním prostředí, vhodným pro určování polohy, je v našem případě právě stožár vedení vysokého napětí. Jako vhodný nástroj pro získání dat z okolního prostředí se pak jeví použití LiDARu.

Předložená práce se zabývá problematikou lokalizace v prostředí stožáru vysokého napětí. Hlavním cílem práce je navržení algoritmu, který bude schopen v datech ze senzoru hledat rovné části konstrukce stožáru. Dalším cílem je zakomponování tohoto algoritmu do programu, který bude ověřovat možnosti určení polohy dronu ve 2D na základě konstrukcí stožáru, nalezených v datech z LiDARu.

Práce je členěna do šesti hlavních částí. První část byla věnována úvodu do problematiky a motivaci k řešení cílů této práce. Druhá část obsahuje popis LiDARu a vlastností použitého senzoru. Třetí část je věnována ukázce a analýze nasnímaných dat. Čtvrtá část se zabývá navrženým algoritmem pro identifikaci konstrukce stožáru v nasnímaných datech. Obsahem páté části je zakomponování navrženého algoritmu do programu, který ověřuje možnosti určení polohy dronu ve 2D. Poslední část je věnována analýze výsledků programu na reálných datech.

Kapitola 2

LiDAR

Název LiDAR vznikl z anglického Light Detection And Ranging nebo také Laser Imaging Detection And Ranging. Jde o metodu měření vzdáleností okolních předmětů od senzoru, založenou na výpočtu doby šíření pulsu laserového paprsku odraženého od předmětu. Z naměřeného času se zjednodušeně získá vzdálenost pomocí následujícího vzorce:

$$vzdálenost = \frac{t * c}{2}$$

kde t je čas naměřený od vyslání paprsku, do zachycení jeho odrazu, c je rychlost světla, jehož přesná hodnota se uvádí jako

$$c = 299\,792\,458 \frac{m}{s}$$

[14].

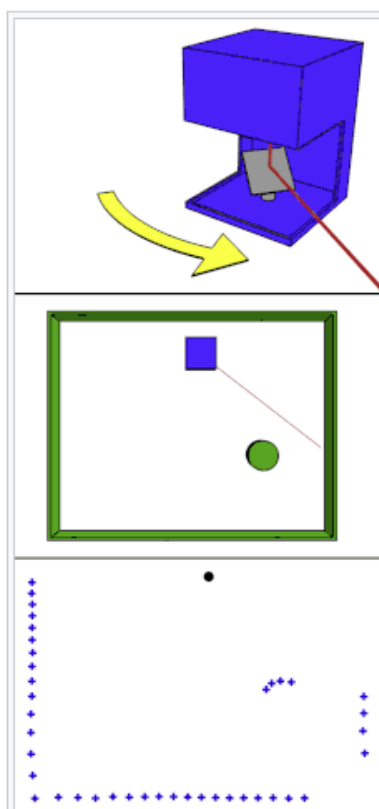
LiDARy mají poměrně široké využití, mezi ta nejčastější patří měření vzdáleností, mapování terénu či zkoumání dna oceánů [16]. Hlavními součástkami LiDARU jsou:

1. **Laser** - pro použití v LiDARu jsou nejčastěji používány lasery o vlnové délce 600 - 1000 nm. Hranice možných vlnových délek jsou limitovány tak, aby nebyly lasery při použití nebezpečné pro osoby v okolí. (Hlavním nebezpečím je poškození rohovky či čočky oka vlivem zahřátí při absorpci intenzivního laserového paprsku. Lasery jsou pro tento případ opatřeny pojistkou, která zamezuje použití laseru pokud by byl nastaven na nebezpečné vlnové délky.)

2. LiDAR

2. **Zrcadla** - umožňují namíření paprsku laseru do různých směrů. Díky různým kombinacím otáčení a nahlýbání systému zrcadel je možné mít v celém přístroji jen jeden zdroj laserového paprsku a ten postupně namířit do různých směrů.
3. **Čidla** - detekují energii, předanou senzoru při nasvícení odraženým paprskem. Podle naměřeného času je pak vypočtena vzdálenost od nasvíceného objektu.

Fungování LiDARu znázorňuje následující ilustrace:



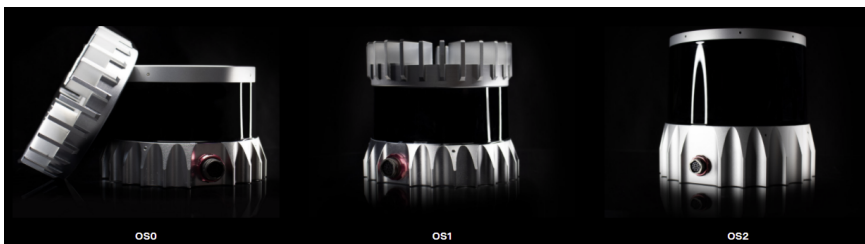
Obrázek 2.1: Ilustrace způsobu fungování LiDARu [31]

2.1 Ouster LiDAR

Jednou z významných společností, zaměřujících se na vývoj a výrobu LiDARů je společnost Ouster. Jde o americkou společnost se sídlem v San Franciscu v Kalifornii zaměřující se na výrobu 3D senzorů s vysokým rozlišením, fungujících na principu LiDARu se širokými možnostmi využití v autonomních vozidlech, robotech, dronech, při mapování či v bezpečnostních systémech.

Přesný princip fungování LiDARů od společnosti Ouster není veřejný a je možné dohledat jen některé z použitých principů. Jedním z nich je zvolení laseru s nižší vlnovou délkou 850 nm. Lasery s nižší vlnovou délkou se často setkávají s problémem velké podobnosti vlnové délce slunečního světla, což může na dobře nasvícených plochách způsobovat falešné zachycení odrazu slunečního světla místo vyslaného laserového paprsku. Společnosti Ouster se podařilo přijít s řešením, které odfiltrovává ambientní sluneční osvětlení. Lze tak čerpat výhod, které laser pracující s kratší vlnovou délkou umožňuje. Těmi jsou například: lepší citlivost čidel pro tuto vlnovou délku, lepší výsledky při práci ve vlhkém prostředí s množstvím vodní páry ve vzduchu či propojení informací z přijatého laserového odrazu a odfiltrovaného ambientního osvětlení pro lepší výkon při zhoršených světelných podmínkách, tedy západech či východech slunce nebo při zatažené obloze. Dalším příkladem zajímavé technologie společnosti Ouster může být přesnější směřování laserových paprsků tak, aby mířil jen přesně takovým směrem, z jakého je možný odraz zpět na čidlo. Není tak plýtvána energie na laserové osvětlování oblastí, odkud není možné paprsek zachytit. [25]

Společnost Ouster nabízí senzory ve třech hlavních modelech nazvaných OS0, OS1, OS2, ty se od sebe liší především úhlem vertikálního snímání a maximální vzdáleností, ze které je odražený paprsek ještě zaznamenán. Typ OS0 poskytuje velký vertikální úhel snímání na kratší vzdálenost, OS2 nabízí naopak menší vertikální úhel na delší vzdálenosti, OS1 je pak kompromisem se středním vertikálním úhlem i vzdáleností snímání. Každý z těchto modelů pak existuje ještě v různých variantách, podle počtu vertikálně vyslaných paprsků. Možné varianty jsou 16, 32, 64 nebo 128 paprsků. [24]



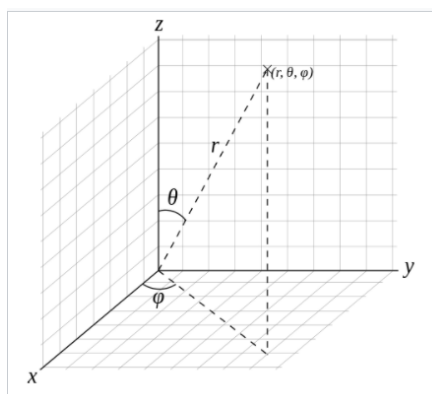
Obrázek 2.2: Ukázka senzorů ze stránek společnosti Ouster [24]

Data z LiDARu se nejčastěji přepočítávají do formy takzvaných mračen bodů (z anglického Point Cloud). V tomto formátu jsou data standardně reprezentována formou tří čísel (například v případě formátu PointCloud z PointCloudLibrary jde o 3 čísla typu float, novější PointCloud2 pak podporuje více datových typů [9]) příslušných souřadnic X , Y , Z . Získání tohoto formátu z dat ze senzoru je možné díky znalosti vertikálního a horizontálního úhlu otočení zrcadla a vzdálenosti bodu. Přepočítání si pak můžeme představit jako převod bodu ze sférických souřadnic pomocí známých rovnic.

$$X = r \sin \theta \cos \Phi$$

$$Y = r \sin \theta \sin \Phi$$

$$Z = r \cos \theta$$



Obrázek 2.3: Sférické souřadnice [4]

Kde r je změřená vzdálenost bodu od senzoru, umístěného v počátku soustavy souřadnic, θ je úhel mezi osou z a spojnici bodu s počátkem soustavy souřadnic, daný indexem měřeného paprsku, a Φ je úhel mezi osou x a průmětem spojnice bodu s počátkem soustavy souřadnic do roviny xy , daný rotací zrcátka LiDARu. Společnost Ouster dává k dispozici vlastní modul pro framework ROS¹, který zajišťuje komunikaci se senzorem. Jeho výstupem, se kterým můžeme dále pracovat, je již přímo mračno bodů, v souřadném systému senzoru.

¹<https://www.ros.org/>

2.2 Parametry a nastavení použitého zařízení

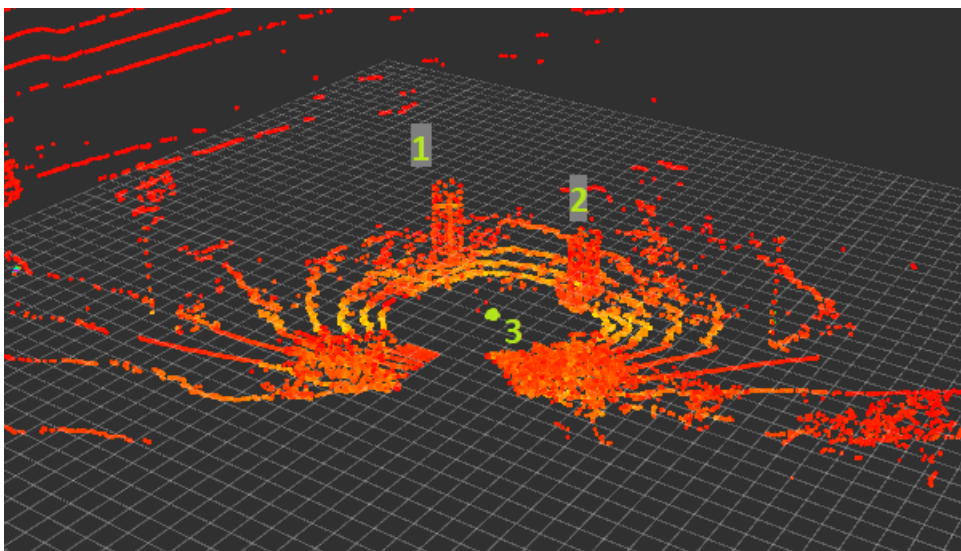
Pro sběr dat využívaných v této práci byl použit LiDAR od společnosti Ouster typu OS0. V následujícím seznamu jsou uvedeny parametry zařízení a zvolené nastavení. Nastavení bylo voleno tak, aby bylo co nejlépe možné zachytit stožár vysokého napětí a jeho konstrukci tvořilo v zachyceném mračnu co možná nejvíce bodů.

- **Vertikální rozlišení** - jde o počet paprsků, které jsou vyslány a zaznamenány ve vertikální směru, tento parametr je pevně dán modelem LiDARu a není možné jej nastavovat. V našem případě jde o **16 paprsků**.
- **Horizontální rozlišení** - jde o počet vertikálních snímaní provedených za jedno otočení LiDARu. Je možné volit mezi 512, 1024 a 2048 snímaními. V našem případě, kdy se snažíme zaznamenat v nasnímaných datech stožár vysokého napětí, který má kvůli způsobu konstrukce jen velmi malé plochy, od kterých se může paprsek odrazit, potřebujeme co největší množství bodů, abychom zajistili, že se některé paprsky do plochy stožáru treří. Zvolenou možností je tedy **2048 snímaní za jedno otočení**.
- **Maximální dosah** - vyjadřuje maximální možnou vzdálenost od objektu tak, aby se od jeho plochy vyslaný paprsek odrazil a byl poté úspěšně zaznamenán v alespoň 80 % případech. Tato hodnota je u modelu OS0 pevně dána na **50 metrů**. V našem případě, kdy uvažujeme pohyb bezpilotní helikoptéry v bezprostřední blízkosti stožáru, je tato vzdálenost naprosto dostačující.
- **Minimální dosah** - jde o nejmenší vzdálenost, ve které je bod ještě zaznamenán. Nastavení je opět pevně dáno typem senzoru na **0.25 metru**.
- **Vertikální rozhled** - udává, jaký úhel paprsky pokrývají ve vertikálním směru. Toto nastavení je také pevně dáno a jeho hodnota je **90°**, což znamená 45° nad vodorovnou osu a 45° pod vodorovnou osu.
- **Frekvence otáčení** - udávána je v Hz, tedy jako počet otočení za 1 sekundu, je možné volit mezi nastavením na 10 Hz nebo 20 Hz. I v tomto případě chceme, aby bylo zaznamenáno co nejvíce bodů a byla velká pravděpodobnost, že alespoň některý z paprsků treří malou plochu stožáru. Volíme tedy nastavení na **20 Hz**.
- **Počet zaznamenaných bodů za sekundu** - tento údaj můžeme snadno dopočítat jako součin vertikálního rozlišení, horizontálního rozlišení a frekvence otáčení, pro náš senzor a zmíněné nastavení tak dostaneme hodnotu **655 360**.

Kapitola 3

Ukázka a analýza nasnímaných dat

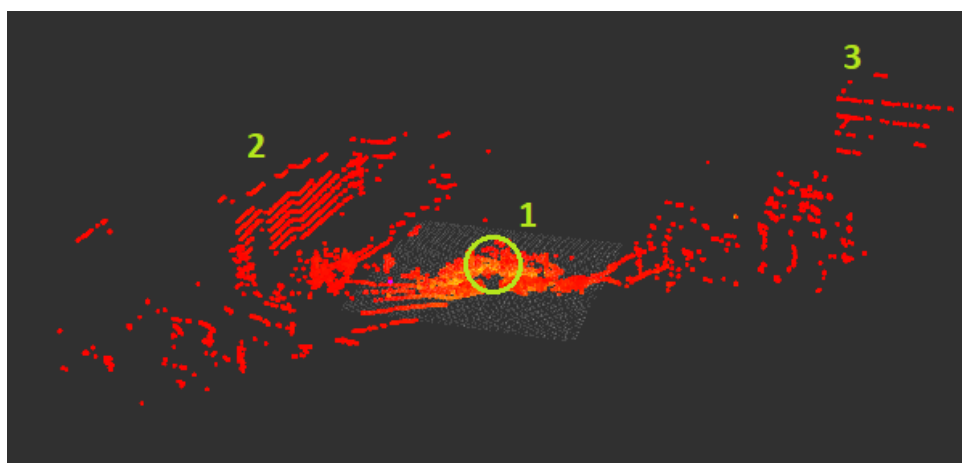
Na následujících obrázcích jsou ukázána data, nasbíraná pomocí LiDARu typu OS0 od společnosti Ouster. Jeho vlastnosti a použité nastavení byly podrobně popsány v kapitole 2.2. Zobrazená data byla nasnímaná v prostředí se dvěma stožáry vysokého napětí, a to přibližně ze vzdálenosti šesti metrů od stožárů. V obrázcích se pro lepší představu nachází bílá mřížka, její střed (v obrázku 3.1 označen číslem **3**) odpovídá místu, kde se nacházel senzor, strana jednoho čtverečku odpovídá vzdálenosti jednoho metru. Vizualizace dat proběhla pomocí nástroje rviz¹.



Obrázek 3.1: Ukázka dat s přiblíženými stožáry

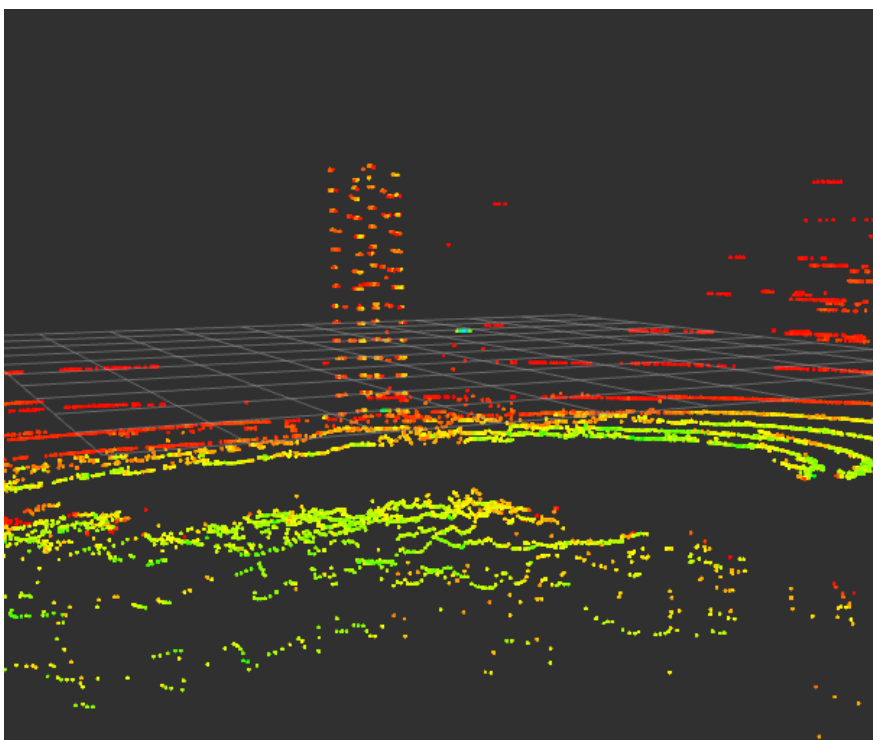
¹<http://wiki.ros.org/rviz>

Z obrázku 3.1 můžeme vidět, že se nám se zvoleným nastavením LiDARu úspěšně podařilo zachytit oba stožáry vysokého napětí (v obrázku označené čísly **1** a **2**, tečka u čísla **3** označuje střed mřížky, ve kterém se nacházel senzor), a to i přes malou plochu jejich konstrukce. Stožáry jsou v tomto přiblíženém záběru v okolní krajině jasně zřetelné a i lidský pozorovatel je dokáže při pohledu na záběr bez větších problémů rozeznat.

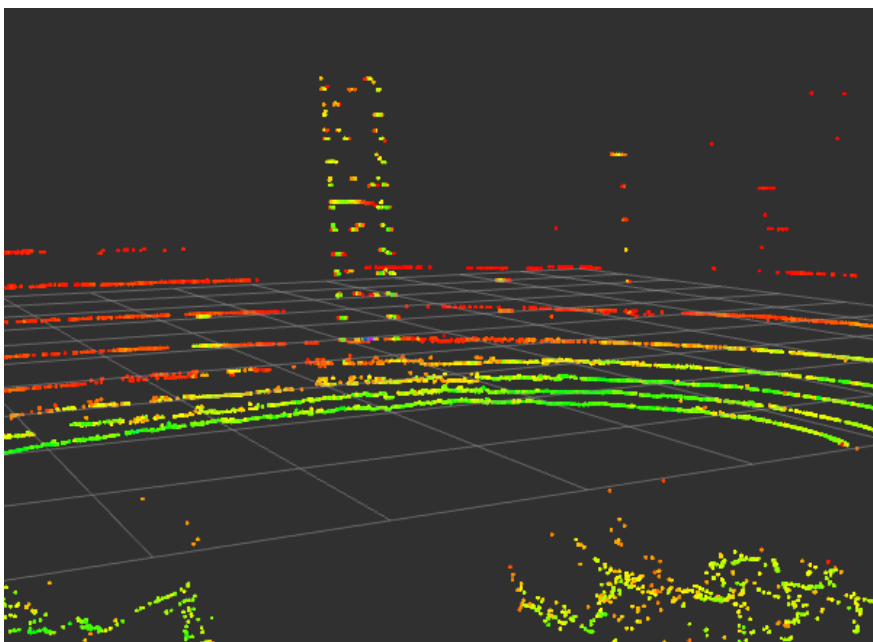


Obrázek 3.2: Ukázka dat s celým záběrem LiDARu

Na obrázku 3.2 se nachází oddálený záběr snímku LiDARu. V kroužku označeném číslem **1** se nachází oba stožáry vysokého napětí a střed souřadnic, ve kterém se senzor nacházel (jedná se o stejný snímek, ze kterého pochází přiblížený záběr v obrázku 3.1). Je vidět, že dosah LiDARu je pro účely lokalizace v blízkosti stožáru vysokého napětí více než dostatečný. Čísly **2** a **3** jsou označeny přilehlé domy, ty se objevují jen v některých snímcích a nejsou vždy jasně viditelné, pro lokalizaci je tak zřejmě nebude možné využít. Naopak by svou rovnou a vzpřímenou konstrukcí mohly být mylně zaměněny za části stožáru. Tento problém je možné vyřešit oříznutím okrajových částí snímku. V okolí vedení vysokého napětí totiž platí ochranné pásmo do vzdálenosti 30 metrů od vedení či konstrukce stožárů, ve kterém se budovy nemohou nacházet [1]. Plánovaná vzdálenost pro použití lokalizace je menší než 30 metrů, nehrozí tak, že by byly oříznuty části snímku důležité pro její funkci.



Obrázek 3.3: Přiblížený záběr na stožár



Obrázek 3.4: Přiblížený záběr na stožár z úhlu, ve kterém vidíme jen jeho přední stranu

Kapitola 4

Identifikace stožáru vysokého napětí

Jak bylo zmíněno v kapitole 3, okolní krajina je ve snímcích z LiDARu nejasná a zašumělá a nebude ji tedy možné pro lokalizaci použít. Cílem části práce, popsané v této kapitole, bude identifikovat v mračnu bodů, přijatém ze senzoru, samotnou konstrukci stožáru a případně ji vhodně upravit pro použití v další části práce, zabývající se samotnou lokalizací. Vzhledem k tomu, že pracujeme se senzory a práce se zabývá lokalizací dronů, řízených systémem ROS, byl program práce vyvinut jako balíček¹ ve frameworku ROS².

4.1 Spuštění programu

Ještě než přejdeme ke spuštění samotného programu pro identifikaci stožáru, je potřeba spustit uzel (z anglického node)³, který se stará o komunikaci se senzorem a o převod dat do formátu mračna bodů. Prvním krokem je instalace ROSu a potřebných modulů ze stránek MRS ČVUT - Multi-robot Systems Group UAV system a následně UAV modules (instalace se provede snadno podle instrukcí přímo na stránkách). Následně zadáme do příkazového řádku příkaz

```
roslaunch ouster_ros uav.launch
```

¹<http://wiki.ros.org/Packages>

²<https://www.ros.org/>

³<http://wiki.ros.org/Nodes>

Příkaz `roslaunch` za nás spustí `roscore`⁴(v případě, že ještě není zapnutý), a další uzly, popsané s nastavením jejich parametrů v souboru `uav.launch`. Pro nás nejdůležitější je uzel `/uav1/os_cloud_nodelet` s následujícími vlastnostmi:

```
~$ rosnode info /uav1/os_cloud_nodelet
-----
Node [/uav1/os_cloud_nodelet]
Publications:
 * /rosout [rosgraph_msgs/Log]
 * /tf_static [tf2_msgs/TFMessage]
 * /uav1/os_cloud_nodelet/imu [sensor_msgs/Imu]
 * /uav1/os_cloud_nodelet/is_alive [std_msgs/Bool]
 * /uav1/os_cloud_nodelet/points [sensor_msgs/PointCloud2]

Subscriptions:
 * /uav1/os_nodelet/imu_packets
 * /uav1/os_nodelet/lidar_packets
```

Z odebíraných témat je pro nás zajímavé téma `/uav1/os_nodelet - /lidar_packets`, ze kterého uzel získává data ze senzoru ve zprávách⁵ typu `ouster_ros/PacketMsg`. Tato data následně převádí a uspořádává do zpráv typu `sensor_msgs/PointCloud2`⁶, obsahujících vždy celý záběr z jednoho otočení LiDARu. Tyto zprávy následně publikuje do tématu `/uav1/os_cloud_nodelet/points`, právě z tohoto tématu je budeme odebírat a identifikovat v nich konstrukci stožáru.

V případě, že chceme přehrávat již dříve nahraná data, použijeme následující variantu příkazu:

```
roslaunch ouster_ros uav.launch replay:=True
```

Ta nám umožní přehrávat data zaznamenaná ve formátu tzv. `bag`⁷, znaného koncovkou `.bag`. Jedná se o primární metodu ukládání záznamů v ROSu, pomocí ukládání jednotlivých zpráv z vybraných témat⁸, kterými mezi sebou uzly komunikují [30]. Pro práci s `bagy` slouží nástroj `rosbag`⁹. Přehrávání zaznamenaných dat v `bagu` spustíme následujícím příkazem:

```
rosbag play filename.bag
```

⁴<http://wiki.ros.org/roscore>

⁵<http://wiki.ros.org/msg>

⁶http://docs.ros.org/en/noetic/api/sensor_msgs/html/msg/PointCloud2.html

⁷<http://wiki.ros.org/Bags>

⁸<http://wiki.ros.org/Topics>

⁹<http://wiki.ros.org/rosbag>

Po spuštění začne program rosbag publikovat všechny zprávy, uložené v souboru, podle času, kdy byly zaznamenány. Všechny programy v systému ROS tedy běží přesně tak, jak by běžely, pokud by byly připojeny k reálným sensorům.

Program, vyvíjený v této práci, se nachází v balíčku `loc` (zkráceno z anglického `localization`), který obsahuje všechny moduly, potřebné k lokalizaci dronu v okolí stožáru vysokého napětí. Pro spuštění je kód nejdříve potřeba zkompileovat zavoláním příkazu

```
catkin_make
```

v adresáři pracovního prostředí. Tím se vytvoří dva spustitelné soubory - `localization` a `identification`. V této části se budeme dále věnovat programu `identification` (části `localization` je věnována kapitola 5). Uzel následně spustíme zavoláním příkazu

```
roslaunch loc identification
```

Spuštěný uzel `\identification` má následující vlastnosti:

```
~$ roslaunch loc identification
-----
Node [/identification]
Publications:
 * /loc/ground [sensor_msgs/PointCloud2]
 * /loc/pilar [sensor_msgs/PointCloud2]
 * /loc/pilar_lines [sensor_msgs/PointCloud2]
 * /loc/pilar_vertical [sensor_msgs/PointCloud2]
 * /rosout [roscpp_msgs/Log]

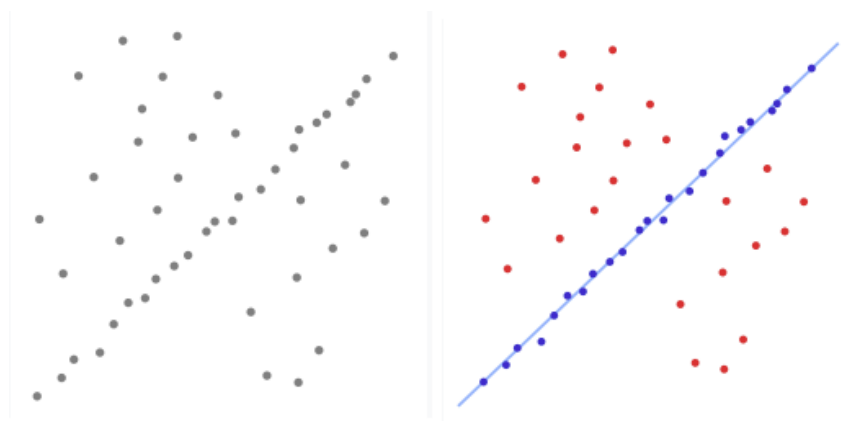
Subscriptions:
 * /uav1/os_cloud_nodelet/points [sensor_msgs/PointCloud2]
```

Vidíme, že uzel odebírá naskenovaná data ve formátu `sensor_msgs/PointCloud2` a následně ve stejném formátu publikuje jednotlivé rozpoznané objekty v různých úpravách a variantách. Přesný popis dat, publikovaných ve zprávách těchto témat, bude následovat dále v této kapitole.

4.2 Algoritmus RANSAC

Základem programu `identification` je algoritmus RANSAC. RANSAC (zkráceno z anglického RAndom SAmple Consensus) je iterativní algoritmus, sloužící k odhadu parametrů matematického modelu z množiny bodů. Byl prezentován M. A. Firscherem a R. C. Bollesem v roce 1981 [17]. Jeho hlavní výhodou oproti známým metodám pro odhad parametrů modelu, jakou je například metoda nejmenších čtverců, je jeho robustnost vůči datům obsahujícím větší množství bodů zatížených chybou, které do modelu nepatří. Ostatní metody povětšinou využívají k odhadu parametrů všechny, nebo co největší množství bodů. RANSAC oproti tomu počítá s tím, že data obsahují jak body zapadající do modelu (z anglického *inliers*) tak body zatížené chybou, které do modelu nezapadají (z anglického *outliers*). Využívá proto pro odhad parametrů modelu jen minimální potřebné množství bodů, náhodně vybraných z dat. Přesné kroky algoritmu jsou následující [10]:

1. Z dat náhodně vybereme M bodů, kde M je minimální počet bodů nutných k určení parametrů modelu.
2. Určíme parametry modelu pro vybrané body.
3. Určíme kolik bodů zapadá do modelu při předem známe toleranci ϵ .
4. Pokud je poměr bodů zapadajících do modelu oproti celkovému počtu bodů větší než předem daná hodnota τ , vybereme tento model a pokračujeme na krok 6.
5. Opakujeme kroky 1-4 až N -krát, kde N představuje předem nastavený maximální počet iterací. Pokud ani po N iteracích algoritmus neskončí, vybereme nejlepší do té doby nalezený model a pokračujeme na krok 6.
6. Znovu určíme parametry modelu, tentokrát jen za použití všech bodů, zapadajících do vybraného modelu. To jsou naše výsledné parametry a body.



Obrázek 4.1: Ukázka fungování RANSAC na příkladu proložení přímky ve 2D. [22]

Mezi nevýhody algoritmu patří jeho větší časová náročnost, způsobená množstvím iterací a fakt, že se nejedná o deterministický algoritmus, neboť body, ze kterých jsou odhadovány parametry jednotlivých modelů, jsou voleny náhodně. Není tedy jisté, zda do vybraného modelu zapadají všechny body, které by měly, případně zda model neobsahuje body, které by do něj zapadat neměly. Pravděpodobnost vybrání ideálního modelu (tedy modelu, který obsahuje právě všechny body nezatížené chybou) je možné kontrolovat nastavením počtu iterací N , avšak pouze v případě, kdy známe pravděpodobnost, že bod není zatížen chybou [10].

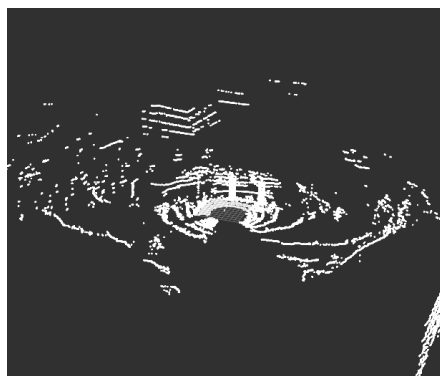
4.3 Využití algoritmu pro identifikaci stožáru vysokého napětí

Jak je patrné z popisu algoritmu, RANSAC představuje díky své schopnosti správně odhadovat parametry modelu i v případě dat, obsahujících body zatížené chybou, vhodné řešení pro identifikaci stožáru v mračnu bodů, přijatém z LiDARu. Vzhledem k tomu, že v přijatém mračnu bodů tvoří body samotné konstrukce stožáru jen malou část (viz. 3), není možné hledat v datech rovnou přímky, reprezentující jednotlivé svislé části stožáru. V opačném případě by algoritmus našel lepší proložení přímek ve velkém množství bodů zachycujících okolní krajinu, či v budovách částečně zachycených na některých snímcích. Pro úspěšnou detekci konstrukce stožáru byl proto navržen následující algoritmus postupného filtrování mračna bodů:

1. Z mračna bodů odebereme body, nacházející se v příliš velké vzdálenosti

pomoci s detekováním konstrukce stožáru. Naopak mohou obsahovat části okolních budov či jiných objektů, které by mohly být mylně označeny za části konstrukce stožáru. Zároveň jejich odstraněním snížíme celkový počet bodů, se kterými budeme pracovat, čímž zrychlíme práci dalších částí algoritmu. Vzdálenost, od které budou body z mračna odebrány, byla určena na 30 metrů (viz 3). Odebereme tedy každý bod, jehož vzdálenost od počátku soustavy souřadnic (odpovídá pozici umístění senzoru) je větší než 30 metrů.

Dalšími body, které je nutné z mračna bodů odebrat, jsou body, které nebyly zaznamenány. Jsou to body odpovídající těm paprskům, které se neodrazilily od žádného objektu. V otevřeném prostoru v okolí stožáru vysokého napětí se může množství takových bodů blížit k polovině všech zaznamenaných bodů. Tyto body jsou po převedení ze zprávy v naší reprezentaci mračna bodů stále zaznamenány, a to jako body s hodnotou $[0, 0, 0]$. Je jasné, že pokud bychom tyto body v mračnu ponechali, znemožnily by úspěšnou detekci jakýchkoliv objektů v mračnu, neboť by každý detekovaný objekt vzhledem k množství nezaznamenaných bodů musel obsahovat bod $[0, 0, 0]$, což samozřejmě neodpovídá realitě. Vizualizace dat před a po provedení tohoto kroku je vidět na obrázku 4.2.



(a) : Data před filtrováním bodů



(b) : Data s odfiltrovanými body

Obrázek 4.2: Data před a po použití tohoto kroku filtrace

Jak můžeme vidět, úspěšně se nám podařilo odfiltrovat okolní šum a nepotřebné části mračna bodů. Zároveň nebyla nijak ovlivněna důležitá část dat, obsahující samotné konstrukce stožárů vysokého napětí.

4.3.2 Detekce krajiny pod stožárem

Tato část práce se věnuje popisu druhého kroku algoritmu, popsaného v sekci 4.3. Do této části přichází mračno bodů již filtrované z první části, ve formátu ukazatele na mračno bodů z PCL. Mračno bodů je v tuto chvíli z většiny tvořené krajinou, nacházející se okolo a pod stožárem vysokého napětí. Cílem tohoto kroku algoritmu je krajinu detekovat a následně ji z mračna odebrat.

Krajinou pod stožárem je možné proložit rovinu. Za krajinu pod stožárem pak budeme považovat všechny body, ležící v této rovině a v předem definované vzdálenosti od ní. Správné proložení roviny přijatým mračnem bodů nalezneme pomocí algoritmu RANSAC. Jde o úlohu, kterou je algoritmus schopen poměrně snadno vyřešit, neboť většina bodů při vhodně nastavené toleranci vzdálenosti zapadá do modelu. Mezi body nezapadající do modelu pak bude patřit jen šum, výrazné odchylky v krajině a konstrukce stožárů vystupujících nad krajinu.

Pro spuštění algoritmu využijeme jeho implementaci z PCL následujícím způsobem:

```
pcl::PointIndices::Ptr indices (new pcl::PointIndices);
pcl::ModelCoefficients::Ptr
    coefficients(new pcl::ModelCoefficients);

pcl::SACSegmentation<pcl::PointXYZ> seg;

//nastaveni modelu roviny
seg.setModelType(pcl::SACMODEL_PLANE);
//zvoleni algoritmu RANSAC
seg.setMethodType(pcl::SAC_RANSAC);
//nastaveni vstupnich dat
seg.setInputCloud(cloud);
//nastaveni meze pro vzdaleness od stozaru
seg.setDistanceThreshold(GROUND_THRESHOLD);
//nastaveni maximalniho poctu iteraci
seg.setMaxIterations(100);
//spusteni algoritmu a ulozeni vysledku do vystupnich dat
seg.segment(*indices, *coefficients);
```

Nastavíme tedy nejdříve jako model rovinu a jako algoritmus k jejímu nalezení zvolíme RANSAC. Pravděpodobnost, že bod není zatížen chybou v tomto případě neznáme. Musíme proto určit hodnotu N počtu opakování experimentálně. Čím větší bude hodnota N , tím větší je pravděpodobnost,

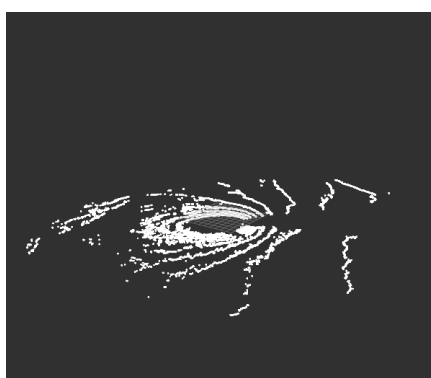
že algoritmus najde správné řešení, zároveň je tím ale větší časová náročnost algoritmu. Na základě experimentů byla určena hodnota 100 opakování. Stejně tak byla experimentálně nastavena hodnota proměnné `GROUND_THRESHOLD`, která určuje toleranci pro vzdálenost bodu od modelu, při které je bod ještě považován za zapadající do modelu. Proměnná byla nastavena na hodnotu 0.9, což odpovídá 0.9 metru. Tato hodnota je dostatečná pro zachycení naprosté většiny nerovností v krajině, zároveň však neodebere příliš velkou část ze stožárů, vysokých více než 10 metrů, výrazně vystupujících z krajiny.

Poměr počtu bodů, zapadajících do modelu, k bodům nezapadajícím se v každém nasnímaném mračnu liší v závislosti na množství šumu, množství bodů zachycujících stožáry a na nerovnosti krajiny. Nemůžeme tak určit poměr, při kterém by měl být algoritmus zastaven. Algoritmus tak proběhne vždy právě N-krát a vybrán bude nejlepší nalezený model. V proměnné `indices` po provedení algoritmu nalezneme indexy bodů, zapadajících do modelu. Proměnná `coefficients` obsahuje koeficienty vybraného modelu.

Body, vybrané jako okolní krajina, jsou publikovány do tématu `/loc/ground` ve formátu `sensor_msgs/PointCloud2`. Pro další zpracování pak naopak ponecháme v mračnu bodů jen body, které nebyly vybrány jako okolní krajina. Ukázkou dat, označených jako krajina, můžeme vidět na obrázku 4.3. Obrázek 4.4 pak ukazuje data po provedení tohoto kroku filtrace.

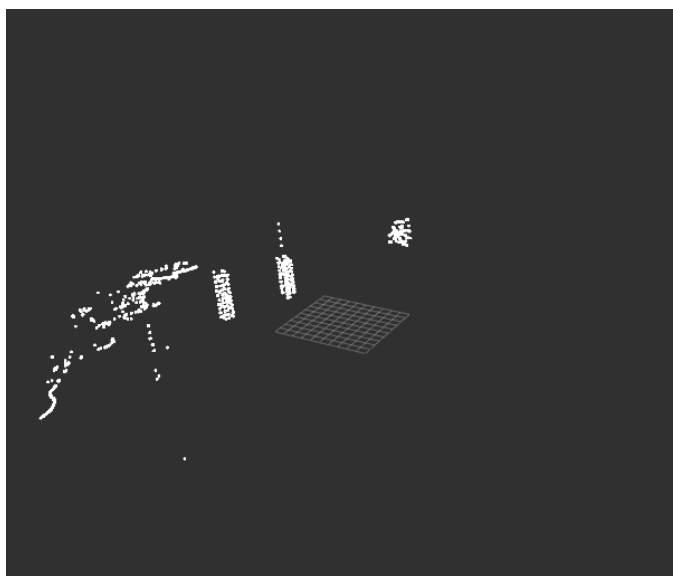


(a) : Data před filtrováním bodů



(b) : Body označené jako krajina

Obrázek 4.3: Data před a po použití detekce krajiny



Obrázek 4.4: Ukázka dat po tomto kroku filtrace

4.3.3 Detekce svislých částí stožáru

Tato část práce se věnuje popisu třetího kroku algoritmu, popsaného v sekci 4.3. Do této části přichází mračno bodů (opět ve formátu ukazatele na mračno bodů z PCL) již vyfiltrované tak, že se v něm mimo konstrukcí stožárů vysokého napětí nachází jen drobný šum a větší nerovnosti okolní krajiny. Z takto vyfiltrovaného mračna jsme již schopni při správném nastavení podmínek detekovat přímo svislé části konstrukce stožárů. Jednotlivými svislými částmi stožárů se budeme postupně snažit prokládat přímky a za detekované stožáry budeme považovat všechny body, zapadající s tolerancí do modelů těchto přímek.

Co nejlepší proložení přímek budeme i v této části hledat pomocí algoritmu RANSAC. V tomto případě však budeme muset přizpůsobit nastavení parametrů algoritmu tomu, že body, zapadající do modelu, budou vždy tvořit jen jednu svislou část stožáru, bude jich tedy znatelně méně, než bodů do modelu nezapadajících. Opět využijeme implementace algoritmu RANSAC z knihovny PCL. Algoritmus použijeme následujícím způsobem:

```
pcl::PointIndices::Ptr indices (new pcl::PointIndices);

pcl::ModelCoefficients::Ptr
    line_coefficients(new pcl::ModelCoefficients);

pcl::SACSegmentation<pcl::PointXYZ> seg;
```

```

seg.setModelType(pcl::SACMODEL_LINE);
seg.setMethodType(pcl::SAC_RANSAC);
seg.setInputCloud(pilar_from);
seg.setDistanceThreshold(PILAR_THRESHOLD);
seg.setMaxIterations(1000);
seg.segment(*indices, *line_coefficients);

```

Jako model zde nastavíme přímku. Ani v tomto případě neznáme pravděpodobnost, že je bod zatížen chybou. Počet opakování byl určen experimentálně na hodnotu 1000 opakování. Počet je v tomto případě výrazně vyšší, než při prokládání roviny, což je způsobeno malým množstvím bodů, zapadajících do jednotlivých modelů přímek. Cílový poměr počtu bodů zapadajících do modelu ku počtu bodů nezapadajících nemůžeme ani v tomto případě předem odhadnout. Algoritmus tak vykoná vždy všechna opakování a následně vybere nejlepší model. Hodnota `PILAR_THRESHOLD` byla nastavena podle šířky konstrukce stožáru na hodnotu 0.15 (což odpovídá 0.15 m), při této hodnotě do modelu zapadají body z celé šířky svislé části stožáru, nezapadají do něj případně zaznamenané vodorovné spojnice stožáru.

Algoritmus RANSAC však ve vstupním mračnu bodů často nalezne i proložení přímek ve zbytcích okolní krajiny či v šumu. Abychom zabránili označení těchto bodů jako součástí konstrukce stožáru, byly zavedeny dvě dostatečné podmínky, které musí být pro označení bodů jako částí konstrukce stožáru splněny zároveň. První podmínka říká, že do modelu přímkou musí zapadat alespoň 10 bodů. Hodnota počtu bodů byla určena experimentálně. Tato podmínka zajistí, že nebudou vybrány krátké přímkou proložené skrze části šumu či skrze drobné zbytky okolní krajiny. Podmínka zároveň nezabraňuje správnému označení částí stožáru, neboť ty jsou oproti šumu a zbytkům krajiny dlouhé, jimi proložená přímkou tak obsahuje větší množství bodů. Druhá podmínka říká, že proložená přímkou musí svírat úhel větší než 60° rovinou $z = 0$. Tato podmínka vychází z faktu, že konstrukce stožáru je kolmá na povrch země, tolerance vychýlení 30° je ponechána kvůli náklonům senzoru během letu dronu. Do modelů přímek, proložených zbytky okolní krajiny, v některých případech může zapadat dostatečné množství bodů pro splnění první podmínky, krajina však není dostatečně vysoká, a proto jsou přímkou spíše vodorovné a tedy nesplňují tuto druhou podmínku.

Pro korektní detekování všech svislých částí konstrukce je nutné spouštět algoritmus RANSAC opakovaně. Body, tvořící nalezenou část konstrukce, vždy překopírujeme do výsledného mračna bodů a odebereme je z mračna, ve kterém konstrukci hledáme. Tím zabráníme opakovanému nalézání nejlepšího modelu přímkou v mračnu. Podobně odebereme i přímkou, které nesplňují dvě podmínky, uvedené v předchozím odstavci, jen s tím rozdílem, že je nepřekopírujeme do výsledného mračna. Opakování ukončíme ve chvíli, kdy

ani nejlepší nalezená přímka nesplňuje podmínku o minimálním množství bodů, zapadajících do modelu. V tu chvíli předpokládáme, že všechny části konstrukce byly nalezeny a odebrány z mračna, ve kterém je hledáme.

Algoritmus hledání konstrukce stožáru je tak následující:

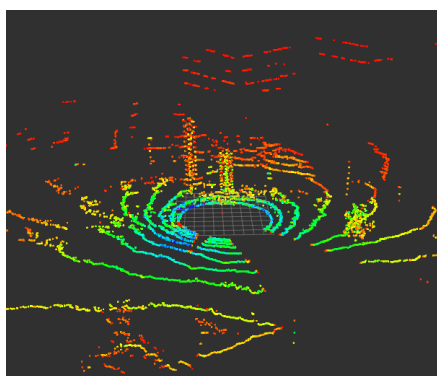
Algorithm 1 Algoritmus pro nalezení konstrukce stožáru ve filtrovaném mračnu

```

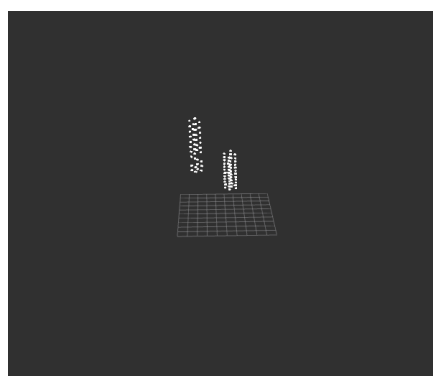
input_cloud
final_cloud  $\leftarrow \emptyset$ 
line  $\leftarrow$  RANSAC(input_cloud)
while size(line)  $\geq$  10 do
  input_cloud -= line
  if CORRECT_ANGLE_WITH_GROUND(line) then
    final_cloud += line
  end if
  line  $\leftarrow$  RANSAC(input_cloud)
end while
return final_cloud

```

Body, označené jako konstrukce stožáru, jsou publikovány do tématu `/loc/pilar` ve formátu `sensor_msgs/PointCloud2`. Tyto body jsou zároveň také ponechány v mračnu za účelem dopravení pro lepší fungování dalších částí programu. Výsledky detekce stožáru jsou ukázány na obrázku 4.5.



(a) : Snímek z LiDARu



(b) : Detekované stožáry vysokého napětí

Obrázek 4.5: Ukázka detekce stožárů vysokého napětí

Jak můžeme vidět, algoritmus úspěšně detekuje svislé části stožáru vysokého napětí ve snímku z LiDARu. V některých snímcích se však algoritmu nepodaří detekovat všechny části obou stožárů vysokého napětí. V těchto

případech většinou není detekována vzdálenější část konstrukce stožáru, která je částečně zastíněná bližší částí konstrukce a obsahuje tak jen malé množství bodů. Obecně lze říct, že úspěšnost detekování stožáru se zhoršuje, se zhoršující se kvalitou zachycení stožáru na snímku. Vzhledem k povaze algoritmu, používaného v lokalizační části, je však chybějící část konstrukce stožáru menším problémem, než jakým by byly případné špatné detekce, tedy ponechání bodů, které do konstrukce nepatří.

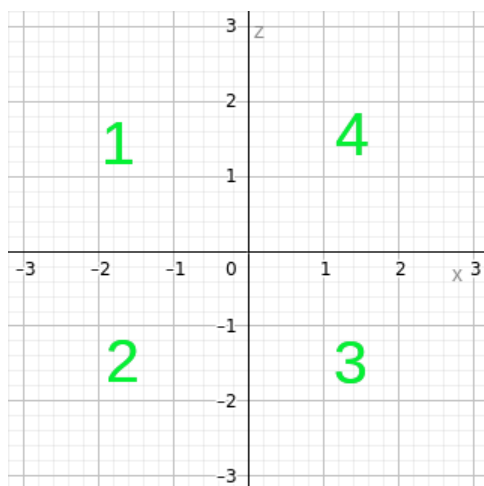
4.3.4 Úprava nalezených bodů pro další část programu

Tato část práce se věnuje popisu čtvrtého kroku algoritmu, popsaného v sekci 4.3. Data pro zpracování v této části přicházejí ve formátu ukazatele na mračno bodů z PCL a obsahují již jen samotné detekované konstrukce stožárů vysokého napětí. Samotný senzor se při snímání dat většinou nenachází v přesně vodorovné poloze a detekované konstrukce stožárů jsou tak různě nakloněné. Pro lokalizační část programu by však bylo vhodné, aby konstrukce stožárů byly vždy svislé (tedy svíraly úhel 90° s rovinou $z = 0$). Právě narovnání detekovaných konstrukcí stožáru bude cílem této části práce.

Narovnání konstrukce stožáru dosáhneme pomocí kombinace rotací okolo os x a y . Jako referenční část konstrukce stožáru, podle které budeme rotace provádět, použijeme nejvýraznější, tedy první proloženou přímkou určenou algoritmem RANSAC v části 4.3.3. Z parametrického popisu přímky, skládajícího se ze směrového vektoru a bodu, jsme schopni určit úhel, který svírá přímka s rovinami $x = 0$ a $y = 0$. Následně dopočítáme úhel α , o který otočíme body proti směru hodinových ručiček okolo osy x , a úhel β , o který otočíme body proti směru hodinových ručiček okolo osy y . Se znalostí těchto úhlů můžeme sestavit rotační matice[15]:

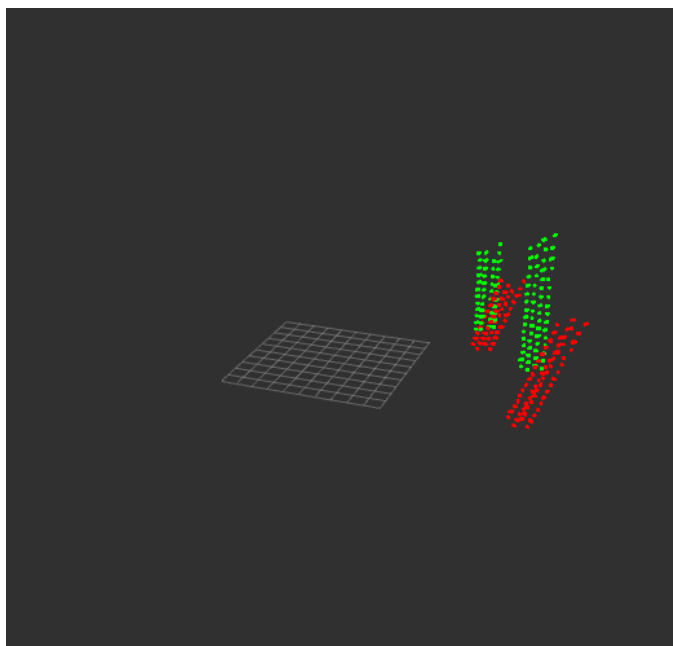
$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix} R_y = \begin{pmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{pmatrix}$$

Pro určení úhlu α nejdříve spočítáme úhel, který svírá přímka s rovinou $x = 0$, označme jej γ . Následně dopočítáme úhel α podle kvadrantu, ve kterém leží promítnutí směrového vektoru do roviny xz . Označení kvadrantů můžeme vidět v obrázku 4.6. V kvadrantu 1 je $\alpha = \gamma$, v kvadrantu 2 je $\alpha = 180 - \gamma$, v kvadrantu 3 $\alpha = \gamma + 180$ a v kvadrantu 4 je $\alpha = (180 - \gamma) + 180$. Obdobným způsobem dopočítáme hodnotu β z úhlu svíraného s rovinou $y = 0$ a polohy promítnutí směrového vektoru do roviny yz .



Obrázek 4.6: Označení kvadrantů

Pomocí rotačních matic pak zrotujeme všechny body v mračnu a tím získáme svislé konstrukce stožárů vysokého napětí. Tyto body jsou publikovány do tématu `/loc/pilar_vertical` ve formátu `sensor_msgs/PointCloud2`. Činnost této části algoritmu je viditelná v obrázku 4.7.



Obrázek 4.7: Ukázka dat před převedením do svislé polohy (červeně) a po něm (zeleně)

■ Experiment s přímkami

Cílem tohoto experimentu bylo rotovat do svislé polohy jednotlivé proložené přímkami z části 4.3.3 namísto bodů, spadajících do modelů těchto přímk. Lokalizační část práce pak měla pracovat s průsečíky přímk s rovinou $z = 0$. Tato změna měla snížením množství bodů zajistit větší rychlost a přesnost lokalizační části. Tato očekávání se však nenaplnila a lokalizační část tak pracuje s body publikovanými v části 4.3.4. Průsečíky jsou pro případné další užití nadále publikovány do tématu `/loc/pilar_lines` ve formátu `sensor_msgs/PointCloud2`.

Kapitola 5

Lokalizace na základě zpracovaných dat ze senzoru

Cílem této části práce je ověřit možnosti lokalizace v prostředí stožáru vysokého napětí, na základě zpracovaných dat ze senzoru. Tato část programu přijímá identifikované svislé části stožáru vysokého napětí (viz. 4) a následně z nich určuje relativní polohu senzoru vzhledem k místu, ve kterém se senzor nacházel při spuštění programu.

5.1 Spuštění programu

Lokalizační část programu se nachází společně s identifikační částí v balíčku `loc`, který obsahuje všechny potřebné soubory. Pro správnou činnost této části je potřeba nejdříve zkompilevat kód, spustit uzly, zajišťující komunikaci se senzorem a převod dat do vhodného formátu, a následně spustit identifikační část programu. Všechny nutné kroky jsou popsány v části 4.1. Lokalizační část programu následně spustíme pomocí příkazu

```
roslun loc localization
```

Spuštěný uzel `\localization` má následující vlastnosti:

```
~$ roslun info /localization
```

```
-----  
Node [/localization]
```

Publications:

```
* /loc/matched_clouds [sensor_msgs/PointCloud2]
* /loc/position [visualization_msgs/Marker]
* /rosout [rosgraph_msgs/Log]
```

Subscriptions:

```
* /loc/pilar_vertical [sensor_msgs/PointCloud2]
```

Vidíme, že uzel odebírá detekované části konstrukce stožáru, otočené do svislé polohy z tématu `/loc/pilar_vertical` ve formátu `sensor_msgs/PointCloud2` a následně publikuje na sebe přiložené snímky ve stejném formátu. Dále publikuje odhadované pozice senzoru do tématu `/loc/position` ve formátu `visualization_msgs/Marker`. Přesný popis publikovaných dat bude následovat dále v této kapitole.

5.2 Algoritmus ICP

Pro řešení lokalizace je vhodné použít algoritmus, který umožňuje položit přes sebe dvě mračna bodů tak, aby byl po dokončení algoritmu minimální rozdíl mezi mračny. Známým a v jeho různých verzích hojně využívaným algoritmem, který nabízí řešení této úlohy je algoritmus ICP. ICP algoritmus (z anglického Iterative Closest Point) byl navržen a poprvé prezentován dvojicí Paul J. Besl a Neil D. McKay v roce 1992, je možné jej použít pro přesné přiložení mračen bodů ve 2D i ve 3D. [6] Jeho hlavní předností jsou robustní a velmi stabilní výsledky, které poskytuje.

Vstupem pro algoritmus jsou dvě mračna bodů. První z nich je označováno jako referenční nebo také cílové (target), toto mračno zůstává fixní. Cílem algoritmu je k cílovému mračnu co nejpřesněji přiložit druhé mračno, označované jako zdrojové (source). Pro přiložení hledá algoritmus co nejlepší transformaci, skládající se z posunutí a rotace. Kvalita transformace je hodnocena pomocí nejmenších čtverců. Po dokončení algoritmu jsme schopni pomocí transformace, použité pro přiložení mračen, získat změnu polohy, ke které došlo mezi porízením snímku zdrojového a cílového mračna.

Označme cílové mračno bodů jako P , zdrojové mračno jako Q a požadovanou přesnost jako σ . Kroky algoritmu jsou pak následující[21]:

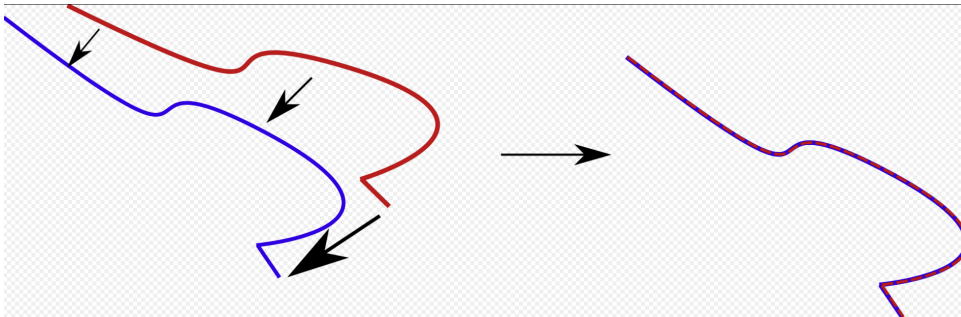
1. Každému bodu $p_i \in P$ přiřadíme příslušný nejbližší bod $q_i \in Q$.

2. Nalezneme transformační matici T , která kombinací transformace a rotace minimalizuje součet vzdáleností přiřazených dvojic bodů. Nalezení matice je možné například použitím SVD rozkladu.
3. Matici T použijeme pro transformaci zdrojového mračna Q .
4. Při dosažení maximálního počtu iterací nebo při naplnění podmínky $d(P^*, Q^*) < \sigma$ ukončíme algoritmus, v opačném případě opakujeme krok 1. Hodnota $d(P^*, Q^*)$ je vypočtena jako:

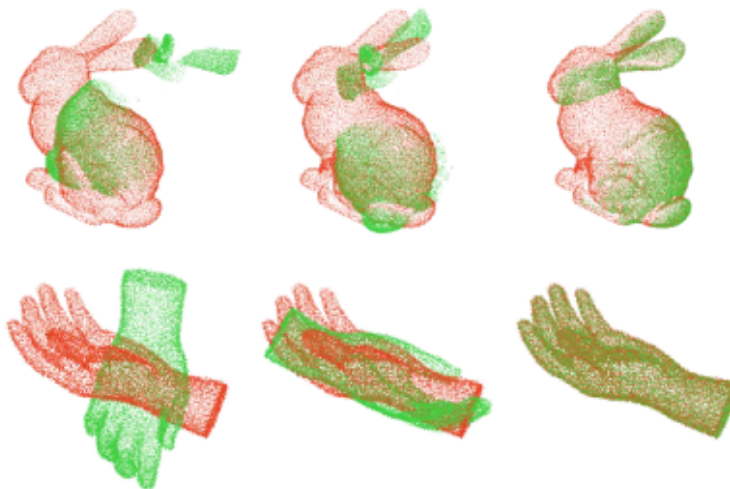
$$d(P^*, Q^*) = \sqrt{\frac{1}{n} \sum_{i=1}^n (\|p_i - q_j\|)^2}, (1 \leq j \leq m)$$

kde q_j je bod z Q odpovídající bodu p_i z P , n je velikost P a m je velikost Q .

Činnost algoritmu je vizualizována na obrázcích 5.1 a 5.2.



Obrázek 5.1: Vizualizace ICP ve 2D [7]



Obrázek 5.2: Vizualizace ICP ve 3D [33]

5.2.1 Varianty algoritmu ICP

Algoritmus ICP se vyskytuje v různých variantách. Výše popsaný způsob, kdy k sobě přiřazujeme dvojice bodů, se nazývá **point to point**. Další známé varianty jsou například:

- **point to line** - sousední body v cílovém mračnu jsou propojeny úsečkami, body zdrojového mračna pak nepřibližujeme k bodům, ale k těmto úsečkám.
- **point to plane** - body v cílovém mračnu jsou proloženy rovinami, zdrojové mračno se pak přibližuje k těmto rovinám.
- **feature to feature** - jedna ze složitějších variant, z obou mračen jsou vybrány výrazné části jako například hrany objektů či různé linie. V algoritmu pak přikládáme pouze tyto výrazné části (na podobném principu pak funguje také varianta point to feature) [12].

Existuje také mnoho různých vylepšení, která se snaží o zrychlení či zlepšení algoritmu. Příkladem takových vylepšení jsou:

- Filtrování dvojic bodů pomocí omezujících podmínek za účelem zrychlení algoritmu a zlepšení přesnosti jeho výsledků - jsou odebrány dvojice, o kterých se usuzuje, že by konvergování algoritmu zpomalovaly [32].
- Využití polárních souřadnic, jakožto formátu dat ve kterém LiDAR přímo pracuje [12].
- Přidělování váhy jednotlivým bodům v modelu, kdy je váhou určována důležitost přesnějšího přiložení některých bodů, což může zrychlit nalezení řešení i přesnost tohoto řešení.
- Využití k-d stromů za účelem rychlejšího vyhledávání nejbližších bodů [23].

5.2.2 Faktory ovlivňující činnost algoritmu

Výsledek algoritmu nemusí být vždy takový, jaký očekáváme. Může se stát, že dostaneme přibližně správné řešení, ale s větší chybou. V horším případě

pak může algoritmus uváznout v lokálním optimu, tedy u takového přiložení mračen, které se pro algoritmus jeví jako dostatečně dobré, ale přitom může být od správného řešení diametrálně odlišné. V následujícím seznamu jsou vyjmenovány některé faktory, které v různé míře ovlivňují správnou činnost algoritmu [21]:

- **Míra překrytí** - vyjadřuje, jaká část zdrojového a cílového mračna je shodná, obecně platí, že čím je shodná část menší, tím je pravděpodobnější, že algoritmus nedojde ke správnému řešení.
- **Úhel** - říká, o kolik stupňů jsou vůči sobě mračna pootočená. V případě, že je otočení menší, dokáže si s ním algoritmus postupným iterováním poradit. Vždy však existuje hraniční hodnota, po jejímž překročení algoritmus zdrojové mračno přetočí a uvázne v lokálním optimu.
- **Vzdálenost** - vzhledem k tomu, že jsou v algoritmu přiřazovány nejbližší body, hraje různá vzdálenost mračen také jistou roli. Nejde ani tak o její velikost jako o směr posunu, který také může způsobit uváznutí v lokálním optimu.
- **Ostatní faktory** - těmi může být například šum či tvar pozorovaných objektů.

5.3 Lokalizace pomocí ICP algoritmu

Také v této části programu využijeme pro práci s mračnem bodů knihovnu PCL. Po přijetí zprávy, obsahující detekované části konstrukce stožáru vysokého napětí, převedeme data do formátu ukazatele na mračno bodů, obsahující body popsané souřadnicemi XYZ (viz. 4.3.1). Vzhledem k tomu, že bude určována poloha pouze ve 2D (nebudeme určovat výšku, ve které se dron nachází), avšak přijatá data jsou trojrozměrná, nabízí se otázka, jakým způsobem data vhodně upravit pro dosažení co nejlepších výsledků. Mezi testované možnosti patřilo například: ponechat data ve 3D a zanedbávat hodnotu z , promítnutí všech bodů do roviny $z = 0$, výběr jedné vrstvy snímku, či použití průsečíků přímek, reprezentujících svislé části konstrukce stožáru (viz. 4.3.4). Jako nejlépe fungující řešení se nakonec ukázala kombinace některých uvedených přístupů. Z přijatých dat jsou vybrány pouze body vzdálené mezi -1 m a 3 m od roviny $z = 0$, ty jsou následně promítnuty do této roviny nastavením hodnoty z na nulu. Uplatnění tohoto postupu na zarovnanou konstrukci stožáru zachová celou šířku jeho jednotlivých částí, zároveň však neobsahuje příliš velké množství bodů. To jednak snižuje šum v datech, dále pak nižší množství bodů snižuje časovou náročnost ICP algoritmu. Oproti práci ve

3D také algoritmus viditelně méně končí nalezením pouze lokálního optima. Snímky, ve kterých se nepodařilo najít žádné části konstrukce stožáru a jsou prázdné, pro určení polohy nevyužíváme a jsou automaticky přeskočeny.

Pro spuštění ICP algoritmu využijeme jeho implementaci z PCL následujícím způsobem:

```
pcl::IterativeClosestPoint
  <pcl::PointXYZ, pcl::PointXYZ> icp;
icp.setMaximumIterations (500);
icp.setTransformationEpsilon (1e-9);
icp.setInputSource (cloud_in);
icp.setInputTarget (cloud);
icp.align (*cloud_in);
float score = icp.getFitnessScore();
```

Maximální počet iterací byl vzhledem k náročnosti úlohy na těchto datech nastaven na 500. Možnost zastavení algoritmu při dostatečně malém rozdílu mezi mračny zde nevyužíváme, neboť detekované části konstrukce stožáru v různých snímcích jsou pro správné nastavení této podmínky příliš rozdílné. Místo toho využíváme podmínku, která algoritmus zastaví, jestliže je rozdíl ve zdrojovém (a tedy v jednotlivých iteracích transformovaném) mračnu mezi dvěma iteracemi menší než $1 * 10^{-9}$. Algoritmus je tak zastaven ve chvíli, kdy by další iterace nepřinášely výrazné zlepšení výsledku, což snižuje časovou náročnost algoritmu. Dále stojí za povšimnutí, že jako cílové mračno používáme starší přijaté mračno, ke kterému přikládáme jako zdrojové nově přijaté mračno `cloud_in`. Kvůli tomuto rozdělení dostáváme transformaci od novějšího snímku ke staršímu, pro správné vyjádření změny polohy mezi snímky tak musíme pracovat s inverzní transformací.

Přijatá data však mohou obsahovat také méně kvalitní snímky, ve kterých se například nemuselo podařit úspěšně identifikovat dostatečné množství bodů z konstrukce stožáru a algoritmus při práci s nimi nenalezne dostatečně dobré přiložení či uvázne v lokálním optimu. Tyto snímky by způsobily výrazné chyby v odhadování polohy. Pro řešení tohoto problému byl navržen algoritmus pracující s historií předchozích snímků. Algoritmus si ukládá pět posledních přijatých mračen bodů společně s pro ně určenými polohami v prostoru (pro první snímek je tato poloha nastavena do bodu [1, 1], který umožňuje zaznamenání rotace okolo souřadnicové osy). K nově přijatému mračnu následně přiložíme pomocí ICP algoritmu postupně všechna tato mračna a vybereme transformaci takového přiložení, ve kterém algoritmus dosáhl nejlepšího skóre. Následnou nově určenou polohu získáme pomocí této transformace z předchozí polohy. Jako předchozí polohu volíme polohu příslušící k mračnu, jehož přiložením k nově přijatému mračnu jsme dosáhli

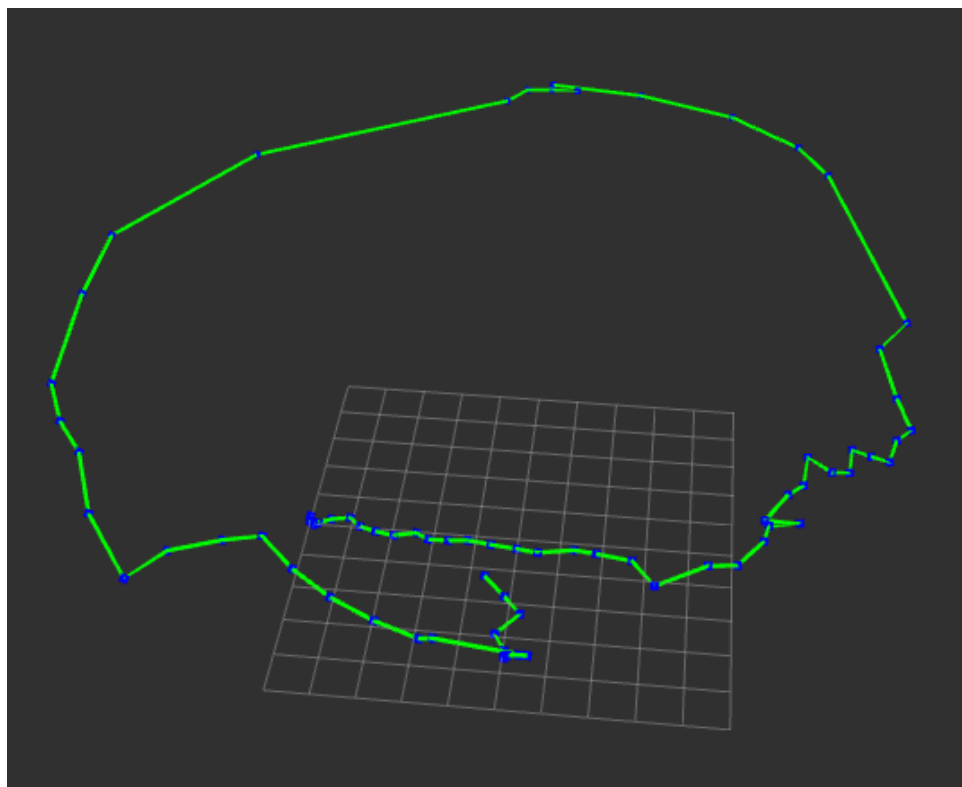
nejlepšího skóre. Skóre je vypočítáno jako součet druhých mocnin vzdáleností bodů ze zdrojového mračna od jim přiřazeným nejbližším bodům z cílového mračna. V případě, že je toto skóre menší než 15 (čím menší skóre tím lepší příložený bylo), je mračno bodů považováno za dostatečně dobré a je společně s pro něj určenou polohou přidáno mezi uložená mračna, pokud je více než pět uložených mračen, je odebráno nejstarší mračno. V případě, že je skóre menší než 5, můžeme považovat příložený za správné a jako poloha senzoru je určena poloha získaná na základě tohoto mračna. Jestliže je skóre mezi 5 a 15, je jako poloha prozatím určena poloha příslušící mračnu, se kterým jsme našli nejlepší příložený. Nová poloha je však stále uložena do historie společně s přijatým mračnem a může se s ní dále pracovat. Toto opatření zamezuje náhlým prudkým výchytkám v určení polohy, způsobeným ojedinělým méně kvalitním snímkem. Údaj o poloze však zůstává zaznamenán a ve chvíli, kdy by se tento snímek ukázal jako správný (ve chvíli, kdy byl přijat byla historie zaplněna méně kvalitními snímky) a dosáhl by v dalším běhu algoritmu nejlepšího skóre, bude se nadále pracovat právě s touto polohou. Hodnoty pro požadované skóre se mohou zdát příliš velké oproti běžným případům použití algoritmu ICP, musíme si však uvědomit, že v přijatých datech může být mezi dvěma snímky rozdíl v celé detekované části stožáru, čímž skóre rychle narůstá.

Ve chvíli, kdy přijaté mračno bodů nedosáhlo při přiřazování k uloženým snímkům v žádném případě skóre pod 15, není mračno s určenou polohou vůbec uloženo. V případě, že by se toto opakovalo s pěti po sobě jdoucími mračny, je historie vymazána a je do ní vloženo poslední přijaté mračno s poslední určenou polohou. Tím je zajištěno, že algoritmus bude schopen fungovat i v situaci, kdy se například dron nakloní a senzor zaznamená sérii horších snímků, následný kvalitní snímek pak bude díky tomu příliš pootočený pro správné příložený. V tomto případě bude algoritmus po vymazání historie pokračovat přiřazování k novému snímku a určování polohy bude pokračovat od poslední určené polohy. Fungování algoritmu je popsáno v pseudokódu 2.

Algoritmus je takto nastaven pro rekonstrukci celé trasy pohybu. V praxi by jej bylo možné využít také jen pro určení změny polohy oproti předchozím snímkům bez nutnosti záznamu celé historie pohybu.

Byla vyzkoušena také varianta algoritmu, ve které se do historie místo posledních snímků ukládal model prostředí, skládaný postupným příložením nových mračen bodů se skóre menším než 5. V tomto modelu se však vlivem různé struktury detekované konstrukce stožáru v různých snímcích postupně střídala chyba a lokalizace dosahovala horších výsledků než při ukládání posledních snímků.

Přiložená mračna jsou pro vizuální kontrolu publikována do tématu `/loc/matched_clouds` ve formátu `sensor_msgs/PointCloud2`. Určené polohy jsou společně s celou historií publikovány do tématu `/loc/position` ve formátu `visualization_msgs/Marker`. Tento formát lze dobře vizualizovat v programu rviz, zároveň se z něj dají snadno získat souřadnice poloh pro případnou další práci. Vizualizace určené polohy je na obrázku 5.3. Jak můžeme vidět, algoritmus zvládl správně určit celý okruh pohybu senzoru okolo stožáru vysokého napětí.



Obrázek 5.3: Vizualizace určené polohy při pohybu okolo stožáru vysokého napětí

Algorithm 2 Algoritmus pro určení polohy z přijatého mračna

```

history ← ∅
misses ← 0
function CALLBACK(cloud_in)
  if size(cloud_in) == 0 then
    return
  end if
  if size(history) == 0 then
    history += < cloud_in, [1, 1] >           ▷ Initial position is set
    return
  end if
  score, prev_location, transformation ←
  GET_BEST_ICP_MATCH(cloud_in, history)
  ▷ Finds best match by matching
  cloud_in one by one on all clouds in
  history using ICP. Match with lowest
  score is selected. Gives us the score,
  location corresponding to the best
  matched cloud and the used trans-
  formation.

  if score < 15 then
    misses ← 0
    new_location ← APPLY_INVERSE_TRANSFORMATION(prev_location)
    history += < cloud_in, new_location >
    if size(history) > 5 then
      REMOVE_OLDEST(history)
    end if
    if score < 5 then
      PUBLISH_LOCATION(new_location)
    else
      PUBLISH_LOCATION(prev_location)
    end if
  else
    misses += 1
    if misses > 5 then
      last_location ← GET_LAST_LOCATION(history)
      clear(history)
      history ← < cloud_in, last_location >
    end if
  end if
  return
end function

```

Kapitola 6

Analýza funkčnosti programu

Tato kapitola je věnována závěrečné analýze správné činnosti programu v závislosti na vzdálenosti senzoru od konstrukce stožáru vysokého napětí. Měření prezentovaných dat probíhalo na čtyřech různých záznamech z LiDARu, zachycujících postupné vzdalování senzoru od konstrukce stožáru vysokého napětí. Výsledná data byla rozdělena podle intervalu vzdálenosti, do kterého spadá vzdálenost nejbližší části stožáru vysokého napětí od senzoru v měřeném snímku. Intervaly vzdálenosti byly stanoveny následovně: méně než 15 metrů, 15-20 metrů, 20-25 metrů, 25-30 metrů a pro zjištění limitů správného fungování programu také interval více než 30 metrů. Pro naměření dat v posledním zmíněném intervalu byla dočasně posunuta hranice filtrování dat na základě polohy (viz. 4.3.1) na 40 metrů. Tento krok si můžeme pro účely analýzy fungování programu dovolit, neboť se v žádném z použitých záznamů nenachází do vzdálenosti 40 metrů od senzoru žádný objekt, který by mohl být chybně identifikován jako část konstrukce stožáru.

Protože nebyla k dispozici informace o skutečné poloze senzoru, byly pro algoritmus, zajišťující identifikaci konstrukce stožáru v mračnu bodů (viz. 4.3), naměřeny čtyři statistiky - celkový počet snímků, přijatých pro zpracování; počet snímků, ve kterých byla proložena alespoň jedna přímka tvořící část konstrukce stožáru; průměrný počet bodů ve snímku, označených jako součást konstrukce stožáru (započítávají se pouze snímky, ve kterých byla nalezena alespoň nějaká část konstrukce stožáru) a průměrná časová náročnost běhu algoritmu pro jeden snímek. Výsledky jsou zaznamenány v tabulce 6.1.

Jak můžeme vidět v tabulce 6.1, navržený algoritmus spolehlivě zvládá identifikovat konstrukci stožáru vysokého napětí v přijatém mračnu bodů

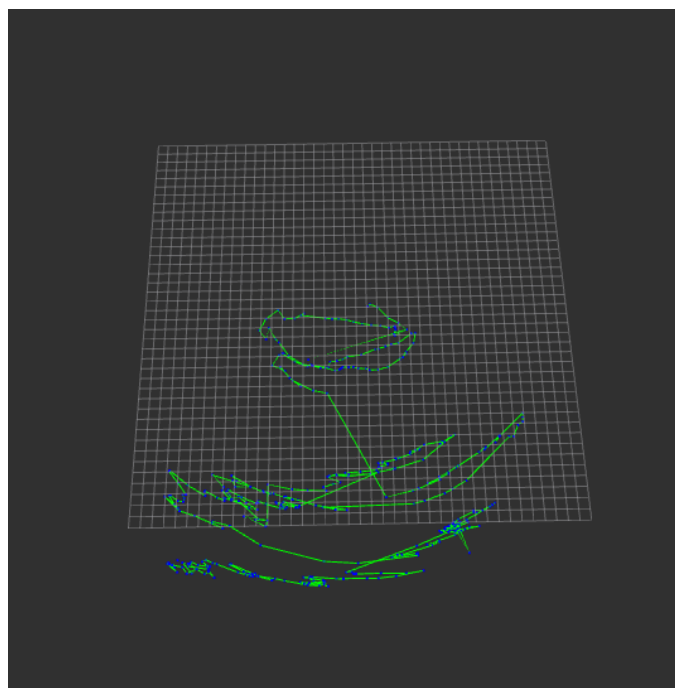
Vzdálenost:	<15 m	15-20 m	20-25 m	25-30 m	>30 m
Celkový počet příkládaných snímků	442	308	444	395	297
Počet příložených se skóre < 15	415	279	430	351	188
Z toho počet příložených se skóre < 5	391	265	416	330	172
Množství snímků se skóre < 15 vzhledem k počtu příkládaných snímků v %	93.9%	90.6%	96.8%	88.9%	63.2%
Množství snímků se skóre < 5 vzhledem k počtu příkládaných snímků v %	88.5%	86.0%	93.7%	83.5%	57.9%
Průměrná časová náročnost	75 ms	21 ms	16 ms	11 ms	12 ms

Tabulka 6.2: Statistiky fungování lokalizace v různých vzdálenostech

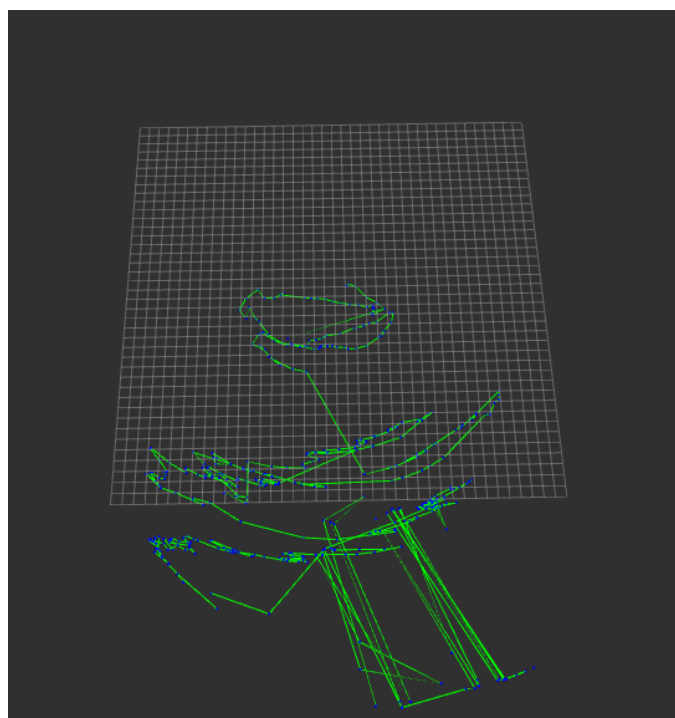
Také v tomto případě klesá časová náročnost lokalizačního algoritmu, což je způsobeno rychlejším během algoritmu ICP při nižším množství bodů.

6.1 Rekonstrukce trasy postupného vzdalování od stožáru vysokého napětí

Na závěr práce byl program testován na datech, zachycujících postupné vzdalování senzoru od stožáru vysokého napětí. Cílem programu bylo co nejlépe rekonstruovat trasu pohybu senzoru. Program při tomto testování ukázal větší citlivost na prudké pohyby senzoru, které mají za důsledek velkou vzdálenost a pootočení po sobě jdoucích snímků, což může způsobit nižší přesnost určení změny polohy mezi těmito snímky. Z reálných dat byl program schopen věrně rekonstruovat celou trasu pohybu, a to až do vzdálenosti blízké se 25 metry, čímž byl dokonce mírně překonán očekávaný výsledek práce. Vizualizaci je možné vidět na obrázku 6.1, hranice mřížky označuje vzdálenost 20 metrů od stožáru. Na obrázku 6.2 je pak vidět také počátek špatného určení polohy v příliš velké vzdálenosti od stožáru.



Obrázek 6.1: Vizualizace rekonstruované trasy pohybu



Obrázek 6.2: Vizualizace rekonstruované trasy pohybu včetně špatných určení polohy ve větší vzdálenosti



Kapitola 7

Závěr

Předložená práce se zabývá problematikou lokalizace v prostředí stožáru vysokého napětí. Hlavním cílem práce bylo navržení algoritmu, schopného v datech z LiDARu identifikovat rovné části konstrukce stožáru vysokého napětí. Dalším cílem pak bylo zakomponování tohoto algoritmu do programu, ověřujícího možnosti určení polohy dronu ve 2D na základě nalezených konstrukcí stožáru.

V úvodu práce byla popsána motivace pro zkoumání této problematiky. V kapitole 2 následoval popis fungování LiDARu a popis senzoru OS0 od společnosti Ouster, použitého v této práci. Kapitola 3 byla věnována ukázce a analýze nasnímaných dat. V kapitole 4 byl následně představen navržený algoritmus pro identifikaci konstrukce stožáru vysokého napětí v nasnímaných datech. Kapitola 5 popisuje činnost programu, který ukazuje možnosti určování polohy dronu ve 2D na základě identifikované konstrukce stožáru vysokého napětí. Kapitola 6 pak analyzuje funkčnost programu v závislosti na vzdálenosti od stožáru. Program ukázal schopnost určení polohy v prostředí stožáru vysokého napětí, a to až do vzdálenosti blížíící se 25 metrům od konstrukce stožáru. Byly tak splněny všechny stanovené cíle a body zadání této bakalářské práce.



Literatura

- [1] Zákon o podmínkách podnikání a o výkonu státní správy v energetických odvětvích a o změně některých zákonů (energetický zákon), ve znění účinném k 1.1.2018. *458/2000 Sb.*
- [2] Statista 2022. Net electricity consumption worldwide in select years from 1980 to 2019. <https://www.statista.com/statistics/280704/world-power-consumption/>.
- [3] AERIAL-CORE. Project objectives. <https://aerial-core.eu/objectives/>, 2019.
- [4] Andeggs. Bod ve sférické soustavě souřadnic cc by-sa 4.0 <<https://creativecommons.org/licenses/by-sa/4.0>> via wikimedia commons. https://cs.wikipedia.org/wiki/Sf%C3%A9rick%C3%A1_soustava_sou%C5%99adnic#/media/Soubor:Spherical_with_grid.svg, 2008.
- [5] Carlos Barbosa, Carlos Severino, Edson Matsumoto, Liu Guijun, Lucila Labaki, Marcos Filho, Paulo Barbosa, Paulo Vatauvuk, Rogerio Camargo, and Tiago Forti. Windbreaks working as barriers to high speed winds for protection of electric power transmission lines and towers. *Engineering Failure Analysis*, 82, 06 2017.
- [6] Paul Besl and H.D. McKay. A method for registration of 3-d shapes. *iee trans pattern anal mach intell. Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14, 03 1992.
- [7] Biggerj1. Idea behind the iterative closest point algorithm, cc by-sa 4.0 <<https://creativecommons.org/licenses/by-sa/4.0>>, via wikimedia

- commons. https://en.wikipedia.org/wiki/Iterative_closest_point#/media/File:Idea_closest_point_algorithm.svg, 3 2020.
- [8] Point Cloud Library. Point cloud library documentation. <https://pointclouds.org/>.
- [9] Point Cloud Library documentation. Pcl overview. [http://library.isr.ist.utl.pt/docs/ros/wiki/pcl\(2f\)Overview.html](http://library.isr.ist.utl.pt/docs/ros/wiki/pcl(2f)Overview.html), 11 2011.
- [10] Konstantinos G. Derpanis. Overview of the ransac algorithm. 2010.
- [11] Wendong Ding, Shenhua Hou, Hang Gao, Guowei Wan, and Shiyu Song. Lidar inertial odometry aided robust lidar localization system in changing city scenes. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.
- [12] A. Diosi and L. Kleeman. Laser scan matching in polar coordinates with application to SLAM. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.
- [13] Ahmed Elhattab and Mostafa Rabah. Fachbeitrag rabah/el-hattab, investigating the impact of high voltage power lines on gps signal. 06 2011.
- [14] Ray Essen. Determining the speed of light. *Photonics Day - 21 October 2016*, 10 2016.
- [15] Philip Evans. Rotations and rotation matrices. *Acta crystallographica. Section D, Biological crystallography*, 57, 11 2001.
- [16] Andrew Filisetti, Andreas Marouchos, Andrew Martini, Tara Martin, and Simon Collings. Developments and applications of underwater lidar systems in support of marine science. 10 2018.
- [17] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. 24(6), 1981.
- [18] Joao Fonseca, António Baptista, Ma Martins, and João Paulo Torres. Distance measurement systems using lasers and their applications. *Applied Physics Research*, 07 2017.
- [19] Walker Jearl Halliday David, Resnick Robert. Fyzika iii: Elektřina a magnetismus. *Prometheus*, 2000.
- [20] Alberto Hata and Denis Wolf. Feature detection for vehicle localization in urban environments using a multilayer lidar. *IEEE Transactions on Intelligent Transportation Systems*, 17, 09 2015.
- [21] Peng Li, Ruisheng Wang, Yanxia Wang, and Wuyong Tao. Evaluation of the icp algorithm in 3d point cloud registration. *IEEE Access*, 8, 2020.

- [22] Msm. Fitted line with ransac, cc by-sa 3.0 <<http://creativecommons.org/licenses/by-sa/3.0/>>, via wikimedia commons. https://en.wikipedia.org/wiki/Random_sample_consensus#/media/File:Fitted_line.svg.
- [23] Andreas Nuchter, Kai Lingemann, and Joachim Hertzberg. Cached k-d tree search for icp algorithms. 2007.
- [24] Inc Ouster. <https://ouster.com/products/>, 2021.
- [25] Angus Pacala. How multi-beam flash lidar works. <https://ouster.com/blog/how-multi-beam-flash-lidar-works/>, 11 2018.
- [26] G. Silano, D. Liuzza, L. Iannelli, and M. Saska. A framework for power line inspection tasks with multi-robot systems from signal temporal logic specifications. In *SIDRA (Italian Society of Automatic Control), National Meeting*, September 2020.
- [27] Giuseppe Silano, Tomáš Báča, Robert Pěnička, Davide Liuzza, and Martin Saska. Power line inspection tasks with multi-aerial robot systems via signal temporal logic specifications. *IEEE Robotics and Automation Letters*, 6, 04 2021.
- [28] J.M. Silva and R.G. Olsen. Use of global positioning system (gps) receivers under power-line conductors. *IEEE Transactions on Power Delivery*, 17(4), 2002.
- [29] De Cian E. Sue Wing I. van Ruijven, B.J. Amplification of future energy demand growth due to climate change. *Nat Commun* 10, 06 2019.
- [30] ROS wiki. Bags. <http://wiki.ros.org/Bags>, 2 2022.
- [31] Wikipedia. Lidar animation. <https://en.wikipedia.org/wiki/Lidar#/media/File:LIDAR-scanned-SICK-LMS-animation.gif>.
- [32] Wei Xin and Jiexin Pu. An improved icp algorithm for point cloud registration. *Proceedings - 2010 International Conference on Computational and Information Sciences, ICCIS 2010*, 12 2010.
- [33] Jiaolong Yang, Hongdong Li, and Yunde Jia. Go-icp: Solving 3d registration efficiently and globally optimally. In *2013 IEEE International Conference on Computer Vision*, 2013.

Příloha A

Struktura přiloženého programu

Obsahem přílohy je balíček `loc`.

```
loc
├── include
│   └── loc
├── CMakeLists.txt
├── package.xml
├── identification.cpp
└── localization.cpp
```