



České Vysoké Učení Technické v Praze

Fakulta dopravní

Bc. Marek Pavelka

Detekce objektů v blízkosti vozidla pomocí LiDARu

Diplomová práce

Vedoucí práce: Ing. Václav Jirovský, Ph.D.

Obor: Inteligentní dopravní systémy

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta dopravní

d ě k a n

Konviktská 20, 110 00 Praha 1



K623Ústav bezpečnostních technologií a inženýrství

ZADÁNÍ DIPLOMOVÉ PRÁCE (PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení studenta (včetně titulů):

Bc. Marek Pavelka

Kód studijního programu a studijní obor studenta:

N 3710 – IS – Inteligentní dopravní systémy

Název tématu (česky): **Detekce objektů v blízkosti vozidla pomocí LiDARu**

Název tématu (anglicky): LiDAR Based Object Detection Near Vehicle

Zásady pro vypracování

Při zpracování diplomové práce se řiďte osnovou uvedenou v následujících bodech:

- Seznámení se s problematikou měření vzdálenosti pomocí LiDARu
- Návrh a výroba systému pro detekci překážek pomocí 1D LiDARu
- Návrh algoritmů pro detekci objektů
- Naprogramování aplikace pro vyhodnocení detekce objektů
- Otestování systému a algoritmů



Rozsah grafických prací: dle pokynů vedoucího diplomové práce

Rozsah průvodní zprávy: minimálně 55 stran textu (včetně obrázků, grafů a tabulek, které jsou součástí průvodní zprávy)

Seznam odborné literatury: A. Eskandarian, Handbook of Intelligent Vehicles, 2012
K. Klasing, D. Wollherr, M. Buss, A Clustering Method for Efficient Segmentation of 3D Laser Data, 2008
J. Levinson, Towards Fully Autonomous Driving: Systems and Algorithms, 2011

Vedoucí diplomové práce: **Ing. Václav Jirovský, Ph.D.**

Datum zadání diplomové práce: **31. července 2016**
(datum prvního zadání této práce, které musí být nejpozději 10 měsíců před datem prvního předpokládaného odevzdání této práce vyplývajícího ze standardní doby studia)

Datum odevzdání diplomové práce: **30. května 2017**
a) datum prvního předpokládaného odevzdání práce vyplývající ze standardní doby studia a z doporučeného časového plánu studia
b) v případě odkladu odevzdání práce následující datum odevzdání práce vyplývající z doporučeného časového plánu studia



.....
doc. Ing. Václav Jirovský, CSc.
vedoucí

Ústavu bezpečnostních technologií a inženýrství

.....
prof. Dr. Ing. Miroslav Svítek, dr. h. c.
děkan fakulty

Potvrzuji převzetí zadání diplomové práce.

.....

Bc. Marek Pavelka
jméno a podpis studenta

V Praze dne..... 31. července 2016

Poděkování

Rád bych poděkoval panu Ing. Jirovskému, Ph.D. za vedení práce, za nespočetné podněty a rady. Dále děkuji všem pracovníkům z Laboratoře speciálních projektů na FD ČVUT, kde diplomová práce vznikala, jmenovitě: panu Ing. Vondráčkovi, panu Ing. Honzíkovi, Ph.D., panu Fukarovi, panu Majeríkovi. Děkuji své přítelkyni Kateřině za obrovskou trpělivost. A především děkuji svým rodičům za morální a finanční podporu během celého studia.

Prohlášení autora práce

Nemám závažný důvod proti užívání školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).
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.

Dne

Podpis autora práce

Abstrakt

Byl navržen a vyroben systém na detekci překážek před vozidlem pomocí tří levných 1D lidarů, které byly nasazeny na krokové motory, aby tak mohly skenovat určitý výsek před vozidlem. Systém je řízen pomocí CompactRIO, v němž probíhá i zpracování dat z měření v reálném čase. Vyroběný systém je vhodný do intravilánu, kde se vozidla pohybují malými rychlostmi, jelikož lidary mají dosah 40 m. Byla navržena kalibrační metoda, která umožnila nastavit lidary vůči sobě tak, aby bylo možné fúzovat data z jednotlivých lidarů. Zároveň byl navržen algoritmus pro segmentaci dat z lidarů založený na eliptické prahové funkci a hledání nejbližších sousedů. Algoritmus byl otestován pomocí simulace a srovnán s jinými přístupy. Celý systém byl otestován při venkovním měření. Navržený systém detekoval překážky před vozidlem a potvrdil tak hypotézu, že lze použít více levných lidarů pro detekci překážek v intravilánu pomocí fúze dat z jednotlivých lidarů.

Klíčová slova

ADAS, Detekce objektů, Lidar, Segmentace, Shlukování

Abstract

A system for obstacle detection based on three low cost 1D lidars was designed and developed. The lidars were mounted onto stepper motors, so that the lidars could scan the area in front of vehicle. The system is controlled with CompactRIO, which also analyses measured data in real-time. The developed system is suitable for urban environment, where the vehicles move slowly, since the lidars can measure a distance up to 40 m. A calibration method was developed, so that the lidars position and pose could be adjusted in a way that could allow data fusion from the lidars. Moreover, a segmentation algorithm was proposed based on the elliptical threshold function and on the nearest neighbour search. The algorithm was compared in a simulation framework with other threshold functions. The system was tested in real use in the outside environment. During the testing the system of three lidars was able to detect obstacles in front of the vehicle. The results confirmed the idea of using more low cost lidars for obstacle detection in the urban environment via data fusion from the lidars.

Keywords

ADAS, Clustering, Object detection, Lidar, Segmentation

Obsah

1	Úvod	1
1.1	Cíle práce	2
1.2	Struktura práce	2
2	Lidar	3
2.1	Laserový skener	5
2.2	Solid-State Lidar	7
2.3	MEMS Lidar	11
2.4	Vnější vlivy na funkci lidaru	11
2.5	Srovnání s ostatními senzory měření vzdáleností	15
3	Algoritmy zpracování 2D dat z lidaru	17
3.1	Detekce vozovky	17
3.2	Parametrické metody	18
3.3	Shlukování	19
3.4	Detekce diskontinuity	21
3.5	Occupancy Grid	22
4	Hardware architektura	23
4.1	Lidar	24
4.2	Krokový motor	26
4.3	cRIO	26
4.4	Ostatní příslušenství	26
4.5	Komunikace	27
4.6	Technické parametry	29
4.7	Konstrukce	30
4.8	Umístění lidarů na vozidle	31
5	Software architektura	32
5.1	LabVIEW	33
5.2	Popis aplikace	33
6	Kalibrace	38
6.1	Korekce offsetu	38
6.2	Korekce polohy lidarů	38

6.3	Určení výšky lidarů od vozovky	39
6.4	Nastavení sklonu lidarů	39
6.5	Určení naklopení lidarů	40
6.6	Odhad nejistot	41
7	Detekce objektů	43
7.1	Pre-processing	44
7.2	Segmentace	47
7.3	Výsledky simulace	49
8	Testování	54
8.1	Test 1	54
8.2	Test 2	56
8.3	Komentář výsledků	60
9	Závěr	62

Seznam zkratek

- ACC** – adaptivní tempomat, (z angl. Adaptive Cruise Control)
- A/D** – Analog/digitální převodník
- ADAS** – Pokročilé podpůrné systémy řidiče (z angl. Advanced Driver Assistance Systems)
- APD** – Lavinová fotodioda (z angl. Avalanche Photo Diode)
- ASC** – Advanced Scientific Concepts, americká společnost
- CC** – Connected Components
- CCD** – Z angl. Charged Coupled Device, typ přijímače kamery
- CMOS** – Z angl. Complementary Metal Oxide Semiconductor, typ přijímače kamery
- DBSCAN** – Z angl. Density-based Spatial Clustering of Applications with Noise, shlukovací algoritmus
- DMA** – Z angl. Direct Memory Access, způsob čtení dat z paměti
- FIFO** – Z angl. First In First Out, typ paměti
- FPGA** – Programovatelná logická hradla, (z angl. Field Programmable Gate Array)
- HIL** – Z angl. Hardware-in-the-Loop, simulační a testovací technika
- I2C** – Z angl. Inter-Integrated Circuit, synchronní komunikační protokol
- KF** – Kálmánův filtr
- MEMS** – Z angl. Microelectromechanical System
- NI** – National Instruments, americká společnost
- OCDMA** – Optical Code Division Multiple Access
- OPA** – Optické fázové pole, (z angl. Optical Phased Array)
- OG** – Occupancy Grid
- PWM** – Pulzně šířková modulace (z angl. Pulse Width Modulation)
- RANSAC** – Z angl. Random Sample Consensus, regresní iterativní metoda
- RBNN** – Z angl. Radially Bounded Nearest Neighbour, shlukovací metoda
- RT** – Systém reálného času, (z angl. Real-time)
- SPAD** – Fotonová lavinová fotodioda, (z angl. Single Photon Avalanche Diode)
- SSL** – Z angl. Solid State Lidar
- SVD** – Singulární rozklad matice, (z angl. Singular Value Decomposition)
- SVM** – Support Vector Machine
- TTL** – Z angl. Transistor-Transistor Logic
- UART** – Z angl. Universally Asynchronous Receiver/Transmitter, zařízení umožňující sériovou asynchronní komunikaci
- UI** – Uživatelské rozhraní, (z angl. User Interface)

Seznam veličin

Veličina	Jednotka	Popis
ρ_{123}	$[^\circ]$	podélný sklon lidarů L1, L2 a L3
β_{12}	$[^\circ]$	příčný sklon lidarů L1 a L2
h_{123}	$[m]$	výška lidarů nad vozovkou
a_{12}	$[m]$	vzdálenost krajních lidarů od prostředního lidarů
R	$[m]$	naměřená vzdálenost bodu lidarem
λ	$[nm]$	vlnová délka lidarů, 905 nm
$\Delta\phi$	$[^\circ]$	velikost kroku motoru a zároveň rozlišení lidarů
f_{scan}	$[Hz]$	frekvence skenování

1 Úvod

V dnešní době je téměř nemožné si nevšimnout velkého zájmu o autonomní vozidla a pokročilé asistenční systémy (ADAS - z angl. Advanced Driver Assistance Systems) ze strany výrobců vozidel a výzkumných týmů. Za posledních deset let bylo představeno několik přístupů, jak dodat vozidlům větší autonomii, či inteligenci, které by zvyšovaly bezpečnost, spolehlivost a komfort vozidel. Jednotlivé přístupy se od sebe lišily mimo jiné tím, jaké senzory byly použity, což dále určuje to, jaké další metody pro zpracování dat a řídicí algoritmy se použijí a k jaké aplikaci je lze využít, např. kamery pro rozpoznání dopravních značek, nebo radar pro adaptivní tempomat. Posledních dvacet let se začalo experimentovat s použitím lidarů jako doplněk či náhrada dosavadních senzorů, které by měly umožnit dát vozidlům větší inteligenci.

Lidar (z angl. **L**ight **D**etection and **R**anging) je senzor pro bezkontaktní měření vzdálenosti pomocí laserového paprsku. Na základě doby letu paprsku k objektu a zpět je schopen vypočítat vzdálenost objektu. Lidar je už dlouhou dobu používán pro měření a průzkum zemského povrchu z letadla. Avšak na sklonku tohoto milénia začali vědci a vývojáři vyvíjet lidar, který by se použil pro vozidla, ať už jako náhradní alternativa k radaru, nebo jeho komplement. Jako první reálné použití lidarů pro vozidla lze považovat jeho použití pro vývoj adaptivního tempomatu v roce 1995 skupinou japonských vědců [68]. O deset let později byl lidar nasazen do autonomních vozidel v rámci soutěže *DARPA challenge*, která se poprvé konala v roce 2004 a v roce 2005 většina soutěžících vozidel používala lidar [61], kde role lidarů byla skenovat okolí vozidla a určovat oblast, kde je pro vozidlo bezpečné jet. Motivace k nasazení lidarů do vozidel je především jeho schopnost většího prostorového rozlišení než je radar schopen, čímž poskytne detailnější informaci o detekovaném objektu. Doposud ale jeho největší slabinou byla vysoká výrobní cena, která naprosto zabránila jeho většímu uplatnění v automobilovém průmyslu [8] a dlouhou dobu byl lidar používán jen ve vědeckých projektech. Až v posledních letech jeho cena dramaticky poklesla a automobilový výrobci začaly postupně nasazovat lidar do jejich vozidel, aby sloužil jako senzor umožňující funkce adaptivního tempomatu, nebo třeba systému zabraňujícímu srážce. V budoucnosti lze s lidary počítat jako s jedním ze senzorů, které jsou součástí autonomních vozidel.

Dosavadní vědecké projekty zabývající se použitím lidarů se soustředily buď na samotný vývoj lidarů, nebo na algoritmy detekce překážek, senzorickou fúzi lidarů a jiných senzorů, např. radaru, kamery, a následné řízení vozidla. Tyto vědecké týmy často měly drahé lidary např. od firmy SICK nebo Velodyne. Avšak tato diplomová práce byla založená na myšlence využití relativně levného lidarů Lidar Lite, který je hlavně využíván v robotice. Tento lidar je 1D, což znamená, že míří jedním směrem a tedy získává vzdálenost jen jednoho bodu v prostoru. V této práci se tedy řešil návrh a výroba systému, který by použil tento 1D lidar a fungoval jako 2D lidar, který by poskytoval vzdálenosti bodů ve skenovací rovině. Dále tato práce řešila algoritmy detekce překážek z naměřených dat.

1.1 Cíle práce

Pokud bychom vyjádřili jednotlivé úlohy autonomního řízení na percepci, plánování a řízení, pak tato práce se zaměřila na percepci, jež měla tři hlavní cíle.

1. Navrhnout, vyvinout a otestovat systém detekce překážek před vozidlem pro rychlosti do 50 km/h pomocí lidarů Lidar-lite. A tím ověřit myšlenku, zda-li je možné použít více levných lidarů, které by skenovaly okolí vozidla a pomocí fúze dat tak bylo možné detekovat překážky okolo (360°) vozidla .
2. Vyvinout aplikaci v LabVIEW, jež by řídila celý systém a umožnila zpracování dat v reálném čase pomocí CompactRIO (cRIO), která by byla prezentována na osobním počítači.
3. Navrhnout algoritmy, které by měly být schopny z malého počtu naměřených dat rozlišit od sebe jednotlivé objekty. Tyto algoritmy by měly být otestovány na reálných datech, případě i simulací.

1.2 Struktura práce

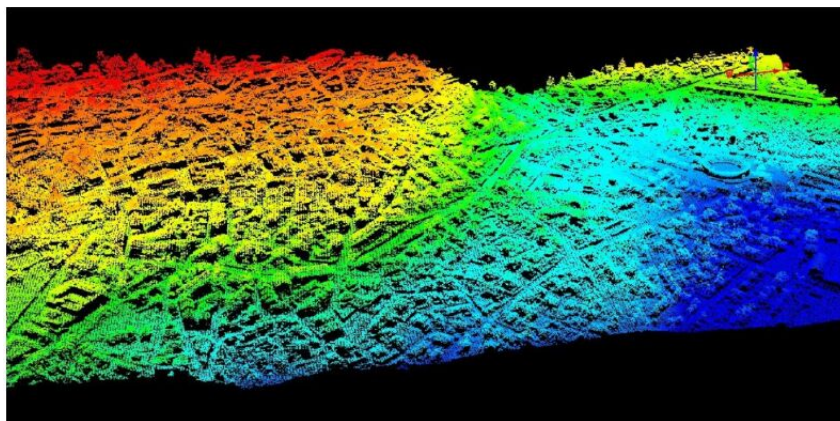
Práce je členěna následovně: sekce 2 shrnuje poznatky o lidarů, je zde popsán princip měření vzdálenosti, uvedené jednotlivé typy lidarů, závěr této části se zabývá různými vlivy na funkci lidarů a srovnáním lidarů s jinými senzory v automobilech používanými pro detekci překážek. Sekce 3 se věnuje popisu jednotlivých tříd algoritmů na zpracování dat z lidarů a detekci překážek. Sekce 4 je věnována popisu hardware architektury navrženého systému. Sekce 5 popisuje software architekturu aplikace a poskytuje manuál k použití. Kalibrace systému pro reálné použití je popsána v sekci 6. Navržený algoritmu pro detekci objektů popisuje sekce 7. Sekce 8 popisuje testování systému a shrnuje jejich výsledky. Sekce 9 zakončuje práci shrnutím výsledků a návrhy dalšího vývoje navrženého systému.

2 Lidar

Jak bylo v úvodu zmíněno, lidar je senzor pro bezkontaktní měření vzdálenosti založeném na měření doby letu světelného pulsu. Existuje velké množství druhů lidarů lišících se parametry, metodami zpracování signálů, konstrukcí aj., což má vliv na aplikace, pro které je daný druh lidarů vhodný. Lidary pro skenování zemského povrchu se používají už několik desítek let, ale ty se liší od lidarů používaných v automobilovém průmyslu např. vlnovou délkou laseru, místo pulzního laseru používají tzv. CW (Continuous Wave) světelný signál, jenž vysílá světelný signál nepřetržitě, také mají větší dosah.



Obrázek 1: Lidar pro skenování zemského povrchu – na spodní ploše trupu mezi křídly. Zdroj [9].



Obrázek 2: Výsledek skenování zemského povrchu laserovým skenerem. Zdroj [34].

Lidary se postupně začínají integrovat do současných vozidel. Vlivem zatím vyšší ceny je jeho uplatnění u vozidel vyšší třídy, kde se uplatňují v systémech ACC, v systémech pro vyhnutí a zabránění srážce. Jen pro představu Velodyne lidar HDL 64E stojí přes 70 000 USD. Nejlevnější lidar od Velodyne je Lidar Puck, jehož cena je 8000 USD [7]. Již vyráběny a také automobilovými výrobci používán je lidar

SCALA od firmy Valeo, jenž se prodává v řádu stovek USD [59]. Již dnes některé firmy pracují na tzv. Solid-state lidarů, jehož prodejní cena je údajně pod 200 USD, např. firma Quanergy. Pochopitelným požadavkem automobilových výrobců je to, aby lidar byl malý a kompaktní a šel elegantně skrýt např. za přední mřížku na kapotě vozidla. To dlouhou dobu nebylo možné, jelikož rozměry lidarů byly velké. I tento problém se ale v současnosti daří vyřešit a lze již dnes nalézt lidary malé velikosti, které lze do vozidel instalovat, aniž by byla snížena estetičnost vozidla. S tím jak se snižuje cena lidarů a jejich rozměr, lze očekávat jejich uplatnění v systémech pro detekci mrtvého úhlu, či jako parkovací asistenty. Lidar však nikdy nebude schopen zajistit všechny funkce asistenčních systémů, ale vždy bude nutné jej doplnit kamerovým a radarovým systémem.

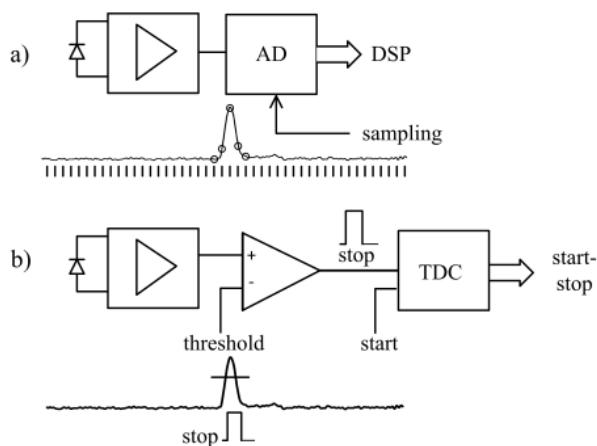
Základní dělení lidarů je na 1D, 2D a 3D. 1D lidar poskytuje vzdálenost jen jednoho bodu (toho, na který zrovna míří). 2D lidar je schopen nějakým způsobem laserový paprsek vychylovat a tím vytvářet skenovací rovinu a měřit vzdálenost více než jednoho bodu. 3D lidar dokáže měřit vzdálenost bodů ve více rovinách, je to takzvaný vícevrstvý lidar. Pochopitelně, aby mělo smysl použít lidar pro detekci překážek před vozidlem, je potřeba použít 2D nebo 3D lidar.

Základní 1D lidar se skládá z laserového zdroje vysílajícího světelné pulzy. Z optiky vysílače, která vytváří rovnoběžné světelné svazky. Z optiky přijímače, která směřuje odražený světelný signál na přijímač světelného signálu a z řídicí a výpočetní jednotky. Lidary používané pro automobilový průmysl měří vzdálenost na základě tzv. přímé metody měření času (dTOF - direct Time-of-flight). Laserový zdroj vysílá v krátkých intervalech (několik kHz) pulzy světla, které musí být krátké, aby energie signálu nepřekročila normou danou hodnotu, což zaručuje bezpečné použití v místech, kde se vyskytuje člověk (Eye Safe Class 1 Laser). Typická vlnová délka používaná pro lidary v automobilovém průmyslu je 905 nm a délka pulsu je v řádech nanosekund. Výjimečně trvá tento pulz déle, pak ale laserový zdroj musí být méně výkonný, aby byla norma dodržena, což snižuje vzdálenost, na kterou je lidar schopen měřit. Jakmile je pulz vyslán začne se měřit čas. Až se odražený signál vrátí zpět do přijímače lidarů, zastaví se měření času a z doby trvání Δt od vyslání po přijetí pulzu se vypočítá vzdálenost d pomocí známé rychlosti světla c jako:

$$d = c \cdot \frac{\Delta t}{2} \quad (1)$$

Přijímač odraženého laserového signálu je buď PIN foto dioda, nebo častěji lavinová foto dioda (APD – z angl. Avalanche Photo Diode), případně SPAD (Single Photon Avalanche Diode), což je APD pracující v Geiger módu a dokáže počítat jednotlivé přijaté fotony. Foto diody převádí světelný signál na elektrický signál. Dopadající fotony generují elektrický proud vzniklý pohybem elektronů a děr ve foto diodě. Následně je tento proud konvertován na napětí, které je dále zesíleno. Existují dva základní přístupy, jak je tento výstupní signál dále zpracován, obr. 3. Buď se tento signál vzorkuje A/D převodníkem, který

jej převádí na digitální signál. A/D převodník vzorkuje signál s určitou vzorkovací frekvencí. Čím je tato frekvence vyšší, tím přesnější je měření vzdálenosti. Např. 1 GHz odpovídá periodě 1 ns, za tuto dobu světlo urazí vzdálenost 30 cm. Pokud je požadovaná vyšší přesnost v jednotkách cm, pak musí být i frekvence převodníku vyšší [60]. Druhý přístup využívá komparátor, do nějž je vstupní signál přiveden. V něm se porovnává vstupní napětí s prahovou hodnotou, která je-li překročena, je vyslán pulz do TDC (Time-to-Digital Converter), který zastaví měření času [1, 22, 29].

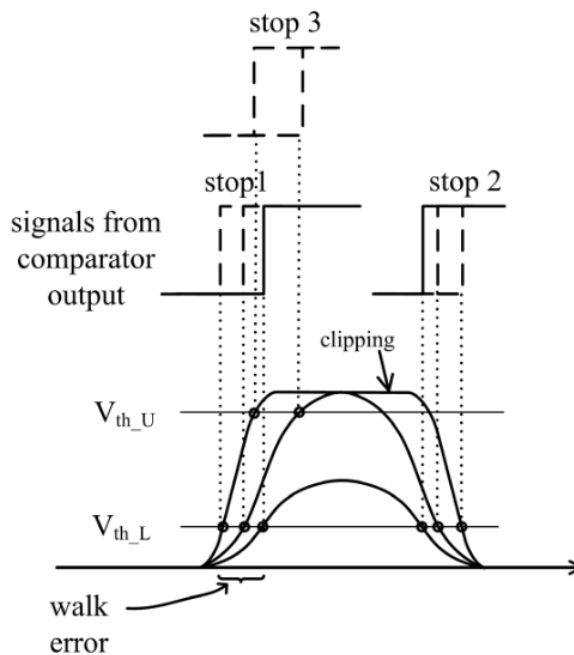


Obrázek 3: Dva přístupy zpracování výstupu z foto diody. a) vzorkování výstupního signálu. b) detekce pulzu založená na detekci události. Zdroj [29].

Jeden z problémů u komparátoru je, že vstupní signál není dokonalý obdélník, ale je zašuměný, deformovaný a má různé velikosti amplitudy v závislosti na vzdálenosti objektu či jeho reflexivitě. Což způsobuje, že část tohoto napěťového pulzu, která je nad prahovou hodnotou, může být různě dlouhá a náběžná hrana může být různě strmá, což způsobuje tzv. *timing walk error*, obr. 4, jež zhoršuje přesnost určení počátku překročení prahové hodnoty, což zvyšuje nepřesnost měření (chyba v určení času o 1 ns odpovídá chybě 30 cm). Např. Kurtti et al. v [29] řeší tento problém tak, že používají 2 komparátory každý s jinou hodnotou prahového napětí, obr. 4. Pomocí dvou komparátorů tak zjišťují dobu vzestupné hrany a délku pulzu. Tyto dvě hodnoty pak slouží ke kompenzaci *timing walk error* a jsou tak schopni snížit chybu v časové oblasti až na 50 ps.

2.1 Laserový skener

Aby bylo možné získat prostorovou informaci o tom, co se nachází v okolí lidarů, musí se nějakým způsobem paprsek laseru vychylovat, směřovat známým směrem – úhel natočení musí být znám k správnému přiřazení naměřené vzdálenosti konkrétnímu bodu v prostoru. V současnosti existují dva přístupy, jak směřovat laserový paprsek u laserových skenerů.



Obrázek 4: Ilustrace problému s detekcí pulzu v komparátoru. V_{th} je prahové napětí. Zdroj [29].

Pohyblivý laser

Tato kategorie lidarů vychyluje paprsek mechanickým natočením laserového zdroje uchyceném na elektromotoru, ať už krokovém motoru nebo DC motoru s dekodérem polohy. Ze známého natočení motoru je znám úhel, kam laser míří, čímž je získána informace o poloze bodu v prostoru – úhel a vzdálenost. Na tomto principu je navržen i systém v této práci, nebo Velodyne HDL-64E (obr. 6), Velodyne Puck (obr. 5). I při použití DC motoru, kdy se motor otáčí kontinuálně, není problém s příjmem odraženého paprsku od překážky. Doba od vyslání po přijetí světelného signálu je tak krátká, že za tu dobu se lidar s přijímacím ústrojím pootočí jen o malý úhel, a tak podstatná část odraženého paprsku má šanci dopadnout do přijímače.

Velodyne Puck obsahuje až 16 páru vysílač/přijímač. Je schopen skenovat 360° v horizontální rovině s úhlovým rozlišením 0.1° a 20° ve vertikální rovině s úhlovým rozlišením 1.33° . Měří na vzdálenost 100 m s přesností 3 cm. Během jednoho skenu naměří až 300 000 bodů. Frekvence skenu je od 5 do 20 Hz. [65]. Velodyne HDL-64E lidar má ve svém ústrojí až 64 páru vysílač/přijímač. V horizontální rovině skenuje prostor v rozsahu 360° s úhlovým rozlišením 0.08° . Ve vertikální rovině skenuje výše 26.9° s úhlovým rozlišením 0.35° . Je schopen detekovat objekty na vzdálenost 120 m s přesností na 2 cm. Během skenu je schopen vyprodukovat až přes 1 milion bodů. Frekvence skenu je od 5 do 20 Hz [63]. Jeho největší nevýhodou pro větší použití v automobilovém průmyslu je kromě rozměru jeho cena, jež je přes 70 000 USD [7].

Statický laser s pohyblivým zrcadlem

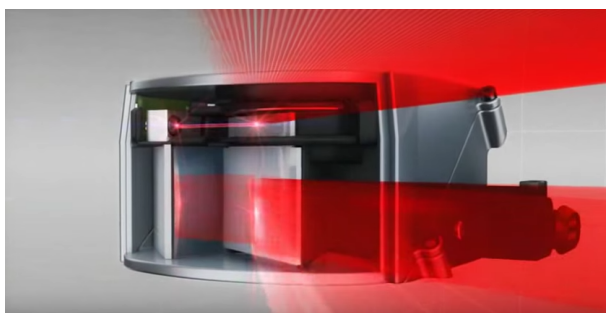
V tomto případě je laser statický a míří na zrcátko, jež se konstantní rychlostí otáčí a vlivem synchronizace otáček zrcátka a pulzů laseru je vyslaný laserový paprsek vychýlen o známý úhel natočení zrcátka. Zrcátko je přichyceno na elektromotor. Příkladem tohoto typu lidaru jsou lidar SCALA od firmy Valeo (obr. 7), lidary SICK, nebo LUX lidar od firmy Ibeo (obr. 8). Zrcátko musí být vysoce reflexivní, aby se minimalizovala absorpce. Na obr. 7 lze vidět, jak laser ostřeluje malé zrcátko, které kontinuálně rotuje, a zpět dopadá odražený paprsek na větší zrcátko, jež směřuje paprsek na detektor (APD). SCALA i LUX lidary jsou již navrženy tak, aby je šlo zaintegrovat do vozidla a propojit s dalšími systémy ve vozidle přes CAN, či FlexRay sběrnici.



Obrázek 5: *Velodyne Puck lidar. Zdroj [64].*



Obrázek 6: *Velodyne HDL-64E lidar. Zdroj [62].*



Obrázek 7: *SCALA lidar od firmy Valeo. Zdroj [73].*



Obrázek 8: *LUX lidar of firmy Ibeo. Zdroj [4].*

2.2 Solid-State Lidar

Solid-State Lidar (SSL) je nejnovější typ lidaru v současnosti vyvíjen. Neobsahuje žádnou pohyblivou část. Existují dva typy lidarů tohoto druhu lišících se podle toho, jakým způsobem směřují paprsek. Výhodami SSL jsou nižší rozměry, nižší výrobní náklady a delší životnost, jelikož neobsahují žádnou

pohyblivou část [71]. Dosavadními problémy byly užší zorné pole, krátká měřitelná vzdálenost a pomalé měření. Avšak i tyto problémy se daří postupně odstraňovat.

OPA

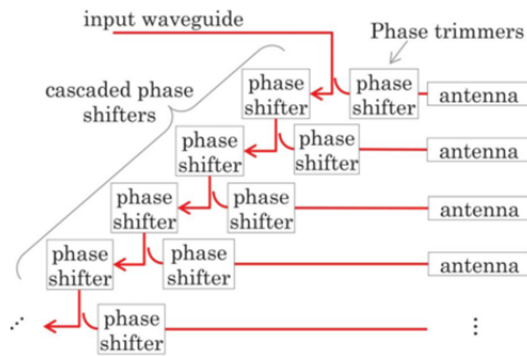
Laserový paprsek je zde směřován pomocí tzv. optického fázového pole (optical phased array) [36]. Hulme et al. v [11] popisují jejich architekturu SSL, kde koherentní úzký laserový paprsek je rozdělen v multi modálním interferometru na více svazků, které jsou přivedené do optických vláken, kde jsou jednotlivé svazky různě zpomalené. Toto zpomalení je způsobeno např. řízeným zahřátím jednotlivých optických vláken, čímž se změní jejich optické vlastnosti, např. index lomu. Tyto různě zpomalené paprsky jsou pak přivedeny dohromady v demodulátoru, z něž vystupuje opět jeden paprsek, ale vlivem fázového posunu jednotlivých sub-svazků se výstupní paprsek pohybuje již vychýleným směrem. Řízení úhlu v druhém, kolmém směru se realizuje pomocí řízení vlnové délky laseru. Tím je možno docílit směřování laserového paprsku jak v horizontálním, tak vertikálním směru.

Yaacobi et al. [71] navrhli SSL, které je oproti [11] rychlejší a méně energeticky náročné, zároveň řízení směru paprsků je jednodušší, jelikož používají pouze jeden řídicí signál a velikost napájecího napětí není závislá na počtu vysílacích antén, což umožňuje jednoduché rozšíření architektury o další antény. Avšak oproti [11], jejichž design je 3D SSL (řídí paprsek ve vertikálním a horizontálním směru), je SSL od Yaacobi 2D (řídí paprsek v horizontálním směru). Avšak tvrdí, že lze na tentýž čip přidat další optická pole s jiným nastavením optického filtru, čímž se ovlivňuje propustnost jednotlivých vlnových délek, což umožní vytvoření více skenovacích rovin. Jejich SSL má navíc zorné horizontální pole 51° , což je údajně v té době nejširší zorné pole, kterého zatím bylo dosaženo.

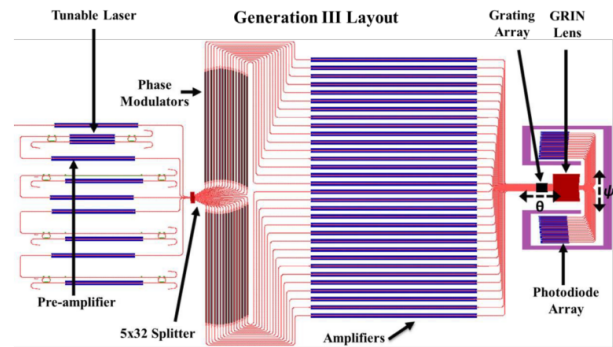
Jiný přístup k modulaci fáze světelného signálu je pomocí likvidních krystalů, které se vloží do optického vlákna a dle potenciálu elektrického pole okolo vlákna se natáčí tyto krystaly, které tak prodlužují optickou dráhu, čímž je dosaženo požadovaného fázového posunu. Detailněji o tom pojednává [36, 58, 66]. Nevýhodou této modulační techniky je, že změna natočení krystalů trvá v řádu desítek milisekund [37], což snižuje frekvenci skenu.

Oproti pohyblivým lidarům je SSL mnohem menší. Cena je mnohonásobně nižší, přední výrobci tohoto typu uvádějí hodnoty mezi 50 až 200 USD, což je podstatný rozdíl oproti lidarům od Velodyne, jež se pohybují v řádech tisíců dolarů za kus. Tento fakt dělá z těchto lidarů cenově, rozměrově a výkonnostně přijatelný výrobek, jenž automobilový výrobci začnou brzy instalovat do svých vozidel. Příkladem tohoto typu je lidar S3 od firmy Quanergy (obr. 11).

Další výhodou SSL s OPA detektorem je jejich schopnost směřovat paprsek do libovolného směru s různým rozlišením. Místa s očekávaným výskytem překážek (např. zjištěných po předešlém skenu) mohou



Obrázek 9: *Yaacobi design SSL*. Zdroj [71].



Obrázek 10: *Hulme design SSL*. Zdroj [11].

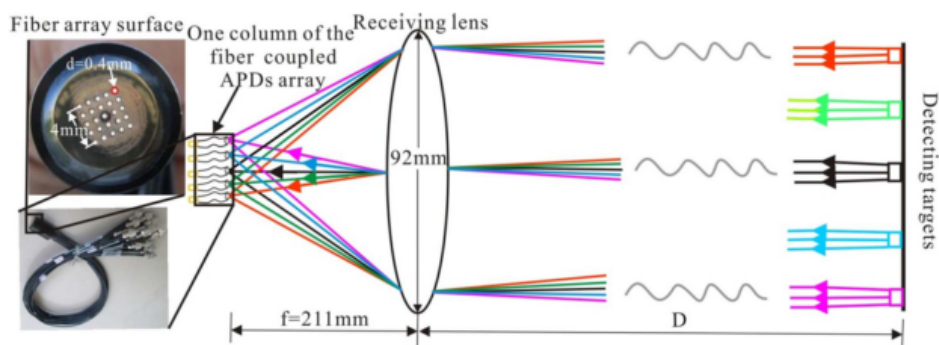


Obrázek 11: *Quanergy S3 lidar*. Zdroj [48].

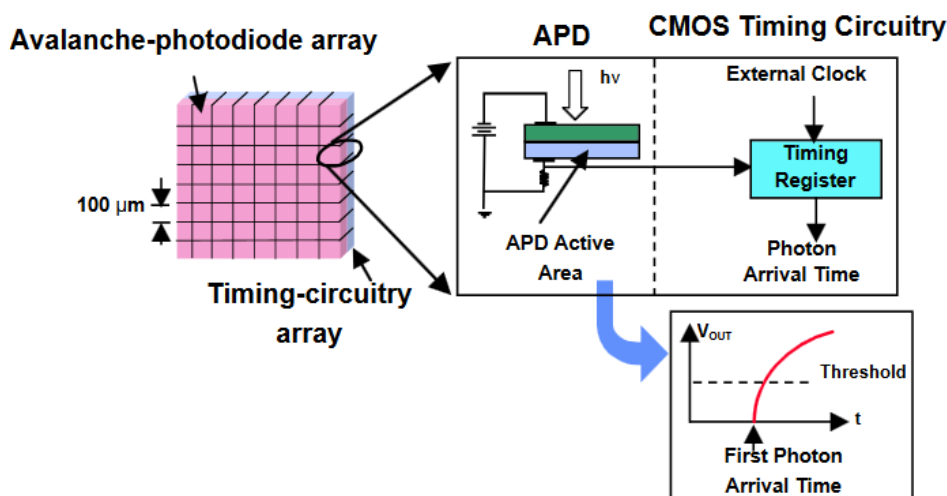
být skenována s mnohem větším rozlišením a prázdná místa s menším, čímž lze efektivně využít potenciál lidarů během jednoho skenu [37].

Flash 3D lidar

Flash lidar je unikátní v tom, že zatímco všechny předešlé typy lidarů potřebovaly nějakým způsobem směřovat laserový paprsek a tak skenovat prostor před lidarem, tento typ lidarů osvětluje širším paprskem větší prostor před lidarem a jednotlivé odražené paprsky od překážek pak dopadají na pole, případně maticí detektorů (většinou SPAD). Paprsek z laseru je optikou rozšířen a kolimován. Odražené paprsky se poté vracejí a skrz čočky je přiveden na jednotlivé detektory, viz. obr. 12. SPAD jsou APD, které pracují v tzv. Geiger módu, kdy je možno touto diodou počítat jednotlivé fotony. Tyto diody jsou na čipu integrovány s digitálním obvodem pro velmi přesné měření času. Není zde potřeba A/D převodníku mezi diodou a digitálním obvodem, jelikož výstupní napětí generováno diodou je přímo v úrovních 5 V (TTL) případně 3.3 V, čímž jsou schopny trigovat časovací obvod, obr. 13. Rozlišení a zorné pole je dáno počtem APD detektorů v poli.



Obrázek 12: Optický systém přijímače G Lidaru II. Zdroj [76].



Obrázek 13: APD pole v Geiger módu spojené s CMOS časovacím obvodem. Zdroj [19].

Laserový zdroj vysílá opět pulzy o určité energii, jejichž trvání je v řádu nanosekund. Jelikož je tento laserový svazek rozptýlen na větší plochu, jednotlivé odražené paprsky, které dopadají na detektory, mají o dost nižší energii, což může vést k chybnému měření, proto se často provede několik sérií osvětlení v řádu milisekund a výsledná měření jednotlivých detektorů se průměrují, čímž se eliminuje šum a chyby [22]. Velkou výhodou flash lidarů má být to, že měření není ovlivněno pohybem vozidla, na kterém je lidar umístěn. Jelikož u laserových skenerů vlivem pohybu dochází k distorzi měření právě proto, že sken trvá určitou dobu, za kterou vozidlo urazí určitou vzdálenost. Proto je snaha zkrátit dobu osvětlení u flash lidarů, aby nedocházelo k této distorzi a mohla se maximálně využít výhoda této technologie.

Flash lidar byl prvotně používán pro letecké průzkumy Země a armádou či NASA, jelikož první takové lidary byly velké a drahé. Až v posledních letech se začaly vyvíjet takové lidary, které by bylo možné použít v automobilovém průmyslu. V článkách [1, 3, 19, 33, 43, 57, 76, 77] lze najít detailní popis jednotlivých návrhů flash lidarů a jejich aplikace. Jedním z hlavních komerčních výrobců flash lidarů je firma ASC (Advanced Scientific Concepts), jež úzce spolupracuje s americkou armádou a NASA, jimž

poskytuje flash lidary pro autonomní přistání na vesmírných tělesech. Jednotlivé testy pro tyto účely popisují [3, 24]. ASC zahájilo v roce 2015 spolupráci s firmou Continental a společně vyvíjejí flash lidar pro automobilový průmysl [6]. Mezi další výrobce flash lidarů patří např. firma LeddarTech.

2.3 MEMS Lidar

MEMS je z angl. Microelectromechanical Systems. U MEMS lidarů je tímto MEMS systémem myšlen systém malých zrcátek (velikost od stovek nanometrů po jednotky milimetrů), které jsou elektronicky rozvibrovány tak, aby dokázaly laserový paprsek směřovat požadovaným směrem. Zde se odlišují dva systémy. První obsahuje jedno malé zrcátko, které vibruje, a paprsek je od něj odrážen. Druhý připomíná OPA. Obsahuje pole (matici) několika malých zrcátek, které se různě natáčejí, či vysouvají, tím dochází k fázovému posunu celé světelné vlny a k změně směru paprsku. Příkladem prvního přístupu jsou např. [26] s horizontálním zorným polem 240° a vertikálním 40° s rychlostí skenu až 50 ms a dosahem 30 m. Jako další tohoto prvního typu popisují: [21, 23, 54, 72].

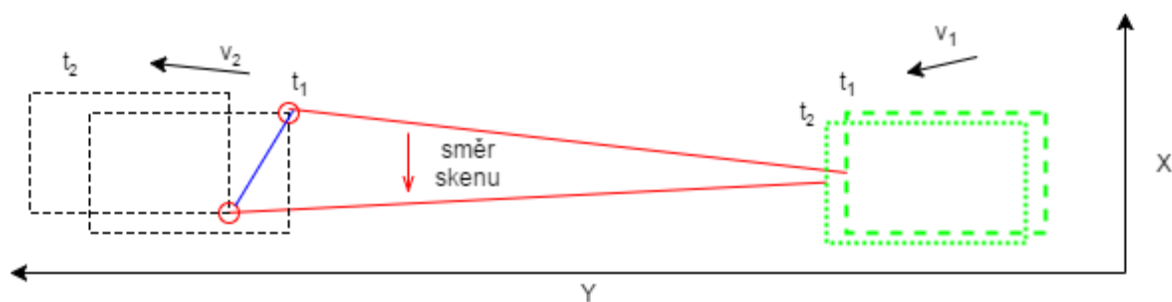
Mnohem zajímavější jsou však MEMS druhého typu. U tohoto typu jsou použity několik malých zrcátek o rozměrech v řádech stovek nanometrů. Jsou mnohem rychlejší a jsou napájeny nižším napětím. Např. Wang a Wu v [67] představili systém, který k pohybu zrcátek potřebuje jen 10 V a rychlost odezvy je 2 μ s s dosahem 100 m. Velikost zorného pole závisí na vlnové délce laseru. Používali dvě, 905 nm odpovídá 22° a 1550 nm odpovídá 40° . Megens et al. navrhli odlišný systém s 32 x 32 matici MEMS zrcátek. Používají také vlnovou délku 1550 nm, dosahují také rychlosti odezvy okolo 2 μ s, avšak k řízení zrcátek potřebují 34 V [37].

2.4 Vnější vlivy na funkci lidarů

Jako každý senzor i lidar je ovlivněn vnějším prostředím, ve kterém funguje. S těmito vlivy se při vývoji lidarů a jeho následným používáním musí počítat. Mezi tyto vlivy patří vliv počasí, konkrétně jak je ovlivněna schopnost lidarů měřit vzdálenost za deště, sněžení, či mlhy. Pohyb vozidla, ve kterém je lidar, také ovlivňuje měření, ani ne tak přesnost měření, jako přesnost určení polohy a rychlosti objektů. Do optického senzoru lidarů nevstupuje pouze odražený laserový paprsek, ale i světlo ze Slunce a protijedoucích vozidel, což způsobuje zašumění vstupujícího světelného signálu. S tímto vlivem se lze vypořádat buď už na úrovni senzoru přijímače, nebo následným zpracováním signálu, nebo obojím. Tyto vlivy jsou níže popsány.

Vliv pohybu na skenování

Tzv. Motion Scanning Effect je problém, se kterým je nutno počítat v případě, že k detekci je použit lidar, kterému trvá určitou dobu, byť krátkou v řádech desítek milisekund, pořídit sken. Takže se to týká všech typů lidarů, kromě flash lidarů. Jelikož se vozidlo a jeho okolí během jednoho skenu pohybují dochází ke zkreslení měření a vzniká systematická chyba. Proto je nutné s tímto efektem počítat a ještě než se data budou zpracovávat je potřeba je předzpracovat tím, že se všechny měřené body transformují do jednoho referenčního souřadného systému, aby vznikl "obraz", jako kdyby se vozidlo nepohybovalo a měření všech bodů proběhlo v jeden okamžik.



Obrázek 14: Ilustrace motion scanning efektu. Zelené vozidlo vybavené laserovým skenerem jede rychlostí v_1 , před ním jede černé vozidlo rychlostí v_2 . V čase t_1 detekuje poprvé lidar vozidlo a v čase t_2 jej detekuje naposledy. Za tu dobu se obě vozidla posunula do nových poloh. Jelikož rozdíl rychlostí těchto dvou vozidel není nulový, vznikne zkreslení tvaru zádě černého vozidla, jelikož jednotlivé měření mezi časy t_1 a t_2 odpovídají modré linii u černého vozidla.

Jen pro představu, jak důležité je dělat tuto korekci, si představme situaci, kdy vozidlo jede po silnici rychlosti 90 km/h (25 m/s) a detekuje vozidlo v protisměru jedoucí toutéž rychlostí, jejich relativní rychlost je tedy rovna 50 m/s. Pokud jeden sken trvá např. 40 ms (jak je to např. u SCALA lidarů), pak rozdíl vzdálenosti mezi prvním měřeným bodem a posledním bodem je 2 m. To nejen vytváří problém s určením polohy a tvaru objektu, ale i s určením jestli je tento objekt statický nebo pohyblivý. Tento problém je tím větší, čím větší je relativní rychlost a čím déle trvá sken.

Většinou se body posouvají v čase dopředu do času na konci skenu, nebo se průměrují na čas uprostřed skenu. Aby taková korekce byla možná, je nutné znát rychlost vozidla s lidarem a jeho natočení. Největší problém představují pohyblivé objekty, které ještě systém v předešlém skenu neviděl a nemá je tak obsažené v trackovacím listu. Jakmile však systém dokáže trackovat objekty, zná jejich rychlost a natočení vůči vozidlu s lidarem a je schopen mnohem přesnějších korekci [50]. Tato časová korekce je nezbytná, pokud má dojít k fúzi lidarů s kamerou, jelikož kamera zaznamená scénu v jeden okamžik. A tak je nutné, aby naměřené body lidarů byly posunuty v prostoru tak, aby odpovídaly bodům z kamery

[53]. Další způsob kompenzace relativního pohybu vozidla a překážek poskytuje Gröll v [16], kdy se snažili vytvořit matematický model situace a simulovat vzájemný pohyb vozidel a schopnost jimi navržené kompenzační techniky. Zároveň se zaměřili na identifikaci modelových parametrů, kdy zjistili, že všechny jsou možné být zjištěny z jednoho skenu.

Vliv atmosféry a počasí

Při průchodu signálu z lidarů vzduchem je tento signál utlumován. Dle rovnice 2 [17] lze vyjádřit energii signálu ve vzdálenosti R od zdroje jako:

$$P_R = \frac{P_T \tau_E \tau_R \tau_F A_R \rho \cos \beta}{\pi R^2} e^{-2 \int_0^L \alpha(r) dr} \quad (2)$$

Kde P_T je energie vyslaného signálu, τ_E je spektrální transmitance optiky vysílací části, τ_R je spektrální transmitance optiky přijímače, τ_F je spektrální transmitance úzkopásmového filtru, ρ je koeficient reflexivity objektu, A_R je obsah plochy přijímače, β je úhel mezi normálovým vektorem plochy objektu a optickou osou přijímače a $\alpha(\lambda, r)$ je koeficient atmosférické absorpce v závislosti na vlnové délce lidarů λ , vzdálenosti r a na částicích nacházejících se v atmosféře (dešť, mlha, atd.).

Kromě útlumu způsobeného atmosférou vytváří problémy částice nacházející se v atmosféře, jako je prach, sněh, dešť, mlha. Je známo, že světlo na optickém rozhraní dvou materiálů (oblastí) se láme, odráží a část se může absorbovat. Proto je tento problém u lidarů nutné řešit. Při dešti, mlze či sněžení totiž může nastat několik situací, co se s daným paprskem stane, když narazí na vodní kapku. Pokud projde a zlomí se, může se stát, že značná část paprsku se nevrátí zpátky do snímače, což způsobí sníženou schopnost detekce. Stejně problematická je situace, pokud se významná část energie paprsku odrazí od vodní kapky zpět do přijímače. To totiž způsobí to, že je detekována jako překážka, a pokud algoritmus na zpracování dat z lidarů s deštěm neumí dobře pracovat, může dešť činit nemalé problémy při zpracování dat z lidarů, aby nevytvářel falešné objekty. Velký problém také představuje nečistota, námraza, či voda na krytu lidarů, z něž vychází laserový paprsek a do nějž se také vrací. Stav krytu musí být monitorován, aby případná nečistota nezpůsobovala chybnou funkčnost. Lidar musí být schopen toto znečištění detekovat a případně řidiče varovat, že není schopen své funkce a tudíž návazné asistenční systémy nelze použít.

Fersch et al. v [12] zkoumali vliv deště na odrazivost a absorpci laserového paprsku. Jejich přístup je čistě simulační s mnoha zjednodušeními. Ze simulačních výsledků vyplynulo, že ztráta energie průchodem paprsku je zanedbatelná. Bohužel neřešili hlavní problém, což je, že jednotlivé kapky mohou vytvářet falešné objekty a tvářit se tak, jako překážka.

Filgueira et al. v [14] uskutečnili reálný pokus s lidarem Velodyne VLP-16. Zkoumali intenzitu vráče-

ného paprsku, počet úspěšných měření a přesnost v závislosti na různých typech povrchů. Z výsledků vyplývá, že čím je intenzita deště větší, tím nižší je intenzita odraženého paprsku, počet úspěšných měření klesá (hlavně u silnice a chodníku). Avšak dle výsledku měření není jasná korelace mezi intenzitou deště a přesnosti měření.

Rasshofer et al. v [49] provedli reálné měření vlivu deště na schopnosti detekce lidarů a následně vytvořili i testovací HIL (z angl. Hardware-in-the-loop) systém. Při jejich testech zjistili, že s rostoucí intenzitou deště klesá měřitelná vzdálenost lidarem a roste počet falešných detekcí. Uvádějí, že s deštěm se lze vypořádat softwarově, což ale má neblahý důsledek na délku zpracování, nebo lze upravit hardware obvodu přijímače, což ale vede ke složitějšímu návrhu a nárůstu ceny.

Interference s jinými lidary

Na optický přijímač lidarů může dopadat kromě odraženého paprsku a okolního světla i laserový signál od jiného lidarů. Což může způsobovat falešnou detekci. Avšak existují způsoby, jak se s takovým cizím signálem vypořádat. Např. lidar SCALA měří výšek 145° v rozlišení 0.25° [74], to znamená, že za jeden sken naměří 580 bodů. Jeden sken trvá 40 ms, z toho plyne, že na jedno měření připadá cca 68.97 μ s. Tento lidar vysílá laserový puls trvající 1 metr světla, tedy přibližně 3.33 ns. Po té se počítá doba než se vrátí do přijímače, jež je závislá na vzdálenosti objektu. Teoreticky je SCALA schopna detekovat objekty až na vzdálenosti 350 m. To znamená, že světlo musí urazit tam a zpátky vzdálenost 700 m. Při přibližné rychlosti světla 300 000 km/s uplyne přibližně doba 2.336 μ s než se vyslaný puls vrátí zpátky do přijímače, což znamená, že za 2.336 μ s od počátku pulsu musí přijít odražený signál zpět. Pak se čeká cca 66.5 μ s než se vyšle další puls. Pokud během této doby přijímač zaznamená jiný signál, je jasné, že musí jít o jiný signál než z vlastního laseru. Avšak existuje zhruba 3.3 % pravděpodobnost, že během doby čekání na odražený signál může přijít do přijímače signál z jiného lidarů za předpokladu, že se v okolí lidarů vyskytuje jeden cizí lidar.

Zajímavou myšlenku, jak se vypořádat s interferencí lidarů s jinými lidary, nabízí Fersch et al. v [13]. Navrhují modulační techniku OCDMA (Optical Code Division Multiple Access) založenou na On-Off Keying principu. Vyslaný puls je zakódován do pulsové sekvence, takže pokud přijde jiná sekvence než byla vyslána, znamená to, že lidar přijal signál od cizího lidarů. Tím, že laserový puls může trvat jen krátkou dobu, aby splňoval bezpečnostní kritéria, je délka zakódovaného signálu omezena a ne všechny kódové kombinace (slova) jsou vhodná. To znamená, že může existovat více laserovým skenerů s téže modulační sekvencí, což nezaručuje nulovou pravděpodobnost interference, avšak výrazně ji snižuje k nule.

Kim et al. v [25] testovali vzájemnou interferenci dvou lidarů SICK od sebe vzdálených 1.2 m umístě-

ných vedle sebe. Interference nastala pouze u 0.05% úhlů, avšak nejdelší souvislá interference obsahovala 25.5°, což není zanedbatelné. Tato metoda testovala interferenci odražených paprsků (ne přímých), které se hůře detekují jako cizí paprsky. Situace, že se dva lidary vyskytnou v této konstelaci vedle sebe, je pokud vozidla jedou vedle sebe na více proudové vozovce.

2.5 Srovnání s ostatními senzory měření vzdáleností

Kromě lidarů se v automobilovém průmyslu používají na detekci objektů i jiné senzory: radar, ultrazvukový senzor, kamera. Každý z těchto senzorů má jiné vlastnosti, které předurčují jejich užití pro danou aplikaci. Lidar není dominantní senzor pro detekci, současné systémy a budoucí systémy jsou postaveny spíše na fúzi jednotlivých senzorů, velmi častá je fúze lidar-kamera, radar-kamera a i lidar-radar-kamera, čímž se robustnost celého systému umocňuje. Zatímco lidar, radar, ultrazvukový senzor jsou aktivní senzory, tj. vyzařují určitou formu energie za účelem detekce jejího odrazu od překážky, kamera je pasivní senzor, který je závislý na světelných podmínkách okolí, ve kterém se vyskytuje.

Lidar vs. Radar

Lidar ve srovnání s radarem má lepší prostorové rozlišení a určení polohy, také přesnost měření vzdálenosti je větší. Také zorné pole lidarů je širší, většinou přes 100°, zatímco tzv. *Long Range Radar* má zorné pole okolo 12° a tzv. *Short Range Radar* má zorné pole okolo 70° [2]. Radar je více náchylný na interferenci od ostatních radarů než lidar [15]. Lidar lépe detekuje chodce než radar [8]. Výhodou radaru je, že dokáže při jednom měření zjistit vzdálenost překážky a rychlost překážky, zatímco lidar potřebuje 2 až 3 měření k určení rychlosti a směru překážky. Funkce radaru je také méně ovlivněna špatným počasím jako je déšť, mlha, sníh. Radar je stále zatím levnější než lidar, avšak pokud se podaří vylepšit SSL, pak takový lidar může být i levnější než radar. Widmann et al. v [70] porovnávali adaptivní tempomat (ACC) založený jednou na radaru a podruhé na lidarů. Dospěli k závěru, že ACC s lidarem trvalo o trochu delší dobu inicializovat trackování objektů právě vlivem toho, že lidar potřebuje více skenů na zjištění rychlosti objektů. Avšak z výsledků vyplynulo, že toto zdržení nemělo vliv na funkčnost ACC a rozdíly ve funkčnosti jsou pro řidiče nepostřehnutelné.

Lidar vs. Ultrazvuk

Ultrazvukové senzory se používají zejména jako parkovací senzory, jelikož jejich detekovatelná vzdálenost je pod 3 m [2]. Na tuto vzdálenost dokáží velmi přesně detekovat překážky zejména flash lidary, avšak jejich cena je stále vyšší než ultrazvukového senzoru. Amditis et al. v [2] předpovídá, že v bu-

doucnosti bude ultrazvukový senzor používán jako parkovací asistent pouze u levnějších vozů, zatímco vozidla vyšší třídy budou používat lidar a kameru.

Lidar vs. Kamera

Kamerové systémy v automobilech lze dělit na ty, které jsou schopny měřit vzdálenost a na ty, které nejsou toho schopny. Infra, CCD a CMOS kamery jsou příkladem druhého typu kamer. Na rozdíl od lidaru, který je schopen poskytnout intenzitu přijatého paprsku, kamera poskytuje barvy, kontury a ostřejší tvary. Je proto vhodná pro čtení dopravních značek, vodorovného dopravního značení a klasifikaci objektů. Proto se lidar a kamera fúzí, kdy lidar poskytne docela přesné vymezení oblasti, kde je překážka a udá i její vzdálenost, a algoritmus na zpracování obrazu z kamery se už zaměří jen na tyto regiony vytipované lidarem, čímž určuje, zdali jde o člověka, auto, nebo jinou překážku. Pochopitelně tato fúze klade velký požadavek na správnou a přesnou kalibraci lidaru a kamery.

Kamerové systémy založené na stereovizi, kdy vzdálenost je měřena ze dvou např. CCD kamer, jsou schopny určit vzdálenost objektu na vzdálenosti do 50 m [2]. To z něj dělá vhodného kandidáta pro aplikaci na detekci chodců, systému pro vyhnutí se srážce, ale i jako parkovací asistent, či pro systém automatického parkování. Všechny kamerové systémy jsou však pasivní senzory, tzn. nevysílají žádný signál, ale pouze přijímají, co do nich přijde. Proto jsou kamerové systémy závislé na okolním osvětlení a povětrnostních podmínkách.

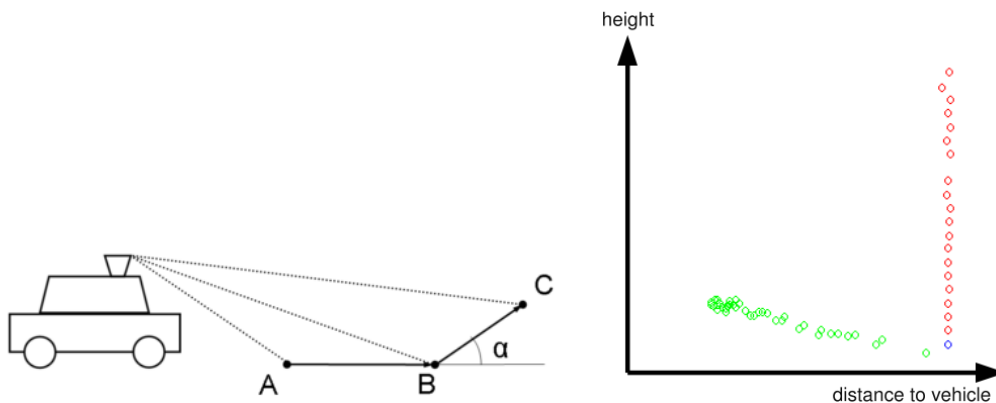
3 Algoritmy zpracování 2D dat z lidarů

K detekci překážek pomocí dat z lidarů existuje velké množství metod, avšak cíl je společný: určit, které body náleží vozovce a obecně zemi, a zbylé body přiřadit daným objektům, tak aby se minimalizovala chyba segmentace. Při segmentaci mohou nastat dva typy chyb: *nadbytečná segmentace* (over-segmentation) a *nedostatečná segmentace* (under-segmentation). Při nadbytečné segmentaci je jednomu objektu přiřazeno více shluků, tj. jeden objekt je rozdělen na více objektů. Nedostatečná segmentace naopak sloučí více objektů v jeden objekt. Existují ale také algoritmy, které nesegmentují body do objektů, ale určují jen místa v prostoru, kam vozidlo může bezpečně jet a kam nemůže. Jednotlivé třídy algoritmů a jejich příklady jsou níže popsány.

3.1 Detekce vozovky

Jako první krok je potřeba určit, které body jsou body, jež mohou patřit překážkám, a které mohou patřit vozovce (zemi), což jsou místa v prostoru, která jsou zpravidla bezpečná, pokud v nich není nějaká velká díra. Tento problém komplikuje především pohyb vozidla po nerovné vozovce, nejen rychlé nerovnosti na vozovce, ale i změny profilu vozovky. Nelze proto použít jednoduchý přístup výpočtu vertikální souřadnice z naměřené vzdálenosti bodu od lidarů a známé výšky lidarů nad vozovkou, tj. $Z_i = h - R_i \cdot \sin(\rho)$, kde h je výška umístění lidarů na vozovkou a ρ je úhel sklonu lidarů. To vede k nesprávnému přiřazení bodů do množiny vozovky P'' .

U vícevrstvých lidarů, které poskytují měření ve více rovinách, lze využít gradientu mezi jednotlivými vrstvami. Pokud více vrstev zaznamená překážku, pak jsou X a Y souřadnice bodů z těchto vrstev přibližně stejné. Avšak pokud jedná vrstva zaznamená překážku, ale druhá již vidí vozovku, projeví se to ve větším rozdílu Y -ových souřadnicích. Další přístup nabízí Steinhauser et al. [56], kdy pomocí lidarů Velodyne HDL64 získávají řez prostorem při každém pootočení a měření lidarů, obr. 15b. Určení, zda bod patří do množiny vozovky, nebo množiny bodů náležících objektům, je založeno na vzdálenosti jednotlivých bodů od odhadnuté roviny vozovky. Odhadnutá rovina vozovky je získána lineární regresí bodů v řezu, kdy se hledají jednotlivé segmenty přímek. Podobný přístup zvolil Himmelsbach et al. [20]. V jejich přístupu si nejdříve okolí vozidla rozdělili na několik stejně velkých výsečí a každou výseč ještě v radiálním směru rozdělili na několik buněk o stejné velikosti. Pak každý bod prostoru přiřadili do jeho odpovídající výseče a buňky. Každá buňka byla reprezentována tzv. prototypem, což byl bod s nejnižší Z -ovou souřadnicí. Pro každou výseč vypočítali rovnice přímek procházejícími prototypy. Body vzdálené od přímek o větší vzdálenost než zvolená hodnota prahu považovali za body nepatřící vozovce. Natale et al. [39] použili 3D flash lidar a hledali body vozovky pomocí algoritmu RANSAC pro hledání roviny. Nutno podotknout, že si vybrali jen 8 m široký pruh před vozidlem, jenž se snažili proložit rovinou, ne



(a) Detekce vozovky pomocí vícevrstvého lidarů. Zdroj [44].

(b) Profilový řez prostorem lidarem. Zelené body jsou vozovka, červené překážka. Zdroj [56].

Obrázek 15: Jeden ze způsobů detekce vozovky u vícevrstevých lidarů.

tedy celé okolí vozidla.

Způsob detekce vozovky velmi záleží na tom, kolik vrstev lidar dokáže poskytnout. Je pochopitelné, že čím více vrstev lidar skenuje, tím snazší je hledat rovinu vozovky, jelikož algoritmus má více dat a lze tak použít jednu z výše uvedených metod. Avšak pokud lidar skenuje jen jednu rovinu případně další 2-3 vrstvy, je mnohem obtížnější určit, které body náleží vozovce. Některé vrstvy mohou být nastaveny rovnoběžně s vodorovnou vozovkou, ale jakmile se vozidlo začne blížit nějakému stoupání vozovky, začne tato vrstva detekovat vozovku. Algoritmus tak musí z toho mála dat, co mu poskytují ještě další vrstvy, vyhodnotit, že skenuje vozovku a ne překážku.

Jakmile jsou jednotlivé body měření klasifikované na body vozovky P^v a tzv. objektové body P^o , lze provádět tzv. segmentaci objektových bodů. Cílem této segmentace je najít shluky bodů, které odpovídají jednotlivým detekovaným objektům. Po správné segmentaci, by měla být data rozdělena do shluku tak, aby jednomu objektu odpovídal jen jeden shluk a aby jeden shluk neodpovídal více než jednomu objektu, jinak dochází k tzv. *přeučení* (angl. over-fitting, over-segmentation), nebo k tzv. *nedoučení* (angl. under-fitting, under-segmentation). Správná segmentace je zásadní pro další krok zpracování, jímž je klasifikace, kdy se algoritmus snaží z nalezených shluků určit, o jaký typ objektu jde, jestli je to vozidlo, nebo člověk, nebo autobus, atd. Jednotlivé třídy segmentace a vybrané příklady jsou níže popsány.

3.2 Parametrické metody

Parametrické metody mají za cíl najít v datech linie, případně roviny, vytvořit rohy ze dvou linií, pokud jsou splněny určité podmínky. Nejčastěji se nejdříve hledá prototyp linie, kdy body, které patří určitému segmentu, jenž linie reprezentuje, nejsou od linie vzdáleny více než zvolený práh. Jakmile jsou tyto body

rozděleny, buď že patří do segmentu reprezentovaným linií, nebo jsou klasifikovány jako šum (outliers), hledá se optimální fit bodů patřících do segmentů nejčastěji pomocí metody nejmenších čtverců (Least Squares). Tím se naleznou parametry přímky reprezentující optimální fit.

Existuje několik iterativních metod, které dokážou najít prototypy linií (rovin). Nguyen et al. [42] poskytli popis několika takových algoritmů a otestovali každý z nich na datech pořízených lidarem SICK umístěném na robotovi pohybujícím se uvnitř budovy. Jeden z velmi oblíbených algoritmů je RANSAC (Random Sample Consensus), jenž se iterativně snaží najít nejlepší fit dané množiny bodů. Vždy vybere několik bodů náhodně, proloží jimi přímku např. metodou nejmenších čtverců. Dále zjišťuje, kolik ostatních bodů z množiny je dostatečně blízko přímky, tzv. inliers. Pokud jich je málo, opakuje celý proces znovu s novou náhodnou množinou, pokud je těchto bodů dostatek, pak se model přímky přepočítá na datech z původní náhodné množiny a nových přidanych bodů (inliers) a algoritmus se ukončí.

RANSAC lze použít i pro nalezení rovin, kdy se nehledá předpis přímky ale roviny. Tento přístup na detekci vozovky použil Natale et al. [39]. Premebida a Nunes v [45] poskytli přehled algoritmů pro detekci kružnice a elipsy a způsob odhadnutí parametrů těchto křivek. Harati a Siegwart v [18] poskytli novou metodu pro detekci zlomů v detekovaném prostředí pomocí diskretní vlnkové transformace (Discrete Wavelet Transform) úhlu mezi dvěma sousedícími body měření. Ve srovnání s ostatními algoritmy, např. těmi popsány v [42], neposkytuje lepší výsledky v detekci linií, ale jeho výhodou je rychlost zpracování a nižší komplexnost, která je $O(n)$, zatímco např. algoritmus Split-and-Merge je $O(n \times \log(n))$, RANSAC je $O(s \times n \times n_{trials})$ [42].

U parametrických metod je potřeba poznamenat, že jsou vhodné zejména pro lokalizaci a navigaci robotů v interiérech budov, kdy se robot snaží vytvořit virtuální mapu prostředí z laserových měření. Uvnitř budov se totiž nalézají pravidelné tvary překážek, jako je zeď, sloup, nábytek. Tyto všechny překážky vytvářejí zřejmé linie bodů z lidarů, které lze jednoduše pomocí algoritmů najít. Avšak pro použití venku, např. v městském provozu, jsou neúčinné, jelikož venkovní prostředí je méně pravidelné, takže se v něm hůře hledají pravidelné linie. Často proto jsou tyto metody použity až po nalezení jednotlivých shluků k extrakci příznaků objektu pro pozdější klasifikaci. V každém shluku je snaha najít nějaké linie či křivky, které by daný objekt popsaly a dalo se z nich získat informaci, např. o šířce a délce vozidla. Tyto informace jsou zmíněné příznaky.

3.3 Shlukování

Metody shlukování jsou postaveny na hledání shluků v závislosti na vzájemné prostorové poloze jednotlivých bodů. Často jsou tyto metody postaveny na hledání nejbližších sousedů. Základní myšlenka těchto metod je, že body, které leží v určité vzdálenosti od sebe by měly patřit do jednoho shluku, tj.

pokud $\|P_i, P_{i+1}\|^2 \leq d_{prah}$, pak body P_i, P_{i+1} by měly patřit do jednoho shluku, jinak by měly být separovány. P_i je bod definován souřadnicemi X, Y případně Z . S touto myšlenkou jsou spjaty dva problémy, které je potřeba řešit, a to najít vhodnou prahovou funkci vzdálenosti a prohledávání nejbližších sousedů. Klasický algoritmus shlukování K-means [32] nelze použít, jelikož ten požaduje apriorní znalost o množství shluků, které se mají v datech nacházet, avšak to v shlukování dat z lidarů snímajícím reálnou scénu před vozidlem nelze použít, protože, jak se mění scéna, mění se i počet shluků, jenž není znám apriori.

Prahová funkce

Zde nastává problém s vhodnou volbou prahu vzdálenosti. Body, které leží blíže lidarů, jsou i blíže k ostatním bodům, s rostoucí vzdáleností, roste i vzdálenost mezi body. Proto hodnota prahu nemůže mít fixní hodnotu, protože buď by algoritmus fungoval správně pro body blízké lidarů, ale na větší vzdálenosti by vytvářel více shluků než je objektů, nebo by na velké vzdálenosti fungoval správně, ale blíže k lidarů by vytvářel méně shluků než je objektů, tzn. dva separátní objekty by spojil v jeden shluk. Tudíž musí být tento práh pro každý bod jiný v závislosti na vzdálenosti onoho bodu od lidarů a na úhlovém rozlišení lidarů, tj. $d_{prah}^i = f(r_i, \Delta\phi, \theta)$. Kde r_i je vzdálenost i -tého bodu naměřená lidarem a $\Delta\phi$ je úhlové rozlišení lidarů a θ jsou další parametry. Dietmayer et al. [10] navrhli práh, který bere v potaz rozlišení lidarů a vzdálenost bodu od lidarů, jako:

$$d_{prah} = C_0 + \min\{r_i, r_{i+1}\} \cdot \sqrt{2 \cdot (1 - \cos(\Delta\phi))} \quad (3)$$

Kde C_0 je parametr pro kompenzaci nepřesnosti měření lidarů. Pro úhly $\Delta\phi$ do cca 17° (většina lidarů má $\Delta\phi < 1^\circ$) lze výraz $\sqrt{2 \cdot (1 - \cos(\Delta\phi))}$ aproximovat jako $\sin(\Delta\phi)$, což je pro implementaci efektivnější. Santos et al. [52] upravili rovnici (3) na:

$$d_{prah} = C_0 + \min\{r_i, r_{i+1}\} \cdot \frac{\tan(\beta) \cdot \sqrt{2 \cdot (1 - \cos(\Delta\phi))}}{\cos(\Delta\phi/2) + \sin(\Delta\phi/2)} \quad (4)$$

Kde parametr β slouží ke snížení závislosti prahu na vzdálenosti bodu od lidarů. Borges a Aldon představili tzv. Adaptivní detektor [5], který bere v potaz kromě vzdálenosti bodu od lidarů i nejhorší možný případ natočení překážky, kam by následující paprsek mohl dopadnout, aby ještě patřil do stejného shluku jako předešlý bod. Toto natočení je vyjádřeno parametrem λ . Adaptivní detektor má předpis:

$$d_{prah} = r_i \cdot \frac{\sin(\Delta\phi)}{\sin(\lambda - \Delta\phi)} + 3 \cdot \sigma_r \quad (5)$$

Parametr σ_r kompenzuje chybu měření lidarů. Kromě prahové funkce založené na vzdálenosti bodů, lze použít i jiná prahová kritéria. Např. Klasing et al. v [27] použily kromě prahu pro vzdálenost ještě práh

pro úhel, který svírají normálové vektory jednotlivých bodů, jež jsou vypočteny pomocí SVD (Singular Value Decomposition) bodů v okolí každého bodu. Druhý problém, který se řeší při shlukování po nalezení vhodné prahové funkce, je způsob hledání nejbližších sousedů a případné spojování shluků.

Hledání nejbližších sousedů

Tato část algoritmu může celý proces analýzy velmi zpomalit, jelikož hledání nejbližších sousedů má komplexnost $O(\log(n))$ [38]. Pokud by se při hledání shluků hledaly nejbližší sousedi pro každý bod, pak je komplexnost celého algoritmu $O(n \times \log(n))$. Přesně tuto komplexnost má základní verze DB-SCAN algoritmu (Density-based Spatial Clustering of Applications with Noise), který právě prochází bod po bodu a hledá jeho sousedy. Celý hledací proces lze urychlit např. tím, že se nebude pro každý bod počítat postupně vzdálenost k ostatním bodům, ale dopředu se již vypočte matice vzdálenosti mezi všemi body, což však na druhou stranu zvyšuje nároky na paměť. Dalšího zrychlení celého algoritmu lze docílit tím, že se nebudou hledat sousedi pro každý bod, např. Klasing et al. v [28]. Jejich algoritmus pojmenovaný jako RBNN (Radially Bounded Nearest Neighbour) funguje tak, že pro body, které již jsou přiřazeny do nějakého hluku, se neprovádí hledání jejich sousedů. To znamená, že více než jeden sousední bod je nalezen při hledání sousedů, takže rádius vyhledávání musí být dostatečně velký, aby obsahoval více sousedů. To však zvyšuje riziko toho, že dva objekty budou spojeny v jeden shluk. Qin et al. v [47] přišli s tzv. časoprostorovým přístupem, aby vyřešili segmentaci a klasifikaci v jednom kroku. Místo aby segmentovali každý sken zvlášť, tak nahromadili více po sobě jdoucích skenů, ve kterých se detekovaly vyskytující se objekty. Také pohyblivé objekty takto mohly být snáze detekovány pomocí časoprostorových příznaků. Segmentaci prováděli na základě vzdálenosti v časové a prostorové oblasti.

3.4 Detekce diskontinuity

Dalším přístupem pro segmentaci dat z lidarů je založen na detekci diskontinuity v datech. Hledají se tzv. zlomové body (break points), které jsou hraniční body objektů a nachází se tam, kde je velká vzdálenost mezi sousedními body. Borges a Aldon v [5] přišli s detektorem zlomových bodů založeném na Kálmánově filtru (KF). KF se používá hlavně na trackování a zpřesňování určování polohy, např. u GPS nebo právě u trackování objektů detekovaných lidarem, radarem, nebo kamerou. Slouží k tomu, že měření není přesné, obsahuje nějaký šum, pokud je známá rychlost vozidla a objektu, jenž je trackován, lze sestavit stavový model. KF predikuje pomocí stavového modelu polohu objektu v následujícím čase a provádí korekci své predikce na základě výsledků měření. Avšak Borges a Aldon nepoužili KF k trackování objektů, ale použili KF jako prediktor dalšího bodu měření, a pokud odchylka jeho skutečné polohy od predikované hodnoty byla větší než práh, pak považovali tyto dva body jako zlomové a tedy, že nepatří

do stejného shluku. K popisu měření vzdálenosti pomocí lidarů, který lze vnímat jako dynamický proces, použili stavový popis:

$$\mathbf{x}_n = \mathbf{A}\mathbf{x}_{n-1} + \mathbf{w}_n \quad (6)$$

$$z_n = \mathbf{C}\mathbf{x}_n + \mathbf{v}_n \quad (7)$$

Kde $\mathbf{x}_n = (r_n, \frac{dr_n}{d\phi})^T$ je stav systému, z_n je měřená veličina, \mathbf{w}_n a \mathbf{v}_n jsou nekorelované šумы s Gaussovým rozdělením s nulovým průměrem a kovariancí \mathbf{Q}_n a R_n . Matice \mathbf{A} a \mathbf{C} jsou:

$$\mathbf{A} = \begin{bmatrix} 1 & \Delta\phi \\ 0 & 1 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad (8)$$

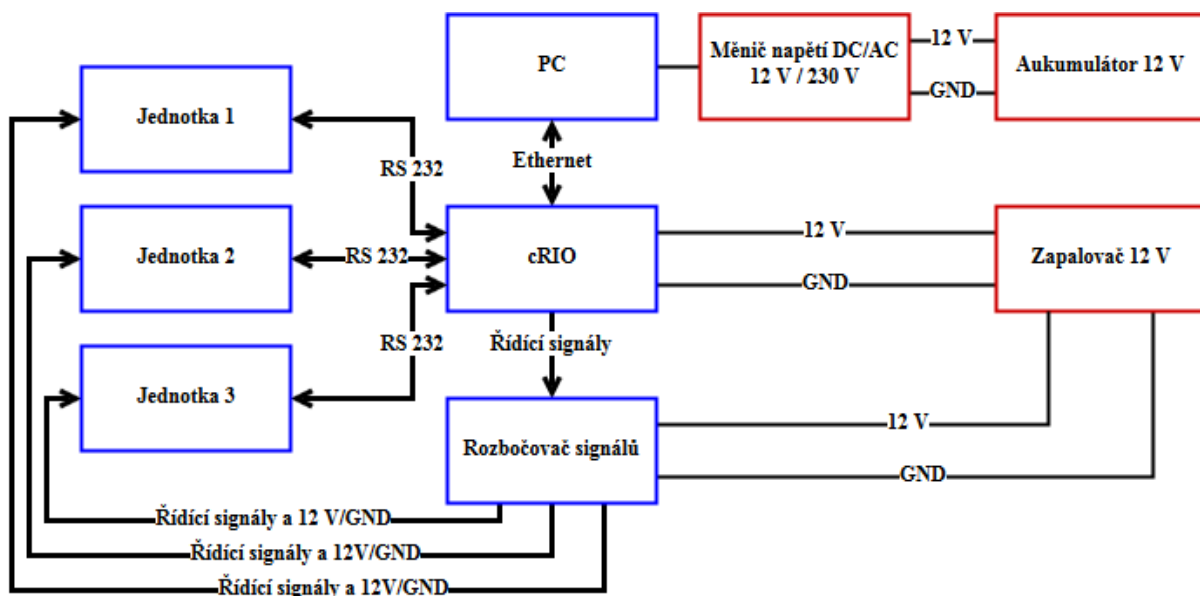
Zhang et al. v [75] použili rozšířený KF (Extended KF – EKF), což je nelineární verze KF pro vypořádání se s nelinearitou systému. Jiný přístup pro detekci zlomových bodů použili Kwak et al. v [30], kdy tuto detekci řešili jako klasifikační problém, a to jestli dva sousední body jsou zlomové body, nebo nejsou. Klasifikaci prováděli pomocí SVM (Support Vector Machine).

3.5 Occupancy Grid

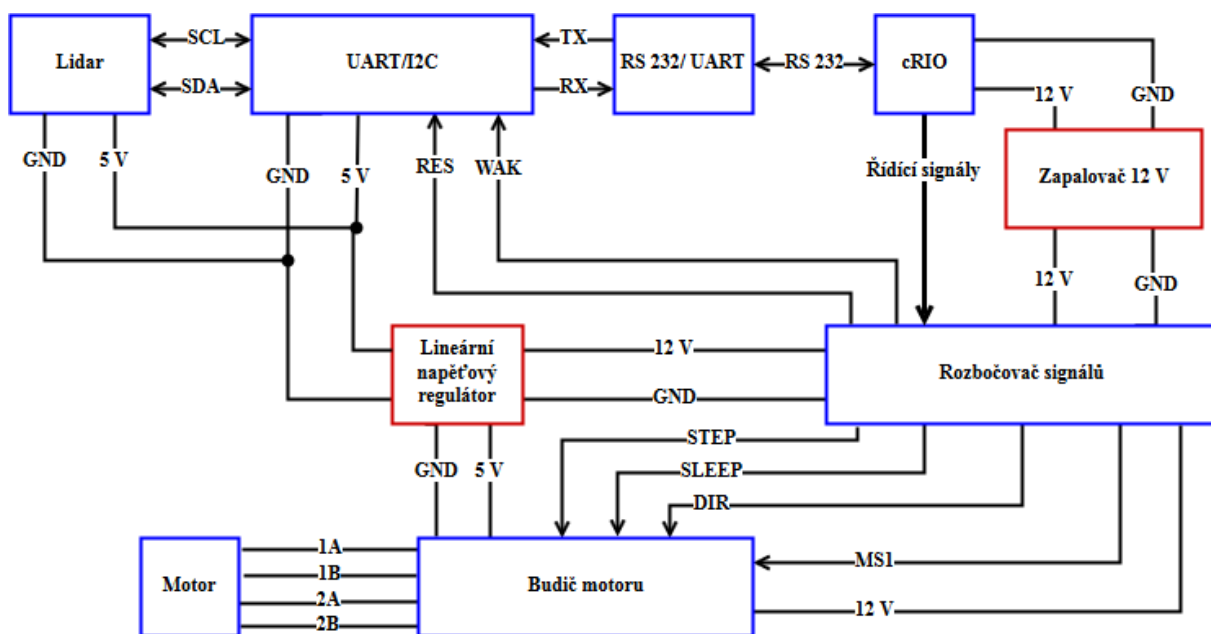
Occupancy grid (OG) lze volně do češtiny přeložit jako rastr obsazenosti. Tato metoda je velmi často používána pro segmentaci dat měření z lidarů. Spočívá v tom, že jednotlivá měření se promítnou do roviny XY – rastr, která je rozdělena na několik stejně velkých buněk tvořících rastr. Jednotlivé obsazené buňky se pak shlukují pomocí propojených komponent (connected components – CC). Výhodou je, že jedna buňka může obsahovat více měření, tudíž se zmenší prohledávací prostor. Problémem může však být pevná velikost všech buněk. To může způsobit opět problémy s over a under-segmentation. Himmelsbach et al. v [20] použili OG tak, že i jediný bod ležící v buňce znamenal, že buňka je obsazená, což je podle nich způsob, jak se vypořádat s nedostatečnou segmentací objektů ve velké vzdálenosti. Rubio et al. v [51] použili také binární OG, avšak zaznamenali problémy s fixní velikostí buněk a navrhují dynamickou velikost buněk podle vzdálenosti buňky od lidarů. Segmentaci pak následně prováděli na binární OG pomocí CC. Další podobné použití OG pro segmentaci uvádí [35]. Weiss et al. v [69] navrhli algoritmus založený na OG, jež nazvali online map. Hodnota každé buňky byla pravděpodobnost, že je daná buňka obsazena. Po každém skenu je tato mapa aktualizována, tzn. předchozí stav mapy byl zkombinován s aktuálním měřením pomocí Bayesova teorému. Tímto dokázali detekovat jak statické, tak dynamické objekty, které trackovali. Zároveň tato mapa sloužila jako indikátor, kde je bezpečné jet a kde ne. Podobné řešení navrhl Lindner et al. v [31].

4 Hardware architektura

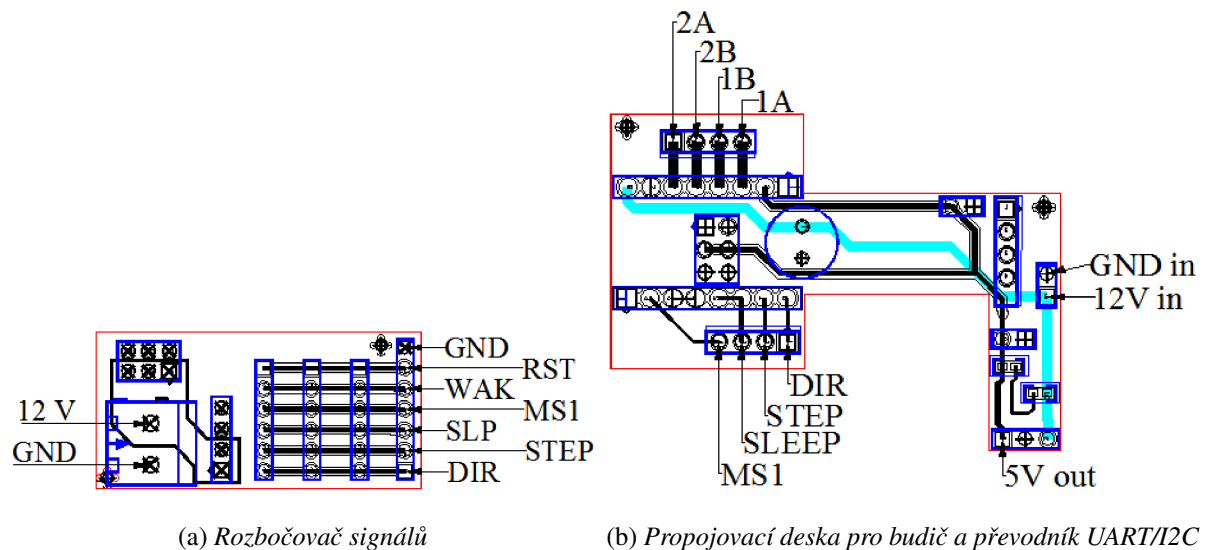
Systém se skládá ze tří jednotek a řídicího členu. Každá jednotka se skládá z: Lidar-Lite V3, obr. 19, krokového motoru NEMA 14 a dalších příslušenství zahrnujících převodníky RS232/UART a UART/I2C, budiče motoru Pololu A4988. Řídicí člen zde představuje cRIO. Obr. 16 popisuje hardware architekturu systému a druhy komunikace mezi jednotlivými komponentami. Detailnější popis signálu a návrh jedné jednotky je znázorněn na obr. 17.



Obrázek 16: Hardware architektura systému.



Obrázek 17: Detailní popis jedné jednotky s připojením k cRIO a zdroji a s řídicími signály a rozvody napájení.



Obrázek 18: PCB návrh rozbočovače signálů a propojovací desky

Z cRIO vedou řídicí signály pro motor a převodník UART/I2C. Aby byly výstupní signály z cRIO pro všechny jednotky stejné, musí se dráty vedoucí z pinů cRIO rozbočit a vést paralelně ke každé jednotce zvlášť. Pro tento účel byl navržen a vyroben rozbočovač signálu znázorněn na obr. 18a, který výstupní signály cRIO rozdělí na tři svazky. Zároveň rozvádí na další tři svazky napájecí kabely 12 V a zem (GND). Další navržená deska (obr. 18b) složí k převodu 12 V na 5 V, k propojení signálů a napájení budiče a převodníku UART/I2C, tím se snížilo množství kabelových vodičů a celý systém tak byl více kompaktnější a robustnější.

4.1 Lidar

V tomto systému je použit Lidar-Lite V3 vyroben firmou Garmin. Tento lidar je hlavně oblíben v komunitě zabývající se robotikou a drony pro jeho poměr cena/výkon. Cena jednoho kusu se pohybuje okolo 150 USD. Maximální měřitelná vzdálenost je 40 m. Nepřesnost roste se vzdálenosti a většinou odpovídá 1 % vzdálenosti. Výrobce uvádí, že je schopen měřit s frekvencí více než 1 kHz pro silné signály, ale pro optimální a robustní měření se frekvence měření pohybuje okolo 250 Hz.

Technická dokumentace, kterou výrobce poskytuje, detailně popisuje princip měření a zpracování signálu pro určení měřené vzdálenosti. Ve stručnosti jej lze popsat takto: Laser v lidarů vystřeluje několik světelných signálů v krátkém sledu za sebou. Fotodioda zaznamenává jednotlivé reflektované signály a následně se každý odražený signál koreluje s původním signálem, pokud nastane shoda, signál se uloží do korelační paměti. Takto se to děje s každým přijatým signálem, dokud nebyl vystřelen určitý počet signálů, např. 120 (toto číslo lze nastavit). Tyto jednotlivé přijaté signály se sčítají, aby výsledná jejich energie byla vyšší než šum. V záznamu se pak objeví vrchol vystupující nad hodnoty šumu a jeho poloha

určuje, vzdálenost objektu. Avšak tímto způsobem by stále nebylo možné změřit vzdálenost s přesností v jednotkách centimetrů, jelikož by bylo potřeba vzorkovat přijatý signál s frekvencí 15 GHz, aby bylo možné dosáhnout přesnosti 1 cm. Použít takto výkonný procesor by bylo nepraktické a drahé, proto vyvinuli algoritmus, který interpoluje signál mezi jednotlivými vzorkovanými body (vzorkovací frekvence je 500 MHz) a dopočítává tak měřenou vzdálenost. Tímto je možno měřit vzdálenost s přesností ± 2.5 cm, ale běžně spíše 1% vzdálenosti.

Lidar lze nastavovat pomocí interních registrů. Tím lze nastavit a ovlivnit tak frekvenci měření, přesnost, maximální měřitelnou vzdálenost. Vyšší rychlost měření snižuje přesnost ve větších vzdálenostech, snižuje maximální měřitelnou vzdálenost a zhoršuje přesnost, pokud přijatý signál je slabý vlivem špatného materiálu měřeného objektu.

Přesnost lidarů je značně ovlivněna tím, jaký má detekovaný objekt povrch, jak je daný objekt (velikost plochy, kterou objekt před lidarem zaujímá) natočen vůči lidarů. Lesklé kovové nebo plastové povrchy způsobují disperzi paprsku a síla reflektovaného signálu je malá, což může způsobit buď sníženou přesnost měření, nebo absolutní neschopnost danou vzdálenost naměřit. Pokud ovšem tyto lesklé objekty jsou vůči lidarů postaveny čelem (normála plochy objekty míří k lidarů), pak nedochází k takové značné disperzi a problém s měřitelností nenastává. Při měření různých ploch vozidla bylo zjištěno, že největší problém Lidaru Lite dělá sklo v oknech vozidla a zadní světla. V těchto případech lidar není schopen změřit vzdálenost. Proto je vhodné nastavit lidary tak, aby nemířily na zadní okno vozidla před nimi.

Lidar s řídicím členem může komunikovat buď přes I2C protokol, nebo pomocí PWM. V navrhovaném systému byla zvolena komunikace přes I2C, jelikož PWM je příliš pomalé pro danou aplikaci, jelikož 1 meter odpovídá délce PWM signálu 1 ms. To by znamenalo, že přenést informaci o tom, že měřená vzdálenost je 30 metrů, by trvalo 30 ms. Zatímco lidar podporuje I2C s přenosovými rychlostmi 100 a 400 kHz, což umožňuje rychlý přenos dat do cRIO.



Obrázek 19: *LiDAR-Lite V3 od Garmin. Zdroj [55].*

4.2 Krokový motor

Návrh systému počítá s použitím krokového motoru NEMA 14, který je řízen budičem A4988 od Pololu. Motor má dostatečný výkon a kroutící moment pro otáčení lidaru požadovanou rychlostí. Motor je schopen krokovat až o frekvenci 20 kHz. Základní délka kroku je 1.8° , ale podporuje i tzv. microstepping, což umožňuje snížit velikost kroku o 1/2, 1/4, 1/8 nebo 1/16. Systém je navržen pro krok 1.8° a 0.9° , s tím že výchozí velikost kroku je nastavena na 0.9° . Minimální napájecí napětí je 10 V a proud na fázi je 500 mA. Budič umožňuje velmi lehké řízení motoru, jelikož k řízení je potřeba jen dva signály: *step* a *dir*, kdy první dává pokyn ke kroku a druhý určuje směr otáčení. Na budiči lze nastavit proudové omezení, kterým lze regulovat kroutící moment motoru, ale hlavně se tímto proudovým omezením zabráňuje tomu, aby se motor zničil. Toto nastavením se provádí pomocí trimmeru na budiči měřením napětí mezi zemí a výstupním pinem referenčního napětí V_{ref} . Přepočítání mezi V_{ref} a proudem na fázi uvádí výrobce budiče Pololu A4988 na webových stránkách i v manuálu budiče.

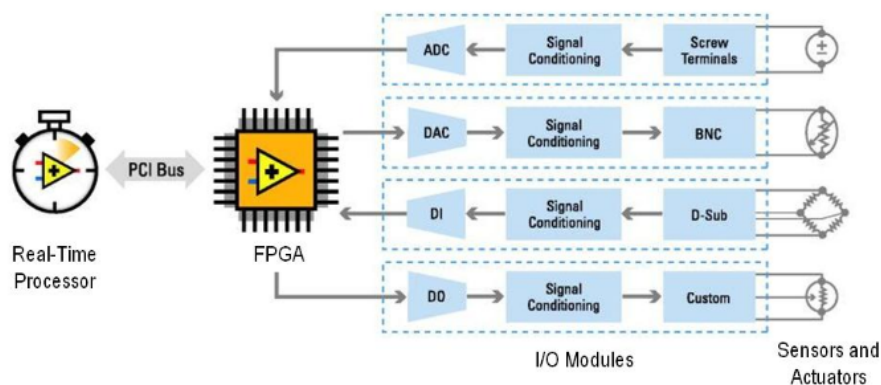
4.3 cRIO

cRIO od National Instruments není mikrokontrolér v pravém slova smyslu, ale v systému stojí v roli mikrokontroléru. Jedná se o průmyslový počítač určený k sběru dat a řízení aktuátorů. Skládá se z real-time procesoru, který je přes vysokorychlostní sběrnici spojen s FPGA. cRIO má modulární architekturu, obr. 20, která umožňuje zasouvání jednotlivých měřicích karet a jiných modulů. Tyto moduly mohou být různé digitální či analogové IO moduly, CAN bus modul, RS 232 modul a mnohé jiné. Pro popisovaný systém byl použit cRIO NI 9025 s DIO modulem NI 9401 a RS 232 modulem NI 9870.

FPGA zde slouží jako procesor s tvrdým real-time determinismem s přesným hodinovým oscilátorem, což umožňuje přesné a rychlé řízení komunikace a např. motoru. RT procesor také nemusí řešit žádné rozsáhlé alokování zdrojů, jak je to běžné u osobních počítačů, a tak umožňuje také relativně tvrdý real-time determinismus. cRIO se programuje pomocí grafického programovacího jazyku LabVIEW.

4.4 Ostatní příslušenství

Jelikož Lidar-Lite podporuje I2C protokol, ale cRIO modul, který byl v laboratoři dostupný, byl pouze pro komunikaci RS 232, bylo potřeba zvolit konverzní prvek, pro převod I2C protokolu do RS 232. Toho mohlo být dosaženo dvěma způsoby. Buď navrhnout vlastní plošný spoj pro přímou konverzi I2C/RS232, nebo koupit již hotové řešení. Důležitým aspektem při volbě řešení byla cena a čas. Jelikož náplní práce nebyla návrh elektrických obvodů, byla zvolena druhá varianta: koupit konkrétní převodník. Jelikož rozhodující byla cena, nakonec se pořídily dva převodníky, jeden pro převod I2C na UART



Obrázek 20: cRIO architektura. Zdroj [40].

a druhý pro převod UART na RS 232. Jelikož převod UART a RS 232 je pouze převod napět'ových úrovní, největší komunikační zpoždění je způsobeno převodníkem I2C/UART.

4.5 Komunikace

Jak předešlá podkapitola napovídá, lidary byly připojeny k cRIO přes převodníky. cRIO se dotazovalo na data z lidarů konstantní frekvencí. Tato frekvence se odvíjí od požadované skenovací frekvence, maximální frekvence dotazování je 130 Hz, jež je limitována rychlostí měření lidarů a zpožděním způsobeným převodníkem I2C/UART.

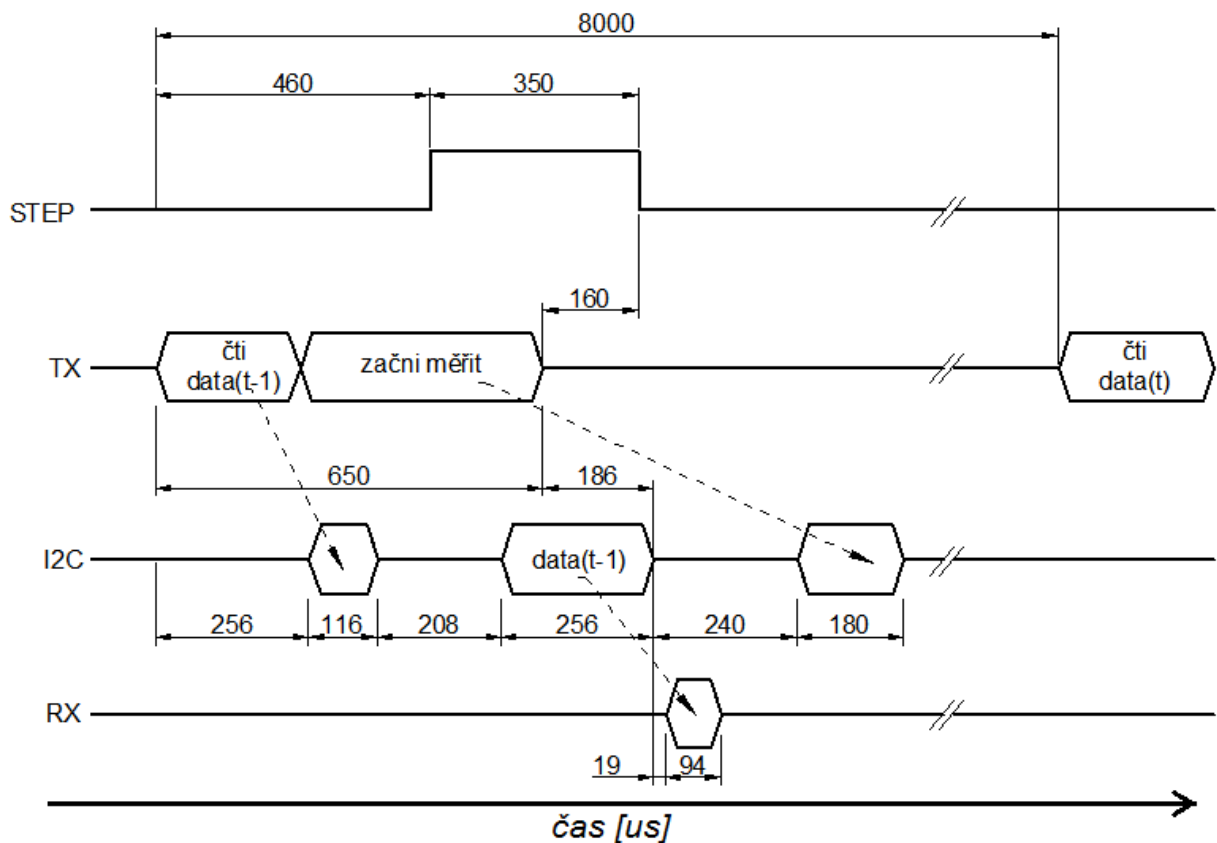
Od okamžiku změření dat lidarů do okamžiku příjmu dat do cRIO uběhne cca 950 μ s (obr. 21). To je způsobeno nejen přenosovými rychlostmi I2C a RS 232, ale i překladem z I2C na UART. Zdržení způsobeno překladem nelze nijak ovlivnit. Jediný způsob, jak ovlivnit velikost zdržení, je měnit přenosové rychlosti I2C a RS 232. Z toho důvodu byla rychlost I2C nastavena na nejvyšší možnou, která je podporována lidarem a umožněna převodníkem I2C/UART, tj. 461 kHz. Přenosová rychlost RS 232 byla nastavena na 230 400 bits/s, což je nejvyšší rychlost podporována převodníkem.

Z tohoto pohledu se užití převodníku I2C/UART jeví nepraktické, avšak existují dva důvody, proč musel být použit. Prvním je to, že I2C modul do cRIO nebyl k dispozici, avšak RS 232 ano. A druhý důvod je ten, že I2C protokol lze použít jen na krátké vzdálenosti, čím delší vedení, tím větší je kapacita, která zaobluje signálové hrany, což v určitý moment může vyústit v nemožnosti komunikace. Jelikož lidary jsou umístěny venku na kapotě a cRIO uvnitř na zadním sedadle, je vedení delší než 3 m, což je pro I2C nevhodná vzdálenost.

Jelikož systém obsahuje tři lidary, každý z nich měl tudíž svůj dedikovaný kanál RS 232 (tedy i svůj port na modulu NI 9870) a vlastní I2C sběrnici. Tato dedikovanost značně zjednodušila komunikaci a její programování. RS 232 modul připojený k cRIO má celkem 4 porty, takže systém lze rozšířit ještě o další

lidar, kdy každý by stále měl svůj vlastní kanál. Pokud by systém měl být rozšířen o více než jeden lidar, musel by se poupravit software pro komunikaci, aby mohlo více lidarů sdílet jednu I2C sběrnici. Avšak to by mělo nepříznivý vliv na dotazovací frekvenci, která by se musela snížit, což by vedlo ke snížení skenovací frekvence, zhoršení systémových parametrů, jelikož lidary jsou trigrovány pro každé individuální měření vzdáleností příkazem přes I2C. Trigrování externím signálem nelze provést, pokud lidar je nastaven pro komunikaci přes I2C.

Na obr. 21 je znázorněn časový diagram pro komunikaci a krokový signál motoru. Na něm lze vidět, že samotný překlad z RS 232 na I2C je rychlý (jen pár μs), ale odvysílání stejných dat přes I2C a pak RS 232 a vice versa celou komunikaci zdržuje. U převodníku I2C/UART nelze nastavit mezi bajtové pauzy pro I2C, které jsou dlouhé ve srovnání s mezi bajtovými pauzami I2C lidarů. Princip vyčítání dat z lidarů a započítání nového měření je takový, že nejdříve se přečtou data naměřená z předešlého kroku, pak se vyšle pokyn lidarů, aby začal měřit a paralelně se vyšle signál do budiče motoru, aby se motor pootočil o jeden krok. Časování těchto procesů je uděláno tak, aby lidar přijal pokyn pro započítání měření až poté, co se motor pootočí o jeden krok do nové polohy, kde by měl lidar měřit nový bod v prostoru.



Obrázek 21: Časový diagram komunikace RS232 (RX a TX), I2C a krokovaní motoru pro skenovací periodu 80 ms. Hodnoty jsou uvedeny v μs .

4.6 Technické parametry

Výkonnost systému byla určena rychlostí měření lidarů. Jelikož rychlost měření ovlivňuje maximální měřitelnou vzdálenost, musel se najít určitý kompromis v nastavení lidarů tak, aby byl lidar schopen měřit rychle a zároveň na co největší možnou vzdálenost. Z testů vyplynulo, že lidar potřebuje alespoň 7 ms k změření vzdálenosti 30 metrů a více, velmi totiž záleží na materiálu cíle, osvětlení, atd. Z rychlosti měření se odvíjely další parametry systému, jako je rychlost I2C, baud rate RS 232 kanálu, počet bodů za sken a úhel skenu.

Velikost kroku, neboli horizontální rozlišení, bylo zvoleno 0.9° proto, aby vzdálenost dvou sousedních bodů na větší vzdálenosti jako je 30 m nebyla příliš velká. V případě 0.9° je vzdálenost mezi dvěma sousedními body měření 47 cm, pokud by krok byl zvolen větší, mohlo by se stát, že menší objekty na větší vzdálenosti nebude možno detekovat. Proto krok byl zvolen co nejmenší, ale zároveň dostatečně velký, aby skenovací úhel nebyl příliš malý.

Jako nejnižší skenovací frekvence byla zvolena 10 Hz, aby systém byl vůbec vhodný pro reálné použití. Při rychlosti měření 7 ms lze za 100 ms naměřit 14 bodů jedním lidarem, avšak nakonec bylo zvoleno 10 bodů, 11. bod je společný pro skeny jdoucí po sobě, jelikož tento bod je krajním bodem. Tímto lze dosáhnout vyšší skenovací frekvence než 10 Hz.

Velikost skenovací frekvence má vliv především na dva aspekty: (i) schopnost systému zaznamenat změnu polohy jednotlivých překážek před vozidlem, neboli citlivost a (ii) velikost reakční doby nadřazeného systému řízení, např. adaptivního tempomatu.

Citlivost skenovacího systému

Čím větší je skenovací frekvence, tím větší je citlivost skenovacího systému, což znamená lepší schopnost detekovat menší změny polohy překážek před vozidlem. Změny polohy za jednu skenovací periodu ovlivňuje relativní rychlost vozidla a překážky. Pokud proti sobě pojedou vozidla rychlostí 20 m/s, pak v případě skenovací frekvence 10 Hz to znamená vzájemnou změnu polohy o 4 metry. Pochopitelně se vzrůstající relativní rychlosti tato změna ještě narůstá, proto skenovací systémy na detekci překážek, by měly mít skenovací frekvenci o dost vyšší, jako např. SCALA lidar s frekvencí skenování 25 Hz.

Reakční doba řízení

Jelikož laserový skener poskytuje pouze vzdálenost, musí se rychlost překážek vypočítat z alespoň dvou po sobě jdoucích skenů této překážky, ze změny naměřené vzdálenosti a známé doby skenování je možné

Tabulka 1: Technické parametry systému

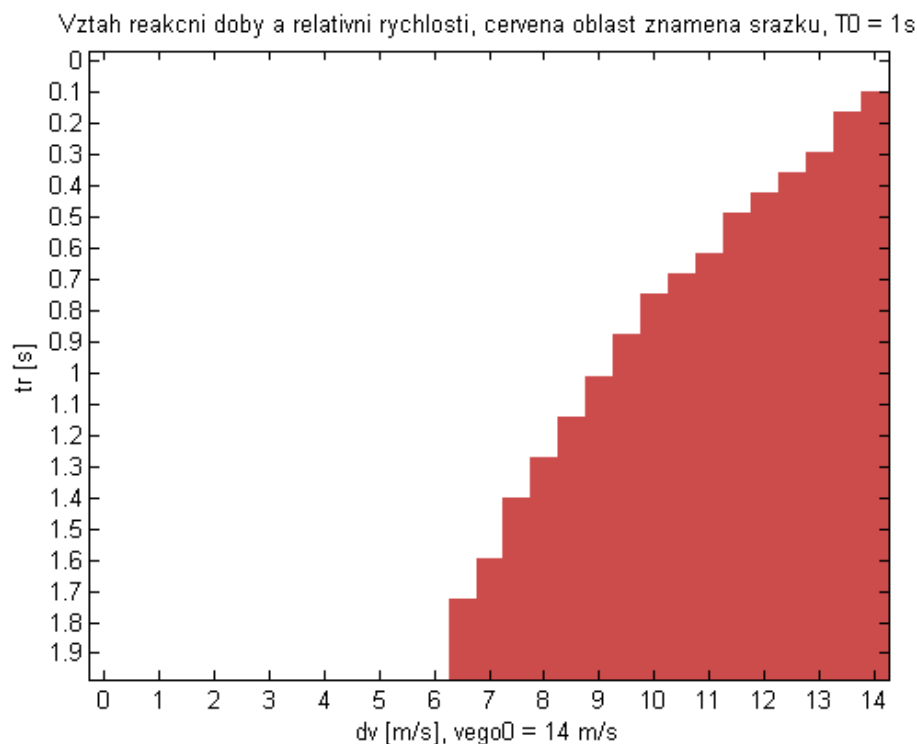
Parametr	Hodnota
Skenovací frekvence	10 – 12,5 Hz
Měřitelná vzdálenost	1 – 40 m
Nepřesnost lidarů	1 % vzdálenosti
Horizontální zorné pole	9°
Horizontální rozlišení	0,9°
Počet bodů/sken/jednotka	11 bodů
RS 232 přenosová rychlost	230 400 bits/s
I2C frekvence	461 kHz
Výkon jedné jednotky	14.4 W

rychlost překážky vypočítat, z dalších skenů je pak možno dopočítat zrychlení a ryv, což jsou požadované veličiny nadřazených řídicích systému vozidla. Čím delší tedy je skenovací perioda, tím déle trvá, než uběhne několik po sobě jdoucích skenů a tím později má řídicí systém dostatečnou informaci k akčnímu zásahu, čímž roste jeho reakční doba. Reakční doba není však dána jen dobou, která je nutná pro pořízení 2-3 skenů, ale je dána i rozhodovacím algoritmem pro vykonání reakce. Představme si situaci, že před vozidlo s lidarem jedoucím v pravém pruhu se rychle zařadí z levého pruhu jiné vozidlo. Rozhodovací algoritmus pro akční zásah může být nastaven tak, že rozhodnutí o reakci nastane, až když přejíždějící vozidlo bude do pravého jízdního pruhu zasahovat alespoň polovinou své šířky [74], což však může trvat i 2 s. O tom, jestli nastane srážka nebo ne, rozhoduje časová mezera mezi vozidly, rychlost laterálního pohybu a relativní rychlost.

Na obr. 22 lze vidět vztah reakční doby systému a relativní rychlosti ego vozidla (vozidlo vybavené řídicím systémem) a vedoucího vozidla před ego vozidlem. Obrázek je výsledkem simulace situace, kdy před ego vozidlo se zařadí vedoucí vozidlo v takové vzdálenosti, že časová mezera mezi nimi je 1 s. Ego vozidlo má počáteční rychlost 14 m/s a vedoucí vozidlo má různé počáteční rychlosti, z čehož plynou různé počáteční relativní rychlosti. Pokud relativní rychlost není nulová, pak je ego vozidlo nuceno začít brzdit, aby předešlo srážce. V simulaci bylo uvažováno, že časová mezera mezi vozidly 1 s je pro řídicí systém malá mezera mezi vozidly, na což reaguje nouzovým brzděním s decelerací $a_{max} = -8 \text{ m/s}^2$ pokud je relativní rychlost nenulová. Výsledkem simulace je vztah mezi reakční dobou a relativní rychlosti vozidel určující, jestli nastane srážka nebo ne.

4.7 Konstrukce

Jelikož Pulsedlight lidar, který byl použit pro navržený systém, je 1D lidar, musel být navržen způsob, jak z něj udělat laserový skener. Prvotní myšlenka byla, připevnit zrcátko na krokový motor a lidar upevnit napevno, tak aby mířil na zrcátko. Nakonec, ale od této varianty bylo upuštěno a lidar byl připevněn na

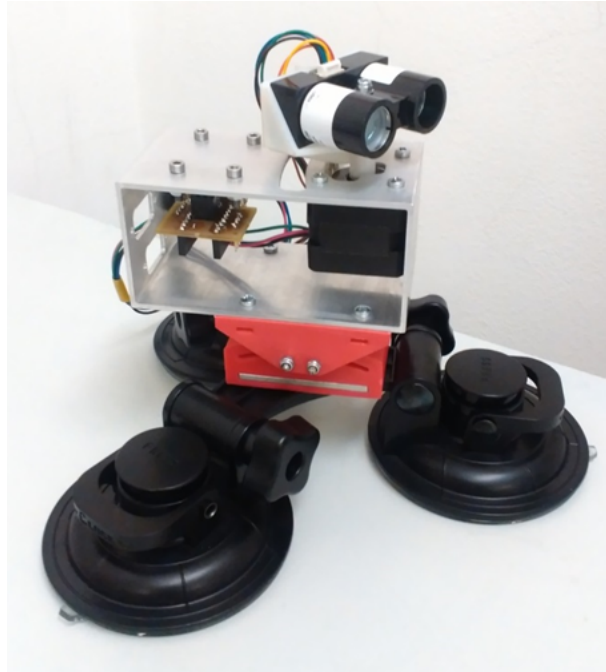


Obrázek 22: Vztah mezi relativní rychlosti d_v ego vozidla a vedoucího vozidla a reakční dobou t_r .

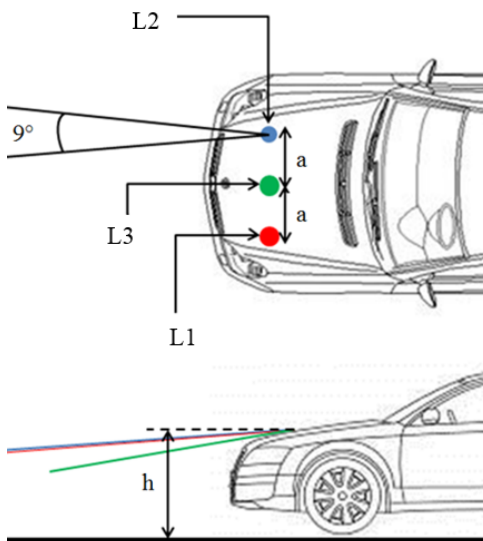
krokový motor. Proto byl navržen a na 3D tiskárně vyroben držák, do kterého se lidar zasadí. Tento celek se pak nasune na hřídel motoru. Dále se navrhl rám systému, ke kterému se připevnil motor a všechny elektronické komponenty. Pro upravení vertikálního sklonu lidarů byla navržena a na 3D tiskárně vyrobena podložka s pohyblivou částí, která spojuje přísavky s rámem. Celá jednotka je zobrazena na obr. 23.

4.8 Umístění lidarů na vozidle

V počátcích vývoje tohoto systému se počítalo s tím, že se použijí tři lidary a umístí se na střechu vozidla. Skenovací roviny by byly sklopeny tak, aby mířila na vozidla před vozidlem. Později ale bylo zjištěno, že když lidar zamíří na zadní okno vozidla, tak nenaměří správnou vzdálenost, lidar vrátí 1 cm měření, což je jeho znak pro chybné měření. Což by znamenalo, že pokud bychom lidary umístili na střechu vozidla, tak by vozidla v určité vzdálenosti od našeho vozidla nebyla detekovatelná, jelikož by se nacházela v takové nevhodné vzdálenosti, že by lidar mířil na zadní okno. Proto bylo rozhodnuto, že se lidar umístí na přední část vozidla na kapotu, tak aby skenovací roviny mířily pod okna vozidel před naším vozidlem. Konstrukce lidarů je uchycena na kapotu pomocí přísavek. Poloha lidarů na kapotě je znázorněna na obr. 24 a 25.



Obrázek 23: Jedna jednotka laserového skeneru s přísavkami pro uchycení na kapotu



Obrázek 24: Poloha lidarů na vozidle



Obrázek 25: Lidary na testovacím vozidle

5 Software architektura

Tato část popisuje programovací jazyk LabVIEW, v němž byla aplikace naprogramována, dále se popisuje jednotlivé součásti aplikace a jejich provázanost.

5.1 LabVIEW

Aplikace byla naprogramována pomocí LabVIEW, což je grafický programovací jazyk vyvíjen společností National Instruments (NI). Důvod, proč bylo zvoleno LabVIEW je ten, že cRIO, také od NI, podporuje pouze LabVIEW.

Základním stavebním kamenem jakékoli aplikace v LabVIEW je tzv. *vi*, což je zkratka pro *Virtual Instrument*. Jakákoli část kódu může tvořit jeden funkční blok a utvořit tak funkci. Tak jako v jiných programovacích jazycích, např. C apod., i v LabVIEW funkce má nějaké vstupní argumenty, tělo funkce a výstupní argumenty. Aplikace v LabVIEW se programuje tak, že se jednotlivé tyto funkční bloky spojují v logickém sledu za sebou. Programování v LabVIEW je založeno na tzv. *Data Flow* paradigmatu, což lze přirovnat k Petriho sítím, kdy přechod nelze odpálit, pokud všechny vstupy nejsou obsazeny. Totéž platí v LabVIEW, tělo funkce se vykoná až poté, co všechny vstupní argumenty jsou dostupné.

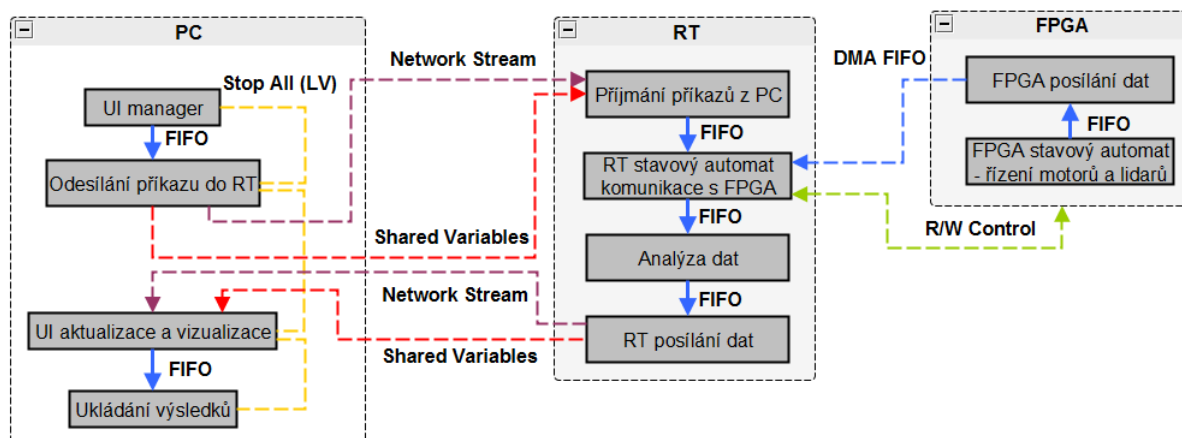
LabVIEW umožňuje velmi jednoduše vytvořit separátní procesy pro jednotlivé části aplikace a automaticky zajistí tzv. *multithreading*, tedy že jednotlivé procesy aplikace jsou exekuvány v oddělených vláknech. To umožňuje např. to, aby příkazy uživatele prostřednictvím GUI neovlivňovaly např. kolekci dat ze senzorů.

5.2 Popis aplikace

Aplikace je vytvořena ze tří významných vrstev, každá běžící na jiné části hardware a každá je reprezentována samostatným programem (*vi*). Obr. 26 znázorňuje tyto tři vrstvy, jež jsou: PC, RT, FPGA. FPGA vrstva zodpovídá za řízení motoru a zajišťuje komunikaci s lidarem, RT zpracovává data z lidaru a posílá je do PC vrstvy k zobrazení uživateli. Na systém lze nahlížet jako na malou síť, jelikož počítač a cRIO jsou k sobě připojeni přes Ethernet. V každé vrstvě běží určité procesy, jež mezi sebou komunikují pomocí LabVIEW metody pro frontu (FIFO – first in - first out), založené na principu producent-konzument, tj. jeden proces produkuje data a posílá je přes FIFO metodu konzumentovi, jenž data přijímá. Zároveň vrstvy mezi sebou komunikují přes network stream, což je komunikační metoda pro síťovou komunikaci mezi cRIO a PC, nebo přes síťově sdílené proměnné (network-published shared variables), jež využívají NI publish-subscribe protokol.

FPGA vrstva

Tato vrstva je nejnižší vrstva aplikace běžící na cRIO. FPGA vrstva programu řídí chod motoru, zajišťuje komunikaci RS 232 modulu v cRIO s lidary, přijímá příkazy z vyšší vrstvy (RT) a posílá přijatá data z lidaru vyšší vrstvě. FPGA posílá příkazy jednotlivým modulům v cRIO, konkrétně modulům RS 232,



Obrázek 26: Software architektura aplikace sestávající ze 3 vrstev, metody komunikace uvnitř vrstev mezi procesy a mezi vrstvami.

DIO (Digital input/output), které přeměňují příkazy na elektrické signály ovládající komunikaci s lidary a chod motorů. FPGA vrstva je logicko-fyzická vrstva, která je rozhraním mezi software a externím hardware, ale zároveň má určitou svou vnitřní logiku, avšak jinak slouží jako vykonavatel požadavků vyšší vrstvy.

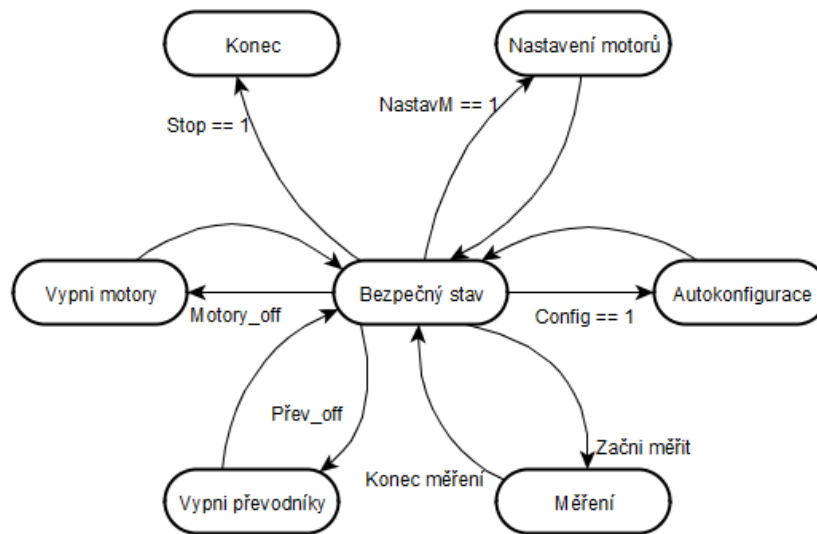
Jelikož FPGA je programovatelný hardware, tak umožňuje hardwarový paralelismus. To velmi zjednodušuje řízení motoru a komunikaci s lidary, kdy tyto dva procesy mohou běžet paralelně, čímž lze docílit tvrdého determinismu, tedy že periodické děje mají pevnou periodu s minimální odchylkou (jitter).

Základem této vrstvy je stavový automat (SA), obr. 27, jehož stavy určuje vyšší vrstva. Bezpečný stav je výchozím stavem a i stavem, do kterého se SA vrací ze všech ostatních stavů. V tomto stavu se motory netočí, ale odebírají proud, proto je lze z tohoto stavu vypnout, taktéž i převodníky.

Při autokonfiguraci se spustí sekvence nastavení systému. Nastavuje se přenosová rychlost převodníků I2C/UART, přenosová rychlost RS 232 cRIO modulu, rychlost hodinového signálu I2C (SCL) v převodníku, nastavují se registry lidarů. Stav nastavení motorů nejdříve přivede budiče motorů ze spánkového režimu do aktivního, čímž se motory začnou napájet. Následně motory vykonají určitou krokovou sekvenci, aby se dostaly do výchozí polohy. Jak je zmíněno v části o kalibraci (sekce 6), motory při zapnutí napájení lehce šklubnou, udělají pár kroků doprava, nebo doleva, a navíc každý motor se pohne ještě jinak, jeden víc, druhý méně, třetí např. jiným směrem. Nastavení motorů, kdy se pohnou do výchozí polohy, se nestará o to, že lidary míří správným směrem, o to se stará kalibrace, ale o to, aby lidary kmitaly přibližně okolo osy konstrukce a nenarážely případně do zářezů v krajních polohách.

Ve stavu měření probíhají paralelní procesy pro řízení chodu motorů a komunikaci s lidary. Lidary měří v době, kdy se motory nehýbou, tedy mezi jednotlivými kroky. Proces měření proto probíhá tak, že na

konci měřicí periody program vyčte data z lidarů, což trvá určitou dobu, během níž už paralelní proces pro řízení motorů začne vysílat příkaz pro pohyb motorů o jeden krok. Data jsou přes FIFO poslána do jiného paralelního procesu, který zajišťuje komunikaci s RT vrstvou přes DMA (Direct Memory Access) FIFO, takže data jsou do RT vrstvy posílána po jednotlivých měřeních, kde se hromadí ve DMA FIFO a až všech 33 bodů je ze všech lidarů získáno, pak se teprve vyčtou. Po ukončení pohybu otočení o jeden krok následuje příkaz pro lidary, aby započaly nové měření, a následuje perioda měření. Detailní časový diagram je znázorněn na obr. 21.



Obrázek 27: Stavový automat FPGA vrstvy s jednotlivými stavy a přechodovými podmínkami. Kde chybí podmínka, tam je bráno, že z daného stavu se automat vrátí do bezpečného stavu po ukončení funkce stavu.

RT vrstva

RT vrstva slouží jako prostředník mezi PC vrstvou a FPGA vrstvou, zároveň řídí vyčítání z DMA FIFO, zpracování dat, řízení FPGA SA a posílá data do PC vrstvy.

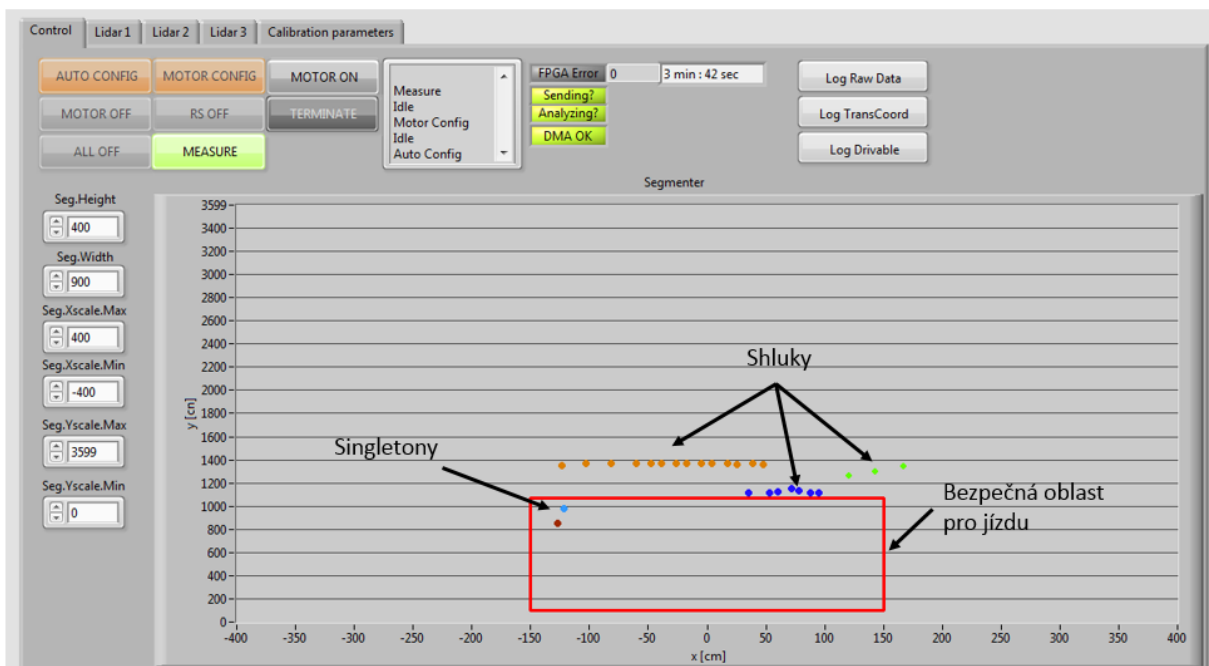
RT vrstva má také SA, který maskuje FPGA SA, tzn. každý stav FPGA SA je i stavem v RT SA, čímž se zjednodušila komunikace mezi PC a FPGA. Stav RT SA určuje příkaz z PC vrstvy, který je poslán přes network stream. V tomto stavu RT vykonává určité procesy, např. vyčítá data z DMA FIFO a posílá je přes FIFO do jiného procesu na zpracování. Data jsou z RT vrstvy posílána do PC vrstvy přes jiný network stream. Výhoda network stream je ta, že lze přes ně posílat datovou strukturu, jež obsahuje různé datové typy. RT vrstva dále komunikuje s PC vrstvou přes již zmíněné sdílené proměnné, jež slouží (i) k přenosu nastavovacích parametru pro FPGA, pro algoritmy analýzy dat, aj., (ii) k indikaci stavu systému, např. jestli přichází data od lidarů, atd.

V RT vrstvě probíhá zpracování dat z měření vždy po získání všech 33 bodů z lidarů, tedy po každém jednom skenu. Výsledek spolu s nezpracovanými daty jsou posílány do PC. Algoritmus zpracování je popsán v sekci *Detekce objektů*. Jakmile je všech 33 bodů vyčteno z DMA FIFO, jsou data opatřena časovou značkou. Do počítače se odesílá datová struktura s těmito prvky:

Tabulka 2: Datová struktura dat posílaných z RT do PC

Prvek	Datový typ	Popis
R1	uint16 array [11 × 1]	Čistá data z lidarů L1
R2	uint16 array [11 × 1]	Čistá data z lidarů L2
R3	uint16 array [11 × 1]	Čistá data z lidarů L3
L1 TransCoords	double Cluster of 3 Arrays	X,Y,Z transformované souřadnice lidarů L1
L2 TransCoords	double Cluster of 3 Arrays	X,Y,Z transformované souřadnice lidarů L2
L3 TransCoords	double Cluster of 3 Arrays	X,Y,Z transformované souřadnice lidarů L3
Analyzed data	double Cluster of N arrays	X,Y souřadnice bodů jednotlivých shluků
Čas	time stamp	Časová značka dat při vyčtení z DMA FIFO

PC vrstva



Obrázek 28: Vzhled hlavního panelu uživatelského rozhraní

Tato vrstva aplikace běží na PC, které slouží jako periférie pro uživatele, jenž přes uživatelské rozhraní (UI) může řídit systém, nastavovat systémové parametry a zobrazovat výsledky analýzy a měření. Dále zde lze nastavit, která data si uživatel přeje ukládat do textového souboru. Aplikace je udělána tak, že ukládání dat není aktivováno jako výchozí stav, ale uživatel jej může aktivovat kdykoli i během měření. Data jsou ukládána do textového souboru oddělená středníkem, (čárkou LabVIEW indikuje desetinou

čárku). Název souboru se skládá z časové značky, kdy bylo ukládání spuštěno a názvu data struktury dat, která má být uložena. Aplikace umožňuje uložit 3 různé data struktury: Původní data přímo z lidarů (Raw data), Transformované souřadnice (Transformed coords) a Bezpečná oblast před vozidlo (Drivable Area), jež je znázorněna červeným obdélníkem na hlavním panelu UI (obr. 28). Prostřednictvím UI uživatel tedy spouští proces měření, může zapnout a vypnout jednotlivé elektronické komponenty (převodníky a budič), spustit auto kalibraci systému aj. UI aplikace je zobrazeno na obrázku 28.

Při spuštění aplikace, se nejdříve musí nastavit parametry pro FPGA, což jsou rychlost skenu a počet vzorkovacích měření. Poté se spustí RT část, která sama spustí automatickou konfiguraci systému.

6 Kalibrace

Po každé změně polohy lidarů na kapotě, je potřeba celý systém zkalibrovat. Kalibrace má několik fází, některé se týkají každého lidarů zvlášť a některé se týkají všech lidarů dohromady, aby bylo možné data z jednotlivých lidarů spojit dohromady, tedy aby si tentýž bod objektu naměřený jedním lidarem odpovídal s bodem naměřeným dalšími lidary. V průběhu kalibrace se zaznamenávají kalibrační parametry uvedené v tabulce 3, které se následně zanesou do systému přes GUI.

Tabulka 3: Kalibrační parametry

Název parametru	Veličina	Jednotka
Offset lidarů L1,L2,L3	$d_{off}^1, d_{off}^2, d_{off}^3$	cm
Výška lidarů L12, L3	h_{12}, h_3	cm
Vzdálenost lidarů L12 od L3	a	cm
Úhel naklonění lidarů L12	β_{12}	úhlový stupeň °
Úhel sklonu lidarů L1,L2,L3	ρ_1, ρ_2, ρ_3	úhlový stupeň °

6.1 Korekce offsetu

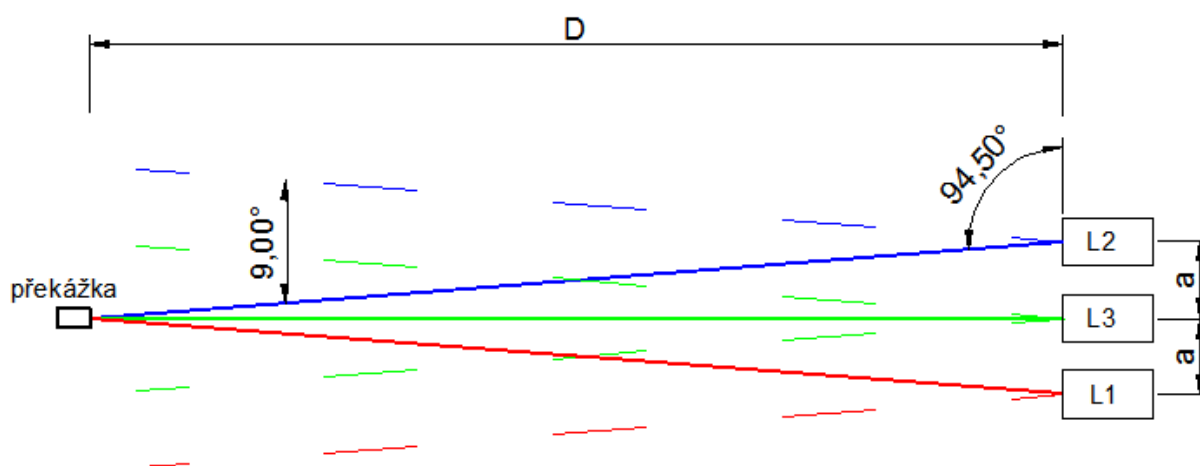
Před dalšími kroky kalibrace je potřeba porovnat hodnoty udávané lidary a jiným laserovým měřicím zařízením, které udává přesnost v milimetrech. Tato měřená vzdálenost by měla být alespoň více než 3 m. Naměří se několik hodnot stejného místa lidary R_i^l , kde $i \in \{1 \dots n\}$ je index vzorku, n je cca 100 a l je index lidarů $l \in \{1, 2, 3\}$ a několik hodnot laserovým měřidlem m_j , $j \in \{1 \dots 10\}$. Vypočtou se průměry pro každý lidar \bar{R}_l a laserové měřidlo \bar{m} . Dále se vypočte rozdíl mezi \bar{m} a \bar{R}_l a daný rozdíl d_{off}^l se uvede do kalibračních parametrů.

6.2 Korekce polohy lidarů

Tato část kalibrace řeší to, že když se lidary umístí ují na kapotu, tak se těžce umístí ují tak, aby prostřední paprsek skenu každého lidarů mířil paralelně s osou vozidla. Když by lidary mířily tak, že jsou jejich osy rovnoběžné, pak se určité paprsky ze tří lidarů potkají v konkrétním místě před vozidlem, obr. 29. Tyto paprsky jsou nazýváme významné paprsky a jsou to: pro pravý lidar L2 je to paprsek v poloze 94.5° , pro prostřední lidar je to prostřední paprsek (90°) a pro levý lidar L1 je to paprsek v poloze 85.5° . Tyto významné paprsky se potkají před vozidlem ve vzdálenosti D , jež záleží na tom, jak jsou lidary od sebe daleko, tedy na vzdálenosti a . Aby bylo možné určit, že se skutečně tyto paprsky v daném místě potkají, je před vozidlo umístěna malá překážka ve vzdálenosti D v ose vozidla. Překážka musí být malá tak, aby na ní dopadaly jen tyto paprsky a žádné jiné. Překážku nelze umístit dokonale do osy vozidla, proto je

uvažována chyba ± 5 cm, která ovlivňuje, s jakou přesností lze nastavit lidary. Pokud jsou boční lidary od prostředního lidaru umístěny ve vzdálenosti $a = 50$ cm, pak $D = 635.3$ cm.

Lidary se zapnou, nechají se kmitat a prostřední lidar se natočí tak, aby směřoval prostředním paprskem na překážku, poté se natočí levý lidar, aby jeho paprsek v poloze 85.5° směřoval na překážku a podobně pro pravý lidar. Přesnost této metody je ovlivněna přesností umístění překážky a lidskou schopností nastavit lidary do požadované polohy.



Obrázek 29: Korekce polohy lidarů pomocí tří významných paprsků – plné čáry

6.3 Určení výšky lidarů od vozovky

Výška jednotlivých lidarů je zjišťována pomocí laboratorního stojanu s prodlouženým ramenem. Konec ramena nastavíme do takové výšky, aby jeho konec byl nad středem optiky lidaru. K bázi ramene přiložíme laserové měřidlo a změříme vzdálenost od země. Tímto způsobem změříme všechny výšky lidarů h_l , které se zanesou do kalibračních parametrů. Krajní lidary L12 se snažíme umístit na kapotu tak, aby byly ve stejné výšce. Laserové měřidlo je velmi přesné, výrobce udává přesnost 1 mm, největší chyba při této kalibrační metodě vznikne při snaze zarovnat konec ramene se středem optiky lidaru, kde se odhaduje, že vznikne chyba 0.5 cm.

6.4 Nastavení sklonu lidarů

Sklon lidarů ρ by měl být pro dva krajní lidary stejný a takový, aby na rovné vozovce mířily na zem ve vzdálenosti větší než 35 m. Prostřední lidar by měl mířit na zem ve vzdálenosti 15 m. Tento sklon se upravuje pomocí konstrukce, jež toto nastavení umožňuje. Velikost tohoto úhlu závisí na výšce, v jaké jsou lidary na kapotě umístěné, zjištěné z předešlého kroku.

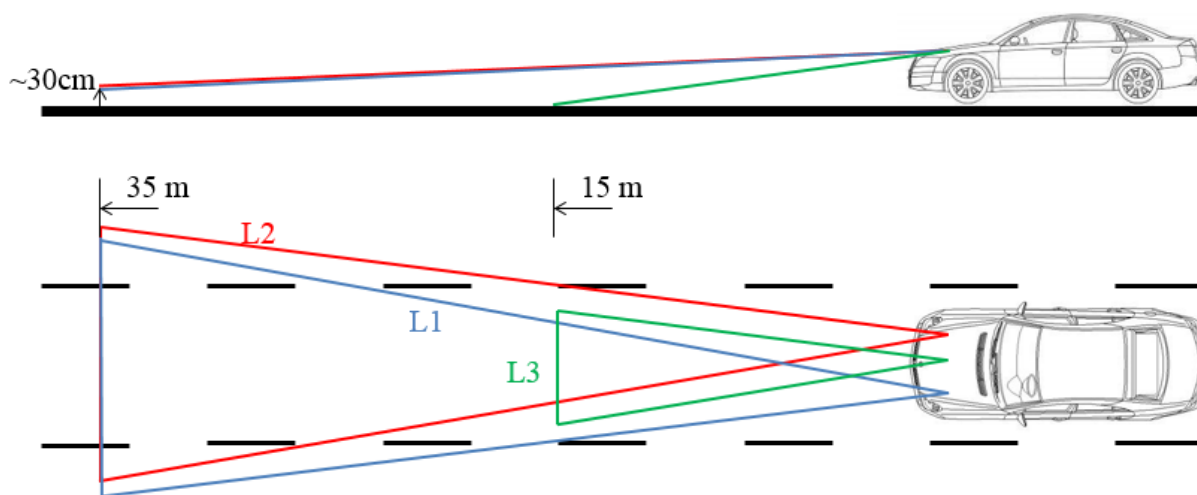
Sklon byl zjišťován akcelerometrem v mobilním telefonu přes aplikaci Accelerometer Meter. Použitý mobilní telefon byl Lenovo A6000 s čipem akcelerometru KXTJ9. Akcelerometr se před použitím musí nakalibrovat. Tato kalibrace byla uskutečněna aplikací Spirit Level pomocí šesti polohové kalibrační techniky. Výběrová směrodatná odchylka tíhového zrychlení měřené akcelerometrem je 0.01 m/s^2 (zjištěno opakovaným měřením) a offset od skutečné hodnoty tíhového zrychlení v Praze je 0.06 m/s^2 . Aplikace umožňuje zobrazení i sférických souřadnic a tedy úhel sklonu a naklonění, výběrová směrodatná odchylka těchto úhlů je 0.08° . Jelikož v určování sklonu a náklonu lidaru se provádí rozdíl mezi referenční hodnotou změřenou ve vozidle a na lidaru, vyruší se efekt offsetu měření akcelerometrem a nemá tak vliv na chybu měření.

Nejprve je potřeba určit sklon a naklonění vozidla (referenční hodnoty), jelikož vozidlo může stát např. do kopce a nakloněné na jednu ze stran. Lidary se musí nastavovat vůči vozidlu, ne vůči zemi. Sklon a náklon vozidla je potřeba určit na ploše uvnitř vozidla, jež lze považovat za rovnou, jako je např. dno kufru. Aplikací takto naměříme sklon vozidla ρ_v a naklonění vozidla β_v . Ideálně by vozidlo při této části kalibrace mělo stát na co možno nejrovnějším povrchu, jako je např. parkoviště, avšak ne vždy je toto možné.

Pro určení sklonu lidaru se musí měřit ve stejné orientaci jako v kufru, aby se nezměnil směr os akcelerometru. Sklon lidaru l se zjistí tak, že se mobil přiloží na tělo lidaru a odečte se sklon ρ'_l . Pomocí konstrukce se upraví tento sklon do požadované polohy, jež je taková, že vzhledem k výšce umístění lidaru je požadováno, aby ve vzdálenosti 35 m byl lidar schopen detekovat objekt o výšce 30 cm. Pokud je výška lidaru 100 cm, pak tyto úhly pro lidary L12 je ρ_1 a ρ_2 je $1,15^\circ$ (obr. 30) v případě, že vozidlo stojí na ideálně rovném povrchu. V případě, že vozidlo nestojí na rovném povrchu (prakticky vždy), pak tento úhel by měl být rozdíl mezi upraveným úhlem ρ'_l a ρ_v , tedy $\rho_l = \rho'_l - \rho_v$. Lidar se tedy nastaví do polohy nejbližší této hodnotě a tento sklon je pak uveden do kalibračních parametrů. Prostřední lidar by ve výšce 100 cm měl být sklopen o $3,8^\circ$, aby mířil 15 m před vozidlo na zem.

6.5 Určení naklonění lidarů

Tato část kalibrace se týká pouze lidarů L12, jelikož prostřední lidar L3 není v náklonu, což je dáno tvarem kapoty testovacího vozidla. Postupuje se podobně jako v předchozím kroku. Z předešlého kroku již známe úhel náklonu vozidla β_v . Akcelerometrem v mobilu změříme úhly náklonu β'_1 a β'_2 na lidarech L12. Skutečný náklon lidarů je pak $\beta_l = |\beta'_1 - \beta_v|$. Při této metodě je snaha o to, aby se lidary umístili tak, aby $\beta'_1 = \beta'_2$. Tyto hodnoty se opět uvedou do kalibračních parametrů.



Obrázek 30: Sklon jednotlivých skenovacích rovin a jejich prostorové rozložení

6.6 Odhad nejistot

Přesnost měření laserového skeneru je ovlivněna několika nejistotami, jež jsou shrnuty v tabulce 4. Jelikož při analýze a zobrazování výsledků se používají kartézské souřadnice, promítnou se do nich při přepočtu tyto nejistoty. Výsledkem transformace do kartézských souřadnic, jež je popsána v sekci *Detekce objektů*, jsou neměřitelné veličiny X, Y, Z . K určení nepřesnosti těchto veličin byl použit zákon o šíření nejistot podle vzorce:

$$u_f = \sqrt{\sum_{i=1}^n \left(\frac{\partial f}{\partial x_i} \cdot u_{x_i} \right)^2}$$

Tabulka 4: Zdroje nejistot laserového skeneru

Zdroj nejistoty	Veličina	Hodnota
Poloha motoru	$u_{\phi M}$	0.05°
Kalibrace natočení	$u_{\phi K}$	$0.5 - 1^\circ$
Kalibrace sklonu	u_{ρ}	0.16°
Měření úhlu naklopení	u_{β}	0.16°
Měření vzdálenosti mezi lidary	u_a	0.01 cm
Měření výšky lidarů od vozovky	u_h	0.5 cm
Nepřesnost lidarů	u_R	1 % vzdálenosti

Nejistota polohy motoru a měření vzdálenosti lidarů je daná výrobcem. Nejistota sklonu a naklopení je dána chybou akcelerometru vypočtenou z výběrové směrodatné odchylky, jež je vynásobena koeficientem $k = 2$ (měřeno více než 100 krát), což je rozšířená nejistota. Nejistota měření vzdálenosti mezi lidary, výšky lidarů od vozovky je dána laserovým měřidlem, jehož přesnost je uvedena výrobcem v manuálu. Nejistota z kalibrace natočení je dána nepřesností umístění terče a lidskou chybou při otáčení platformy a ověřování polohy na počítači. Nejistoty polohy motoru a z kalibrace natočení se sčítají, proto se pro

výpočet nejistot souřadnic používají hodnoty 0.55 a 1.05°.

Nejistoty veličin X, Y, Z rostou se vzdáleností měřeného objektu a zároveň boční lidary jsou ještě ovlivněny naklopením o úhel β do stran. Jelikož nejistotu natočení u_{ϕ_K} lze obtížně určit, byly vypočteny nejistoty pro nejlepší a nejhorší případ v závislosti na vzdálenosti objektu. Z výpočtů vyplynulo, že zanedbatelný vliv na nejistotu souřadnic má aktuální natočení, a to jestli jde o prostřední lidar L3 nebo o krajní lidary L12. Byly tedy vypočteny nejistoty souřadnic v závislosti na vzdálenosti od 1 do 35 metrů s krokem 1 metr a na hodnotách u_{ϕ_K} . Jelikož velikost nejistot je přímo úměrná vzdálenosti, uvedené výsledky v tabulce 5 jsou rozšířené relativní nejistoty o koeficient 2 vzhledem ke vzdálenosti R , tj. $\frac{u_X}{R}, \frac{u_Y}{R}, \frac{u_Z}{R}$. Z výpočtů vyplývá, že nejistoty souřadnic X a Z jsou velmi citlivé na přesnosti, s jakou jsou lidary vůči sobě pootočený a s jakou přesností lze určit sklon lidarů ρ .

Tabulka 5: Rozšířené relativní nejistoty lidarů pro nejhorší a nejlepší variantu

Rozšířená relativní nejistota	$\frac{u_X}{R}$	$\frac{u_Y}{R}$	$\frac{u_Z}{R}$
Hodnota	0.020 - 0.036	0.018 - 0.020	0.006

7 Detekce objektů

Tato část práce popisuje algoritmy a metody, které řeší problém detekce objektů z naměřených dat. Všechny dosavadní přístupy byly navrženy tak, aby byly schopny zpracovat velké množství bodů, jež lidar naměří během jednoho skenu. Zároveň pracovaly s širším zorným polem a s vyšším úhlovým rozlišením než má navržený systém v této práci. Z toho důvodu navržené algoritmy pro detekci překážek byly navrženy s ohledem na to, že navržený systém poskytuje malé množství bodů v malém zorném úhlu. Výchozí hypotéza je taková, že i z malého množství bodů je možno určit, které body náležejí jednomu objektu a které body náležejí již jinému objektu.

Přiřazení jednotlivých bodů jednotlivým objektům odpovídá úloha shlukování. Typické metody shlukování, jako např. *K means* algoritmus, počítají s tím, že je apriori dán počet shlukovaných jader. Avšak v úlohách detekce objektů před vozidlem, kde se počet objektů detekovaných mění, nelze určit počet jader apriori, proto tyto algoritmy, které potřebují určit počet shluků předem, není možné použít. To je jedna z klíčových požadovaných vlastností algoritmů pro segmentaci dat z lidarů.

Navržený algoritmus je navržen pro detekování objektů v reálném čase. V každém skenu je naměřeno 33 bodů. Na základě prostorových vztahů těchto bodů je možné rozpoznat, které body patří kterému objektu. Navržený algoritmus je podobný *RBNN* metodě založené na principu hledání nejbližšího souseda (nearest neighbour) a podobné *DBSCAN* algoritmu. Cílem algoritmu je provést segmentaci dat a najít v nich shluky, které odpovídají jednotlivým objektům.

Pro lepší srozumitelnost je potřeba definovat několik veličin používaných v algoritmu. Z každého lidarů po jednom skenu přijde 11 bodových měření $R^l = [r_1^l, \dots, r_{11}^l]$, $l \in \{1, 2, 3\}$. K nim se přiřadí vektor úhlů $\phi = [85.5^\circ : 0.9^\circ : 94.5^\circ]$ seřazených podle směru pohybu lidarů během tohoto skenu, tj. při pohybu proti směru hodinových ručiček je tento vektor seřazen vzestupně, v opačném směru sestupně. Každý bod měření lze tedy zapsat jako $P_i^l = [r_i^l, \phi_i]$. V algoritmu figuruje několik parametrů, z nichž všechny byly definovány v sekci o kalibraci a jsou uvedeny v tabulce 3. Počátek jednotného kartézského souřadného systému byl zvolen v poloze prostředního lidarů. Souřadný systém je pravotočivý s osou Y mířící od prostředního lidarů před vozidlo. Celý proces detekce objektů sestává z několika kroků:

1. Odstranění chybných měření: $[R_f, TransCoeffs_f] \leftarrow \mathbf{Filter}(R, TransCoeffs)$
2. Transformace souřadnic: $[R_t, X_t, Y_t, Z_t] \leftarrow \mathbf{TransSourad}(R_f, TransCoeffs_f)$
3. Detekce vozovky: $[R_{ng}, X_{ng}, Y_{ng}] \leftarrow \mathbf{DetekceVoz}(R_t, X_t, Y_t, Z_t)$
4. Fúze objektových dat: $[R_o, X_o, Y_o, \phi'] \leftarrow \mathbf{SpojData}(R_{ng}, X_{ng}, Y_{ng})$
5. Segmentace: $Shluky \leftarrow \mathbf{Segmenter}(R_o, X_o, Y_o, \phi')$

Jelikož se v algoritmu počítá transformace ze sférických souřadnic do kartézských, počítají se stále v algoritmu tytéž siny a kosiny úhlů $\phi_i, \rho_1, \beta_{12}$. Aby se algoritmus urychlil, byla vytvořena náhledová tabulka, jejíž hodnoty jsou vypočteny jen při prvním spuštění algoritmu a kdykoli se změní kalibrační parametry, a tak se nemusí tytéž výpočty se stejným výsledkem počítat při každé analýze nového skenu.

7.1 Pre-processing

Prvních sedm bodů lze považovat za před zpracování dat pro segmentaci, které začíná odstraněním chybných měření, za něž lze považovat měření se vzdáleností menší než R_{thrs} , což je práh, jehož hodnota je volena jako 20 cm. V takové vzdálenosti by se před lidary nemělo nic vyskytovat. Všechny takové body jsou odstraněny a spolu s nimi i odpovídající prvky v náhledové tabulce.

Transformace souřadnic

Transformace souřadnic ze sférických do kartézského systému se liší pro krajní lidary a pro prostřední lidar. Pro prostřední lidar je to klasická transformace ze sférických souřadnic do kartézských, avšak pro boční lidary se tato transformace komplikuje ještě náklonem bočních lidarů. Následující rovnice popisují transformaci souřadnic pro prostřední lidar L3.

$$X_i = r_i \cdot \cos(\phi_i) \cdot \cos(\rho_3) \quad (9)$$

$$Y_i = r_i \cdot \sin(\phi_i) \cdot \cos(\rho_3) \quad (10)$$

$$Z_i = h_3 - r_i \cdot \sin(\rho_3) \quad (11)$$

Pro lidary L12 platí následující transformační rovnice:

$$X_i = r_i \cdot \cos(\phi_i) \cdot \cos(\rho_{12}) \cdot \cos(\beta_{12}) \pm a \quad (12)$$

$$Y_i = r_i \cdot \sin(\phi_i) \cdot \cos(\rho_{12}) \quad (13)$$

$$Z_i = h_{12} - r_i \cdot [\sin(\rho_{12}) \pm \cos(\phi_i) \cdot \cos(\rho_{12}) \cdot \sin(\beta_{12})] \quad (14)$$

Pro data z lidaru L1 se v rovnici (4) parametr a (vzdálenost mezi lidary) odečítá, pro lidar L2 přičítá, totéž platí v rovnici (6) pro daný výraz. Všechny výrazy v rovnicích, kde se násobí siny a kosiny, jsou uloženy v náhledové tabulce. Náhledová tabulka tedy obsahuje transformační koeficienty, kterými se při každém zpracování nového skenu jen násobí naměřené vzdálenosti. Transformační rovnice (9-14) převádí všechny body zároveň do jednotného kartézského souřadného systému.

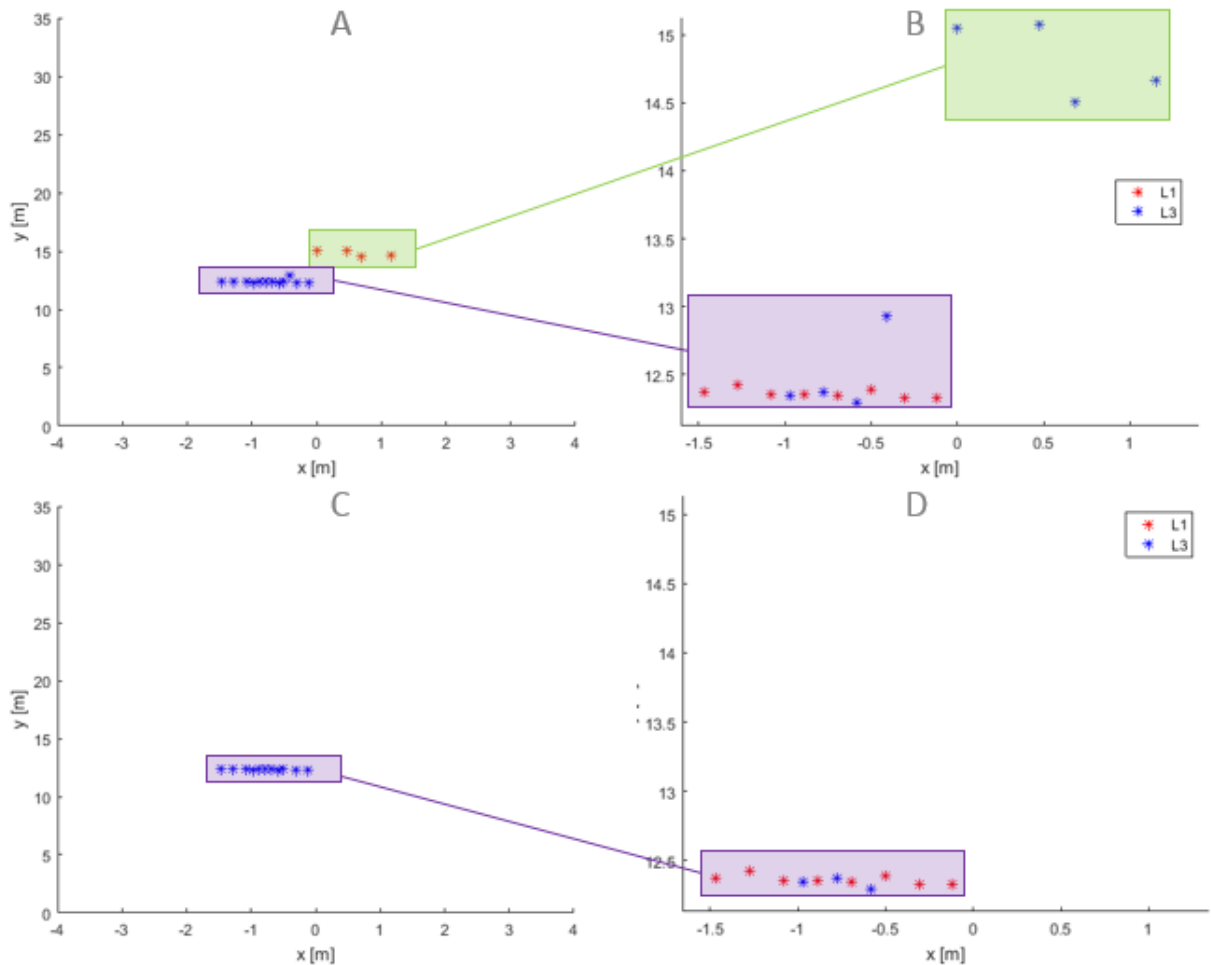
Detekce vozovky

Tato část zpracování dat je nejvíce náchylná k chybám, což způsobuje chybu v segmentaci. Zdroje chyb této části jsou jednak nejistoty vzniklé při kalibraci, hlavně parametrů ρ_l, β_l , jednak neznalosti či neschopnosti odhadovat skutečný profil vozovky před vozidlem. První návrh tohoto algoritmu byl takový, že systém počítá s tím, že před vozidlem je vždy rovna vozovka. Detekce vozovky měla fungovat tak, že jakmile souřadnice Z bodu P_i je nižší než práh $Z_{thrs} = 15$ cm, pak je tento bod zařazen mezi body vozovky do množiny P^v , ostatní body jsou zařazeny do množiny P^o , což jsou potenciální objektová data. Avšak praktické testy potvrdily předpoklad, že takto segmentovat vozovku je neúčinné. I při mírném stoupání si systém může splést jiné vozidlo s vozovkou. Náhle změny gradientu vozovky způsobí dočasné snížení schopnosti správné segmentace.

Proto byl algoritmus detekce vozovky přepracován a postaven na odlišné myšlence. Představme si, že prostřední lidar je sklopen o úhel ρ_3 a krajní lidary jsou nastaveny vodorovně, tj. $\rho_{12} = 0^\circ$. Když bude překážka za místem dopadu paprsku prostředního lidaru na vozovku, pak prostřední lidar bude detekovat vozovku ve vzdálenosti R_3 a krajní lidary budou detekovat překážku ve vzdálenosti R_{12} . Tím, že jsou souřadnice bodů z prostředního lidaru výrazně odlišné od souřadnic bodů z krajních lidarů, je jasné, že to, co detekuje prostřední lidar, musí být vozovka. Pokud bychom překážku umístili blíže k vozidlu, začal by i prostřední lidar detekovat překážku a souřadnice všech tří lidarů by byly více méně podobné, z toho plyne, že to, co detekuje prostřední lidar, není vozovka.

Problémy by mohly nastat v situaci, kdy překážka je těsně za místem, kam míří paprsky z prostředního lidaru, pak by totiž byly body vozovky přiřazeny do shluku překážky, jak se to např. děje na obr. 41. Dále nastanou problémy v případě náhlého stoupání vozovky, jako je např. v podzemních garážích. Při výjezdu z garáží je vozidlo na rovné vozovce a blíží se výjezdu, který prudce stoupá. Nejprve toto stoupání zaznamenají krajní lidary, které to vyhodnotí jako překážku. Jakmile se vozidlo ještě více přiblíží ke stoupání, zaznamená jej i prostřední lidar, souřadnice budou více méně podobné pro všechny lidary a systém tento svah chybně vyhodnotí jako překážku.

K určení tzv. blízkosti, nebo podobnosti souřadnic prostředního lidaru a krajních lidarů je použita vzdálenost jednotlivých bodů z prostředního lidaru, které prošly filtrem chybného měření, ke všem ostatním bodům z krajních lidarů. Pokud není vzdálenost bodů z prostředního lidaru k žádnému bodu z krajních lidarů menší než práh R_z , pak je tento bod klasifikován jako bod vozovky a je zařazen do množiny P^v , jinak do množiny P^o . Hodnota prahu R_z byla zvolena jako 50 cm. Porovnání těchto dvou přístupů, tj. jednoduché filtrace pomocí Z -ové souřadnice a upraveného algoritmu, lze vidět na obr. 31.



Obrázek 31: Srovnání dvou přístupů detekce vozovky na reálných datech. Obr. A a B jsou výsledkem detekce vozovky a následné segmentace na základě jednoduché filtrace dat podle Z-ové souřadnice. Obr. C a D jsou výsledkem detekce země pomocí upraveného algoritmu. Obr. B a D jsou v měřítku a zobrazují body na překážce z jednotlivých lidarů, obr. A a C jsou segmentované shluky. Na obr. A lze vidět červený shluk, který je chybně detekován jako překážka, přestože jsou to body odpovídající vozovce. Upravený algoritmus si s tímto umí poradit, jak lze vidět na obr. C.

Fúze dat

Nyní, když jsou separovaná data objektů od vozovky, je potřeba přepočítat jejich úhel vzhledem k novému jednotnému souřadnému systému, jelikož data krajních lidarů byla posunuta z jejich lokálního souřadného systému do jednotného souřadného systému. Tento výpočet se týká pouze bodů v množině P^o . Nový transformovaný úhel bodu i je $\phi'_i = \text{atan2}(y'_i, x'_i)$. Aby následná segmentace proběhla správně, musí se ještě data z množiny P^o seřadit vzestupně podle úhlu ϕ' .

7.2 Segmentace

Cílem segmentace je odlišit od sebe jednotlivé objekty na základě prostorových vztahů jejich bodů. Segmentaci vystihuje popis funkce **Segmenter**, který je dále detailněji popsán.

1. Vypočti pro každý prvek p_i^o distanční elipsu $DE_{i,j}$: $DE \leftarrow \mathbf{DistancElip}(R_o, X_o, Y_o, \phi', \Delta\phi)$
2. Vytvoř matici sousednosti NN : $NN \leftarrow \mathbf{NajdiSousedy}(DE)$
3. Proveď shlukování: $[C, nC] \leftarrow \mathbf{Shlukovani}(NN)$
4. Vytvoř matici shluků $[MC, nCs] \leftarrow \mathbf{Shluky}(C, nC)$
5. Asociuj body shluků s jejich souřadnicemi: $[X_c, Y_c] \leftarrow \mathbf{Asociace}(MC, nCs, X_o, Y_o)$

Nejprve se vypočte kolem každého bodu distanční elipsa. Její X-ová poloosa je dána předpisem:

$$a_i = r_i \cdot (\sin(\Delta\phi) + e_r) \quad (15)$$

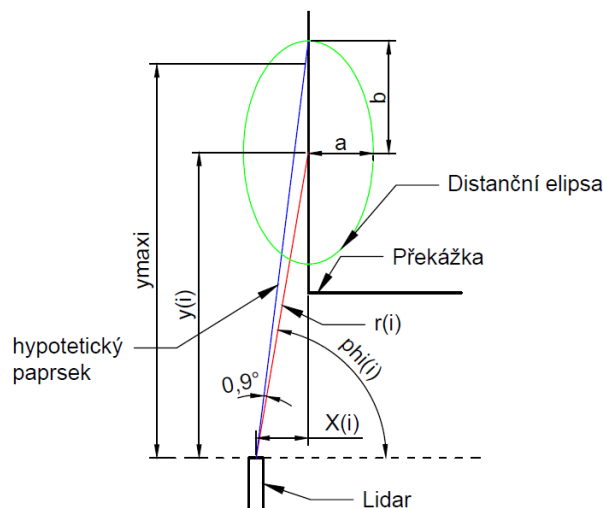
Kde $\Delta\phi = 1.8^\circ$ je uhlový rozsah pro hledání sousedů, jehož hodnota je zvolena, že by měl vidět maximálně 2 nejbližší sousedy. Nastavení této hodnoty má velký dopad na schopnost odlišit od sebe dva objekty, příliš nízká hodnota povede k nedostatečné segmentaci a jeden objekt bude rozdělen mezi více shluku. Vyšší hodnota by naopak zase mohla vést k tomu, že dva objekty budou segmentovány jako jeden shluk. $e_r = 0.01$ je nepřesnost lidarů. Určení délky Y-ové poloosy má více kroků a stojí za ním myšlenka zachycena na obr. 32.

Představme si, že paprsek bodu (červený) i dopadl na překážku v místě o souřadnicích $[x_i, y_i]$ pod úhlem ϕ_i . Následující bod $i+1$ je od něj úhlově vzdálen o velikost kroku motoru, což je 0.9° , tj. leží na paprsku pod úhlem $\phi_i + 0.9^\circ$ (modrý). Pokud je překážka orientována rovnoběžně s osou lidarů jako na obr. 32, pak hypotetický paprsek dopadne na překážku nad předešlý bod i . Pak tedy jeho souřadnice by měly být $[x_i, y_{maxi} = x_i \cdot \tan(\phi_i + 0.9^\circ)]$. Rozdíl Y-ových souřadnic těchto bodů je $\Delta y = y_{maxi} - y_i$, Y-ová délka poloosy elipsy tedy je $b = \Delta y + r_i \cdot e_r$. Čím vzdálenější tyto body jsou od lidarů, tím větší vzdálenost mezi body je a tedy b může být i dost velké na to, aby se dalo s určitou jistotou říct, že tyto dva body patří jednomu objektu. Proto je zaveden práh Y_{max} pro maximální velikost této poloosy, jelikož nad hodnotou tohoto prahu již nelze říct s dostatečnou jistotou, že tyto body jsou sousedé na stejném objektu, a tedy:

$$b = \min\{Y_{max}, \Delta y + r_i \cdot e_r\} \quad (16)$$

Velikost prahu Y_{max} byla zvolena jako 1.5 m.

Tímto se náš přístup liší od jiných algoritmů, které používaly kružnicový práh pro hledání svých nejbližších sousedů, avšak právě natočení překážky má vliv na to, jak blízko si sousední body jsou, a tak s kružnicovým prahem může častěji docházet k nadbytečné segmentaci a tedy rozdělení téhož objektu na



Obrázek 32: Odvození velikosti Y-ové poloosy distanční elipsy

více shluků především v podélném směru. Proto se tato metoda pro prahovou funkci jmenuje EBNN (Elliptically Bounded Nearest Neighbour) jako analogie k RBNN (Radially Bounded Nearest Neighbour).

Když jsou známy poloosy a_i, b_i elipsy $DE_{i,j}$ pro každý bod i z množiny P^o a jeho ostatní sousední body j , pak lze určit předpis takové elipsy jako:

$$DE_{i,j} : \frac{(x_i - x_j)^2}{a_i^2} + \frac{(y_i - y_j)^2}{b_i^2} = 1 \quad (17)$$

Kde x_j, y_j jsou souřadnice všech ostatních $j \in 1 \dots m$, kde m je počet bodů v P^o . Tímto vznikne matice pro distanční elipsu DE , jejíž rozměr je $m \times m$. Po dosazení těchto hodnot do rovnice 17, je hodnota levé strany buď menší než 1, rovna 1, nebo větší než 1. Pokud hodnota je menší nebo rovna 1, pak daný bod j je v sousedství bodu i . Tímto způsobem se určí hodnoty matice $NN : DE_{i,j} \leq 1 \Rightarrow NN_{i,j} = 1$. Rozměr matice NN je stejný jako matice DE .

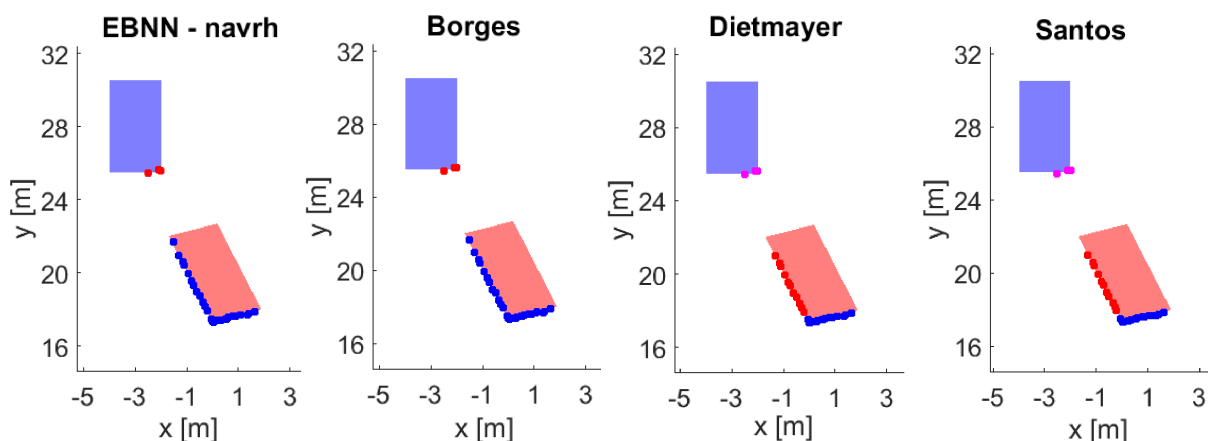
Shlukování je funkce, která přiřazuje jednotlivé body do shluků podle vztahů sousednosti vycházejících z matice NN . Pole C zachycuje, jaký shluk patří kterému bodu, např. $C_3 = 1$ znamená, že bod 3 patří do shluku 1. Pokud bod i má sousední body, tj. existuje $j \in i + 1 \dots k$ takové, že $NN_{i,j} = 1$, pak všechny takové sousední body jsou přidány do jednoho shluku a následuje hledání sousedních bodů posledního přidaného bodu do shluku. Takto jsou postupně přidávány další sousední body. Pokud bod už nemá další sousední body, hledá se další bod, který ještě není ve shluku a následuje nové hledání. Pokud bod i není ještě ve shluku a mezi jeho sousedními body je bod l již ve shluku, pak bod i je přidán do shluku bodu l . Pokud bod nemá žádné sousední body, pak je to tzv. *singleton*. Shlukování je ukončeno, pokud každému bodu je přiřazen shluk, nebo je singleton. Po vykonání funkce *Shlukování* jsou známy pole C a pole nC , které obsahuje počet bodů v daném shluku. Singleton se nepovažuje za objekt, jelikož je tvořen jen jedním bodem a nelze tedy určit, jestli to je např. tyč, nebo jen chyba měření.

Když jsou body rozděleny do shluku, je potřeba vytvořit datovou strukturu, která obsahuje souřadnice bodů daného shluku, což je úlohou funkce *Asociace*, tedy přiřazení indexům bodů ve shluku jejich souřadnice. Na závěr algoritmus nalezne nejbližší shluk, který jakkoli zasahuje do jízdního pruhu vozidla a určí tak bezpečnou oblast, kde vozidlo může jet. Na monitoru uživatele je tato bezpečná oblast znázorněna červeným obdélníkem, např. na obr. 28. Celý algoritmus byl nejprve naprogramován v MATLAB jako prototyp, který byl pak naprogramován v LabVIEW.

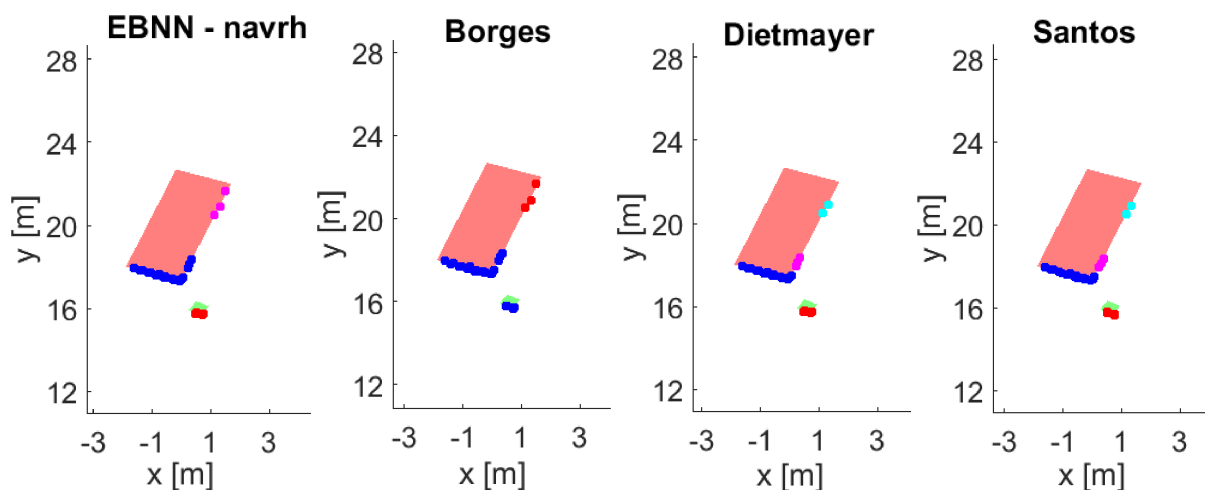
7.3 Výsledky simulace

V podsekcí Shlukování v sekci 3 byly zmíněny 3 prahové funkce, které určují maximální vzdálenost mezi sousedními body, aby byly ještě zařazeny do stejného shluku. Táto podsekcce srovnává navrženou prahovou funkci v algoritmu *EBNN* s ostatními prahovými funkcemi, tj. *Borges*, *Dietmayer*, *Santos*. Srovnání bylo provedeno v simulačním framework v MATLAB. V této simulaci byly vytvořené umělé objekty představující vozidla nebo malé objekty jako je člověk či strom, sloup. Měření lidarů byla simulována s šumem, jenž odpovídá nepřesnosti lidarů. Virtuální měření, která dopadla na objekty, byla segmentována navrženým algoritmem pro jednotlivé prahové funkce. Simulace nebrala v potaz neschopnost lidarů v určitých situacích změřit vzdálenost např. vlivem špatném reflexivity objektů. Dále byl uvažován pouze případ, že všechny lidary jsou sklopené pod stejným úhlem 0° . Parametr $\Delta\phi = 0.9^\circ$ byl použit pro všechny prahové funkce. Parametr *EBNN* byl $Y_{max} = 1.5$ m. Parametr pro funkci *Borges* je $\lambda = 10^\circ$, což je hodnota navržená v *Borges et al.* [5]. Pro *Santos* byl použit parametr $\beta = 35^\circ$, jež dával uspokojivé výsledky. Vozidlo s lidary je umístěno v počátku souřadného systému.

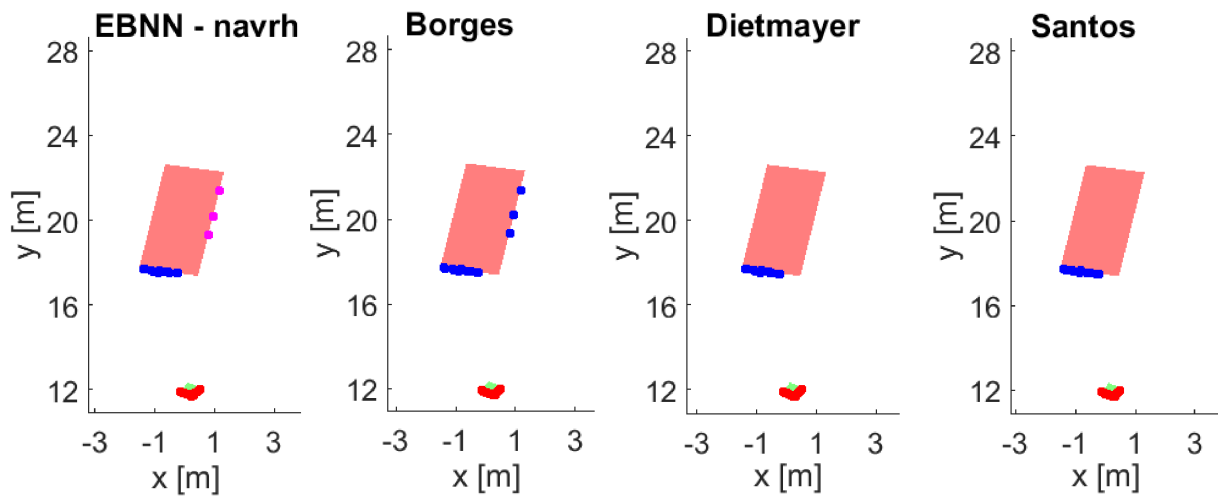
Z výsledků je patrné, že *Dietmayer* a *Santos* neposkytují uspokojivé výsledky segmentace, kdy často vedou k nadměrné segmentaci. Problém nejčastěji mají s podélnou segmentaci, kdy jednoduchý malý kruhový práh selhává. *Borges* se jeví jako velmi dobrá prahová funkce pro podélnou segmentaci a i v případě zakrytí objektu jiným objektem. Je to dáno tím, že vytváří velkou hodnotu prahu, takže dovolí i vzdálenějším bodům patřit do stejného objektu. Což je výhoda pro podélnou segmentaci, ale jak lze vidět na výsledcích (obr. 36 a 40) velká hodnota prahu je nevýhoda, pokud jsou dva odlišné objekty blízko u sebe, což vede k nedostatečné segmentaci a dva objekty jsou reprezentovány jedním shlukem. Prahová funkce *EBNN* selhává ve dvou případech. Za prvé, když je objekt zakryt jiným objektem, pak zakrytý objekt je často rozdělen na více shluků. Za druhé, ve vzdálenostech větších než cca 27 m pokud je objekt natočen, podélná segmentace selhává a opět vzniká více shluků na jednom objektu. Velký vliv na podélnou segmentaci u *EBNN* má parametr Y_{max} . Hodnota 1.5 m je v případě větší vzdálenosti (cca 27 m) objektu od lidarů příliš nízká. Avšak větší hodnota by vedla ke spojení dvou objektů do jednoho shluku tak, jak se to děje v případě prahové funkce *Borges*.



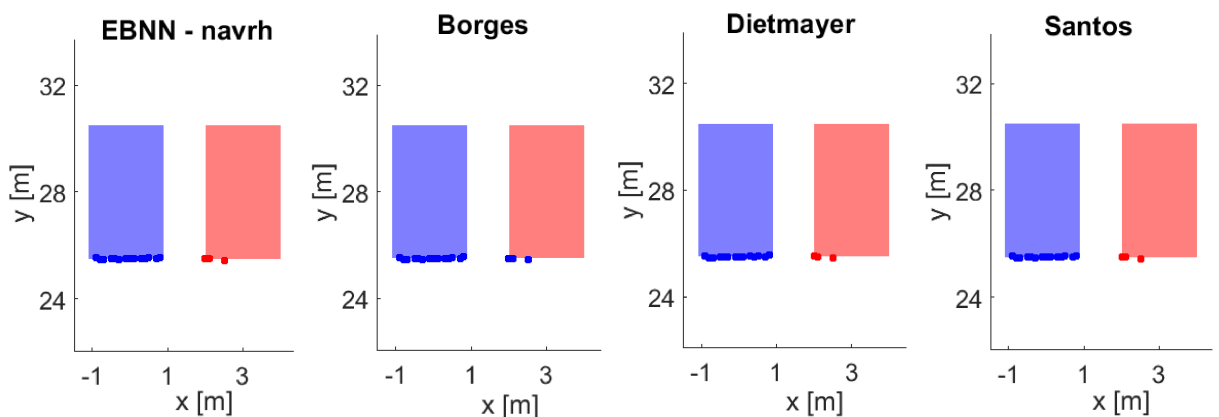
Obrázek 33: Tato situace představuje okamžik, kdy modré vozidlo jede v levé pruhu a červené vozidlo se zařazuje před vozidlo s lidary. EBNN a Borges dokážou provést longitudinální (ve směru osy Y) segmentaci červeného objektu správně, aniž by vznikly dva shluky na jednom objektu. Zatímco Dietmayer a Santos shodně vedou ke vzniku dvou shluků pro jeden objekt (červené a modré body). Všechny prahové funkce byly schopny odlišit od sebe dva objekty – modré a červené vozidlo.



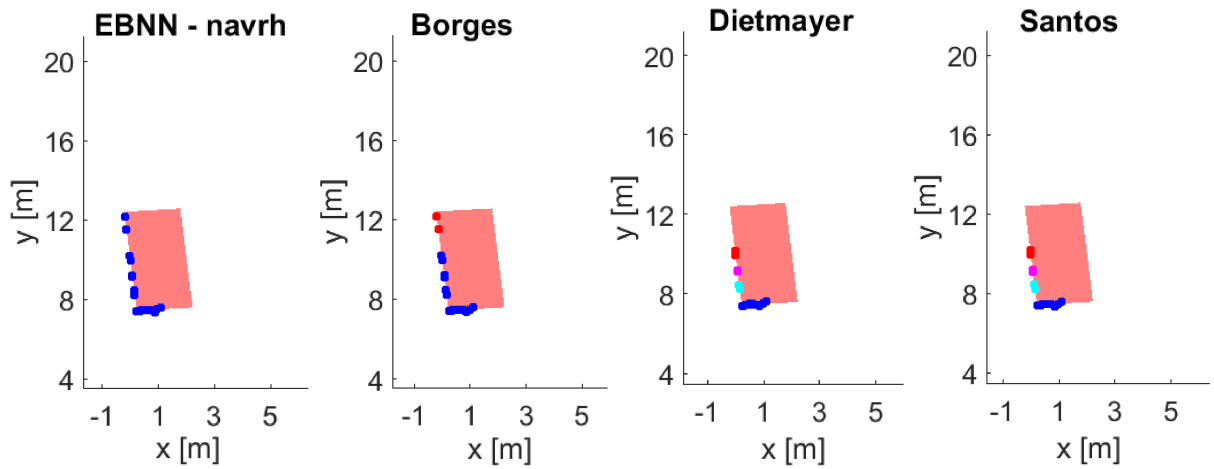
Obrázek 34: Červené vozidlo je částečně zakryto zeleným malým objektem. EBNN vede k nadměrné segmentaci, kdy vlivem zakrytí vznikly modrý a růžový shluk pro jeden objekt. Borges má stejný problém se zakrytím, navíc chybně spojil dva objekty v jeden – modrý shluk obsahuje i malý zelený objekt. Dietmayer a Santos oba sice odlišily dva objekty od sebe, ale červenému objektu přiřadily 3 různé shluky.



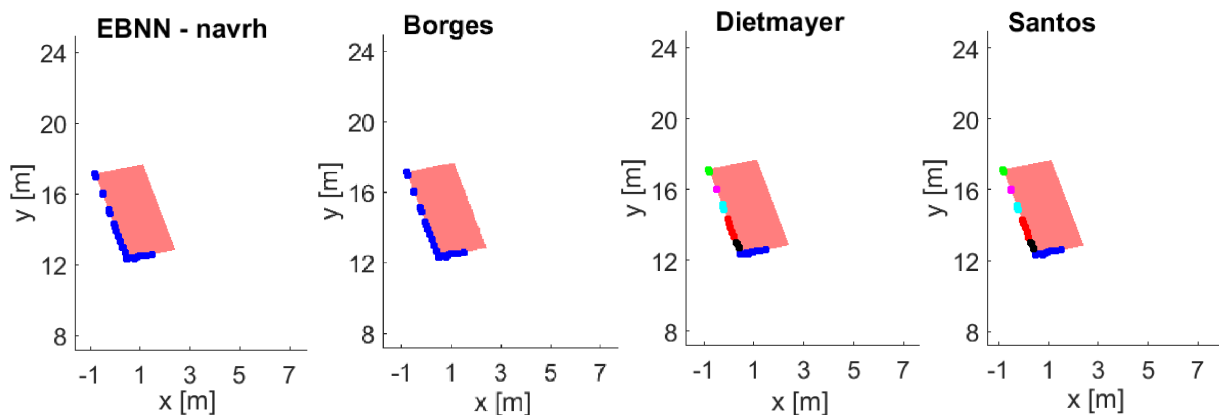
Obrázek 35: V tomto případě je zakrývající objekt blíže k lidarů, než na obr. 34. EBNN má stále problém s longitudinální segmentací a vytváří více shluků pro červený objekt. Borges si poradil správně s zakrývajícím objektem a červený objekt tedy není rozdělen na více shluků. Dietmayer a Santos boční body na červené vozidle nedokázaly segmentovat, tj. každý bod byl klasifikován jako singleton.



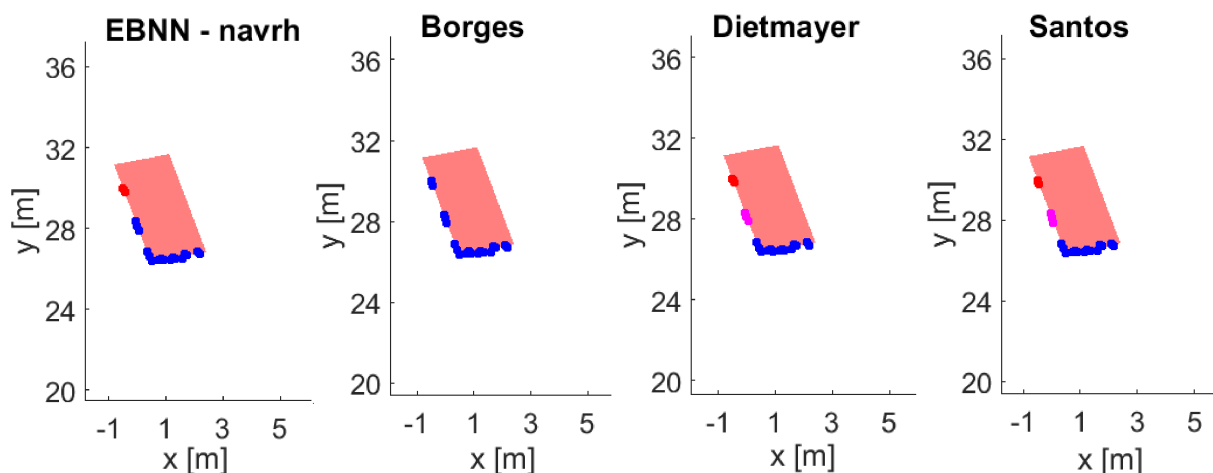
Obrázek 36: Tato situace simuluje polohu dvou aut, kdy modré je ve stejném pruhu jako vozidlo s lidarů a červené je v pravém pruhu. Vzdálenost mezi vozidly je 1 m. EBNN správně tyto dva objekty od sebe odliší dvě shluky – modrý pro modré vozidlo a červený pro červené. Borges chybně sloučí tyto dva odlišné objekty do jednoho shluku. Dietmayer a Santos poskytují stejný výsledek jako EBNN.



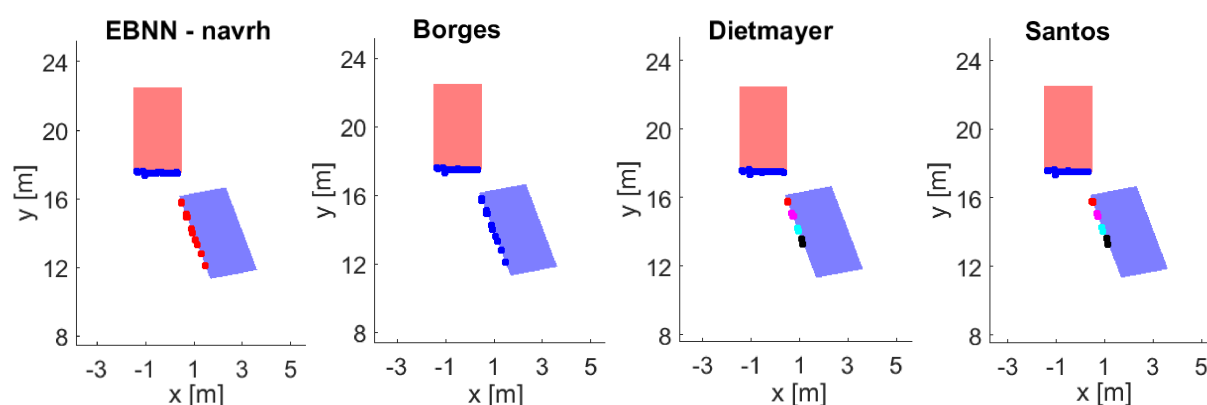
Obrázek 37: Tento případ simuluje situaci, kdy červené vozidlo se připojuje do jízdního pruhu před vozidlo s lidary po úhlem 5° . Zadní část objektu je cca 8 m od lidarů a EBNN správně provedlo longitudinální segmentaci a nerozdělilo červený objekt na více různých shluků. Avšak Borges vedl k vytvoření 2 shluků pro jeden objekt. Dietmayer a Santos vytvořily dokonce více než 2 shluky pro jeden objekt.



Obrázek 38: Podobná situace jako v obr. 37, avšak červený objekt je trochu dále od lidarů a je natočen o úhel 10° . Jak EBNN, tak Borges provedly segmentaci správně. Dietmayer a Santos opět vedly na rozdělení objektu na více shluků.



Obrázek 39: Objekt byl posunut ještě dále od lidarů, natočení bylo zachováno jako v obr. 38. Z výsledku vyplynulo, že jakmile se objekt dostane na vzdálenost větší než 27 m, pak EBNN začíná mít problém segmentovat objekty v podélném směru a rozdělilo objekt na více shluků. Borges s rostoucí vzdáleností naopak nemá problém. Dietmayer a Santos opět vedou k nadměrné segmentaci.



Obrázek 40: Příklad, kdy dvě vozidla jsou blízko u sebe. EBNN správně přiřadí každému objektu jeden shluk. Borges spojí dva objekty v jeden shluk. Dietmayer a Santos opět segmentují nadbytečně.

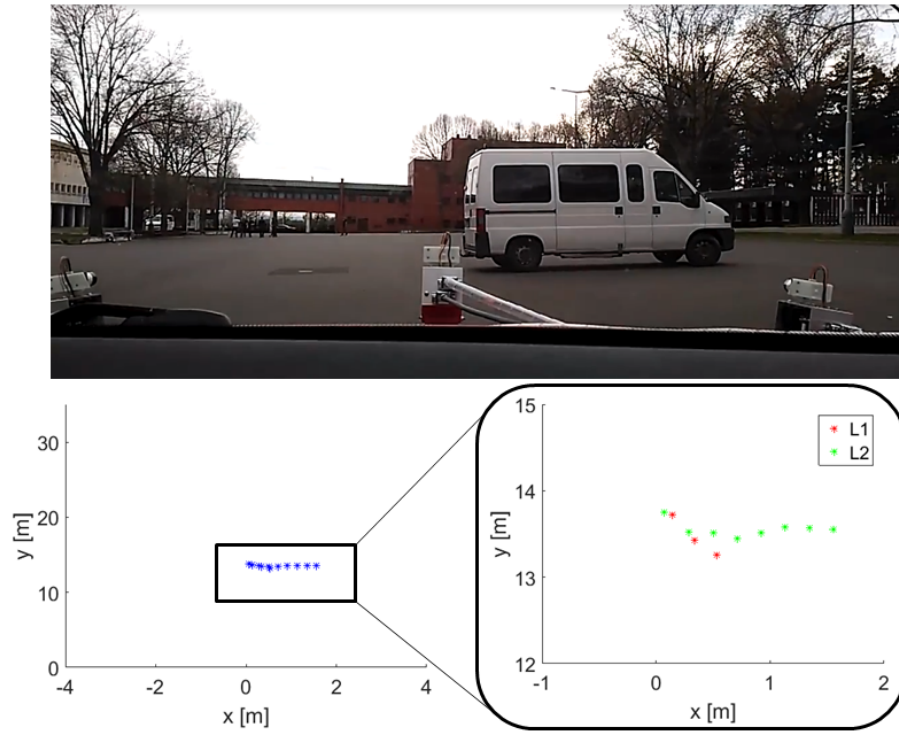
8 Testování

Testování probíhalo na parkovišti na Strahově při rychlostech do 30 km/h. Rychlost skenování byla nastavena na 80 ms. K přednímu sklu byl připevněn mobil s kamerou, který snímal prostor před vozidlem, aby bylo možné pak zpětně se podívat na to, co lidary naměřily. Tato analýza probíhá off-line, kamera není nijak s lidary synchronizována, resp. ukládání dat z lidarů je spuštěno přibližně ve stejný okamžik jako nahrávání. Byly provedeny dva testy, první se sklopeným prostředním lidarem do jiné roviny než zbylé dva, tak jak je popsáno v předešlých sekcích. A druhý s lidary ve stejných skenovacích rovinách. Cílem těchto testů bylo ověřit schopnost navrženého systému detekovat překážky před vozidlem a ověřit funkčnost algoritmu pro segmentaci. Náklon β bočních lidarů vlivem zešikmení kapoty byl změřen akcelerometrem v mobilu jako 7.9° . Boční lidary byly od prostředního lidaru vzdáleny o 50 cm.

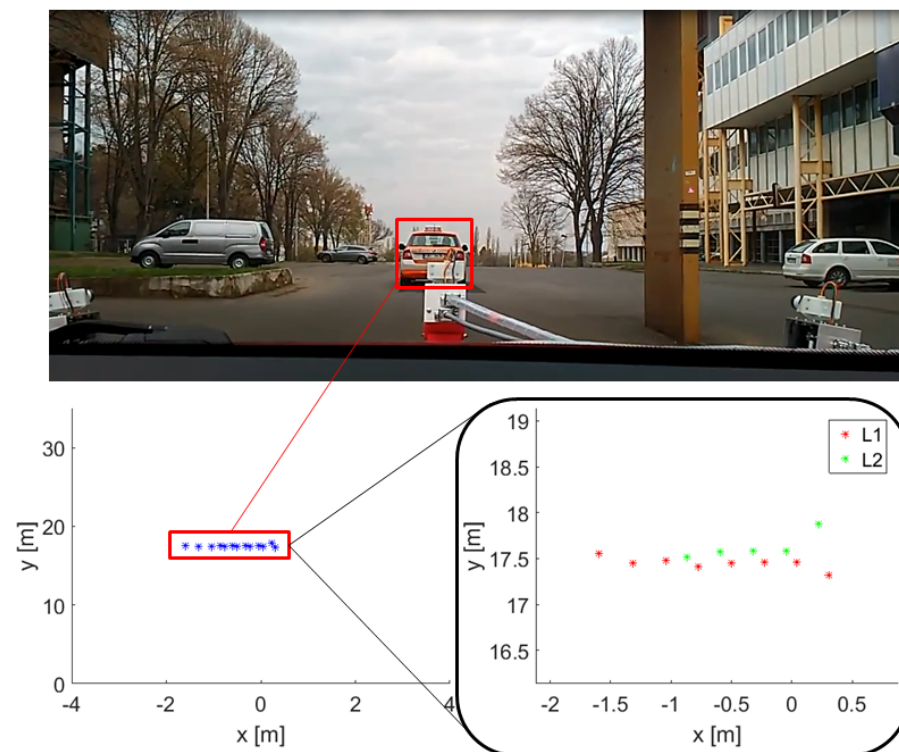
8.1 Test 1

Nejprve byl systém otestován v konfiguraci popsané v předešlých sekcích, tedy že prostřední lidar L3 je sklopen do jiné roviny než dva boční lidary o 3° níže. Krajiní lidary byly nastaveny co možno nejvíce vodorovně, tzn. $\rho_{12} = 0^\circ$, $\rho_3 = 3^\circ$. Tento test byl proveden se základním algoritmem pro filtraci vozovky založeném na filtraci podle Z-ové souřadnice, úprava algoritmu a ověření proběhlo off-line po testování. Zde se projevil očekávaný problém s detekcí země. Jednoduché filtrování země popsané v předešlé sekci funguje pouze na více méně rovné vozovce, hrboly nebo změny průběhu profilu vozovky způsobují to, že vozovka je detekována jako překážka. To je způsobeno nastavením prahu pro Z-ovou souřadnici změřených bodů, pokud tato hodnota není nižší než práh, pak je tento bod považován za bod, který neleží na vozovce. Výsledky zde uvedené jsou však už s upraveným algoritmem pro filtraci vozovky.

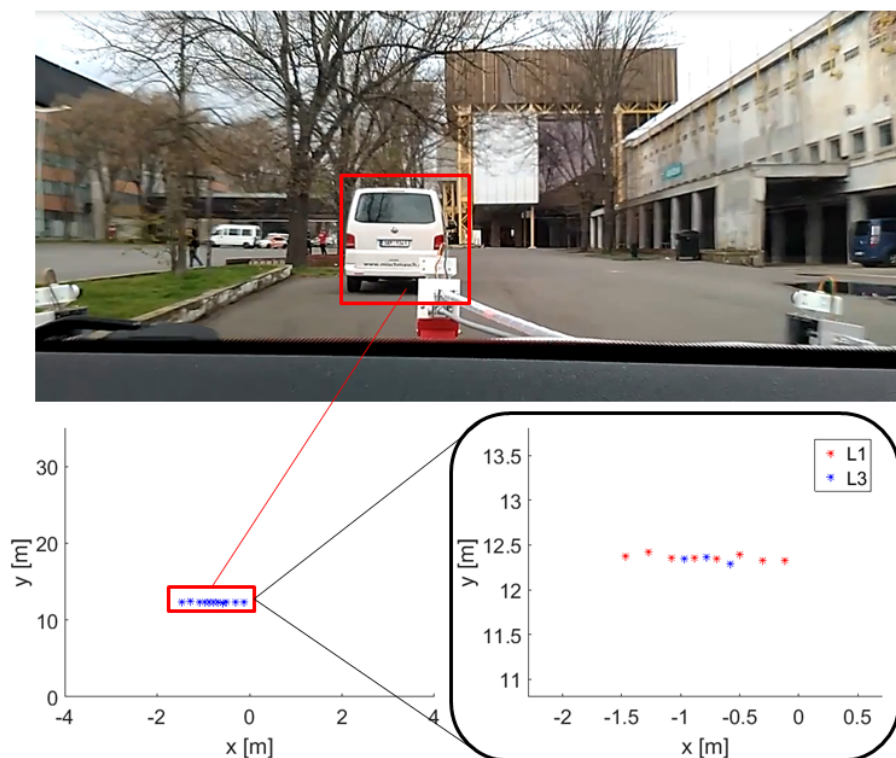
Jedním možným řešením špatné detekce vozovky by bylo vyčítat z lidarů ještě data o intenzitě vraceného paprsku, z něhož lze usuzovat z jakého povrchu se odrazil. Zde by nastaly dva problémy. První problém se týká rozhodování jaká intenzita odraženého světla náleží vozovce a jaká jinému povrchu. Když by byla vozovka mokrá, zasněžená, nebo by na ní bylo více vodorovného značení (např. přechod pro chodce), pak by tato intenzita byla jiná, než na suché vozovce, což by komplikovalo filtraci vozovky. Druhý problém je technický. Vyčítat další data z lidarů o intenzitě by způsobilo zdržení celého detekčního procesu způsobeným trváním překladů a celé komunikace. Další možným řešením by mohlo být upravit současný algoritmus, tak jak bylo popsáno v sekci o detekci vozovky. Na následujících obrazech (41 – 49) je vždy fotografie scény snímané lidary a pod fotografií jsou dva obrázky vizualizace dat z lidarů. Levý obrázek znázorňuje segmentaci scény na jednotlivé shluky barevně odlišené. Pravý obrázek je detail objektů s vyobrazením, kterému lidarů náleží jednotlivé body objektů.



Obrázek 41: *Test 1: Detekce dodávky v takové vzdálenosti, že prostřední lidar ji vůbec nedetekuje, místo toho vidí vozovku, avšak algoritmus pro detekci vozovky je schopen ji odfiltrovat.*



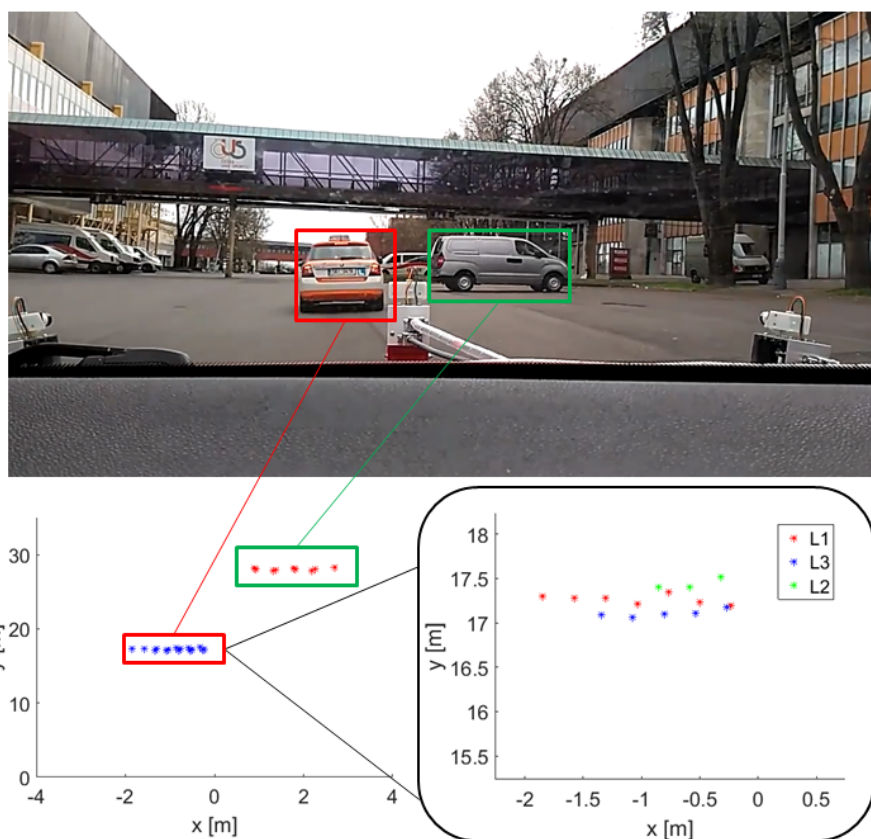
Obrázek 42: *Test 1: Detekce vozidla autoškoly (modrý shluk na levém spodním obrázku). Zde lze opět vidět úspěšnou filtraci vozovky, která je detekována v oblasti cca 14 m před vozidlem.*



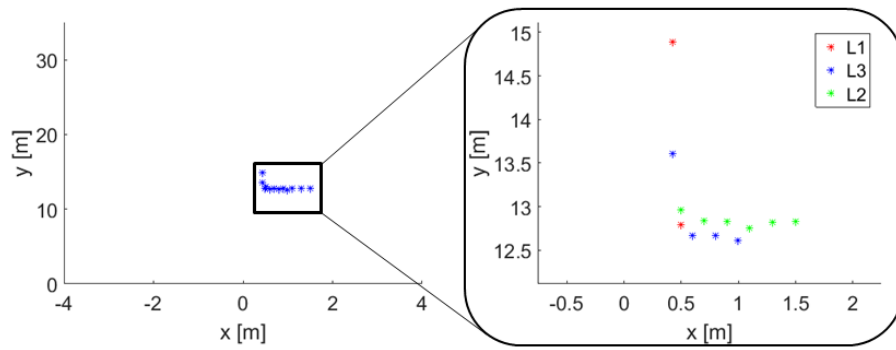
Obrázek 43: *Test1: Detekce bílého minivanu (modrý shluk)*

8.2 Test 2

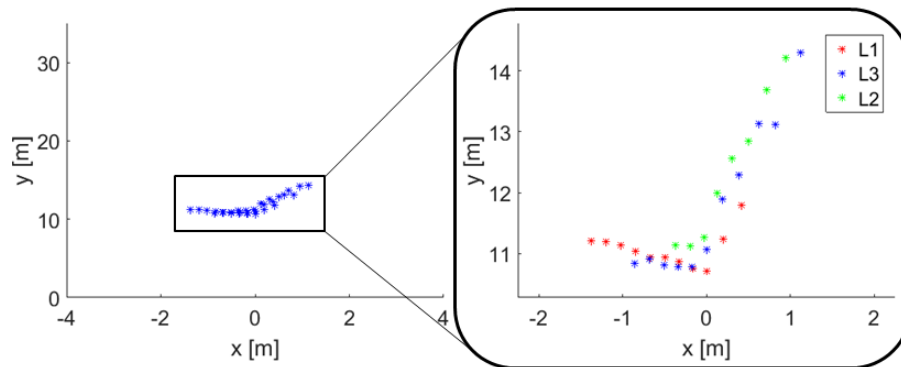
Jedno z možných řešení problému chybné detekce vozovky bylo při testování vyzkoušet variantu, že prostřední lidar je sklopen do stejné roviny jako dva krajní lidary. Prostřední lidar tak doplnil zbylé dva boční lidary, což zvýšilo hustotu bodů skenu. Dalším důvodem ke změně sklonu prostředního lidaru byl fakt, že s detekovanou zemí zatím systém dále nepracuje, a proto v momentě testování bylo rozhodnuto ještě vyzkoušet tuto variantu nastavení. Lidary jsou nastaveny tedy tak, že nedetekují vozovku před sebou, pouze krajní lidary mohou v určitých situacích detekovat vozovku, s čímž ale daný algoritmus nijak nepracuje. Algoritmus segmentace tak běží rychleji, jelikož nemusí řešit filtraci vozovky.



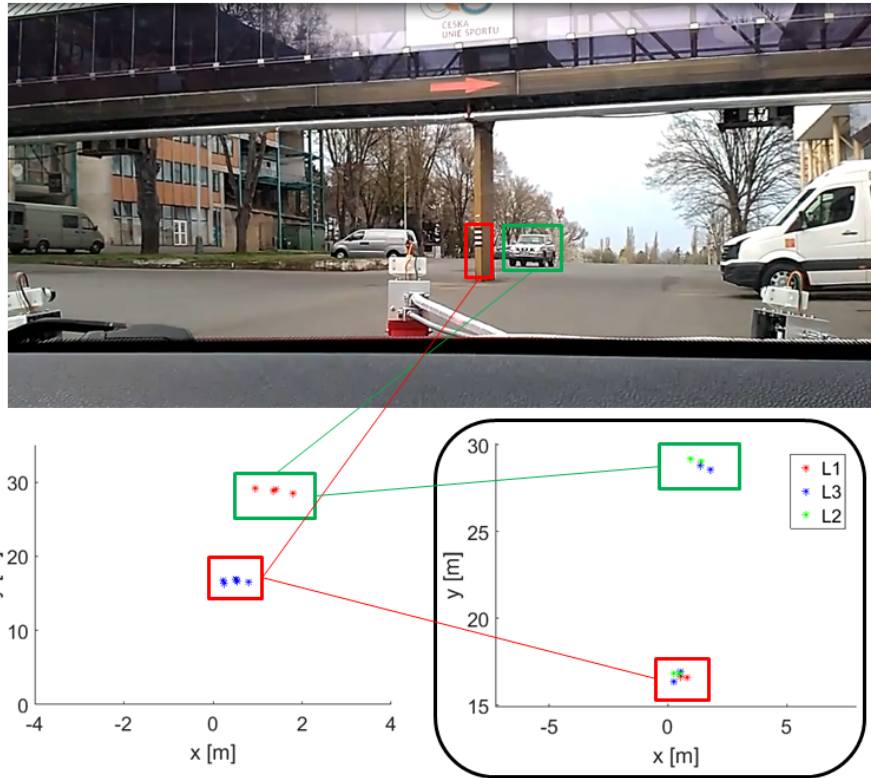
Obrázek 44: Test 2: Detekce dvou odlišných objektů, modrý shluk náleží vozidlu autoškoly a červený shluk šedé dodávce vpravo. V pravé dolním obrázku je detail bodů náležícím vozidlu autoškoly z jednotlivých lidarů. Zde lze vidět, že jednotlivé skenovací roviny se nepřekrývají a je mezi nimi menší odsazení, které je způsobeno nepřesným nastavením natočení jednotlivých lidarů vůči sobě vzniklém při kalibraci systému a také nepřesnosti jednotlivých lidarů.



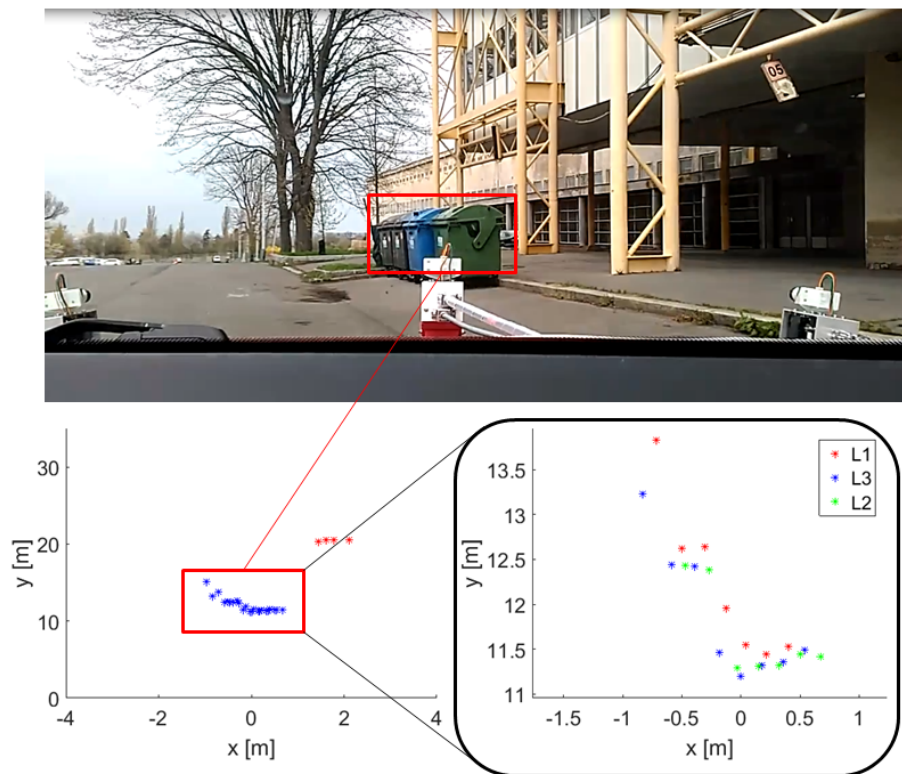
Obrázek 45: Test 2: Typický L tvar skenovaného vozidla



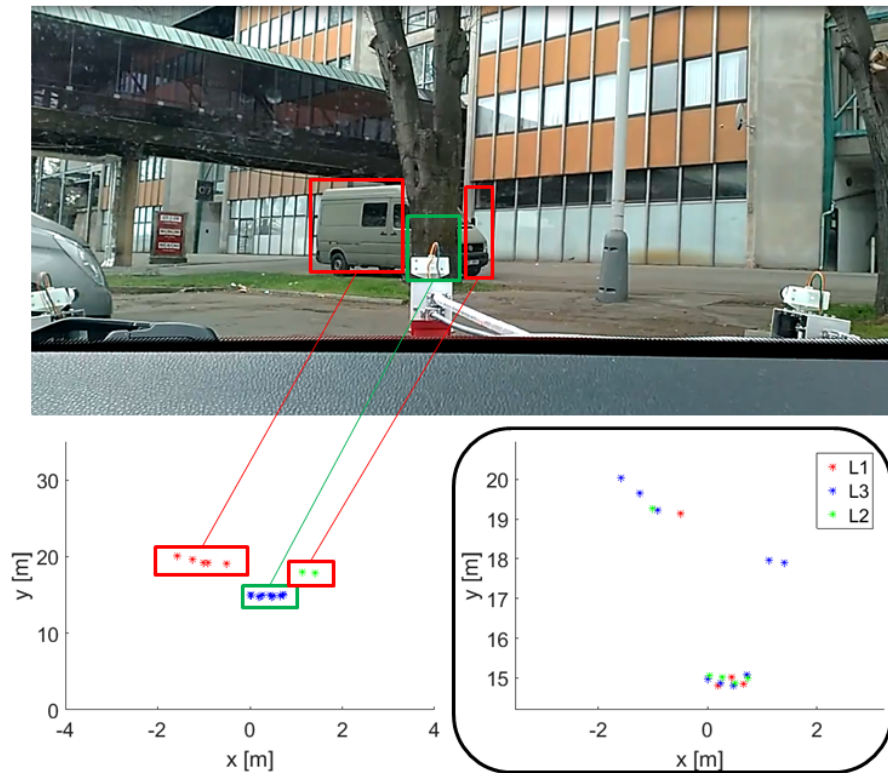
Obrázek 46: Test 2: Detekovaný L tvar bílého minivanu. Opět lze vidět malé odsazení jednotlivých skenovacích rovin.



Obrázek 47: Test 2: Detekce sloupu (červený obdélník) a protijedoucího vozidla (zelený obdélník).



Obrázek 48: Test 2: Detekce popelnice jako jeden objekt (modrý shluk) a sloupů v pozadí (červený shluk)



Obrázek 49: Test 2: Detekce stromu (modrý shluk) a dodávky za ním. Vlivem překrytí dodávky stromem vznikly dva shluky (červený a zelený), které však patří jednomu objektu. Jelikož je strom zrovna v poloze vůči lidarům tak, že zakrývá roh vozidla, nelze ani ze skenovaných bodů poznat, že jde o vozidlo, jelikož typický L tvar zde ani nemohl vzniknout.

8.3 Komentář výsledků

Výsledky ukazují, že i přes nedokonalost kalibrační metody pro nastavení polohy lidarů je systém schopen detekovat překážky do vzdálenosti okolo 35 m. Pochopitelně vlivem nízkého úhlového rozlišení (0.9°) jsou lépe detekovatelné velké překážky, např. vozidlo, stěna. Zatímco sloupce jsou na větší vzdálenosti hůře detekovatelné a jejich segmentace selhává. Avšak širší sloupce detekovány jsou, jak lze vidět na obr. 47. Lepší detekovatelnost menších objektů je v případě, že všechny lidary jsou sklopeny do stejných rovin, jelikož tím vzroste hustota bodů ve skenu. Tím je možno pak pozorovat na vizualizacích z jednotlivých skenů např. L tvar typický pro vozidlo detekované ze strany. Výsledky testování potvrdily počáteční ideu, že by více takových levných lidarů šlo použít pro skenování celého okolí vozidla na základě navrženého prototypového systému v této práci.

Na některých obrázcích lze vidět, že roviny lidarů nejsou ideálně se překrývající, lze tedy pozorovat určité odsazení jednotlivých rovin. Čím lépe se systém nakalibruje, především jde o nastavení polohy lidarů vůči sobě, tím menší bude i toto odsazení. Na druhou stranu i s touto nedokonalostí je algoritmus schopen si poradit a správně segmentuje jednotlivé objekty. Problém překrytí části objektu jiným

objektem, jak tomu je např. na obr. 49, v současné době algoritmus neumí řešit. Možným řešením by bylo analyzovat více po sobě jdoucích skenů a trackovat daný objekt, čímž by pak objekt před zakrytím byl plně detekován a po následném překrytí by trackování zaručilo, že by se dané dva shluky asociovaly s předešlým objektem a došlo by ke spojení těchto dvou shluků. K tomu by bylo třeba implementovat klasifikační a trackovací algoritmus (ať už Kálmánův filtr nebo Bayesův filtr).

Dále se při měření na Strahově ukázalo jako problém detekce vozovky krajními lidary při náklonu vozidla na stranu vlivem odstředivé síly při zatáčení, při prudším brzdění a nebo v případě nerovného terénu po stranách vozidla. Krajní lidary jsou naklopeny na stranu vlivem tvaru kapoty, a tak se ve výše zmíněných situacích může stát, že v krajních polohách skenů může lidar detekovat vozovku. Algoritmus není schopen poznat, že tyto měření náleží vozovce a identifikuje je tak jako překážku. Řešení tohoto problému je také námětem k další práci.

Při běžném měření nějaké veličiny (např. vzdálenosti) se vždy provede několik měření, aby bylo možné určit přesnost a správnost měření. Při měření laserovým skenerem je správnost měření dána tím, že několik po sobě jdoucích měření detekuje překážku v určité vzdálenosti a tím ověří, že několik po sobě jdoucích měření jsou správná. Na obr. 46 lze vidět tvar L představující naskenovanou část vozidla. Tento fakt také potvrzuje správnost měření, jinak by tvar L nebylo možné detekovat. Přesnost laserového skeneru v tomto případě je dána přesností Lidaru Lite.

9 Závěr

V diplomové práci se řešil problém detekce překážek před vozidlem pomocí fúze měření ze 3 levných Lidaru Lite. Byl navržen a vytvořen systém, který z těchto 1D lidarů vytváří 2D lidary, čímž je možno skenovat úzký výsek před vozidlem. Měření z jednotlivých lidarů jsou spojeny tak, aby jednotlivé skeny se doplnily, a tak bylo možné pomocí navrženého algoritmu segmentovat měřená data a detekovat jednotlivé objekty v reálném čase. Výsledky testování potvrzují hypotézu, že pomocí levných lidarů by šlo vytvořit systém, který by detekoval celé okolí vozidla.

Schopnost systému detekce překážek je nejvíce závislá na Lidaru Lite. Ten ovlivňuje především maximální vzdálenost, na kterou je systém schopen detekovat překážky. Dále ovlivňuje rychlost skenování, jelikož mu trvá v řádu jednotek milisekund jedno měření. Tato rychlost měření ovlivňuje rychlost skenu a tím i množství bodů, které za jeden sken je možné změřit, aby byl splněn požadavek na schopnost detekce v reálném čase. Během testování bylo zjištěno, že velký problém pro lidar tvoří skla vozidel a jejich zadní světla. Pokud lidar detekuje tuto část vozidla před ním, pak není schopen změřit vzdálenost. Avšak jiné části vozidla jsou dobře měřitelné. Proto je potřeba lidary nastavit tak, aby neměřily na zadní sklo vozidla. Počítá se však s tím, že některá měření budou neúspěšná při detekci zadních světel.

Přesnost měření závisí na lidarů, ale správnost určení polohy překážky a jejího tvaru je nejvíce závislá na kalibrační metodě. Jednotlivé lidary jsou natočeny vůči sobě tak, že se jejich skenovací roviny překrývají, proto musí být nastaveny do správné známé polohy, aby bylo možné korektně spojit měření z jednotlivých měření. Správnost provedení kalibrace je velmi klíčová pro správnou funkci detekce překážek. Navržená metoda je velmi citlivá na schopnost člověka správně určit polohu lidarů, jejich sklon a náklon. Proto by bylo vhodné v další práci se zaměřit na zlepšení kalibrace, aby byla méně závislá na schopnostech člověka, a vytvořit následně metodu na validaci správnosti měření lidarového systému. Na výsledcích testů lze vidět nedokonalost kalibrace, kdy jednotlivé roviny lidarů jsou vůči sobě mírně natočené, a tak hranice detekovaného objektu jednotlivými lidary se nepřekrývají, ale jsou vůči sobě mírně posunuty. I přes tuto chybu v poloze lidarů je segmentační algoritmus schopen překážku detekovat jako jeden objekt.

Segmentační algoritmus byl vytvořen tak, aby i menší množství dat, které lidary produkují během jednoho skenu, bylo dostatečné k detekci překážky a odlišení od sebe jednotlivých objektů. Byla navržena prahová funkce eliptického tvaru a simulací porovnána s jinými (kruhovými) prahovými funkcemi, které určují maximální vzdálenost sousedních bodů, aby bylo možné tyto body ještě považovat za body náležící stejnému objektu. Z výsledku simulace se navržená metoda jeví jako velmi efektivní. Je schopna od sebe oddělit objekty, které se vyskytují blízko u sebe. Pokud je překážka blíže k lidarům (do cca 27 m), pak není problém s nadbytečnou segmentací v podélném směru. Avšak nad tuto vzdálenost začíná mít

algoritmus někdy problém správně shlukovat data natočených objektů a vytváří více shluků na jednom objektu.

Navržený algoritmus byl otestován na reálných datech při testování na Strahově. Z výsledku z měření venku vyplynulo, že algoritmu také činí problém, když je jeden objekt částečně zakryt jiným objektem. To naruší sekvenci detekovaných bodů na zakrytém objektu a algoritmus jednomu objektu přiřadí dva či více shluků. Tato slabina algoritmu byla ověřena i simulací. K řešení tohoto problému by v budoucnu pomohlo rozšíření algoritmu o klasifikaci a trackování objektů. Tím by se znal přibližný tvar objektu před zakrytím. Jakmile by byl objekt částečně zakryt, trackovací část algoritmu by určila, že předtím detekovaný objekt o určitém tvaru by se měl nacházet za překrývajícím objektem, a tak pokud by na daném zakrytém objektu vzniklo více shluků, bylo by je možné spojit, jelikož se předpokládá, že patří téže objektu, než byl zakryt jiným objektem. Avšak aby bylo možné takové rozšíření algoritmu provést, musel by být použit jiný výpočetní hardware, jelikož tyto operace jsou výpočetně náročné. Zároveň by bylo nutné z vozidla s lidary vyčítat jeho rychlost a natočení volantu a z inerciální měřicí jednotky získávat potřebné informace pro správné trackování objektů. Pokud by tedy byla známá rychlost vozidla a i rychlost ostatních objektů díky trackování, pak by šlo i korigovat vliv pohybu vozidla během skenu a snížit tak zkreslení měření, jak bylo popsáno v pod části 2.4 o vnějších vlivech na lidar.

Při testování na Strahově byly zaznamenány případy, kdy algoritmus detekoval překážku na základě měření, která však náležela vozovce. Krajiní lidary jsou totiž naklopené na stranu vlivem tvaru kapoty a tak v určitých případech (zatáčení vozidla, brzdění, nerovnost terénu) jsou krajiními lidary detekovány body vozovky, avšak algoritmus je klasifikuje jako překážku. Úprava algoritmu pro detekci vozovky a její filtraci je také námětem pro další práci.

Přínosem této práce je za prvé vytvoření funkčního systému pro detekci překážek před vozidlem pomocí lidarů. Tento systém je možné dále rozšiřovat o více lidarů, což by poskytlo možnost skenovat celé okolí vozidla. V případě rozšíření systému o kamerový systém by umožňoval mnohem lepší detekci překážek a v případě implementace klasifikace a trackování objektů, by sloužil lidarový systém jako velmi schopný systém pro detekci překážek v intravilánu. Za druhé, navržený algoritmus se osvědčil jako schopný segmentační nástroj a může tak sloužit jiným projektům jako jeden z možných způsobů segmentace dat z lidarů.

Reference

- [1] M. A. Albota, B. F. Aull, D. G. Fouche, R. M. Heinrichs, D. G. Kocher, Ri. M. Marino, J. G. Mooney, N. R. Newbury, M. E. O'Brien, B. E. Player, B. C. Willard, J. J. Zayhowski, "*Three-Dimensional Imaging Laser Radars with Geiger-Mode Avalanche Photodiode Arrays*", Lincoln Laboratory Journal, vol. 13, no. 2, 2002; 351 – 370
- [2] A. Amditis, P. Lytrivis, E. Portouli, "*Sensing and Actuation in Intelligent Vehicles*", Handbook of Intelligent Vehicles, Springer London, 2012; 31 – 60
- [3] F. Amzajerdian, D. Pierrottet, L. Petway, G. Hines, V. Roback, "*Lidar Systems for Precision Navigation and Safe Landing on Planetary Bodies*", v Proc. SPIE 8192, 2011
- [4] Autonomoustuff, "*ibeo Standard Four Layer Multi-Echo LUX Sensor*", [Online]. [Zobrazeno 18. dubna 2017]. <https://autonomoustuff.com/product/ibeo-lux-standard/>
- [5] G.A. Borges, M.J. Aldon, "*Line Extraction in 2D Range Images for Mobile Robotics*", Journal of Intelligent and Robotic Systems, vol 40, 2004, 26 – 297
- [6] Continental Corp., "*Hi-Res 3D Flash LIDAR will supplement Continental's existing portfolio for automated driving*", [Online]. [Zobrazeno 20. dubna 2017]. http://www.continental-corporation.com/www/pressportal_com_en/themes/press_releases/3_automotive_group/chassis_safety/press_releases/pr_2016_03_03_3dlidar_en.html
- [7] A. Davies, "This Palm-Sized Laser Could Make Self-Driving Cars Way Cheaper", Wired, 2014. [Online]. [Zobrazeno: 20. dubna 2017]. <https://www.wired.com/2014/09/velodyne-lidar-self-driving-cars/>.
- [8] S. Debattisti, "Perception Tasks: Obstacle Detection", in *Handbook of Intelligent Vehicles*, Springer, London, 2012; pp. 1033–1041
- [9] Diamond Aircraft, "*Diamond Aircraft successfully integrated photo-grammetry and laser-scanning on one single aircraft*", [Online]. [Zobrazeno 18. dubna 2017]. http://www.diamond-air.at/en/media-center/press-releases/news/article/diamond_aircraft_successfully_integrated_photo_grammetry_and_laser_scanning_on_one_single_aircraft/
- [10] K. C. J. Dietmayer, J. Sparbert, and D. Streller, "*Model Based Object Classification and Object Tracking in Traffic Scenes from Range Images*" v Proc. 4th IV, Tokyo, 2001
- [11] J. C. Hulme, J. K. Doylend, M. J. R. Heck, J. D. Peters, M. L. Davenport, J. T. Bovington, L. A. Coldren, J. E. Bowers, "*Fully Integrated Hybrid Silicon Free-Space Beam Steering Source with 32 Channel Phased Array*", v Proc. Smart Photonic and Optoelectronic Integrated Circuits, Vol. 8989,

2014; 898907-01 – 898907-15

- [12] T. Fersch, A. Buhmann, A. Koelpin, R. Weigel, "*The Influence of Rain on Small Aperture LiDAR Sensors*", v Proc. GeMiC, 2016; 84 – 87
- [13] T. Fersch, R. Weigel, A. Koelpin, "*A CDMA Modulation Technique for Automotive Time-of-Flight LiDAR Systems*", IEEE Sensors Journal, 2017
- [14] A. Filgueira, H. González-Jorge, S. Lagüela, L. Díaz-Vilariño, P. Arias, "*Quantifying the Influence of Rain in LiDAR Performance*", Measurement, vol. 95, 2017; 143 – 149
- [15] M. Goppelt, H.-L. Blöcher, W. Menzel, "*Automotive Radar – Investigation of Mutual Interference Mechanisms*", Advances in Radio Science, vol. 8, 2010; 55 – 60
- [16] L. Gröll, A. Kapp, "*Effect of Fast Motion on Range Images Scquired by Lidar Scanners for Automotive Applications*", IEEE Transactions on Signal Processing, vol. 55, no. 6, 2007; 2945 – 2953
- [17] J. Guo, H. Zhang, X. Zhang, "*Propagating Characteristics of Pulsed Laser in Rain*", International Journal of Antennas and Propagation, 2015; 1 – 7
- [18] A. Harati, R. Siegartm "*A New Approach to Segmentation of 2D Range Scans into Linear Regions*", v Proc. IROS, 2007, 2083 – 2088
- [19] R. Heinrichs, B. F. Aull, R. M. Marino, D. I. G. Fouche, A. K. McIntosh, J. J. Zayhowski, T. Stephens, M. E. O'Brien, M. A. Albota, "*Three-Dimensional Laser Radar with APD Arrays*", v Proc. SPIE, vol. 4377, Laser Radar Technology and Applications VI, 2001; 106 – 117
- [20] M. Himmelsbach, F. V. Hundelshausen, H. J. Wuensche, "*Fast Segmentation of 3D Point Clouds for Ground Vehicles*", v Proc. IV 2010; 560 – 565
- [21] U. Hofmann, F. Senger, F. Soerensen, V. Stenchly, B. Jensen, J. Janes, "*Biaxial Resonant 7mm-MEMS Mirror for Automotive LIDAR Application*", v Proc. OMN, 2012; 150 – 151
- [22] R. Horaud, M. Hansard, G. Evangelidis, M. Clément, "*An Overview of Depth Cameras and Range Scanners Based on Time-of-Flight Technologies*", Machine Vision and Applications Journal, vol. 27, no. 7, 2016; 1005 – 1020
- [23] K. Ito, C. Niclass, I. Aoyagi, H. Matsubara, M. Soga, S. Kato, M. Maeda, M. Kagami, "*System Design and Performance Characterization of a MEMS-Based Laser Scanning Time-of-Flight Sensor Based on a 256x 64-pixel Single-Photon Imager*", IEEE Photonics Journal, vol. 5, no. 2, 2013
- [24] J. A. Keim, S. Mobasser, D. Kuang, Y. Cheng, T. Ivanov, A. E. Johnson, H. R. Goldberg, G. Khanoyan, D. B. Natzic, "*Field Test Implementation to Evaluate a Flash Lidar as a Primary Sensor for Safe Lunar Landing*", v Proc. IEEE Aerospace Conference, 2010; 1 – 14

- [25] G. Kim, J. Eom, Y. Park, "An Experiment of Mutual Interference between Automotive LIDAR Scanners", v Proc. ITNG, 2015; 680 – 685
- [26] K. Kimoto, N. Asada, T. Mori, Y. Hara, A. Ohya, S. Yuta, "Development of Small Size 3D LIDAR", v Proc. ICRA, 2014; 4620 – 4626
- [27] K. Klasing, D. Wollherr, M. Buss, "Realtime Segmentation of Range Data Using Continuous Nearest Neighbors", v Proc. ICRA, 2009; 2431 – 2436
- [28] K. Klasing, D. Wollherr, M. Buss, "A Clustering Method for Efficient Segmentation of 3D Laser Data", v Proc. ICRA, 2008; 4043 – 4048
- [29] S. Kurtti, J. Nissinen, J. Kostamovaara, "A Wide Dynamic Range CMOS Laser Radar Receiver With a Time-Domain Walk Error Compensation Scheme", IEEE Transactions on Circuits and Systems–I: Regular Papers, vol. 64, no. 3, 2017; 550 – 561
- [30] K. Kwak, J.S. Kim, J. Min, Y.W. Park, "Unknown Multiple Object Tracking Using 2D Lidar and Video Camera", Electronics Letters, vol. 50, no. 8, 2014; 600 – 602
- [31] P. Lindner, G. Wanielik, "3D Lidar Processing for Vehicle Safety and Environment Recognition", v Proc. CIVVS, 2009; 66 – 71
- [32] J. MacQueen, "Some methods for classification and analysis of multivariate observations", v Proc. Fifth Berkeley Symp. on Math. Statist. and Prob., Vol. 1, 1967; 281 – 297
- [33] R. M. Marino, W. R. Davis, Jr., "Jigsaw: A Foliage-Penetrating 3D Imaging Laser Radar System", Lincoln Laboratory Journal, vol. 15, no. 1, 2005; 23 – 36
- [34] A. M. Martín, J. Domínguez, J. Amador, "Applying LIDAR datasets and GIS based model to evaluate solar potential over roofs: a review", AIMS Energy, 2015, 3(3); 326 – 343
- [35] J. R. McBride, J. C. Ivan, D. S. Rhode, J. D. Rupp, M. Y. Rupp, "A Perspective on Emerging Automotive Safety Applications, Derived from Lessons Learned through Participation in the DARPA", Journal of Field Robotics, vol. 25 no. 10, 2008; 808 – 840
- [36] P. F. McManamon, T. A. Dorschner, D. L. Corkum, L. J. Friedman, D. S. Hobbs, M. Holz, S. Lieberman, H. Q. Nguyen, D. P. Resler, R. C. Sharp, E. A. Watson, "Optical Phased Array Technology", v Proc. IEEE 84.2, 1996; 268 – 298
- [37] M. Megens, B.W. Yoo, T. Chan, W. Yang, T. Sun, C. J. Chang-Hasnain, M. C. Wu, D. A. Horsley, "High-Speed 32×32 MEMS Optical Phased Array", OPTICS EXPRESS, vol. 21, no. 10; 12238 – 12248
- [38] A.W. Moore, "Efficient Memory-Based Learning for Robot Control", Disertační práce, Cambridge,

- [39] D. J. Natale, R. L. Tutwiler, M. S. Baran, J. R. Durkin, "*Using Full Motion 3D Flash LIDAR Video for Target Detection, Segmentation, and Tracking*", v Proc. SSI AI, 2010; 21 – 24
- [40] National Instruments, "*Meet the CompactRIO Controller, a Rugged Embedded System for Your Control or Monitoring Application*", [Online]. [Zobrazeno 20. dubna 2017].
<http://www.ni.com/white-paper/52251/en/>
- [41] National Instruments, "*cRIO-9031*", [Online]. [Zobrazeno 20. dubna 2017].
<http://www.ni.com/cs-cz/support/model.crio-9031.html>
- [42] V. Nguyen, A. Martinelli, N. Tomatis, R. Siegwart, "*A Comparison of Line Extraction Algorithms using 2D Laser Range Data for Indoor Mobile Robotics*", v Proc. IROS, 2005
- [43] P. Cho, H. Anderson, R. Hatch, P. Ramaswami, "*Real-Time 3D Ladar Imaging*", Lincoln Laboratory Journal, vol. 16, no. 1, 2006; 147 – 164
- [44] A. Petrovskaya, S. Thrun, "*Model Based Vehicle Detection and Tracking for Autonomous Urban Driving*", Autonomous Robots 26, 2009; 123 – 139
- [45] C. Premebida, U. Nunes, "*Segmentation and Geometric Primitives Extraction From 2D Laser Range Data for Mobile Robot Applications*", Robotica, 2015; 17 – 25
- [46] PulsedLight3D, "*Lidar-Lite v2 Documentation*", [Online]. [Zobrazeno 29. ledna 2017].
<https://github.com/PulsedLight3D/LIDAR-Lite-Documentation/tree/master/Docs/LIDAR-Lite-v2-Docs.pdf>
- [47] B. Qin, Z. J. Chong, S. H. Soh, T. Bandyopadhyay, M. H. Ang Jr., E. Frazzoli, D. Rus, "*A Spatial-Temporal Approach for Moving Object Recognition with 2D LIDAR*", Experimental Robotics, vol. 109, M. A. Hsieh, Springer, 2016; pp. 807 – 820
- [48] Quanergy Inc., "*S3*", [Online]. [Zobrazeno 20. dubna 2017]. <http://quanergy.com/s3/>
- [49] R.H. Rasshofer, M. Spies, H. Spies, "*Radio Science Influences of Weather Phenomena on Automotive Laser Radar Systems*", Advances in Radio Science, vol. 9, 2011; 49 – 60
- [50] J. Rieken, M. Maurer, "*Sensor Scan Timing Compensation in Environment Models for Automated Road Vehicles*", v Proc. ITSC, 2016; 635 – 642
- [51] D. O. Rubio, A. Lenskiy, J.H. Ryu, "*Connected Components for a Fast and Robust 2D Lidar Data Segmentation*", v Proc. 7th AMS, 2013; 160 – 165
- [52] S. Santos, J.E. Faria, F. Soares, R. Araújo, U. Nunes, "*Tracking of Multi-obstacles with Laser Range*

- Data for Autonomous Vehicles*", Robotice, 2003; 59 –65
- [53] S. Schneider, M. Himmelsbach, T. Luettel, H.J. Wuensche, "*Fusing Vision and LIDAR - Synchronization, Correction and Occlusion Reasoning*", v Proc. IV, 2010; 388 – 393
- [54] R.Sudharsanan, "*Low Cost Scanning LiDAR Imager*", LiDAR Magazine, Vol. 3 No. 2, 2013; 1 – 4
- [55] Sparkfun, "*Lidar Lite V3*", [Online]. [Zobrazeno 20. dubna 2017].
<https://www.sparkfun.com/products/14032>
- [56] D. Steinhauser, O. Ruepp, D. Burschka, "*Motion Segmentation and Scene Classification from 3D LIDAR Data*", v Proc. IV, 2008; 398 – 403
- [57] R. Stettner, H. Bailey, S. Silverman, "*Three Dimensional Flash Ladar Focal Planes and Time Dependent Imaging*", Advanced Scientific Concepts, [Online].
<http://www.advancedscientificconcepts.com/technology/white-papers.html>
- [58] J. Stockley, S. Serati, "*Advances in Liquid Crystal Beam Steering*", v Proc. The International Society for Optical Engineering, vol. 5550
- [59] B. Templeton, "All You Need to Know About Self-Driving Cars from CES", Robotics Trends, 2016. [Online]. [Zobrazeno: 20. dubna 2017].
http://www.robotictrends.com/article/all_you_need_to_know_about_self_driving_cars_from_ces/P2.
- [60] Texas Instruments, "*LIDAR System Design for Automotive/Industrial/Military Applications*", [Online]. [Zobrazeno 29. ledna 2017]. <http://www.ti.com/lit/an/snaa123/snaa123.pdf>
- [61] S. Thrun, M. Montemerlo, et. al, "*Stanley: The Robot that Won the DARPA Grand Challenge*", Journal of Field Robotics vol. 23 no. 9, 2006; 661 – 692
- [62] Velodyne Lidar, "*Velodyne Lidar HDL 64E*", [Online]. [Zobrazeno 18. dubna 2017].
<http://velodynelidar.com/hdl-64e.html>
- [63] Velodyne Lidar, "*Velodyne Lidar HDL 64E – datasheet*", [Online]. [Zobrazeno 18. dubna 2017].
http://velodynelidar.com/docs/datasheet/63-9194_Rev-F_HDL-64E_S3_Data%20Sheet_Web.pdf
- [64] Velodyne Lidar, "*Velodyne Lidar Puck*", [Online]. [Zobrazeno 18. dubna 2017].
<http://velodynelidar.com/vlp-16.html>
- [65] Velodyne Lidar, "*Velodyne Lidar Puck – datasheet*", [Online]. [Zobrazeno: 18. dubna 2017].
http://velodynelidar.com/docs/datasheet/63-9286_Rev-F_Puck%20LITE_Spec%20Sheet_Web.pdf

- [66] X. Wang, B. Wang, P. J. Bos, J. E. Anderson, P. F. McManamon, "*2-D Liquid Crystal Optical Phased Array*", v Proc. IEEE Aerospace Conference, 2004; 905 – 913
- [67] Y. Wang, M.C. Wu, "*Micromirror Based Optical Phased Array for Wide-Angle Beamsteering*", v Proc. MEMS, 2017; 897 – 900
- [68] T. Watanabe, N. Kishimoto, K. Hayafune, K. Yamadad, N. Maede, "*Development of an Intelligent Cruise Control System*", v Proc. Steps Forward. Intelligent Transport Systems World Congress, 1995; 1229 – 1235
- [69] T. Weiss, B. Schiele, K. Dietmayer, "*Robust Driving Path Detection in Urban and Highway Scenarios Using a Laser Scanner and Online Occupancy Grids*", v Proc. IV, 2007; 184 – 189
- [70] G. R. Widmann, M. K. Daniels, L. Hamilton, L. Humm, B. Riley, J. K. Schiffmann, D. E. Schnellker, W. H. Wishon, "*Comparison of Lidar-Based and Radar-Based Adaptive Cruise Control Systems*", SAE 2000 World Congress, 2000
- [71] A. Yaacobi, J. Sun, M. Moresco, G. Leake, D. Coolbaugh, M. R. Watts, "*Integrated Phased Array for Wide-Angle Beam Steering*", OPTICS LETTERS, Vol. 39, No. 15, 2014; 4575 – 4578
- [72] L. Ye, G. Zhang, Z. You, C. Zhang, "*A 2D Resonant MEMS Scanner with an Ultra-Compact Wedge-Like Multiplied Angle Amplification for Miniature LIDAR Application*", v Proc. IEEE Sensors, 2016; 1 – 3
- [73] Youtube, "*Valeo SCALA Laser Scanner for Emergency Braking*", [Online]. [Zobrazeno 18. dubna 2017]. <https://youtu.be/4B2D8NctjS8>
- [74] J. Zeisler, H.G. Maas, "*Analysis of the Performance of a Laser Scanner for Predictive Automotive Applications*", v Proc. ISPRS, Volume II-3/W5, La Grande Motte, 2015; 49 – 56
- [75] S. Zhang, L. Xie, M. Adams, F. Tang, "*Geometrical Feature Extraction Using 2D Range Scanner*", v Proc. ICCA, 2003; 901 – 905
- [76] G. Zhou, X. Zhou, J. Yang, Y. Tao, X. Nong, O. Baysal, "*Flash Lidar Sensor Using Fiber-Coupled APDs*", IEEE Sensors Journal, vol. 15, no. 9, 2015; 4758 – 4768
- [77] G. Zhou, J. Yang, X. Li, X. Yang, "*Advances of Flash Lidar Development Onboard UAV*", v Proc. XXII ISPRS Congress, Melbourne, 2012; 193 – 198

Příloha A

Uživatelský manuál

Instalace a síťové nastavení

Aplikaci je možno nainstalovat na jakýkoli počítač s ethernetovým portem a s operačním systémem Windows XP SP 3 a novější. Po instalaci je vytvořen adresář s EXE souborem pro spuštění aplikace. Před samotným spuštěním je ještě potřeba nastavit IP adresu lokální sítě, aby se počítač spároval s cRIO. Cesta k nastavení je: **Ovládací panely » Sít' a Internet » Sít'ová připojení**. Zde pravým tlačítkem kliknout na ikonu **Připojení k místní síti** a vybrat **Vlastnosti**. V seznamu vybrat **Protokol IPv4** a zmáčknout **Vlastnosti**. V otevřeném okně vybrat možnost **Použít následující IP adresu**. Do pole pro IP adresu zadat **192.168.222.81** a masku podsítě zvolit **255.255.255.0**.

Spuštění aplikace

Po spuštění aplikace se otevře okno se záložkami. Výchozí záložka je *Calibration Parameters*. Zde je potřeba nejprve nastavit FPGA parametry, pokud výchozí hodnota nevyhovuje. Těmi jsou rychlost motoru, počet vzorku měření. I když uživatel nechce žádný parametr před spuštěním systému měnit, musí potvrdit nastavení parametru zmáčknutím tlačítka *CALIB OK*. Tato operace proběhla úspěšně, pokud se rozsvítila kontrolka nad tlačítkem. Nyní uživatel může přejít na záložku *Control*, odkud se systém ovládá. Po stisknutí tlačítka *CALIB OK* je potřeba vyčkat cca 2 s, během kterých systém prochází automatickou konfigurací. To uživatel vidí na panelu, kde je uveden stav, ve kterém se nachází RT stavový automat. Při této konfiguraci může dojít k chybě. Nejčastější chyba je chyba s číslem 65572. Pokud tato chyba nastane, je potřeba aplikaci vypnout tlačítkem *Terminate* a znovu ji zapnout. Vše by pak mělo být již v pořádku. Tato chyba se hlavně vyskytuje pokud je cRIO napájeno z jiného zdroje než je napájen modul NI 9870.

Aby fúze dat lidarů byla správná, musí se lidary připojit do cRIO správně, tj. L1 do portu RS 232 číslo jedna, L2 do portu 2 a L3 do portu 3. Je to proto, že lidary neví, kde na vozidle jsou, a tak se tato informace do systému musí zanést právě správným zapojením jednotek do cRIO.

Nastavení motoru

Před spuštěním měření je potřeba udělat několik kroků. Nejprve je potřeba nastavit motory do výchozí polohy. Ručně se musí motory otočit protisměru hodinových ručiček do úvratí. Pak se musí zmáčknout tlačítko *Motor Config*. Tím se motory pootočí o 40 kroků po směru hodinových ručiček. Nyní jsou

motory zapnuty a v poloze, z níž lze zahájit kalibraci systému.

Měření

Jakmile jsou motory ve výchozí poloze, lze začít s měřením, kterému však předchází kalibrace. Měření i kalibrace se začne stisknutím tlačítka *Measure*. Nyní by motory měly být v pohybu a měla by přicházet data z lidarů. Pokud měření, zpracování a přenos dat funguje, objeví se data na hlavní obrazovce a rozsvítí se jednotlivé kontrolní indikátory. Nyní lze začít s kalibrací popsanou v sekci Kalibrace.

Jakmile je kalibrace ukončena, je důležité nevypínat měření opětovným stiskem tlačítka *Measure*. Pokud se tak stane, tak nelze znovu začít s měřením. Za prvé jsou motory ve špatné poloze, za druhé mohlo v DMA FIFO zůstat několik bodů, takže by pak bylo čtení z FIFO a měření vůči sobě asynchronní. Takže pokud z nějakého důvodu uživatel vypne měření, je potřeba aplikaci vypnout tlačítkem *Terminate*. Vypnout napájení a znovu jej zapnout, pak zapnout aplikaci, potvrdit kalibrační parametry, počkat až se vykoná automatická konfigurace, znovu nastavit motory do úvrati a zmáčknout tlačítko *Motor Config*. Kalibrace již by neměla být nutná. Pokud je tedy systém nakalibrován a měření nebylo vypnuto systém je ve stavu měření. Uživatel během měření může sledovat zobrazení zpracovaných dat na jednotlivých záložkách. Může i zapnout ukládání zpracovaných dat. Může si vybrat, co vše chce ukládat. Data, jež lze ukládat, jsou: Původní data přímo z lidarů (Raw data), transformované souřadnice (Transformed coords) a Bezpečná oblast před vozidlem (Drivable Area).

Měření uživatel vypne stisknutím tlačítka *Measure*. Pokud aplikaci ještě nechce vypnout, může vypnout ale napájení motoru stisknutím tlačítka *Motor OFF*. Pokud chce vypnout i napájení převodníků, provede to stisknutím tlačítka *ALL OFF*. Tímto se vypnou jak převodníky, tak i motory (pokud ještě nejsou vypnuty). Pokud si uživatel přeje celou aplikaci vypnout, provede to stisknutím tlačítka *Terminate*. Tímto se kromě aplikace vypne i napájení převodníků a motorů.