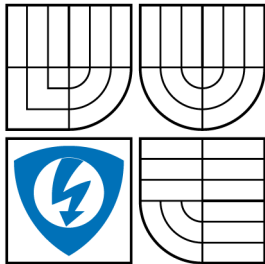


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY  
A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND  
COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

## TVORBA MULTISPEKTRÁLNÍCH MAP V MOBILNÍ ROBOTICE

MULTISPECTRAL MAP BUILDING IN MOBILE ROBOTICS

DIZERTAČNÍ PRÁCE  
DOCTORAL THESIS

AUTOR PRÁCE  
AUTHOR

Ing. FRANTIŠEK BURIAN

VEDOUCÍ PRÁCE  
SUPERVISOR

doc. Ing. LUDĚK ŽALUD, Ph.D.

BRNO 2014

## **ABSTRAKT**

Dizertační práce pojednává o využití multispektrálního optického měření pro datovou fúzi, kterou je možné využít pro vizuální teleprezenci a mapování uvnitř budov i ve vnějším prostředí v systému složeném z heterogenních mobilních robotů. Pro fúzi jsou použity dálkoměrné optické snímače, termovizní kamery a barevné kamery. Popsané algoritmy jsou optimalizovány pro práci v reálném čase a implementovány na robotický systém CASSANDRA vytvořený naší robotickou skupinou.

## **KLÍČOVÁ SLOVA**

mobilní robot, termovize, multispektrální datová fúze, 3D mapování, vykreslování, octree

## **ABSTRACT**

The dissertation deals with utilisation of multispectral optical measurement for data fusion that may be used for visual telepresence and indoor/outdoor mapping by heterogeneous mobile robotic system. Optical proximity sensors, thermal imagers, and tricolour cameras are used for the fusion. The described algorithms are optimised to work in real-time and implemented on CASSANDRA robotic system made by our robotic research group.

## **KEYWORDS**

mobile robot, Thermovision, multispectral data fusion, 3D mapping, rendering, octree

BURIAN, František *Tvorba multispektrálních map v mobilní robotice*: dizertační práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2014. 88 s. Vedoucí práce byl doc. Ing. Luděk Žalud, Ph.D.

## PROHLÁŠENÍ

Prohlašuji, že svou doktorskou práci na téma „Tvorba multispektrálních map v mobilní robotice“ jsem vypracoval samostatně pod vedením vedoucího doktorské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené doktorské práce dále prohlašuji, že v souvislosti s vytvořením této doktorské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Brno .....

.....

(podpis autora)

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu disertační práce panu doc. Ing. Luďkovi Žaludovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno .....

.....

(podpis autora)

# OBSAH

Úvod	11
<b>1 Bezdotykové optické měření vzdáleností v robotice</b>	<b>14</b>
1.1 Rozdělení proximálních snímačů v robotice	14
1.2 Planární skener	15
1.2.1 Základní matematický model	16
1.2.2 Kalibrovaný matematický model	16
1.3 Vícepaprskový rotační skener	17
1.3.1 Základní matematický model	18
1.3.2 Jednobodový kalibrační model	19
1.3.3 Dvoubodový kalibrační model	20
1.3.4 Duální dvoubodový kalibrační model	21
1.4 Naklápěný planární skener	21
1.4.1 Kalibrovaný matematický model	22
1.5 3D kamera	22
1.6 Aspekty použití paprskových senzorů	24
1.6.1 Princip pulzního měření vzdálenosti	24
1.6.2 Efekt rozbíhavosti paprsku	26
1.6.3 Efekt šikmého odrazu	27
1.6.4 Efekt odrazu na hraně, sklo	28
<b>2 Tvorba map v mobilní robotice</b>	<b>31</b>
2.1 Mračna naměřených bodů	31
2.2 Mřížka obsazenosti (Occupancy Grid)	31
2.2.1 Bayes	32
2.2.2 Odd ratio	33
2.2.3 Log odds ratio	34
2.2.4 Dempster-schafer	34
2.3 Mřížka evidence (Evidence Grid)	35
2.4 Octree	36
2.5 Výběr vhodné metody uložení mapy	38
<b>3 Robotický systém CASSANDRA</b>	<b>39</b>
3.1 Roboty	40
3.1.1 Orpheus-AC2	40
3.1.2 Orpheus-X3	41
3.1.3 Scorpio	44

3.1.4	Brontes . . . . .	45
3.1.5	Perseus . . . . .	45
3.1.6	Uranus-HEXA . . . . .	45
3.1.7	EnvMAP . . . . .	47
3.1.8	Ostatní roboty . . . . .	47
3.2	Ovládací stanice . . . . .	47
<b>4</b>	<b>Tvorba multispektrálních map</b>	<b>49</b>
4.1	Reprezentace mřížky evidence v paměti . . . . .	49
4.2	Reprezentace souboru s Octree mapou . . . . .	50
<b>5</b>	<b>Distribuce mapy mezi roboty</b>	<b>53</b>
5.1	Začlenění map do robotického systému . . . . .	53
5.2	Systém blokového dělení globální mapy . . . . .	54
5.3	Interpretace dat uvnitř bloku . . . . .	55
5.3.1	Výšková mapa (Leveled Map) . . . . .	55
5.3.2	Mřížka obsazenosti (Occupancy Grid) . . . . .	55
5.3.3	Mřížka evidence (Evidence Grid) . . . . .	55
5.3.4	Mapa význačných bodů (Feature Map) . . . . .	56
5.4	Systém distribuce mapy mezi více robotů . . . . .	56
5.4.1	Mapový server . . . . .	57
<b>6</b>	<b>Zobrazení a prezentace dat</b>	<b>60</b>
6.1	Fúzování video dat . . . . .	60
6.1.1	Kalibrace paprsků TOF kamery (CAL) . . . . .	61
6.1.2	Mapování paprsků TOF na CCD kameru resp. termokameru (MAP) . . . . .	61
6.1.3	Výsledná fúze dat termokamery a CCD kamery (FUSION) . . . . .	62
6.2	Vykreslování mapy . . . . .	63
6.2.1	Compute shader (CS) . . . . .	64
6.2.2	Vertex Shader (VS) . . . . .	65
6.2.3	Geometry Shader (GS) . . . . .	66
6.2.4	Pixel Shader (PS) . . . . .	66
<b>7</b>	<b>Dosažené výsledky</b>	<b>67</b>
7.1	Experiment . . . . .	68
7.2	Tvorba Octree mapových podkladů . . . . .	69
7.2.1	Vykreslování mapy . . . . .	71
7.2.2	Výsledky fúze dat . . . . .	74

<b>8</b>	<b>Zhodnocení a závěr</b>	<b>75</b>
	<b>Literatura</b>	<b>77</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>80</b>
	<b>Seznam příloh</b>	<b>81</b>
<b>A</b>	<b>Matematické pomůcky</b>	<b>82</b>
	A.1 Odvození inverze symetrické matice 3x3 . . . . .	82
	A.2 Metoda nejmenších čtverců . . . . .	83
	A.2.1 Příklad . . . . .	84
	A.3 Rekurzivní metoda nejmenších čtverců . . . . .	84
<b>B</b>	<b>Statistické pomůcky</b>	<b>85</b>
	B.1 N-rozměrné normální rozdělení . . . . .	85
	B.1.1 Dvourozměrné normální rozdělení [25] [21] . . . . .	85
	B.1.2 Trojrozměrné normální rozložení . . . . .	86
	B.2 Zjištění parametrů elipsy . . . . .	86

# SEZNAM OBRÁZKŮ

1.1	Příklady planárních skenerů: (zleva) URG-04-LX, LMS-200, PLS-200	15
1.2	Planární skener	16
1.3	Princip kalibrace planárního skeneru	17
1.4	Příklady vícepaprskových skenerů (zleva Velodyne HDL-64 a HDL-32)	17
1.5	Vícepaprskový skener	18
1.6	Význam kalibračních parametrů	20
1.7	Naklápěný planární skener	21
1.8	Příklady 3D kamer – zleva IFM O3D200, Mesa SR3000 a Mesa SR4000	23
1.9	3D kamera	23
1.10	Princip pulzního měření	25
1.11	Rozbíhavost paprsku	26
1.12	Odražený impuls	27
1.13	Šikmý odraz paprsku	27
1.14	Šikmý odraz	28
1.15	Odraz na hraně	29
1.16	Odražený impuls od dvou překážek blíže než 1.5m	29
1.17	Dva odrazy	30
2.1	3D mřížka obsazenosti	32
2.2	Princip Octree	36
2.3	Octree	37
3.1	Roboty v robotickém systému Cassandra	39
3.2	Robotický systém CASSANDRA	40
3.3	Robot Orpheus-AC2	41
3.4	Robotický systém Orpheus-AC2	41
3.5	Robot Orpheus-AC	42
3.6	Robot Orpheus-X3	42
3.7	Hlavice	43
3.8	Zapojení robotu pro experimenty	43
3.9	Zapojení robotu pro praktické využití	44
3.10	Robot Scorpio	44
3.11	Robot Brontes	45
3.12	Robot Perseus	46
3.13	Robot Uranus - HEXA	46
3.14	Robot Uranus - ALU	46
3.15	Robot EnvMap	47
3.16	Ostatní roboty	48
3.17	Ovládání systému CASSANDRA	48

4.1	Hlavice Orpheus-X3 . . . . .	49
4.2	Formát serializovaného Octree, vlevo binární datový stream uložený v souboru, vpravo nahoře síťová reprezentace dat, vpravo dole výsledná rekonstruovaná mapa . . . . .	52
5.1	Spojení mezi aplikacemi . . . . .	53
5.2	Dělení globální mapy na oblasti, se zvýrazněním zpracovávaného okolí robotu R, se senzorem s maximálním dosahem měření d zelenou barvou. . . . .	55
5.3	Popis distribuce mapy mezi roboty . . . . .	56
5.4	Základní operace pro distribuci mapy robotu . . . . .	58
5.5	Pokročilé operace pro distribuci mapy robotu . . . . .	58
6.1	Hlavice robotu . . . . .	60
6.2	Fúzování video dat . . . . .	61
6.3	Ideové schéma fúze dat . . . . .	63
6.4	DirectX 11.0 pipeline . . . . .	64
6.5	Fraktálové uložení mapy . . . . .	65
6.6	Činnost Compute Shaderu . . . . .	65
6.7	Činnost Vertex Shaderu . . . . .	65
6.8	Činnost Geometry Shaderu . . . . .	66
7.1	Měření mapy . . . . .	68
7.2	Pohled na měřenou scénu shora s červeně vyznačenou trajektorií robotu . . . . .	69
7.3	Mapa - pohled do jedné z měřených laboratoří . . . . .	69
7.4	Závislost zaplněnosti mapy v závislosti na rozlišení mapy pro všechny buňky měření . . . . .	70
7.5	Časová náročnost konverze mapy mezi Octree a Evidence Grids v závislosti na rozlišení. . . . .	71
7.6	Pohled na celou vytvořenou mapu shora . . . . .	72
7.7	Výsledky optimalizace vykreslování . . . . .	73
7.8	Různé způsoby osvětlení . . . . .	73
7.9	Mapování thermo . . . . .	73
7.10	Fúze dat (snímky z videa na přiloženém datovém nosiči) . . . . .	74
B.1	Parametry elipsy . . . . .	86

# SEZNAM TABULEK

1.1	Parametry dostupných senzorů . . . . .	15
1.2	Parametry dostupných senzorů . . . . .	18
1.3	Parametry dostupných senzorů . . . . .	22
4.1	Struktura jednoho voxelu mřížky evidence . . . . .	50
4.2	Hlavička . . . . .	51
4.3	List . . . . .	51
6.1	Parametry kamer . . . . .	60
7.1	Parametry počítačů použitých pro dynamické měření . . . . .	67
7.2	Statické výsledky . . . . .	70
7.3	Vliv optimalizace vykreslování celé mapy . . . . .	72

# ÚVOD

Tato dizertační práce pojednává o využití multispektrálního optického měření v mobilní robotice. Hlavním cílem bylo vytvořit systém pro multispektrální datovou fúzi, kterou by bylo možné využít pro mapování a teleprezenční režim v multirobotickém systému.

Tato fúze je realizována nad daty tří typů snímačů – proximitního prostorového skeneru, termovizní kamery a barevné kamery ve viditelném spektru. Zde popsané metody jsou obecně použitelné pro jakýkoli typ prostorového (tzv. 3D) optického snímače, v našem případě byla vzhledem k rozměrům a snímkovací frekvenci použita tzv. dálkoměrná kamera.

Pomocí těchto snímačů je možné zaznamenat realitu okolí robotu, která zjednodušeně řečeno obsahuje prostotovou informaci, barevnou informaci a teplotu objektů. Tato informace se pak může použít jednak pro automatickou tvorbu digitálních map okolí robotu, a také pro přímé zobrazení operátorovi ve formě tzv. teleprezence. V obou případech je hlavní snahou a motivací podstatné rozšíření informační hodnoty pro operátora – ať už je hlavním cílem vyhledávání osob v neznámém terénu nebo například vyhodnocení kvality provedené stavby včetně tepelné izolace.

Důležitou součástí mé práce je dále optimalizace uložení a distribuce získaných map v robotickém systému. Zde je důležité si uvědomit, že systém sestává z několika vzdálených robotů, které spolu komunikují bezdrátově a mají tedy velmi omezený vzájemný datový tok. Proto jsem se obsírně zabýval nejen bezeztrátovou kompresí mapových dat, ale také poměrně sofistikovaným systémem umožňujícím efektivní výměnu dat mezi roboty a operátorskou stanicí s možností vzájemného doplňování nových informací.

Konečně jsem se zabýval i zobrazením dat operátorovi. Zde se opět jako hlavní problém ukázalo především obrovské množství dat, která je třeba při zobrazování zpracovávat. Bylo tedy nutné přistoupit k masivní paralelizaci výpočtů, především pomocí nástrojů dostupných na moderních grafických kartách.

Algoritmy popsané v praktické části této práce byly prakticky realizovány na robotickém systému CASSANDRA, který je naší skupinou vyvíjen od roku 2010.

CASSANDRA je komplexní robotický systém, který je vyvíjen několika pracovníky Skupiny umělé inteligence a robotiky na Ústavu automatizace a měřicí techniky a Robotické laboratoře skupiny RG2-2 STI CEITEC. Nyní budou popsány činnosti jednotlivých pracovníků v rámci projektu:

- **Luděk Žalud** – vedení projektu, ideový a systémový návrh, teleprezence, mechanický návrh, uživatelské prostředí, sensorický subsystém,
- **František Burian** – mapování ve vnitřním a vnějším prostředí, multispektrální datová fúze, návrh řídicích jednotek, komunikace, programová optimali-

zace,

- Lukáš Kopečný – silová elektronika, elektrické pohony, netradiční pohony, mechanický návrh,
- Petra Kocmanová – kalibrace optických snímačů,
- Tomáš Jílek – sebelokalizace a navigace pomocí fúze dat GNSS, INS a odometrie,
- Tomáš Florián – malé roboty (Brontes, Perseus), navigace pomocí optických tagů,
- Aleš Jelínek – sebelokalizace pomocí proximitních skenerů, 2D SLAM,
- Petr Gábrlík – multikoptéry, 3D skenování pomocí vzdušných prostředků,
- Vlastimil Kříž – multikoptéry, stabilizace pomocí inerciálních snímačů.

Z výše uvedeného tedy vyplývá, že moje práce je pouze součástí širšího celku. Zde uvedené algoritmy a postupy jsou však mým dílem, pokud není uvedeno jinak.

Tato práce je rozdělena do následujících kapitol:

1. Bezdotykové měření vzdáleností v robotice – Kapitola pojednává obecně o měření vzdáleností optickými metodami v mobilní robotice s maximálním důrazem na poarmetry, které jsou důležité pro mou práci.
2. Tvorba map v mobilní robotice – Tato část se zabývá problematikou vhodné reprezentace dat optických proximitních snímačů a tvorby map z těchto dat. Hlavní důraz je kladen na podrobný popis metod, které jsou implementovány v praktické části.
3. Robotický systém CASSANDRA – Kapitola popisuje zevrubně robotický systém, který naše skupina vyvíjí a pro který jsou jmenované metody vyvíjeny. Krátce jsou popsány všechny důležitější roboty i řídicí stanice. Hlavní důraz je kladen na popis robotu Oprheus-X3, který je použit v měřeních a experimentech této dizertační práce.
4. Tvorba multispektrálních map – V kapitole je popsán způsob uložení dat v multispektrální mapě který jsem zvolil a implementoval tak, aby bylo možné s nimi efektivně pracovat i při jejich obrovském množství.
5. Distribuce mapy mezi roboty – Zde popisují zvolený způsob dělení globální mapy do bloků a komplexní systém, který umožňuje efektivní distribuci obsáhlých map mezi jednotlivými roboty s přihlédnutím k faktu, že každý z robotů má jiné snímače, a vlastní výpočetní prostředky. Velmi důležitým faktem je také důraz kladený na minimalizaci datových toků mezi roboty, jelikož se předpokládá prakticky výhradně bezdrátová komunikace.
6. Zobrazení a prezentace dat – V této kapitole popisují způsob zobrazení fúzování dat, a dále se věnují optimalizaci a masivní paralelizaci zobrazovacích algoritmů tak, aby bylo možné tato vysoce objemná data zobrazit i pomocí současných běžných výpočetních prostředků.

7. Dosažené výsledky – Zde shrnuji dosažené výsledky a provádím jejich analýzu.
8. Zhodnocení a závěr.

# 1 BEZDOTYKOVÉ OPTICKÉ MĚŘENÍ VZDÁLENOSTÍ V ROBOTICE

Pro správnou funkčnost mapovacích a sebelokalizačních metod je nutné nejprve získat data z okolí robotu. Ke snímání okolí robotu jsou v této práci uvažovány optické proximitní (bezdotykové) skenery. Tyto snímače dávají robotu informaci o vzdálenostech mezi senzorem a okolními objekty.

## 1.1 Rozdělení proximitních snímačů v robotice

V současné době se na trhu nachází velké množství snímačů založených na různých principech měření. Mezi základní principy snímačů, se kterými bude dále v práci pracováno patří následující:

**Time-of-flight** snímače patří bezesporu k nejdražším. Princip spočívá v měření doby, která uplyne mezi odesláním paprsku k měřenému objektu a příjmem odraženého paprsku od měřeného objektu. Senzor založený na tomto principu měří v diskrétních krocích, což se může jevit výhodné, zejména při rozítání paprsku. Senzor založený na tomto principu má principiálně minimální vzdálenost, od které měří, což ztěžuje jeho univerzální použití.

**Triangulační** snímače jsou levnější variantou snímačů oproti předchozí skupině. Princip snímání je založen na měření úhlu dopadajícího paprsku na fotocitlivý senzor. Přesnost těchto snímačů je ovlivnitelná nežádoucím způsobem cizím osvětlením, proto se nehodí pro použití ve vnějším terénu.

**Fázové** snímače jsou založeny na principu měření fáze vysílané a přijímané vlny světelného paprsku. Jsou tedy analogií time-of-flight snímačů, avšak tyto pracují kontinuálně, bez diskrétních pulsů. Pro jejich správnou funkci je potřeba, aby po dobu vyhodnocování fáze nebyl snímač v pohybu, protože pak mohou generovat data neodpovídající realitě. Jejich obrovskou výhodou je, že je lze vytvořit na čipu podobném CCD struktuře, a za pomoci optiky lze snímat "3D obraz" podobně jako snímají obraz obyčejné videokamery.

Veškeré metody popsané v této disertační práci jsou nezávislé na použitém principu snímání, a jsou navrženy tak, aby bylo možné použít i jejich vzájemnou kombinaci, která křížově eliminuje nedostatky druhého principu.

Téma této disertační práce je zaměřeno na 3D mapování, proto v následujícím výčtu je kladen důraz pouze na takové skenery, kterými je možné snímat 3D data

## 1.2 Planární skener

Prvním z typů skenerů popsaných v této rešerši je planární skener. Významnými zástupci této skupiny skenerů jsou skenery URG-04-LX firmy Hokuyo a skenery LMS-200 i PLS-200 firmy SICK [24]. Tyto přístroje jsou k dispozici v Laboratoři Teleprezence a robotiky. Pro základní seznámení se s nimi poslouží obrázek 1.1 a tabulka ?? s jejich základními parametry.



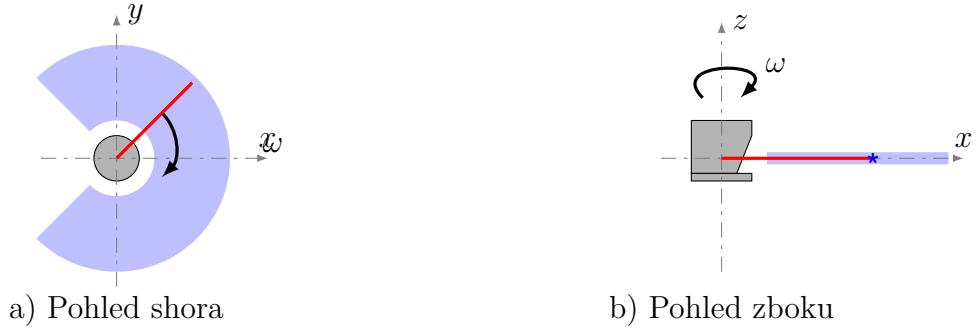
Obr. 1.1: Příklady planárních skenerů: (zleva) URG-04-LX, LMS-200, PLS-200

Tab. 1.1: Parametry dostupných senzorů

Parametr	URG-04-LX	LMS-200	PLS-200
Princip	Triangulace	Time-of-flight	Time-of-flight
Dosah	0.02 m – 4.00 m	0.10 m – 80.00 m	0.10 m – 50.00 m
Rozlišení	1 mm	5 mm	10 mm
Statistická chyba	10 mm	10 mm	20 mm
Úhel měření (horizontálně)	240°	180°	180°
Krok v úhlu	0.36°	0.5°	1°
Frekvence otáčení (max)	10 Hz	20 Hz	20 Hz

Princip funkce planárního skeneru je zřejmý z obrázku 1.2. Bodový měřič vzdálenosti je rotován v ose  $z$ , rychlostí  $\omega$ . V povolené výšce pro měření paprsek měří vnější okolí senzoru.

Tento snímač se nehodí pro přímé skenování ve 3D, avšak lze jím řízeně pohybovat (viz kap. ), čímž se docílí efektu 3D snímání. Proto je nutné zjistit kalibrační vztahy ze kterých bude plynout výsledná přesnost snímku.



Obr. 1.2: Planární skener

### 1.2.1 Základní matematický model

Trajektorii paprsku z obrázku 1.2 je možné jednoduše popsat transformačním vztahem rotace a translace 1.1. Parametr  $d$  zde představuje změřenou vzdálenost od překážky v jednom bodě, a úhel  $\varphi$  představuje úhel natočení měřicího systému v tomto bodě. Matice  $R_z$  představuje matice rotace kolem osy  $z$

$$P(d, \varphi) = R_z(\varphi)X(d) \quad (1.1)$$

Po rozepsání vztahu 1.1 do matic je možné odvodit jednoduchý vztah 1.2

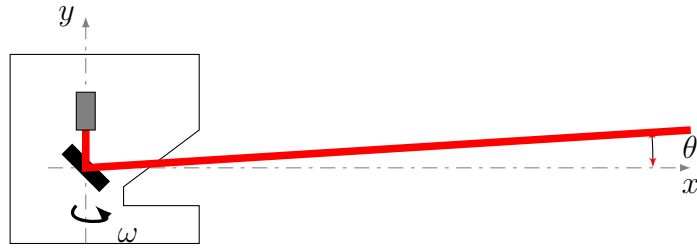
$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} d \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} d \cos \varphi \\ d \sin \varphi \\ 0 \end{pmatrix} \quad (1.2)$$

Tento vztah je možné považovat za nekalibrovaný model chování senzoru. Praktickým měřením bylo zjištěno, že jeho přesnost je nedostatečná pro mapovací potřeby, proto byl navržen model lepší, který zohledňuje parazitní vlastnosti uložení měřicího subsystému. Ten je popsán v následující kapitole.

### 1.2.2 Kalibrovaný matematický model

Jak je zřejmé z obrázku 1.3, kalibrace těchto senzorů spočívá zejména v parametrizaci odchylky rotačního zrcadla vůči rovině skenování. To je možné zapsat matematicky nejlépe jako 1.3

$$P(d, \varphi) = R_z(\varphi + \alpha)R_x(\theta)Y(d) \quad (1.3)$$



Obr. 1.3: Princip kalibrace planárního skeneru

Po rozepsání do matic je možné vyjádřit kalibrovaný model jako vztah 1.5:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos(\varphi + \alpha) & -\sin(\varphi + \alpha) & 0 \\ \sin(\varphi + \alpha) & \cos(\varphi + \alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} 0 \\ d \\ 0 \end{pmatrix} \quad (1.4)$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -d \sin(\varphi + \alpha) \cos \theta \\ d \cos(\varphi + \alpha) \cos \theta \\ d \sin \theta \end{pmatrix} \quad (1.5)$$

### 1.3 Vícepaprskový rotační skener

Dalším ze skenerů, které je možné v práci použít je skupina vícepaprskových 3D skenerů. Mezi významné zástupce této skupiny se řadí skenery firmy Velodyne, konkrétně typy HDL-64 a HDL-32. Tyto přístroje byly k dispozici během řešení této disertační práce. Pro základní přehled o parametrech těchto přístrojů přikládám obrázek 1.4 a tabulku 1.2 s parametry těchto skenerů.

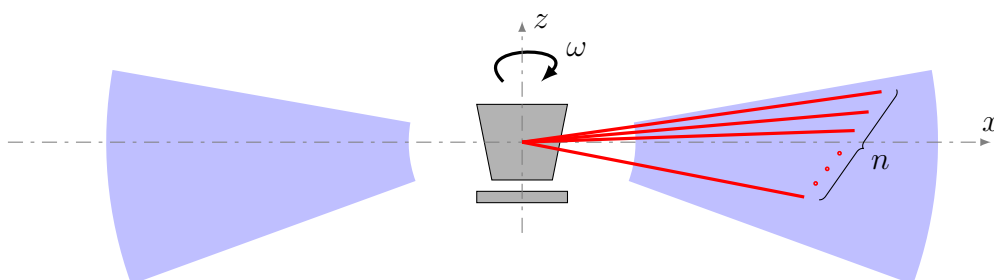


Obr. 1.4: Příklady vícepaprskových skenerů (zleva Velodyne HDL-64 a HDL-32)

Tab. 1.2: Parametry dostupných senzorů

Parametr	HDL-64	HDL-32
Princip	Time-of-flight	Time-of-flight
Dosah	1m - 50m (70m)	1m - 30m (70m)
Rozlišení	20 mm	20 mm
Statistická chyba	50 mm	40 mm
Úhel měření	360°	360°
Krok v úhlu	0.09° při 10Hz	0.16°
Frekvence otáčení	5 Hz - 20 Hz	10 Hz
Počet laserů	64	32

Princip funkce je shodný s obyčejným planárním skenerem, popsáným v předcházející kapitole. navíc však skener obsahuje více laserových vysílačů i přijímačů, takže dokáže simultánně měřit celý vertikální výřez skrz 3D prostor. Rozložení laserů je zřejmé z obrázku 1.5.



Obr. 1.5: Vícepaprskový skener

### 1.3.1 Základní matematický model

Trajektorii každého z paprsků je možné popsat posloupností transformací vyjádřených rovnicí 1.6. Parametr  $\varphi$  určuje aktuální rotaci daného paprsku okolo osy skeneru, a parametr  $\theta$  určuje úhel sklonění právě vybraného paprsku vůči ose rotace.

$$P(d, \varphi, \theta) = R_z(\varphi)R_y(\theta)X(d) \quad (1.6)$$

Po rozepsání do maticové formy je možné dospět k výsledku 1.7.

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \begin{pmatrix} d \\ 0 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} d \cos \theta \cos \varphi \\ d \cos \theta \sin \varphi \\ -d \sin \theta \end{pmatrix} \quad (1.7)$$

Tento vztah lze považovat za nekalibrovaný model chování senzoru. Přesnost tohoto modelu je nedostatečná pro mapovací, ani navigační účely, proto se v dostupné literatuře [29], [10] uvádí různé kalibrační vztahy, kterými lze měření ještě více zpřesnit.

### 1.3.2 Jednobodový kalibrační model

Předchozí vztah (1.7) lze rozšířit na posloupnost transformací vyjádřenou rovnicí 1.8.

$$P(d, \varphi, \theta) = \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix} R_z(\varphi + \alpha) R_x(\theta) [Y(d) + \Delta] \quad (1.8)$$

Ve vztahu (1.8) jsou považovány za kalibrační parametry proměnné  $\alpha$ ,  $\theta$ , a vektor  $\Delta = (H, D, V)^T$ . Tyto proměnné jsou konstantní, pro každý laser různé. Násobení vektorem  $(-1, 1, 1)^T$  je nutné, aby výsledek byl v souladu s kalibračním vztahem, který dodává výrobce [29]. Vektor slouží jako transformace mezi levotočivou a pravotočivou soustavou, aby všechny vztahy uváděné v této práci byly konzistentní.

Rovnici (1.8) lze maticově rozepsat, a po naznačených úpravách lze získat rovnici (1.9).

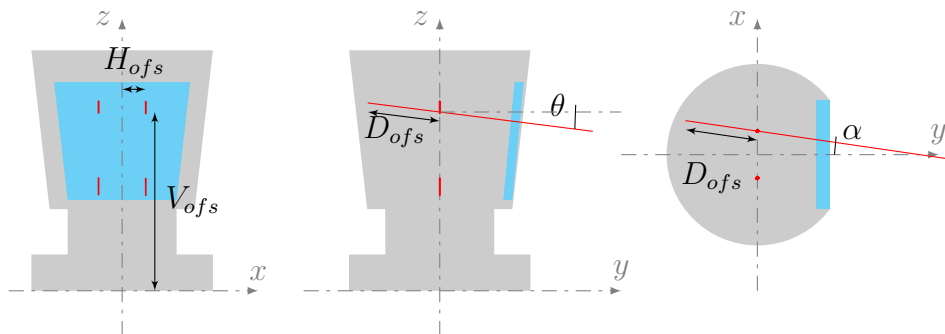
$$\begin{pmatrix} -x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos(\varphi + \alpha) & -\sin(\varphi + \alpha) & 0 \\ \sin(\varphi + \alpha) & \cos(\varphi + \alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{pmatrix} \left[ \begin{pmatrix} 0 \\ d \\ 0 \end{pmatrix} + \begin{pmatrix} H \\ D \\ V \end{pmatrix} \right]$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} ((d + D) \cos \theta - V \sin \theta) \sin(\varphi + \alpha) - H \cos(\varphi + \alpha) \\ ((d + D) \cos \theta - V \sin \theta) \cos(\varphi + \alpha) + H \sin(\varphi + \alpha) \\ (d + D) \sin \theta + V \cos \theta \end{pmatrix} \quad (1.9)$$

Jednotlivé kalibrační parametry jsou znázorněny na obrázku 1.6. V obrázku jsou vyznačeny polohy laserů a jednotlivé kalibrační parametry:

$\alpha$  Úhel odklonu aktuálního laseru od aktuální měřicí osy v horizontální rovině.

- $\theta$  Úhel sklonu aktuálního laseru vůči horizontální rovině.  
 $D$  Korekce změřené vzdálenosti laserem.  
 $H$  Horizontální vybočení laseru vůči ose měření v rovině otáčení senzoru.  
 $V$  Vertikální posunutí laseru vůči referenční rovině otáčení laseru



Obr. 1.6: Význam kalibračních parametrů

V pracích [16],[17] a [20] je popsán postup, jakým se v naší skupině podařilo vytvořit kalibraci pro dostupný laserový skener.

### 1.3.3 Dvoubodový kalibrační model

Tento model vychází z vlastností měřicího ústrojí laserového měření vzdálenosti. Vychází z předpokladu, že jednotka pro odměřování vzdálenosti laserem nemá přesně definovanou konstantu  $D$  (v obrázku 1.6). Respektive, že se hodnota konstanty  $D$  mění s měřenou vzdáleností. Pro adaptaci tohoto zkusení je možné upravit model na vztah:

$$P(d, \varphi, \theta) = \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix} R_z(\varphi + \alpha) R_x(\theta) [Y(d) + \Delta + Y(\varepsilon(d))] \quad (1.10)$$

$$\varepsilon(d) = \frac{d_1 - d}{d_1 - d_0} (D_0 - D_1) \quad (1.11)$$

kde jednotlivé veličiny mají význam:

$d$  Aktuální vzdálenost změřená měřicím ústrojím laseru

$d_0$  Kalibrační vzdálenost 2.4 m

$d_1$  Kalibrační vzdálenost 25.04 m (základní)

$D_0$  Kalibrační konstanta pro vzdálenost 2.4m

$D_1$  Kalibrační konstanta pro vzdálenost 25.04m (základní)

Po rozeepsání a matematických úpravách je možné odvodit výsledný kalibrační vztah:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} ((d + D + \varepsilon(d)) \cos \theta - V \sin \theta) \sin(\varphi + \alpha) - H \cos(\varphi + \alpha) \\ ((d + D + \varepsilon(d)) \cos \theta - V \sin \theta) \cos(\varphi + \alpha) + H \sin(\varphi + \alpha) \\ (d + D) \sin \theta + V \cos \theta \end{pmatrix} \quad (1.12)$$

Tento model byl vybrán jako nejlepší model s nejlepší odezvou.

### 1.3.4 Duální dvoubodový kalibrační model

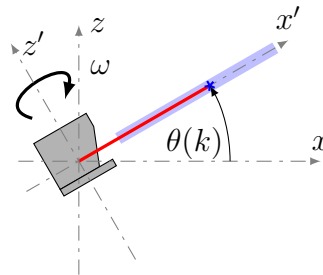
Tento model je publikován výrobcem skeneru jako duální dvoubodový model skeneru. Tento model se nepodařilo odvodit jako soustavu postupných transformací a veličiny v něm obsažené mají jiný význam oproti předchozím modelům.

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} ((d + D + \varepsilon_x(d)) \cos \theta) \sin(\varphi + \alpha) - H \cos(\varphi + \alpha) \\ ((d + D + \varepsilon_y(d)) \cos \theta) \cos(\varphi + \alpha) + H \sin(\varphi + \alpha) \\ (d + \sqrt{(D + \varepsilon_x(d))^2 + (D + \varepsilon_y(d))^2}) \sin \theta + V \cos \theta \end{pmatrix} \quad (1.13)$$

Na tento model se nepodařilo provést kalibraci a výsledky byly dlouhodobě horší, než u dvoubodového modelu, proto byl ve všech následných odvozeních této disertace použit Dvoubodový kalibrační model. (viz kap. ??)

## 1.4 Naklápěný planární skener

3D vzdálenostní data lze získat též naklápěním planárního skeneru. To je graficky znázorněno na obrázku 1.7



Obr. 1.7: Naklápěný planární skener

Jelikož je tento zdroj dat kompozicí klasického planárního skeneru, jehož model již byl popsán, a vlastnostmi rotační transformace, bude proto popsán již jen kalibrovaný model.

### 1.4.1 Kalibrovaný matematický model

Tento model vznikl z planárního skeneru jeho následnou rotací podle osy  $y$ .

$$P(d, \varphi, \theta) = R_z(\varphi + \alpha)R_y(\theta + \beta)X(d + D) \quad (1.14)$$

Po matematických úpravách

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos(\varphi + \alpha) & -\sin(\varphi + \alpha) & 0 \\ \sin(\varphi + \alpha) & \cos(\varphi + \alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\theta + \beta) & 0 & \sin(\theta + \beta) \\ 0 & 1 & 0 \\ -\sin(\theta + \beta) & 0 & \cos(\theta + \beta) \end{pmatrix} \begin{pmatrix} d + D \\ 0 \\ 0 \end{pmatrix}$$

lze dospět ke kalibrovanému vztahu:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} (d + D) \cos(\theta + \beta) \cos(\varphi + \alpha) \\ (d + D) \cos(\theta + \beta) \sin(\varphi + \alpha) \\ -(d + D) \sin(\theta + \beta) \end{pmatrix} \quad (1.15)$$

kde jednotlivé veličiny mají následující význam:

$\beta$  Kalibrační konstanta náklonu zrcadla skeneru

$\alpha$  Kalibrační konstanta odklonu zrcadla skeneru

$\theta$  Náklon celého skeneru

$\varphi$  Úhel vyslání paprsku laseru

$d$  Změřená vzdálenost

$D$  Kalibrační konstanta změřené vzdálenosti

## 1.5 3D kamera

Dalším ze senzorů, které je možné použít pro proces sebelokalizace je bezesporu snímač 3D Time-of-flight kamery. Významným zástupcem této skupiny snímačů je snímač IFM PMD3D, případně Mesa Swissranger SR3000 a SR4000.

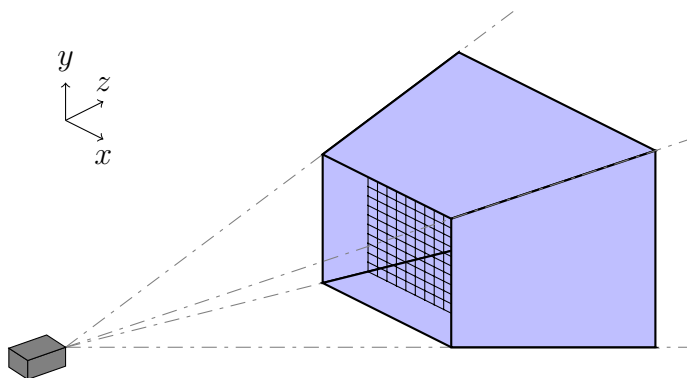
Tab. 1.3: Parametry dostupných senzorů

Parametr	IFM O3D200	Mesa SR3000
Princip	Time-of-flight	Time-of-flight
Dosah	6 m	7.5 m
Rozlišení	0.625°	0.275°
Úhel měření	30°(H) x 40°(V)	47.5° x 39.6°
Počet bodů měření	48 (H) x 64 (V)	176 x 144
Frekvence měření	20 Hz	25 Hz



Obr. 1.8: Příklady 3D kamer – zleva IFM O3D200, Mesa SR3000 a Mesa SR4000

Princip funkce 3D kamery spočívá v měření doby letu paprsku vysílaného LED zářičem na kameře. Přijímaný signál je zpracováván v každém pixelu na kameře, 3D kamera tedy přímo měří 3D vzdálenostní obraz scény před kamerou. To je znázorněno na obrázku 1.9.



Obr. 1.9: 3D kamera

Z tohoto principu plyne, že každá z 3D kamer má vlastní způsob korekcí a kalibrací a nelze určit jedinou kalibrační funkci která by pokryla různé senzory. Pro účely zpracování v této disertační práci se předpokládá, že výrobce dodává knihovní funkci, která vyčítaná data rovnou konvertuje do souřadnicových vektorů  $(x, y, z)^T$ , které jsou kalibrované vztahem dle výrobce.

Na našem pracovišti se podařilo vytvořit vlastní kalibrační vztahy uvedené v pracích [18] [19] [20]. Všechny tyto kalibrace po zobecnění vedou na paprskový model dle rovnice 1.16 a na kalibrační paprskové pole, s vlastní konstantou  $D$ ,  $M$  a  $P_0$  pro každý paprsek jinou.

$$P(d, D, M, P_0) = M(d - D) + P_0 \quad (1.16)$$

kde jednotlivé veličiny mají následující význam:

$P$  Souřadnice dotyku bodu s překážkou

$d$  Změřená vzdálenost v daném bodě

$D$  Kalibrační konstanta pro daný paprsek

$M$  Kalibrační směrový vektor pro daný paprsek

$P_0$  Kalibrační vektor pozice počátku měření daného paprsku

## 1.6 Aspekty použití paprskových senzorů

Veškerá bezdotyková měření okolního prostoru jsou závislá na interakci laserového elektromagnetického vlnění s překážkou. Tato interakce může velmi ovlivňovat přesnost měření a proto je nutné se o některých aspektech zmínit. Tyto nepřesnosti se totiž projevují u všech typů snímačů stejně.

V textu je cílem měření vzdálenosti metodou time-of-flight, protože i dále v disertaci bude pracováno se senzory založenými na této metodě [15].

Dále bude v následujících pasážích předpokládáno užití polarizovaného monochromatického paprsku. Použití multichromatického paprsku celý proces zesložituje a jednotlivé vlastnosti odrazu jsou více skryty.

### 1.6.1 Princip pulzního měření vzdálenosti

Laser na začátku měření každého bodu vyšle puls ve formě elektromagnetické vlny, jejíž intenzitu v každém bodě prostoročasu lze spočítat ze vztahu 1.17 [15].

$$u(t, \tau) = A \cos\left(2\pi \frac{c}{\lambda}(t - \tau)\right) F(t - \tau) \quad (1.17)$$

Kde jednotlivé veličiny značí:

$u(t, \tau)$  Intenzita elektromagnetické vlny paprsku v čase  $\tau$ .

$t$  Čas.

$\tau$  Vzdálenost objektu v časovém měřítku.

$A$  Amplituda elektromagnetických vln odeslaných k překážce.

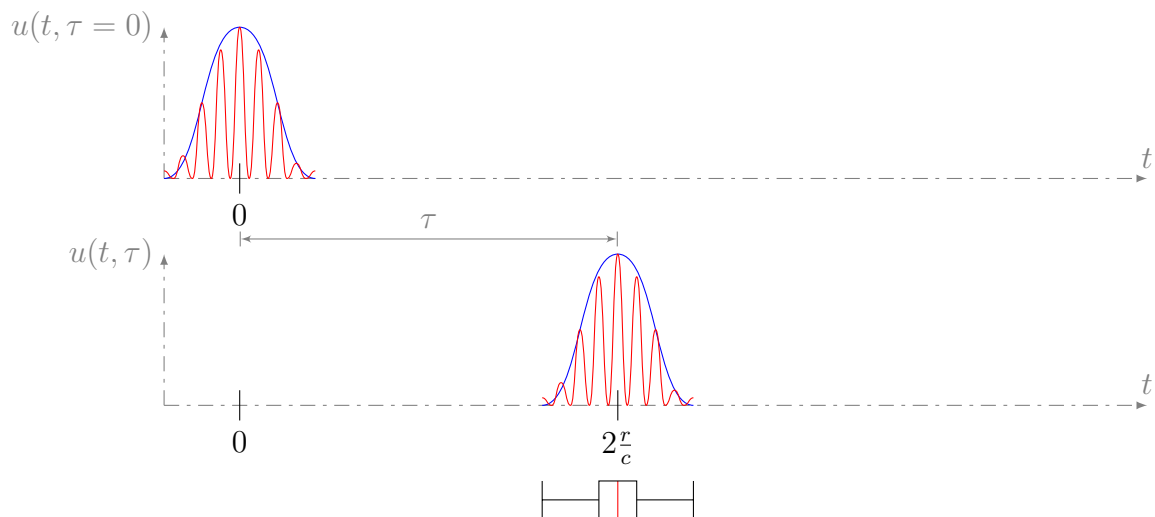
$\lambda$  Vlnová délka elektromagnetických vln.

$c$  Rychlost světla ve vakuu.

$F(x)$  Vnější obálka modulace pulsu.

Tento signál se šíří prostorem, až dosáhne kontaktu s (prozatím pro zjednodušení) bodovou překážkou. Od této se odrazí a šíří se zpět k detektoru. V průběhu letu tedy urazí dvojnásobek vzdálenosti mezi překážkou a měřicím soustrojím.

Na měřicím systému se objeví puls zhruba odpovídající obrázku 1.10. Vertikální osa není v měřítku, protože odražený puls bude utlumen prostředím. Vzhledem k



Obr. 1.10: Princip pulzního měření

zesilovacím stupňům a normalizaci na přijímací straně měřicího řetězce není toto zjednodušení na škodu.

Dalším zjednodušením v tomto a všech následujících obrázcích je znázornění pouze modulační obálky pulsu, s vynecháním interferenčních produktů. Tyto produkty nejsou jednoduše měřitelné, snímač snímá pouze modulační obálku pulsu, proto ani toto zjednodušení není na škodu.

Maximum pulsu se nachází přesně v čase, odpovídajícím dvojnásobku vzdálenosti mezi měřicím soustrojím a bodem, kde se paprsek odrazil. Jelikož se paprsek šíří rychlostí světla, lze psát vztah pro časové zpoždění pulsu v závislosti na vzdálenosti následovně:

$$\tau = 2\frac{r}{c} \quad (1.18)$$

Kde jednotlivé veličiny značí:

$\tau$  Vzdálenost objektu v časovém měřítku.

$r$  Vzdálenost objektu v metrech.

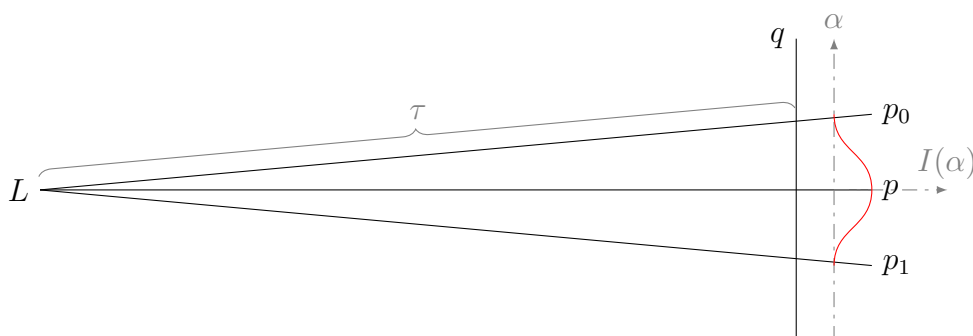
$c$  Rychlost světla ve vakuu.

Nyní záleží na měřicím systému, jakým způsobem bude měřit polohu tohoto pulsu. V současnosti existuje mnoho systémů pro měření doby letu optického pulsu.[15]. Zvolená metoda bude významným způsobem ovlivňovat rozptyl měřených hodnot. Lze však předpokládat, že statisticky významné procento měření bude spadat do oblasti vrcholu pulsu, jak je znázorněno na obrázku 1.10.

## 1.6.2 Efekt rozbíhavosti paprsku

V reálném světě však není technicky možné vysílat pouze nekonečně úzký paprskový zdroj světla. Reálný laserový paprsek má nenulovou rozbíhavost  $\alpha_0$  (např. laserový skener Velodyne HDL-64 má rozbíhavost 2 mrad). Díky této rozbíhavosti je nutné pohlížet i na odraz paprsku komplexněji, než na bodový odraz.

Je třeba též definovat funkci  $I(\alpha)$ , která vyjadřuje rozložení intenzity elektromagnetické vlny v rámci paprsku. Tato funkce slouží více-méně jako váhovací faktor, s vlastností  $\int_{-\infty}^{\infty} I(\alpha) d\alpha = 1$



Obr. 1.11: Rozbíhavost paprsku

Je zřejmé, že na kolmé odrazné ploše se paprsek odrazí zpět do senzoru na každé infinitezimální ploše  $dS$  která je osvětlena. V místě přijímače se všechny paprsky interferenčně skládají do výsledného pulsu, měřeného měřicím soustrojím. Toto interferenční skládání je možné popsat rovnicí:

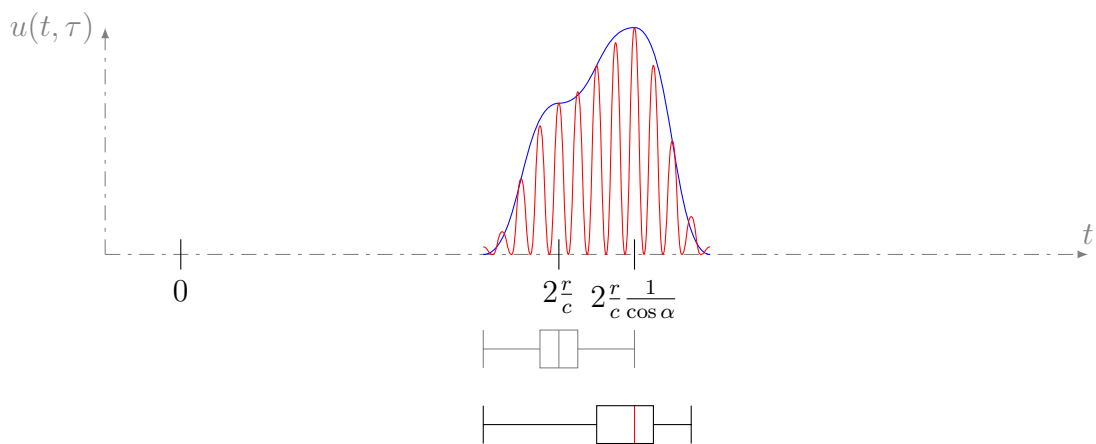
$$u(t) = \int_{-\alpha_0}^{\alpha_0} I(\alpha) u(t, \tau(\alpha)) d\alpha \quad (1.19)$$

kde délka každého z infinitezimálních paprsků je různá, a lze ji popsat následujícím vztahem:

$$\tau(\alpha) = 2 \frac{r}{c \cos \alpha} \quad (1.20)$$

Výsledný puls změřený laserovým soustrojím je uveden na obrázku 1.12. Z tohoto obrázku je zřejmé, že je vyšší procento objemu pulsu obsaženo ve vzdálenosti vyšší, než by odpovídalo kolmé vzdálenosti od překážky.

Tento efekt způsobuje, že změřená hodnota vzdálenosti je vždy nepatrně vyšší než kolmá vzdálenost k překážce u principů vyhodnocení, kdy přijímač hledá amplitudu nebo střed hmoty odraženého pulsu. U metod vyhodnocení, ve kterých je principem funkce prahování, ovlivní tento efekt výslednou měřenou vzdálenost minimálně.

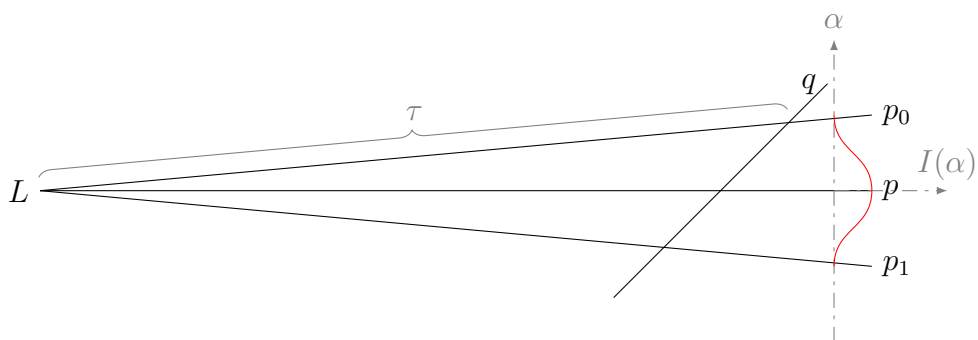


Obr. 1.12: Odražený impuls

U paprskových laserových skenerů bylo ověřeno, že tento efekt zůstává skryt pod kvantizační chybou měřicího soustrojí.

### 1.6.3 Efekt šikmého odrazu

Při měření zejména v urbanistických oblastech se velmi často vyskytují měření dlouhých chodeb, případně koridorů. Je zapotřebí tedy zvážit získání odhadu, jakým způsobem se chová rozptýl měření v takovémto prostředí. Toto prostředí lze definovat obrázkem 1.13, kde přímka  $q$  odpovídá šikmé ploše překážky.



Obr. 1.13: Šikmý odraz paprsku

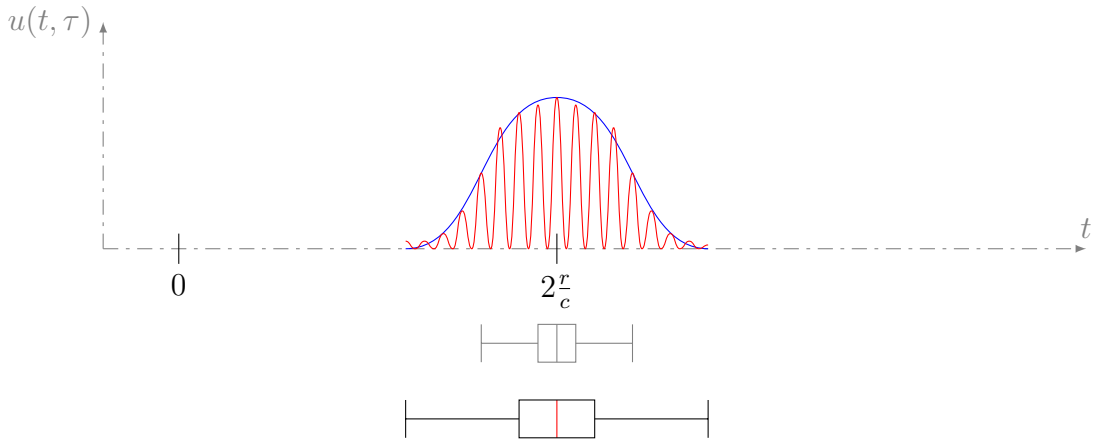
V tomto zobrazení lze definovat parametrické rovnice přímek  $p$  a  $q$  dle vztahu (1.21)

$$p : \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} d \\ 0 \end{pmatrix} + s \begin{pmatrix} \cos \varphi \\ \sin \varphi \end{pmatrix} \quad q : \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + t \begin{pmatrix} \cos \alpha \\ \sin \alpha \end{pmatrix} \quad (1.21)$$

Analýzou průsečíků přímek  $p$  a  $q$  a vzdáleností těchto průsečíků od měřicího systému lze dospět ke vztahu (1.22)

$$\tau(\alpha) = 2 \frac{r}{c \cot \alpha - \cot \varphi \sin \alpha} \quad (1.22)$$

Pokud se do rovnice (1.22) dosadí za  $\varphi = 90^\circ$ , pak celý vztah degeneruje ke vztahu (1.20). Je vidět, že (1.22) bude více ovlivňovat výslednou přesnost. Zakreslením tvaru výsledného pulsu je možné získat obrázek 1.14, ze kterého plyne jak šikmý odraz zvětší rozptyl měřené hodnoty.



Obr. 1.14: Šikmý odraz

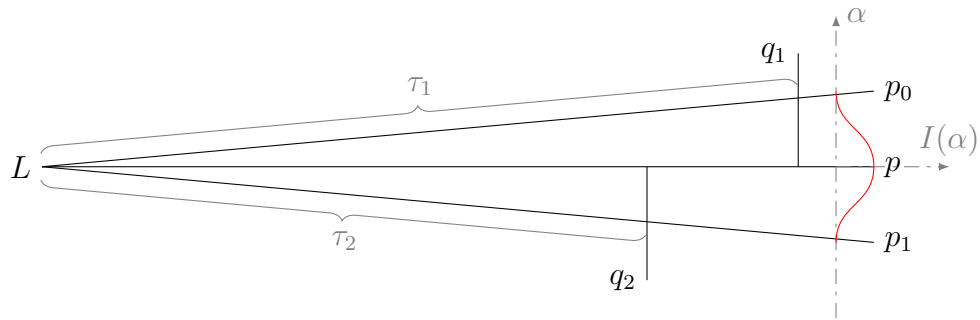
Zde opět záleží na metodě vyhodnocení pulsu, zdali bude ovlivňovat i střední hodnotu. Měřením bylo ověřeno že skener Velodyne HDL-64 je ovlivněn sklonem překážky, kdy u skloněných ploch vykazuje vyšší systematickou chybu (pro úhly  $\varphi$  menší než cca  $20^\circ$ )[11]

#### 1.6.4 Efekt odrazu na hraně, sklo

Dalším velmi častým problémem, který se v měřených datech objevuje, je měření na hraně předmětu, rohu místnosti, případně měření skrz sklo [32][31] [30]. Paprsek se v tomto případě odrazí od dvou objektů nezávisle. To názorně ilustruje obrázek 1.15.

V obrázku 1.15 jsou umístěny dvě překážky  $q_1$  a  $q_2$  do cesty paprsku. Překážky jsou od sebe vzdálené  $d$ . Paprsek se odrazí od obou překážek současně, a vzhledem k různým vzdálenostem od jednotlivých překážek k měřicímu soustrojí jsou různé i časy průchodu paprsku prostředím  $\tau_1$  pro překážku  $q_1$  a  $\tau_2$  pro překážku  $q_2$ .

Výsledné chování laserového měřicího soustrojí se liší podle toho, jak jsou vzdáleny překážky. V okamžiku, kdy  $d \leq cT$  (kde  $T$  značí šířku laserového pulsu a  $c$



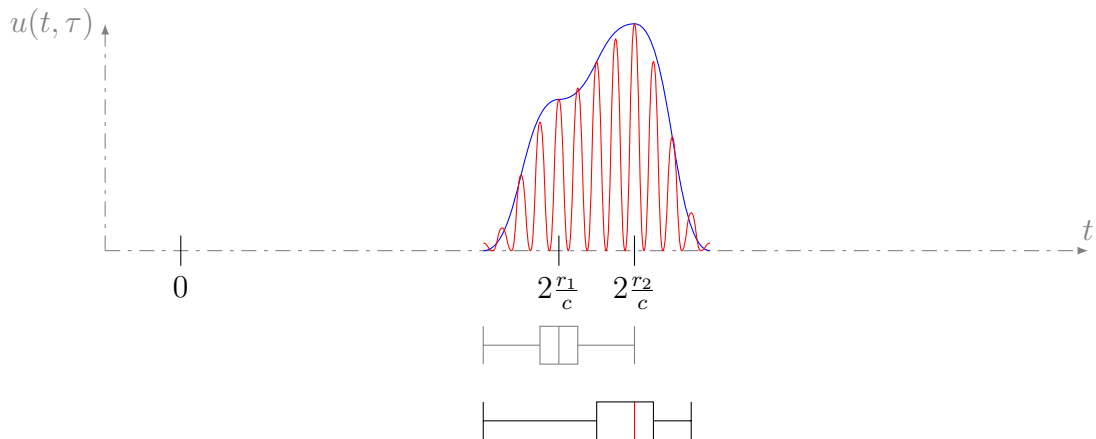
Obr. 1.15: Odraz na hraně

rychlost světla) budou oba pulsy, odražené od překážek splývat, a proto tato konfigurace ovlivní zejména rozptyl naměřených hodnot. Pro případ  $d \geq cT$  bude měřicí systém přijímat sadu pulsů, a s nejvyšší pravděpodobností navrátí první odraz.

U všech v laboratoři dostupných laserových skenerů je délka laserového pulsu rovna hodnotě 5 nanosekund (všechny dostupné skenery zřejmě používají nedestruktivní lavinový průraz tranzistoru s tranzitním kmitočtem blízko 200MHz, jako zdroj rychlých pulsů pro buzení laseru) proto tato kritická vzdálenost dvou překrývajících se překážek  $cT$  vychází kolem 1.5 metru.

### Mezera kratší než 1.5m

Za předpokladu, že je mezera mezi překážkami kratší, než je 1.5m (pro 5ns laserové pulsy), je přijatý puls na přijímací straně možné vyjádřit obrázkem 1.16.



Obr. 1.16: Odražený impuls od dvou překážek blíže než 1.5m

Jak je vidět, oba pulsy od obou překážek se smíchaly, a nejde rozlišit, který z pulsů odpovídá které z částí překážky. Měřicí soustava většinou produkuje jako

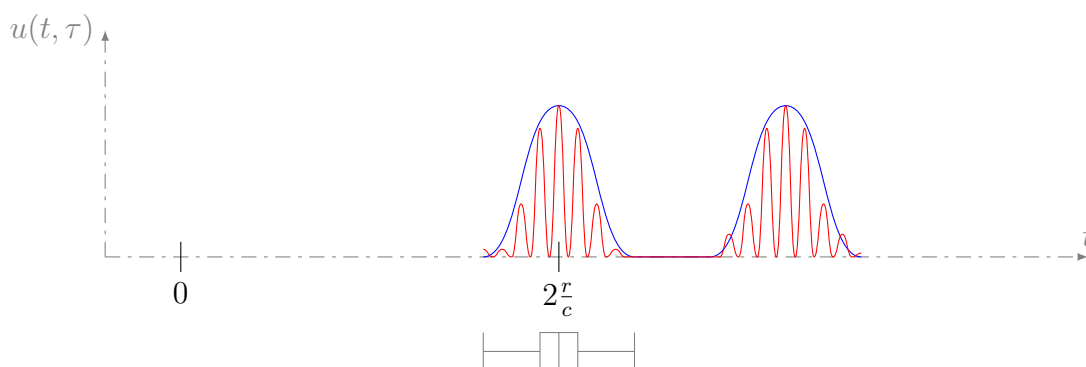
výslednou změřenou vzdálenost hodnotu blízko středu mezi překážkami, vychýlenou blíže k překážce, která byla více osvětlena laserovým paprskem.

V praxi se tento problém v datech vyskytuje v urbanistických oblastech velmi často, proto je nutné data získaná laserovým snímačem filtrovat.

### Mezera delší než 1.5m

V okamžiku, pokud jsou překážky od sebe vzdáleny více než 1.5m (pro 5ns laserový puls) je možné zachytit měřicím soustrojím pulsy podobné obrázku 1.17. Každá překážka odpoví „svým“ impulsem. Tyto impulsy jsou jednoznačně separabilní, a výslednou změřenou vzdálenost neovlivňují na přesnosti.

Zde je třeba rozlišit, jakým způsobem odraz vznikl. Pokud vznikl dvojitý odraz průchodem sklem, lze předpokládat, že intenzita odraženého paprsku od skla bude mnohem slabší, než intenzita odraženého paprsku od překážky za sklem. V tomto případě měřicí subsystém navrátí hodnotu času druhého odrazu (protože bude amplitudou výrazně vyšší). V okamžiku pokud je dvojitý odraz výsledkem částečného překrytí paprsku jiným objektem (například rohem místnosti), bylo pozorováno, že měřicí subsystém navrátí hodnotu blíže položeného objektu, nezávisle na ploše překrytí paprsků.



Obr. 1.17: Dva odrazy

Dostupné laserové skenery mají většinou dostupný alternativní firmware, který umožňuje navrácení časových údajů obou impulsů z obou odrazů. Tento alternativní firmware se u skeneru jmenuje „Dual Return“, a zvýší datový tok z laserového skeneru na dvojnásobek. Vzhledem k výkonově-kapacitním důvodům byl použit v této disetační práci standardní firmware a možnost počítání s vícenásobnými odrazy nebyla využita.

## 2 TVORBA MAP V MOBILNÍ ROBOTICE

V následující kapitole budou popsány jednotlivé problémy spjaté s tvorbou map v mobilní robotice a jejich možnými řešeními.

### 2.1 Mračna naměřených bodů

Základním způsobem jak ukládat data je metoda ukládání přímo naměřených dat ve formátu mračen bodů. (angl. Point Clouds)[23].

Data změřená laserovým skenerem jsou přes kalibrační vztahy převedena do formátu XYZ souřadnic bodů dotyku laserového paprsku s překážkou, a tyto body jsou uloženy do souboru pro každý celistvý sken do samostatného souboru. Pro každý soubor jsou uloženy dodatečné informace z lokalizační jednotky mobilního robotu, aby bylo možné zpětně dohledat orientaci a natočení laserového skeneru v okamžiku měření.

Hlavní výhodou tohoto způsobu ukládání naměřených dat je vysoká rychlost zpracování, kdy obsluhující počítač pouze provádí kalibraci laserem naměřených dat a tyto následně ukládá ve strojově čitelném tvaru na paměťové médium v počítači.

Hlavní nevýhodou tohoto způsobu ukládání naměřených dat je nízká hustota informace uložené v paměťové jednotce, a nízká rychlost prohledávání takto uložené mapy. Aby případný následný algoritmus mohl z mapy vyjmout informaci o tom, zdali se v určitém místě nachází nebo nenachází překážka, musí projít úplně všechny soubory a úplně všechny naměřené body, a na každém naměřeném bodě provést test, zdali se nachází ve vyšetřované oblasti.

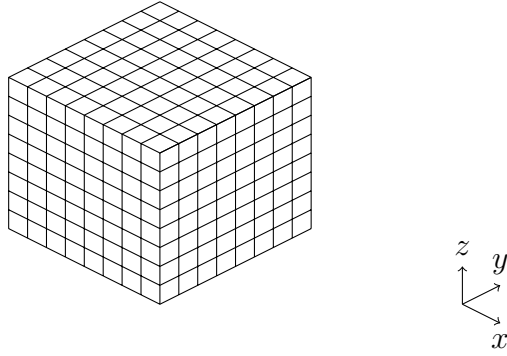
### 2.2 Mřížka obsazenosti (Occupancy Grid)

Řešením problémů popsaných v předcházející metodě je v dnešní době hojně používaná metoda ukládání mapy ve formě mřížky obsazenosti. Mapou v tomto smyslu se rozumí popis uspořádání obsazených a neobsazených míst v rámci mapy.

Nechť je celý prostor, který je potřeba zmapovat rozdělen na pravidelné oblasti. Tyto oblasti nazveme jednotlivými buňkami mapy. Jako příklad takového rozdělení slouží obrázek 2.1.

Nechť každá buňka mřížky obsazenosti obsahuje informaci udávající míru obsazenosti  $\mathcal{M}_i$  dané buňky. Míru obsazenosti dané buňky lze spočítat za předpokladu znalosti údajů z právě proběhlého měření  $\mathcal{R}_i$ , a vlastností světa  $\mathcal{W}_i$

$$\mathcal{M}_i = f(\mathcal{R}_i, \mathcal{M}_i, \mathcal{W}_i) \quad (2.1)$$



Obr. 2.1: 3D mřížka obsazenosti

Problémem této metody je, jakým způsobem je možné matematicky vyjádřit míru obsazenosti buňky  $\mathcal{M}_i$ . Existuje mnoho přístupů, některé zajímavé jsou popsány v následujících kapitolách.

### 2.2.1 Bayes

Tento přístup, jehož základy položil Alberto Elfes v [5] je v současnosti nejpoužívanějším přístupem pro řešení dané problematiky. Vychází ze základních vztahů podmíněné pravděpodobnosti.

**Definice 1.** *Nechť je podmíněná pravděpodobnost výskytu jevu  $B$  za předpokladu, že nastal jev  $A$  označena  $P(B|A)$ . Tuto pravděpodobnost je možné vyjádřit symetricky z pravděpodobnosti  $P(A|B)$  výskytu jevu  $A$  za předpokladu jevu  $B$  následujícím způsobem (definice podmíněné pravděpodobnosti):*

$$P(B|A) = \frac{P(B, A)}{P(A)} = \frac{p(A|B)P(B)}{P(A)} \quad (2.2)$$

Za předpokladu použití v mřížkách obsazenosti je možné přepsat vztah s použitím jednotných veličin používaných v této práci:

$$P(\mathcal{M}_i^O|\mathcal{R}) = \frac{p(\mathcal{R}|\mathcal{W}_i^O)P(\mathcal{M}_i^O)}{p(\mathcal{R}|\mathcal{W}_i^O)P(\mathcal{M}_i^O) + p(\mathcal{R}|\mathcal{W}_i^E)P(\mathcal{M}_i^E)} \quad (2.3)$$

Každá buňka v mřížce obsazenosti (dále jen OG) tedy obsahuje pravděpodobnost obsazenosti dané buňky  $P(\mathcal{M}_i)$  a v každém cyklu měření je v každé buňce tato hodnota aktualizována za pomocí vzorce 2.3

Tento způsob estimace světa má oproti jiným velkou výhodu v možnosti experimentálního i analytického určení parametrů senzoru  $p(\mathcal{R}|\mathcal{W}_i^O)$  a  $p(\mathcal{R}|\mathcal{W}_i^E)$ .

Nevýhodou tohoto přístupu je složitý vztah, který je nutné počítat pro každou buňku měření. Následující metoda výpočty zjednodušuje.

## 2.2.2 Odd ratio

Tento přístup vychází z předpokladu zjednodušení vzorce 2.3 o odstranění výpočtu věty o úplné pravděpodobnosti.

**Definice 2.** Necht funkce  $\mathbb{O}(A)$  představuje odd ratio (poměr šancí <sup>1</sup>), že nastane událost  $A$  oproti tomu, že nenastane:

$$\mathbb{O}(A) = \frac{p(A)}{p(\bar{A})} \quad (2.4)$$

Necht je podmíněné odd ratio události  $B$  za podmínky že nastal jev  $A$  definován jako:

$$\mathbb{O}(B|A) = \frac{P(B|A)}{P(\bar{B}|A)} \quad (2.5)$$

Necht  $\lambda(B, A)$  odpovídá koeficientu věrohodnosti jevu  $B$ , za předpokladu, že nastal jev  $A$

$$\lambda(B|A) = \frac{p(B|A)}{p(\bar{B}|A)} \quad (2.6)$$

Pokud se aplikuje na předchozí postuláty Bayesův vzorec 2.2, je možné po jistých matematických úpravách dosáhnout následujícího vztahu:

$$\mathbb{O}(B|A) = \lambda(A|B)\mathbb{O}(B) \quad (2.7)$$

*Důkaz.*

$$\mathbb{O}(B|A) = \frac{P(B|A)}{P(\bar{B}|A)} = \frac{\frac{P(B,A)}{P(A)}}{\frac{P(\bar{B},A)}{P(A)}} = \frac{p(A|B)P(B)}{p(A|\bar{B})P(\bar{B})} = \lambda(A|B)\mathbb{O}(B)$$

□

**Definice 3.** Pokud jev  $B$  závisí na více jevech  $A \in \{A_1, A_2, \dots, A_n\}$ , pak lze psát

$$\mathbb{O}(B|A_1, A_2, \dots, A_n) = \mathbb{O}(B) \cdot \prod_k \lambda(A_k|B) \quad (2.8)$$

V případě použití teorie odds ratio pro aplikaci na mřížku obsazenosti, je možné jednotlivé veličiny v předchozích vzorcích přeznačit.

$$\mathbb{O}(\mathcal{M}_i) = \frac{P(\mathcal{M}_i^O)}{P(\mathcal{M}_i^E)} \quad (2.9)$$

$$\lambda(\mathcal{R}|\mathcal{W}_i) = \frac{p(\mathcal{R}|\mathcal{W}_i^O)}{p(\mathcal{R}|\mathcal{W}_i^E)} \quad (2.10)$$

<sup>1</sup>Volný překlad, český výraz pro daný pojem není přesný.

Výsledné odd ratio obsazenosti  $\mathbb{O}(\mathcal{M}_i)$  buňky za předpokladu výskytu souboru měření  $\mathcal{R}_k$  je možné vyjádřit jako:

$$\mathbb{O}(\mathcal{M}_i|\{\mathcal{R}\}) = \mathbb{O}(\mathcal{M}_i) \cdot \prod_{\{\mathcal{R}\}} \lambda(\{\mathcal{R}\}_k|\mathcal{W}_i) \quad (2.11)$$

Každá buňka v mřížce obsazenosti v tomto případě implementace obsahuje hodnoty *odds ratio*  $\mathbb{O}(\mathcal{M}_i)$ .

Výhodou tohoto přístupu je jednoduchý výpočet. Vytvořit model senzoru  $\lambda(\mathcal{R}|\mathcal{W}_i)$  lze stejně jako v předchozí kapitole.

### 2.2.3 Log odds ratio

Pro ještě vyšší možnost zrychlení výpočtů na každé buňce OG (případně pro možnost výpočtů na strojích bez hardwarové akcelerace násobení, čehož bude v této disertaci využito) je možné přistoupit k logaritmování předchozího vztahu. Logaritmováním rovnice 2.11 je možné dospět k jednoduchému výpočtu log odds ratio <sup>2</sup>.

$$\log \mathbb{O}(\mathcal{M}_i|\{\mathcal{R}\}) = \log \mathbb{O}(\mathcal{M}_i) + \sum_{\{\mathcal{R}\}} \log \lambda(\{\mathcal{R}\}_k|\mathcal{W}_i) \quad (2.12)$$

Každá buňka v OG v tomto případě implementace obsahuje hodnoty *log odds ratio*  $\log \mathbb{O}(\mathcal{M}_i)$ .

Mezi výhody tohoto přístupu dozajista patří výpočetní nenáročnost, avšak problémem je přesnost a stabilita výsledků.

### 2.2.4 Dempster-schafer

Narozdíl od předchozích realizací, které problém obsazenosti buňky aproximovaly dvěma disjunktními jevy obsazeno/volno, tedy  $P(\mathcal{M}_i^E) + P(\mathcal{M}_i^O) = 1$  (nebo jejich poměrem), je teorie Dempster Schafer (DS) založena na úvaze věrohodnosti tvrzení, že je daná buňka obsazená a tvrzení že je daná buňka volná jako dvou nezávislých jevů. [28]

Pro výpočty uvnitř OG je tedy třeba pro každou buňku ukládat dvě hodnoty.

$$Bel(\mathcal{M}_i^O) = p(\mathcal{M}_i^O)$$

$$Bel(\mathcal{M}_i^V) = p(\mathcal{M}_i^V)$$

Tento přístup je popisován zejména v [2].

Vzhledem k výpočetní a datové náročnosti tato metoda nebude dále rozebírána a je uvedena jen pro celistvost řešerše problematiky.

---

<sup>2</sup>logaritmický poměr šancí. Český překlad pro daný pojem není přesný

## 2.3 Mřížka evidence (Evidence Grid)

V kapitole 2.2 popisovanou metodu uložení míry obsazenosti do jednotné ekvidistntní mřížky s daty lze rozšířit o způsob uchování rozšířené informace o daném koncovém voxelu. Koncový voxel tedy nebude obsahovat pouze míru obsazenosti/volnosti (nebo hodnotu odds ratio), ale bude obohacen o informace navíc, například o barvu bodu, teplotu, materiál povrchu apod.

Tímto způsobem obohacená mapa může sloužit například ke hledání obětí živelních katastrof, k hledání poškozených potrubí ve zdech, unikajících látek apod.

Aby takto vytvořená mřížka měla smysl, je nutné definovat určitá pravidla jak pro přidávané informace, tak pro operace nad nimi. Jedním z důležitých předpokladů pro každou přidanou proměnnou je nalezení integrační funkce, s jejíž pomocí budou informace do jednotlivé buňky integrovány. S tím souvisí i druhý požadavek integrability jednotlivých nově přidávaných položek.

Pro posuzování integrability platí, že požadovaná integrační funkce musí splňovat několik podmínek. První podmínkou je, že při integraci dvou shodných vstupů by měla být výstupem shodná hodnota. Druhou podmínkou je, že dvě různé hodnoty musí funkce převést na třetí hodnotu, která patří do skupiny platných hodnot veličiny. Třetí a poslední podmínkou kladenou na tuto funkci je, že by funkce měla účinně filtrovat outliery.

Například veličina teplota je integrabilní a má několik možných způsobů integrace. Může se do mřížky integrovat jako maximální, či minimální z měřených teplot (v mapě se projeví i okamžité oteplení či ochlazení některé ze zdí), jako medián nebo průměrná teplota (výsledkem je dlouhodobější teplotní mapa prostředí, odfiltrovává pohybující se objekty).

Jako druhý příklad, barva voxelu, je složitější. Barvou bodu totiž rozumíme soustavu několika číselných vzájemně nezávislých parametrů (například složky modelů RGB, HSV, HSB, CMYK apod.) Pro účely integrace do mřížky evidence je vhodné vybrat takový model, který je fyzikálně integrabilní. Z navržených není pro integraci vhodný model RGB, protože prostý průměr jednotlivých složek dává na výstupu úplně jinou barvu než barvy na vstupu. Jako vhodný model se jeví HSV či HSB. Pokud jsou ukládány hodnoty v RGB, je vhodné je převést do HSB či HSV, v tomto modelu provést operace, a pak následovně převést zpátky do RGB pro uložení. Existují i jiné přístupy integrace, založené na histogramech, které jsou ale pro použití v mřížkách evidence naprosto nevhodné z důvodu vysokého záboru paměti.

Dalším problémem tohoto způsobu uložení mapy spočívá v tom, že narozdíl od mřížky obsazenosti není definován profil měření celé trajektorie paprsku, ale zpravidla pouze bod dotyku s překážkou. V praxi se tudíž používá systém, kdy obsazenost se nepočítá klasickým způsobem jako bylo u mřížky obsazenosti, ale počítá se pouze

bod dotyku paprsku s překážkou, případně nejbližší okolí tohoto bodu, současně s integrací hodnot z ostatních snímačů. Tím se docílí oproti klasické mřížce obsazenosti významného zrychlení operací, za cenu pouze aditivního využití mapy (jednou přidaná chybná hodnota měření nebude vymazána více správnými měřeními).

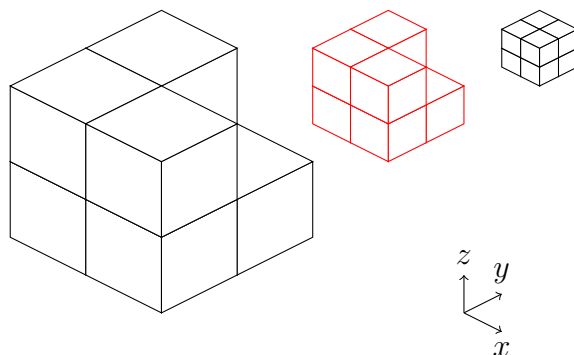
## 2.4 Octree

Problémy s paměťovou náročností metod popsaných v kapitolách 2.3 a 2.2 lze řešit některým ze způsobů komprese.

Při důsledném pohledu na data uvnitř mřížky je vidět, že většina mřížky je zaplněna prázdným prostorem. Klíčovým faktorem vhodné komprese je tedy schopnost dané metody neukládat prázdný prostor, a ukládat pouze obsazené či do měření jinak zapojené buňky.

V praxi nejpoužívanější jsou KD stromy (KD Tree) a oktalové stromy [22] (Octree). KD stromy dosahují vyšších kompresních poměrů, ale pro svou složitost operací nad nimi byly pro účely této práce zavrhnuty.

Principem Octree je schopnost rekurzivního dělení prostoru na poloviny ve všech osách, jak je znázorněno na obrázku 2.2. Toto rekurzivní dělení způsobí vytvoření oktalového stromu.

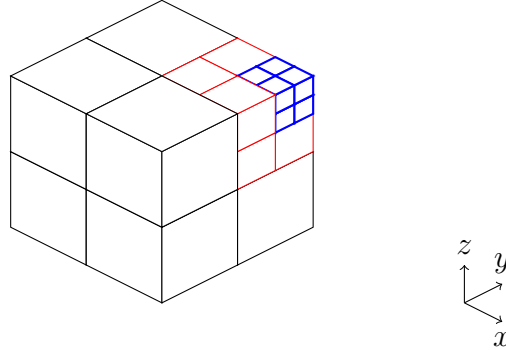


Obr. 2.2: Princip Octree

Systém ukládání zajišťuje, že jsou uloženy pouze neprázdné buňky tohoto stromu. Systém ukládání potřebuje uložit navíc informaci, které z položek jsou prázdné a které jsou plné. Tato informace způsobuje určitý pokles komprese uložených dat, zejména u téměř zaplněného stromu.

Rekurzivní aplikací těchto pravidel umožňuje metoda vytvořit prostorovou stromovou síť, v obecném případě různě hlubokou. Jako příklad takové sítě je uveden

na obrázku 2.3 kdy je obsazen pouze jeden prvek mřížky (pravý horní roh). Re-  
prezentující oktalový strom je hluboký 3 větve a jeden list reprezentuje uloženou  
hodnotu.



Obr. 2.3: Octree

Pro posouzení paměťové náročnosti vyžadované navíc pro plný strom, je nutné  
srovnat paměťovou náročnost plné mřížky evidence s náročností plného oktalového  
stromu. To je srovnáno vztahy 2.13.

$$S_E = S_V 2^{3N} \qquad S_O = S_V 2^{3N} + S_T \sum_{i=0}^{N-1} 2^{3i} \qquad (2.13)$$

kde jednotlivé veličiny značí:

$S_E$  Velikost paměti zabrané mřížkou obsazenosti.

$S_O$  Velikost paměti zabrané oktalovým stromem.

$S_V$  Velikost paměti jedné buňky mřížky obsazenosti.

$S_T$  Velikost paměti pro jednu větev oktalového stromu.

$N$  Hloubka oktalového stromu.

Pro vyjádření relativní redundance komprese v závislosti na obsazenosti stromu  
je možné z 2.13 jednoduše odvodit 2.4

$$\delta_N = \frac{S_T \sum_{i=0}^{N-1} 2^{3i}}{S_V 2^{3N}} = \frac{S_T}{S_V} \sum_{i=1}^N 2^{-3i} \sim \frac{9S_T}{64S_V} \qquad (2.14)$$

Ze vztahu plyne, že velikost uloženého octree stromu, při bitové hloubce  $N = 8$ ,  
velikosti  $S_T = 2$  a velikosti  $S_V = 4$  vychází o cca 7 % vyšší, než je množství skutečně  
obsazených buněk v mapě.

Kompresní poměr tedy kopíruje množství volného prostoru uvnitř mapy. Autor  
[22] uvádí průměrné obsazení prostoru mřížky v urbanistických oblastech na 10%.

V této práci se podařilo prokázat nepřímou úměrnost mezi velikostí voxelu a zaplněností mapy v urbanistických oblastech za shodných podmínek. Z toho plyne, že vyšší bitové hloubky stromu budou dosahovat lepších kompresních poměrů.

Rozdíl v použití Octree pro mřížku evidence a mřížku obsazenosti spočívá v definici prázdného prostoru. Mřížka obsazenosti nikdy neobsahuje prázdný prostor, jen prostor ve kterém je pravděpodobnost výskytu překážky velmi malá. Proto mřížka obsazenosti tvoří od základu plný strom a nedochází u ní k žádné kompresi. Aby bylo dosaženo dobrých kompresních výsledků, je nutné mřížku obsazenosti prahovat, tedy voxely s hodnotou obsazenosti nižší než určitá nastavená hodnota budou považovány za volné. Z tohoto důvodu je možné říci, že tato komprese je bezztrátová pouze pro mřížky evidence, mřížka obsazenost bude vždy komprimována ztrátově, nebo s nízkým kompresním poměrem.

## 2.5 Výběr vhodné metody uložení mapy

V předcházejících kapitolách byly rozebrány jednotlivé možnosti, jakými lze ukládat mapu. Pro účely jednoduchých výpočtů a možnosti masivně paralelizovat výpočty byl jako formát mapy zvolen lineární mřížka obsazenosti využívající principu zjednodušení Odds Ratio.

V okamžiku ukládání mapy na disk, případně odesílání mapy po sdíleném komunikačním prostředku byl vybrán formát uložení ve stromové struktuře Octree, se ztrátovou kompresí.

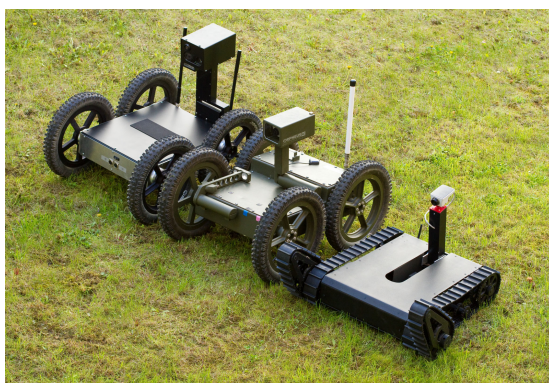
Díky kompresi v Octree formátu je nutné, aby po dobu, kdy bude počítač mapu aktualizovat měřením ji měl v paměti uloženou jako lineární mřížku obsazenosti s buňkami odds ratio. Uložením se ztratí určité množství informace (viz kap. 2.3), které však následně umožní rychlejší běh algoritmů.

### 3 ROBOTICKÝ SYSTÉM CASSANDRA

CASSANDRA je (mobilní) robotický systém vytvářený na pracovištích ÚAMT FEKT VUT v Brně a CEITEC RG2-2 od roku 2010. Jde o systém obsahující několik mobilních robotů s různými vlastnostmi, který je ovládán jedním nebo více operátory. Typicky jde o jeden mapovací robot (např. envMAP), dva větší průzkumné pozemní roboty (Orpheus-X3, Orpheus-AC2), dva malé průzkumné roboty (Perseus, Bron-tes), a jeden létající robot (Uranus HEXA). Všechny části systému, tj. jednotlivé roboty i operátorské stanice, jsou vzájemně propojeny bezdrátovou nebo drátovou komunikační sítí.

Systém je určen především pro vzdálený průzkum oblastí člověku nebezpečných nebo nedostupných. Snahou našeho týmu je vytvořit systém CASSANDRA jako velmi univerzální, přesto jsou definovány hlavní úlohy, které se pomocí něj snažíme řešit:

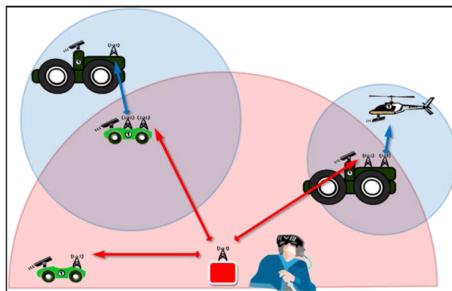
- Multispektrální mapování vnějších a vnitřních prostor.
- Měření environmentálních parametrů – například koncentrace látek ve vzduchu, kontaminace chemickými nebo biologickými látkami, radiace, šíření látek v čase a prostoru, apod.
- Automatické hlídání předem definovaného a známého prostoru – například hlídání koncentrací látek okolí chemického provozu nebo nukleárního zařízení v předem definovaných časových intervalech.
- Vyhledávání osob v nebezpečných nebo nedostupných oblastech – místa živelní katastrofy, bojiště, apod.
- Obecný průzkum neznámých oblastí.



Obr. 3.1: Roboty v robotickém systému Cassandra

Hlavním způsobem ovládání celého systému i jednotlivých robotů je v současnosti tzv. vizuální teleprezence, kterou je možné popsat na základě obrázku 3.2. Operátor

má na hlavě helmu/brýle virtuální reality se snímačem pohybů hlavy. Data z tohoto snímače jsou přenesena k robotu, který má kameru nebo kamerový systém umístěný na manipulátoru s alespoň dvěma rotačními stupni volnosti. Kamera na robotu tak kopíruje pohyby hlavy operátora, a protože obraz z kamery je přenášén do jeho brýlí, ten se cítí, jako kdyby byl na místě robotu.



Obr. 3.2: Robotický systém CASSANDRA

Jedním z hlavních cílů této práce je postupný přechod od teleprezenčního řízení k řízení v rozšířené realitě, kdy se snažíme uživateli prezentovat data, která by jinak nemohl sám vnímat - např. teploty, ale také možnost kombinovaného zobrazení 3D mapy prostředí a reality. Dalším důležitým cílem je postupný přechod od operátorského řízení strojů v reálném čase k alespoň částečné autonomnosti.

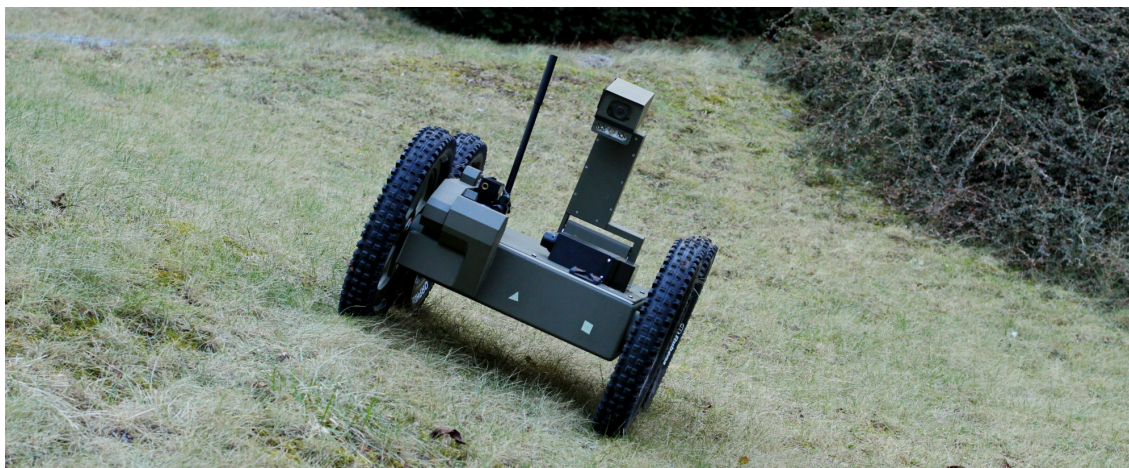
Projekt navazuje na několik předchozích projektů, které však vždy obsahovaly samostatné robotické jednotky. Nejdůležitějšími předchůdci systému CASSANDRA jsou roboty řady Orpheus, především Orpheus-X2 a armádní verze Orpheus-AC.

## 3.1 Roboty

Nyní budou popsány jednotlivé roboty systému CASSANDRA. Pro účely této práce je důležitý robot Orpheus-X3 (a Orpheus-AC2, ze kterého vychází), na jehož tvorbě jsem se podstatnou měrou podílel – bude tedy popsán podrobněji s důrazem na části důležité pro tuto práci. Podílel jsem se však na vývoji všech zde jmenovaných robotů.

### 3.1.1 Orpheus-AC2

Jde o pokročilý průzkumný mobilní robot určený pro armádní chemický a radiační průzkum. Robot byl vyvinut v naší skupině pro Armádu České republiky na objednávku divize VTUO VOP 026 Šternberk. Robot vychází ze stroje Orpheus-AC, který



Obr. 3.3: Robot Orpheus-AC2

podstatným způsobem vylepšuje jak pohonným systémem, tak vnitřní elektronikou a senzorickým vybavením.

Robot je navržen a zkonstruován pro práci v těžkém terénu a za extrémních podmínek, a to jak klimatických, tak z hlediska možné kontaminace chemickými a biologickými látkami a radiačním spadem. Tomu je přizpůsoben jak celý koncept stroje, tak design jednotlivých částí. Stroj prošel 19 MIL-STD zkouškami ověřujícími jeho vlastnosti, včetně environmentálních, elektromagnetické kompatibility a speciálních (vibrace, rázy, dekontaminovatelnost, odolnost na chemické látky, atd).



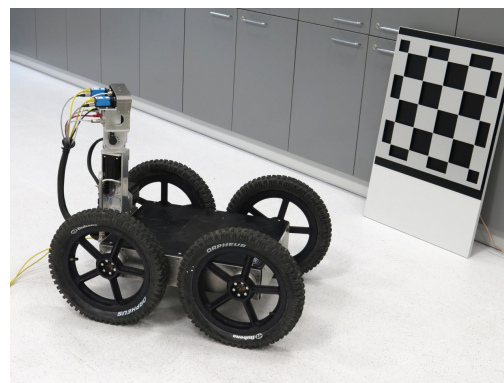
Obr. 3.4: Robotický systém Orpheus-AC2

### 3.1.2 Orpheus-X3

Vychází z robotu Orpheus-AC2, se kterým sdílí stejný pohonný systém a část elektroniky. Naopak značně rozdílná je konstrukce šasi, která je u robotu Orpheus-AC2



Obr. 3.5: Robot Orpheus-AC

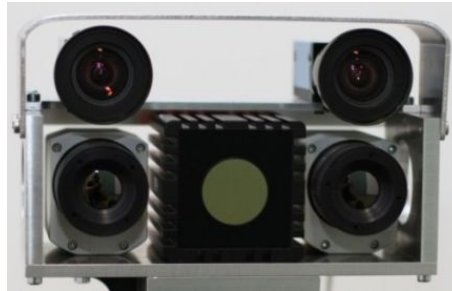


Obr. 3.6: Robot Orpheus-X3

vytvořena ze samonosného svařence z hliníkové slitiny, zatímco u robotu Orpheus-X3 jde o ocelový rám na který jsou přišroubovány desky z lehkých slitin. Toto řešení je mnohem vhodnější pro neustálý vývoj a experimenty – desky je možné snadno upravovat a případně vyměnit. Pro armádní stroj je naopak výhodná nižší celková hmotnost a vodotěsnost svařené konstrukce.

Orpheus-X3 má navíc senzorkou hlavici se třemi stupni volnosti s velmi přesnými motory a převodovkami – zákaznické AC motory TG-Drives s převodovkou harmonic drive od firmy Spinea, včetně speciálních měničů. Tato hlavice je velmi důležitá pro mou práci, protože umožňuje umístění senzorkého systému o hmotnosti až 3kg a jeho velmi přesné a přitom rychlé polohování. Robot může komunikovat bezdrátově, stejně jako Orpheus-AC2, ale i přes dvě metalická propojení Ethernet s rychlostí až 1Gbit/s. Tato variabilita je velmi důležitá při různých typech experimentů.

Robot je vybaven dvěma tzv. embedded počítači s operačním systémem Micro-

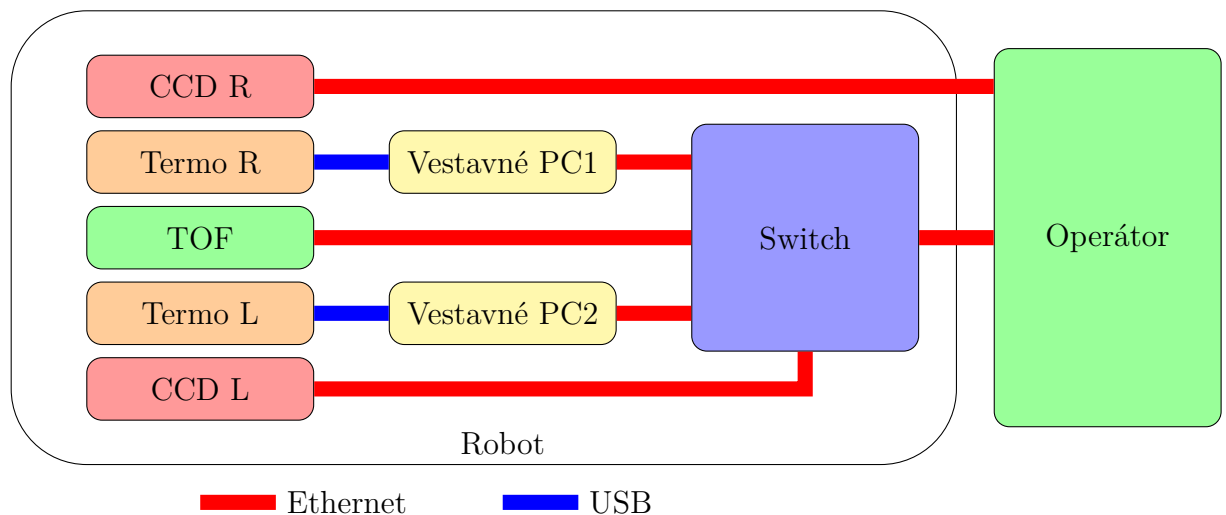


Obr. 3.7: Hlavice

soft Windows, aby bylo možné provádět i přímo na něm složité výpočty ve vyšších programovacích jazycích. Konkrétně jde o počítače Intel NUC s procesorem Intel Core i3.

Vzhledem k využití robotu pro experimentální činnost i pro praktické (např. komerční) využití, používáme dvě základní zapojení.

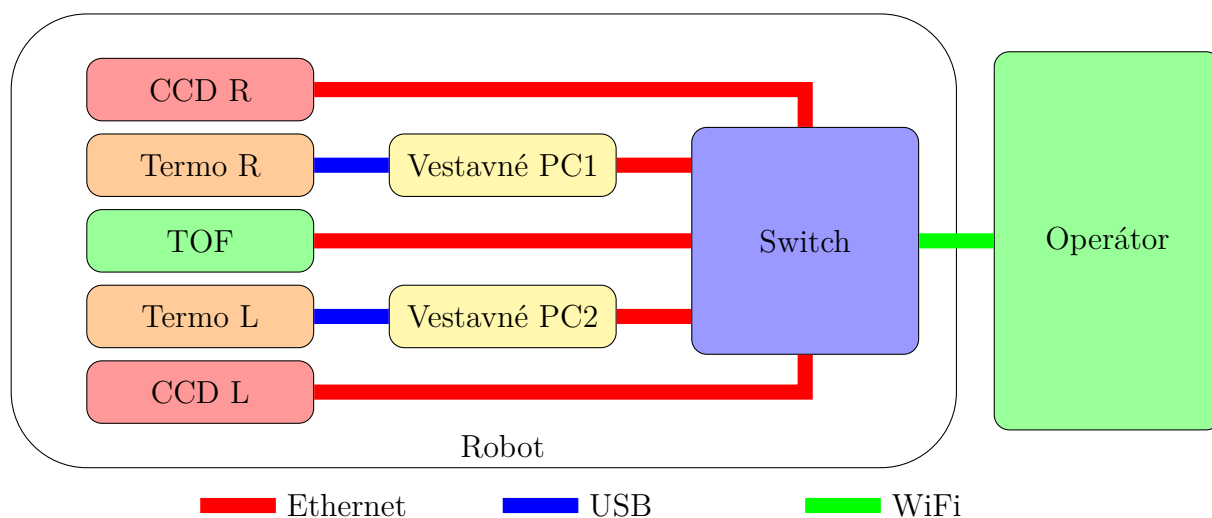
Na obrázku 3.8 je zapojení používané pro experimenty. Dvě vestavné PC zde slouží pouze jako inteligentní převodníky z USB 2.0 na gigabitový Ethernet pro levou a pravou termovizní kameru, včetně primární geometrické kalibrace. Komunikace mezi robotem a operátorskou stanicí je realizována dvěma metalickými Ethernetovými linkami, což představuje značné omezení mobility při praktickém využití, především ve vnějším prostředí. Toto zapojení bylo použito i ve všech experimentech popsaných v této práci.



Obr. 3.8: Zapojení robotu pro experimenty

Na obrázku 3.9 je zapojení pro praktické využití, kde se předpokládá využití

on-board počítačů pro mapování a datovou fúzi s tím, že k operátorovi jdou pouze fúzovaná data a postačuje podstatně pomalejší bezdrátové spojení.



Obr. 3.9: Zapojení robotu pro praktické využití

### 3.1.3 Scorpio



Obr. 3.10: Robot Scorpio

Je větší robot určený především pro práci uvnitř budov. Díky speciálnímu podvozku s tzv. dělenými pásy je schopen vystoupat standardní schody. Mechanika robotu byla zakoupena – jde o robot Dr. Robot Jaguar do kterého byla našim týmem zabudována kompletní elektronika a senzorický subsystém. Robot jsme rovněž dovybavili kamerovou hlavicí se dvěma stupni volnosti, která může být kompletně

skryta do těla robotu. Stroj je tak schopen omezeně pokračovat v misi i v případě převrácení.

### 3.1.4 Brontes



Obr. 3.11: Robot Brontes

Je malý průzkumný robot postavený na podvozku levného RC modelu typu Crawler v měřítku 1:10. Robot má nezávisle hnané obě nápravy, které navíc mohou být nezávisle natáčeny. Díky tomu robot umožňuje několik způsobů řízení a je velmi vhodný do těžkého terénu i do omezených prostor. Na zakoupený podvozek je dobudován elektronický subsystém kompatibilní se systémem CASSANDRA. Robot obsahuje i miniaturní kameru s motorickým natáčením ve dvou stupních volnosti a přísvitem pro případ nedostatku okolního světla. Celkem jsme vyrobili dva kusy tohoto stroje s mírně odlišným sensorickým a elektronickým vybavením (GPS, akcelerometry, retranslační jednotka, ...).

### 3.1.5 Perseus

Patří rovněž do třídy malých průzkumných robotů, oproti robotu Brontes však má odlišnou koncepci podvozku a umožňuje nést objemnější a těžší zátěž. Opět jde o zakoupený podvozek vybavený naší řídicí elektronikou, komunikačním systémem a sensorickým vybavením tak, aby byl kompatibilní se systémem CASSANDRA.

### 3.1.6 Uranus-HEXA

Je létající letový prostředek (tzv. dron) typu multicopter se šesti aktivními rotory. Jde celkem o třetí vývojový prototyp vytvořený na našem pracovišti. Nejdříve byl vytvořen robot Uranus-ALU, což byla kvadrokoptéra s nosností přibližně 300g.



Obr. 3.12: Robot Perseus



Obr. 3.13: Robot Uranus - HEXA

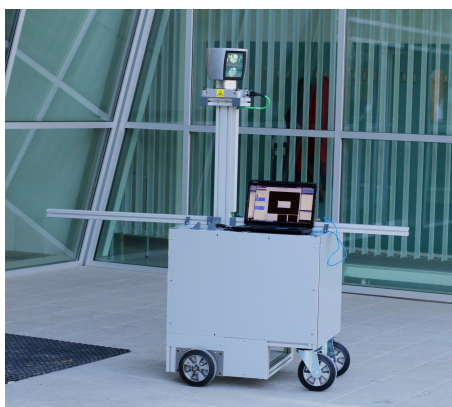


Obr. 3.14: Robot Uranus - ALU

Na tomto prvním prototypu jsme si vyzkoušeli samotný princip, způsob ovlá-

dání, a vyvinuli první řídicí desku pro multikoptéry. Uranus-ALU má konstrukci z profilů z lehkých hliníkových slitin. Dále jsme vyvinuli kavdrokoptéru s konstrukcí z karbonových tyčí. Tento materiál se nám však v průběhu provádění experimentů neosvědčil. Nyní je dokončován vývoj typu Uranus-HEXA, což je šestirotorový stroj s nosností větší než 1500g. To je dostatečné pro předpokládané využití pro průzkum a 3D měření objektů.

### 3.1.7 EnvMAP



Obr. 3.15: Robot EnvMap

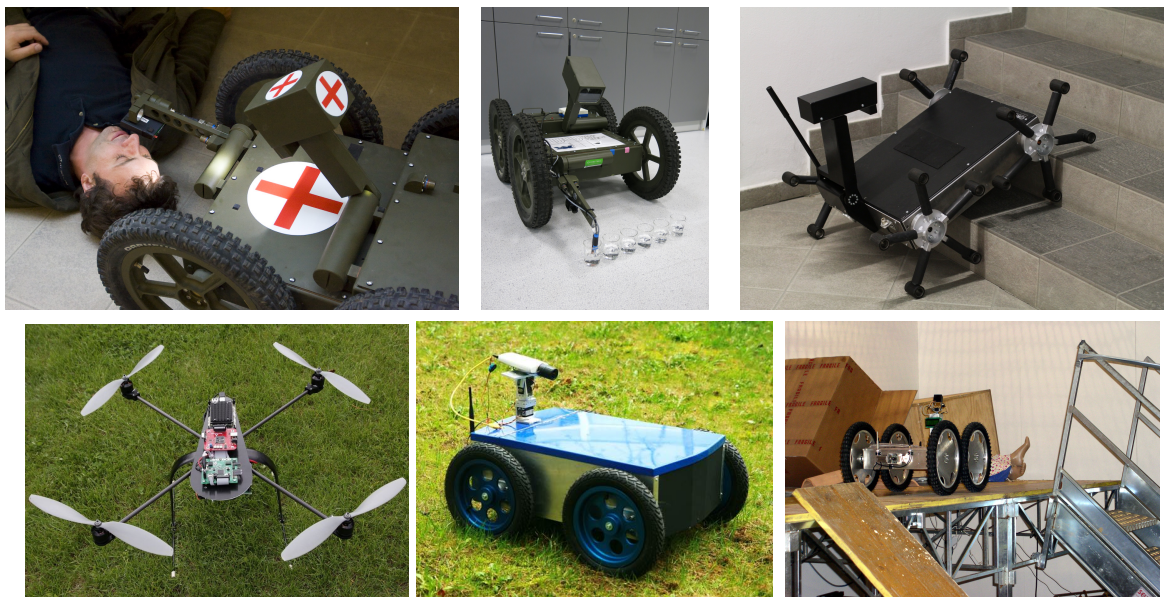
Je první vývojová verze mapovacího robotu. Vzhledem ke značné nejistotě v osazení stroje snímači a výpočetními prostředky, jsme se rozhodli stroj vytvořit jako velmi jednoduchou platformu z tzv. hliníkových profilů s Ackermanovým podvozkem. V současnosti je robot určen převážně pro mapování ve vnějším prostředí a je osazen laserovým skenerem Velodyne HDL-64, inerciálním navigačním systémem a dvoustupovou satelitní navigační jednotkou. Robot dále obsahuje dostatečně výkonný počítač pro ukládání naměřených dat.

### 3.1.8 Ostatní roboty

Do systému CASSANDRA dále může být zapojeno několik dalších robotů, které byly vytvořeny na našem pracovišti – viz. obr. 3.16

## 3.2 Ovládací stanice

Nedílnou součástí robotického systému CASSANDRA je i operátorská stanice. Základem operátorské stanice je výkonný počítač (nebo více počítačů), na kterém



Obr. 3.16: Ostatní roboty

je spuštěna některá verze operátorského systému. Dále brýle nebo helma virtuální reality s podporou stereovize, případně stereovizní monitor nebo projektor. Do operátorského kompletu rovněž patří ovládací zařízení, obvykle tzv. joypad s vibrační odezvou a komunikační modul pro metalické nebo bezdrátové propojení.

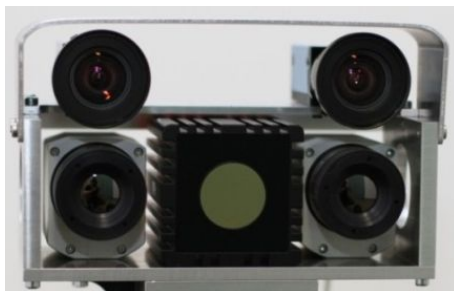
Fyzické uspořádání operátorské stanice je značně variabilní a záleží na typu mise, která je řešena. Typicky jde buď o výkonný komplet s výkonným PC nebo o přenosnou stanici s notebookem. Pro skutečně terénní práci jsme vyvinuli i přenosnou stanici, u které je veškerá elektronika integrována v batohu. (viz obr. 3.17 vpravo)



Obr. 3.17: Ovládání systému CASSANDRA

## 4 TVORBA MULTISPEKTRÁLNÍCH MAP

Pro měření multispektrálních dat byl použit robot Orpheus-X3 vybavený multispektrální stereovizní hlavicí, umístěnou na tříosém manipulačním rameni. Podrobný popis robotu již byl uveden v kap. 3.1.2.



Obr. 4.1: Hlavice Orpheus-X3

Robot, vybavený touto hlavicí je schopen vytvořit trojrozměrný obraz okolí robotu. Omezení pozorovacího úhlu jednotlivých kamer je možné rozšířit za pomoci manipulačních motorů.

Pro zachování přesnosti mapování je nutné zajistit, aby přesnost manipulace s kamerami byla vyšší, než je rozlišení nejjemnějšího snímače. Bylo ověřeno experimentem, že regulační smyčka nastavování motorů je sice nepřesná díky hysterezi, avšak měření okamžité absolutní polohy je velmi přesné (řádově s přesností na tisícinu stupně). Během experimentu bylo též zjištěno, že přejezd do nové polohy včetně ustálení na poloze trvá regulátoru motorů dobu 1 sekundu. Tento čas je nutné též zohlednit při měření.

Se zohledněním výše uvedených zjištění byl navržen systém měření, ve kterém robot stojící v definované poloze periodicky sejme data ze všech senzorů, pokaždé při jiném natočení hlavice. Systém zajišťuje, že řízený pohyb hlavice pokryje během 36 měření celou kulovou plochu okolí robotu, s patřičnými překryvy snímků senzorů. Doba mezi pohybem hlavice a měřením byla zvolena o hodnotě 2 sekundy, aby se hlavice stihla po pohybu ustálit na nastavené hodnotě.

Celý systém je tak schopen měřit v celém rozsahu kulové plochy teplotu, barvu i vzdálenost. Zanášení těchto informací do mapy [26] popisuje následující kapitola.

### 4.1 Reprezentace mřížky evidence v paměti

Jak již bylo uvedeno v kapitole 2.3, mřížka evidence umožňuje uložení více informací v každém voxelu mapy.

V případě robotických map použitých v této práci je nutné ukládat informaci o obsazenosti daného voxelu (log odds ratio), o teplotě daného voxelu a o barvě (RGB).

Integrabilita obsazenosti je daná definicí, integrability teploty je dosaženo za pomoci jednoduchého průměrovacího filtru uvedeného v rov. 4.1, kde  $T_t$  odpovídá hodnotě teploty uložené v buňce před příchodem měření,  $T_{t+1}$  hodnotě v buňce po příchodu měření,  $T_m$  je měřená hodnota teploty a  $K$  je konstanta filtru která určuje rychlost konvergence teploty k nové hodnotě. Během měření byla použita konstanta  $K=4$ .

$$T_{t+1} = \frac{T_m + KT_t}{1 + K} \quad (4.1)$$

Integrabilitu teploty je velmi obtížné zajistit. Během měření byla integrabilita zajištěna převodem z RGB modelu do HSV modelu, integrací průměrem po složkách H,S,V a zpětnou konverzí do RGB pro uložení do mřížky evidence.

Jeden voxel tedy má následující strukturu, uvedenou v tabulce 4.1

Tab. 4.1: Struktura jednoho voxelu mřížky evidence

Offset	Délka	Typ	Název	Popis
0	4B	float	Probability	Log odds hodnota pravděpodobnosti obsazenosti.
4	1B	int	Temperature	Teplota daného bodu (0 – 255°C)
5	3B	int	Color	RGB hodnota barvy daného bodu.

Budoucí verze mřížky evidence již počítají s ukládáním hodnot HSV modelu, čímž odpadne náročný výpočet převodu mezi modely.

## 4.2 Reprezentace souboru s Octree mapou

Jméno souboru obsahujícího mapový podklad je vytvořené tak, aby byla jednoznačně identifikována pozice každého souboru v globální mapě. Jméno obsahuje tři po sobě jdoucí souřadnice (indexy) x,y,z uvozené znaménkem a prodloužené na dvě místa. Tímto způsobem lze uložit globální mapu o velikosti hrany 199 indexů. To za předpokladu jedné buňky mapy o hraně 5.12m definuje uložitelný prostor globální mapy o délce hrany nepatrně větší než 1km, což by mělo postačovat pro veškeré naše experimenty. Uvození znaménka před index umožňuje tento formát rozšířit i na větší mapy s možností zpětného načítání (např při zvětšení o jednu cifru indexu dojde ke zvětšení adresového prostoru mapy na velikost hrany 20km). Například jméno souboru +03-01+00.og3 udává data v buňce se souřadnicemi x=3, y=-1, z=0.

Každý soubor s mapou je ukládán ve formátu Little Endian a obsahuje hlavičku souboru následovanou vlastními daty mapy. Formát souboru og3 obsahuje 32 bajtů dlouhou hlavičku obsahující následující, po sobě jdoucí údaje:

Tab. 4.2: Hlavička

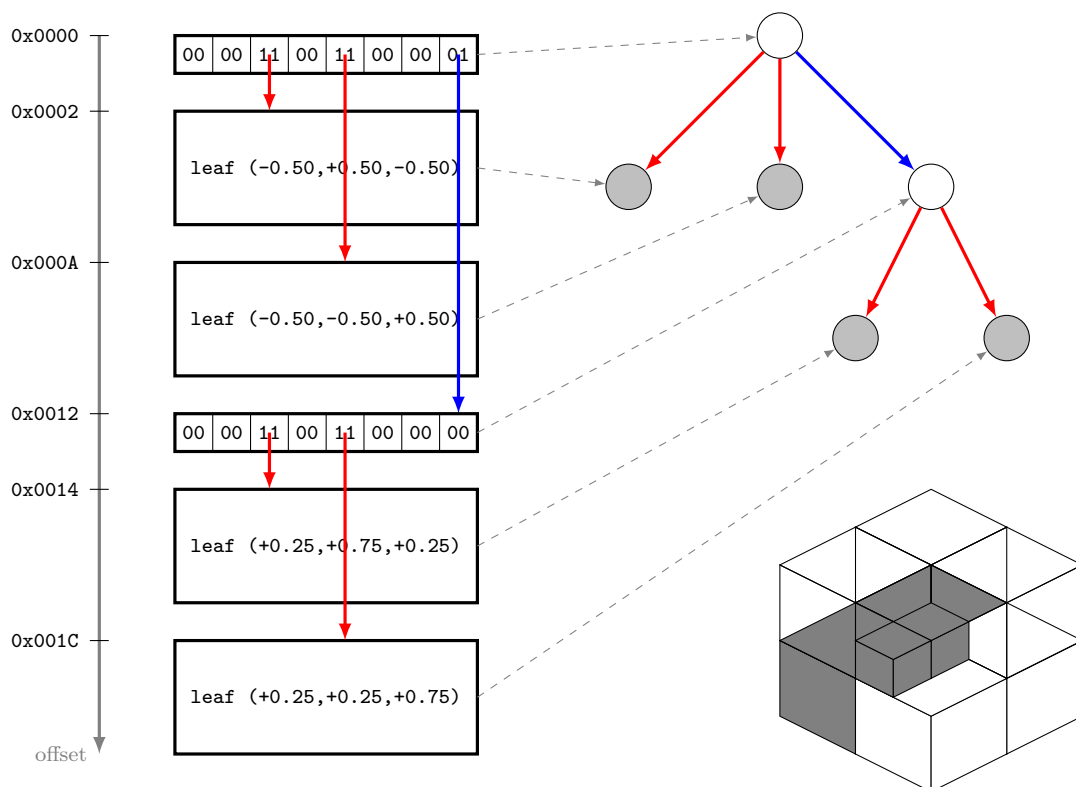
Offset	Délka	Typ	Název	Popis
0	4B	int	Magic	=0xCAFEF00E pro data komprimovaná pomocí Octree.
4	4B	int	HeaderLength	Délka hlavičky, =32 pro tuto verzi.
8	4B	int	GridLength	Délka hrany rastru mřížky v pixelech.
12	4B	float	Size	Velikost lokální mapy, =5.12m pro tuto verzi.
16	4B	float	Resolution	Rozlišení jednoho pixelu mapy [m].
20	4B	float	Origin X	Tři souřadnice levého dolního rohu mapy.
24	4B	float	Origin Y	
28	4B	float	Origin Z	

Vlastní data mapy jsou serializována do souboru s pomocí komprese Octree (pro pole hlavičky Magic s hodnotou =0xCAFEF00E).

U každé větve stromu je uloženo 16 bitové číslo, obsahující 8 dvoubitových hodnot definujících jednotlivé potomky větve. Ty určují, zdali je uložen podstrom (bity odpovídají 01), koncový list (bity 10), nebo prázdný prostor (bity 00). Způsob uložení je zřejmý z následujícího obrázku:

Tab. 4.3: List

Offset	Délka	Typ	Název	Popis
0	4B	float	Probability	Hodnota pravděpodobnosti obsazenosti dané buňky.
4	1B	int	Temperature	Uložená teplota (0 - 255°C).
5	3B	int	Color	Barva bodu.



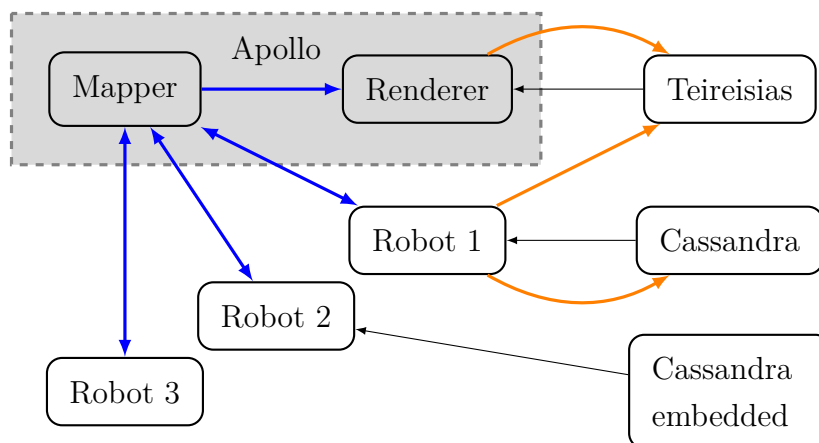
Obr. 4.2: Formát serializovaného Octree, vlevo binární datový stream uložený v souboru, vpravo nahoře síťová reprezentace dat, vpravo dole výsledná rekonstruovaná mapa

## 5 DISTRIBUCE MAPY MEZI ROBOTY

Mapu, vytvořenou jedním robotem s pomocí metod popsanych v předcházející kapitole je nutné distribuovat mezi ostatní roboty, aby i tyto roboty měly informace o měřeních. Systém distribuce map musí zajistit validní začlenění všech měření do globální mapy, a tuto pak distribuovat zpět robotům. Specifickým problémem je omezený datový tok v důsledku použití bezdrátové komunikace.

### 5.1 Začlenění map do robotického systému

Robotický systém sestává z několika částí, které spolu spolupracují (viz obrázek 5.1). Sada mobilních robotů doplněná o mobilní ovládací stanici „*Cassandra embedded*“ tvoří základní hardwarovou část systému.



Obr. 5.1: Spojení mezi aplikacemi

Aplikace „*Cassandra*“ slouží pro vizuální teleprezenční ovládání robotu, dokáže robot řídit i zpracovávat obrazová data z něj. Tato aplikace je určena zejména pro průzkum s použitím pouze jedné kamery.

Aplikace „*Teireisias*“ slouží jako platforma pro napojení na stereovizní vidění, dokáže pouze stereovizně zobrazovat obrazová data z robotu Orpheus-X3 (jako jediný je vybaven stereovizním párem kamer), a z virtuálního robotu mapovací aplikace. Aplikace v současné době provádí i fúzi dat CCD kamery a termokamery s pomocí 3D kamery. Vše v reálném čase. V současné době aplikace umožňuje vzdáleně ovládat pouze virtuální robot, v budoucnu se počítá s plným rozhraním kompatibilním s ostatními roboty.

Aplikace „*Apollo*“ je základním pilířem pro mapovací funkce systému. V současnosti je tato aplikace vnitřně sestavena ze dvou částí, které budou v budoucnosti

odděleny do separátních aplikací z důvodu lepšího využití dedikovaných výpočetních prostředků a možnosti spuštění více instancí části programu. První část, tzv. „*Renderer*“ zprostředkovává rozhraní virtuálního robotu aplikacím Cassandra a Teireisias. Spojení s aplikací Cassandra není v okamžiku psaní práce dokončeno. *Renderer* tedy prezentuje mapové podklady stejně, jako by se jednalo o reálný robot. Po oddělení této části z aplikace Apollo bude možné vytvořit více virtuálních robotů, které zprostředkují náhled do mapy. Druhou částí programu Apollo je vlastní mapovací jádro, tzv. „*Mapper*“. Tato část programu je zodpovědná za integraci naměřených mapových podkladů do globální mapy, distribuci změn globální mapy a vlastní úložiště pro globální mapu. Po vyjmutí z aplikace Apollo bude možné tuto instanci spustit na počítači s nižším výkonem, zejména bez výkonné grafické karty. To umožní spustit tuto instanci přímo na dedikovaném robotu EnvMap, jehož úkolem je podpora mapovacích procesů.

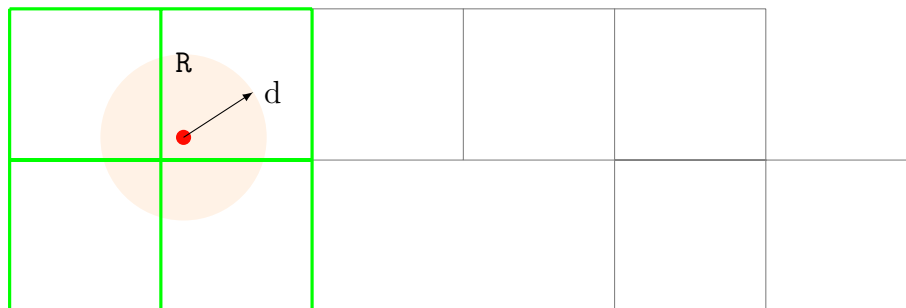
V dalších podkapitolách se zaměřím na popis výměny dat mezi Mapperem a roboty, v následující kapitole bude popsána funkce *Rendereru*.

## 5.2 Systém blokového dělení globální mapy

Základním požadavkem na mapovací subsystém robotu je jeho škálovatelnost při zachování přesnosti. Tyto parametry je nutné volit s přihlédnutím k použití mapy pro vizualizaci i pro strojové vnímání prostředí robotu.

Venkovní prostory jsou dobře popisovány výškovou mapou (leveled map) [14] zatímco prostory uvnitř budov je výhodné popisovat za pomoci mapy obsazenosti (Occupancy grid) [5]. Tento rozdílný formát dat vede na rozdílné, vzájemně nekompatibilní formáty uložení mapy. Za předpokladu, kdy není dopředu specifikováno, zdali se robot bude pohybovat uvnitř nebo vně budov, nebo dokonce plynule přecházet mezi těmito oblastmi je nutné vytvořit hybridní mapu, která umožňuje současné uložení těchto různých mapových podkladů do jednoho heterogenního úložiště.

V této práci byl zvolen systém, kdy se globální mapa rozdělí na očíslované oblasti s pevně danou délkou hrany. Každá z těchto oblastí může obsahovat žádný, jeden nebo více mapových podkladů současně uložených v nezávislých souborech. Tím je zajištěna škálovatelnost systému za předpokladu, že se velikost mapy dynamicky mění. Za předpokladu, že je zvolena délka hrany lokální oblasti dostatečně velká vzhledem k maximálnímu možnému měření robotu, pak není nutné, aby byla mapa uložena v paměti celá, stačí zpracovávat pouze nejbližší oblasti v okolí robotu, což vede k výrazné redukci zpracovávaných dat, a tím pádem ke zvýšení rychlosti odezvy tohoto systému na měření.



Obr. 5.2: Dělení globální mapy na oblasti, se zvýrazněním zpracovávaného okolí robotu R, se senzorem s maximálním dosahem měření  $d$  zelenou barvou.

## 5.3 Interpretace dat uvnitř bloku

Každý blok dat umožňuje mít uloženo několik typů mapy právě pro možnost koexistence externích a interních měření. Pro srovnání jsou zde uvedeny některé typy map, které je do daného bloku možné uložit (vždy pouze jednu od daného typu)

### 5.3.1 Výšková mapa (Leveled Map)

Tato mapa obsahuje výšku terénu rozprostřenou na mřížce. Pro vlastní formát uložení dat se s výhodou dá použít systém bezztrátové komprese Quad-tree, který však vyžaduje dělení rastru s krokem  $2^N$  jednotek.

### 5.3.2 Mřížka obsazenosti (Occupancy Grid)

Tato mapa rozděluje celý prostor bloku na ekvidistantní mříž, ve které je uložena pouze informace o přítomnosti překážek (obsazeno/volno). Lze komprimovat za pomoci Octree. Tento typ mapy vyžaduje pro svou funkci model senzoru laserového skeneru.

### 5.3.3 Mřížka evidence (Evidence Grid)

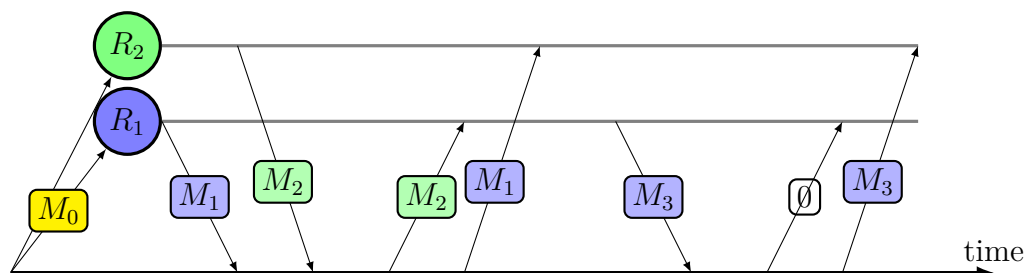
Tato mapa je rozšířením předchozího typu o možnost uložení více informací do každé buňky, například teploty či barvy. Lze komprimovat za pomoci Octree. Vzhledem k heterogennímu obsahu dat a nutnosti udržet komutativitu operací během integrace měření není nutné používat model laserového skeneru. Tento typ uložených dat byl vybrán pro tuto práci.

### 5.3.4 Mapa význačných bodů (Feature Map)

Tato mapa obsahuje pozice význačných bodů, které robot může snadno identifikovat, a na základě těchto bodů se následně navigovat, případně obohatit výstup operátorovi o další užitečné informace. Příkladem jsou například pozice vysílačů sítě, pozice síťových zásuvek, pozice dveří, označení místností, průjezdní koridory, apod. Obsah této mapy je mimo rámec této práce.

## 5.4 Systém distribuce mapy mezi více robotů

Distribuce mapy byla inspirována verzovacím systémem GIT [27] [4], avšak s přihlédnutím k předpokladu, že je třeba znát pouze poslední aktuální verzi, bez nutnosti znát celou historii mapování. Tento předpoklad výrazným způsobem redukuje množství zpracovávaných dat a urychlí operace.



Obr. 5.3: Popis distribuce mapy mezi roboty

Celý proces distribuce znázorňuje obrázek 5.3. Mapový server nejprve distribuuje základní mapu  $M_0$  všem připojeným robotům. Následně přijme sadu měření  $M_i$  od robotu  $R_i$ . Mapový subsystém následně spočítá, že robotu  $R_2$  chybí sada měření  $M_1$  a robotu  $R_1$  chybí sada měření  $M_2$ . V části synchronizace se pokusí dodat chybějící data každému robotu.

Tento systém zaručí, že všechny roboty budou mít informaci o všech měřeních do systému odeslaných prostřednictvím robotů.

V obecném pohledu na tento systém lze pohlížet na jednotlivá měření  $M_i$  jako na agregované soubory změn mapy, identicky jako GIT pohlíží na agregované difference souborů. Při hlubším zkoumání se objeví nutnost online výpočtu difference mezi hlavní mapou, a starší verzí této mapy, což lze jednoduše zajistit fixováním dané mapy v jiné větvi grafu úplně stejným způsobem, jakým GIT provádí větvení repozitáře. Rozdílem však je, že mapovací subsystém nepotřebuje kompletní historická data, stačí pouze poslední verze v hlavní i ve všech podružných větvích.

S výhodou lze využít faktu, že difference map budou z velké části obsahovat prázdná místa, a tudíž budou mít vysoký kompresní poměr při kompresi která se snaží nevyplněná místa neukládat (Octree).

Počítání diferencí mapy vyžaduje mít uloženou starší kopii mapy, což vede k rapidnímu nárůstu vyžadované paměti u obou systémů, jak na robotu tak mapového serveru. Vzhledem k tomu, že roboty jsou většinou postaveny na mobilním a tedy slabším hardwaru, je možné nevyužívat výpočtu diferencí na robotu ale až v mapovacím systému. Ušetří se tím lokální kopie mapy na robotu, ale lze předpokládat dramatický nárůst datové náročnosti na komunikaci.

Z těchto důvodů je v následujících kapitolách uvedena základní sada nástrojů pro robot s nedostatkem paměti, a pokročilá sada nástrojů pro robot s dostatečným množstvím využitelné paměti.

### 5.4.1 Mapový server

Mapový server je program spuštěný na dedikovaném počítači, který zpřístupňuje obsah globální mapy ostatním robotům a umožňuje přijímat a integrovat měření v mapě od ostatních robotů.

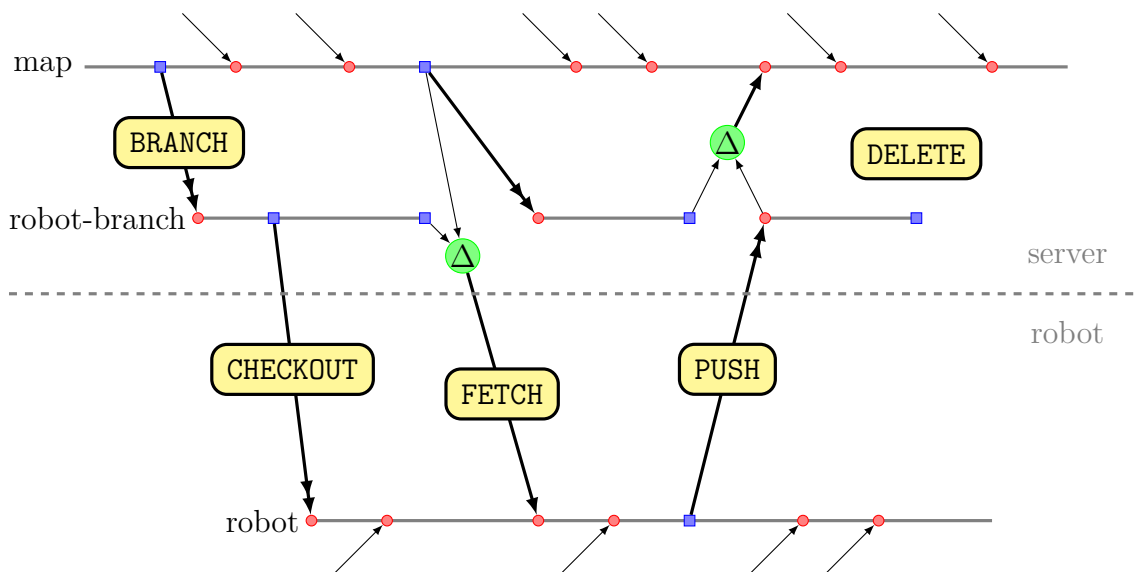
Celá komunikace robotu s mapovým serverem je založena na klasickém protokolu HTTP (standards [6] [7], aktuální [8] [9]), který umožňuje snadno transportovat jednotlivé soubory. Komunikace probíhá na portu 8081 pro odlišení od standardního HTTP protokolu. Jednotlivé operace nad mapou jsou vybírány pomocí požadavku serveru. Cesta k souboru přímo určuje mapu, případně větev mapy. Výsledný dotaz tedy vpadá následovně:

```
[request] http://[server]:8081/[path]
```

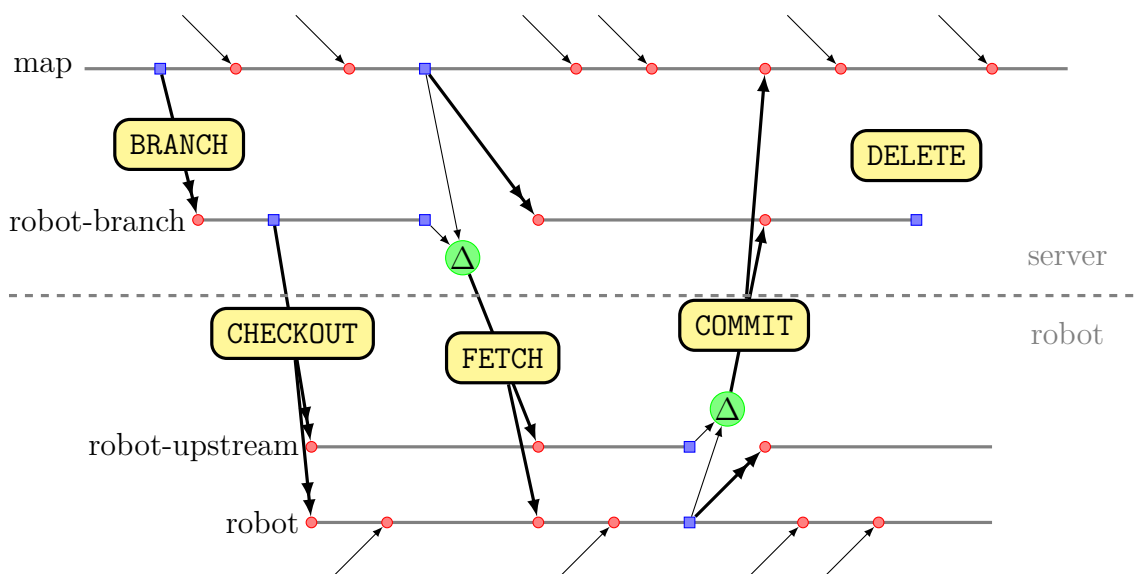
Činnost jednotlivých operací je graficky znázorněna na Obr. 5.4 a Obr. 5.5. Na těchto obrázcích je čas veden zleva doprava. Je zde znázorněn jeden cyklus založení robotu, přenos mapy do robotu, aktualizace mapy v robotu i v mapovém subsystému a ukončení činnosti robotu.

Jednotlivé atomické operace nad stromem jsou shrnuty v následujícím seznamu:

- BRANCH** vytvoří větev z dat hlavní mapy, kvůli trasování změn, které robot již má. Operace slouží k zaevidování nového robotu do mapovacího systému.
- DELETE** odstraní větve robotu ze systému se všemi uloženými mapovými podklady. Operace slouží k odregistrování robotu z mapovacího systému a ukončení jeho činnosti.
- CHECKOUT** stáhne z mapovacího systému celou mapu uloženou ve vlastní větvi robotu. Robot všechna předchozí data mapy zahodí a použije tento čistý mapový podklad. Stažená data mohou být neaktuální vůči hlavní mapě robotu. Operace slouží k počátečnímu stažení mapových podkladů do robotu.



Obr. 5.4: Základní operace pro distribuci mapy robotu



Obr. 5.5: Pokročilé operace pro distribuci mapy robotu

**FETCH** synchronizuje větev robotu s daty hlavní mapy a odešle robotu změny které byly provedeny mezitím v hlavní mapě jinými roboty. Operace slouží k aktualizaci mapy robotu o změny, integrované ostatními roboty do hlavní mapy.

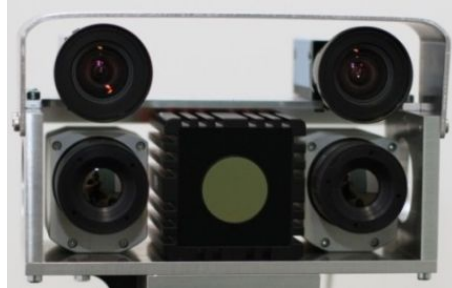
**PUSH** odešle celou mapu robotu do mapovacího systému. Mapovací systém sám spočítá rozdíly, a těmito novými informacemi obohatí hlavní mapu v systému. Operace slouží k odesílání měření robotu do mapovacího systému.

**COMMIT** odešle rozdíly v mapě robotu do mapovacího systému. Tato operace vyžaduje,

aby robot dokázal spočítat rozdíl mezi mapou která mu byla dodána a mapou do které robot zakomponoval nová měření. To vyžaduje dvojnásobné nároky na paměť a proto tato metoda je implementovaná jen pro roboty s dostatkem paměti. Metoda významně redukuje počet přenášených dat po síti.

## 6 ZOBRAZENÍ A PREZENTACE DAT

V této kapitole jsou popsány jednotlivé dílčí optimalizace, provedené v rámci této disertační práce, související s vykreslováním nasbíraných dat. Tato optimalizace je nutná vzhledem k předpokládanému použití celého systému v reálném čase.



Obr. 6.1: Hlavice robotu

Tab. 6.1: Parametry kamer

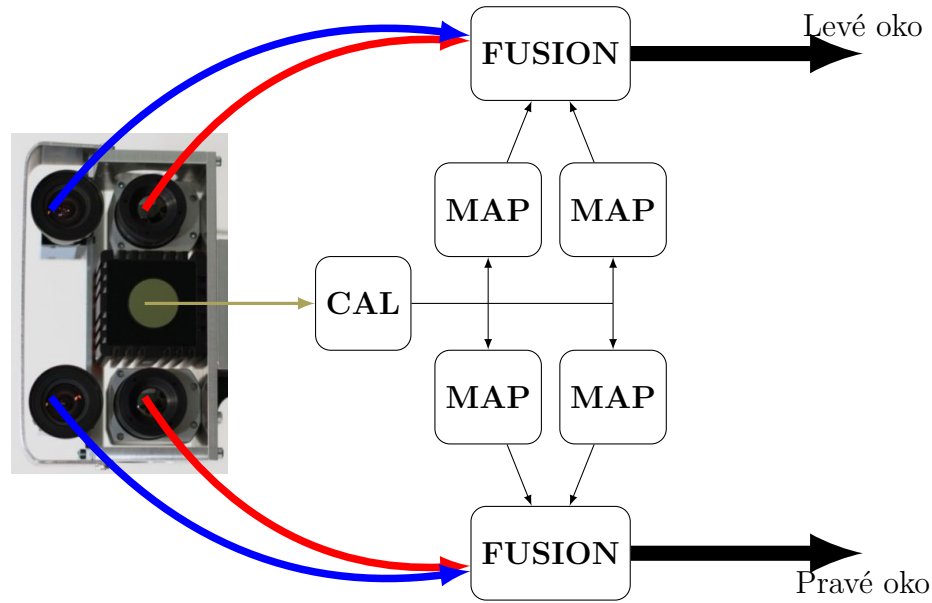
Kamera	Rozlišení	Rychlost	Bitů/pixel	Obrazový tok
TheImagingSource DFK23G445 (2x)	1280x960	30Hz	8	295 Mbps
MicroEpsilon TIM 450 (2x)	382x288	30Hz	16	53 Mbps
Mesa Imaging SR4000 (1x)	176x144	30Hz	64	48 Mbps

Z tabulky 6.1 je patrné, že celkový datový tok všech obrazových dat (bez režie IP protokolu) ze sensorické hlavičky dosahuje hodnoty 744 Mbps. Se započítáním režie síťových vrstev a ostatní datové komunikace s robotem je tato hodnota 916 Mbps. Tyto hodnoty dokazují nutnost masivních optimalizací na výpočetní a zobrazovací straně.

### 6.1 Fúzování video dat

Celý proces kalibrace a fúzování grafických dat byl dokumentován v článkách [3], [18] a v [19].

Rovnice uvedené v těchto článkách byly nadále optimalizovány pro masivní paralelizaci pro získání vyšších přenosových rychlostí a výkonů. Celý proces fúze dat názorně popisuje obrázek 6.2. Jednotlivé bloky budou popsány v následujících podkapitolách.



Obr. 6.2: Fúzování video dat

### 6.1.1 Kalibrace paprsků TOF kamery (CAL)

Vstupem do kalibrační metody je kalibrovaný vektor směrnic paprsků  $v_i$  pro každý pixel TOF dálkoměrné kamery. Souřadnice  $x_i$  a  $y_i$  odpovídají definici kalibrace senzoru popsané v článku [18].

$$v_i = \frac{\begin{pmatrix} x_i & y_i & f \end{pmatrix}^T}{\left| \begin{pmatrix} x_i & y_i & f \end{pmatrix}^T \right|} \quad (6.1)$$

Účelem tohoto modulu je převést surová dálkoměrná data ze vzdáleností do souřadného systému spojeného s dálkoměrnou kamerou. To se děje oproti citovaným článkům pouhým pronásobením kalibračního vektoru  $v_i$  se změřenou vzdáleností  $d_i$  pro každý pixel dle rovnice 6.2.

$$p_i = d_i v_i \quad (6.2)$$

Výhodou tohoto uskupení je, že se násobí konstantní vektor změřenou vzdáleností, a pro následné optimalizace lze tento vztah snadno separovat.

### 6.1.2 Mapování paprsků TOF na CCD kameru resp. termokameru (MAP)

Účelem tohoto procesu je přenést jednotlivé změřené body dálkoměrem v 3D prostoru do souřadného prostoru dané kamery. To se děje za pomoci homogenní trans-

formace, jejíž jednotlivé prvky jsou součástí kalibračních parametrů pro každou kameru.

$$p_{ai} = \begin{pmatrix} x_{ai} \\ y_{ai} \end{pmatrix} = Hp_i \quad (6.3)$$

Po provedení homogenního násobení je provedeno projektivní dělení pro převod z třírozměrného systému do dvojrozměrného souřadného systému obrazu kamery, současně s aplikací ohniskové vzdálenosti  $f$  a korekce pozice centrálního pixelu  $(x_0, y_0)$ . Vztah  $p_{bi} - S$  určuje souřadnice bodu vztažené vůči středu obrazu. Kvadratický profil vyšel dle [18] nejlépe, proto jsem jej použil.

$$p_{bi} = \begin{pmatrix} x_{bi} \\ y_{bi} \end{pmatrix} = \begin{pmatrix} x_{ai}/z_{ai} \\ y_{ai}/z_{ai} \end{pmatrix} f + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \quad (6.4)$$

Dále jsou souřadnice kalibrovány na radiální zkreslení kamery za pomoci následujícího vztahu, kde  $S = (W/2, H/2)$  určuje střed obrazu v pixelech:

$$p_{ci} = \begin{pmatrix} x_{ci} \\ y_{ci} \end{pmatrix} = p_{bi} + \frac{k_1}{f^2} |p_{bi} - S|^2 (p_{bi} - S) \quad (6.5)$$

Jako poslední krok je testováno, zdali výsledné souřadnice bodu  $x_{ci}$  změřeného dálkoměrem se nachází na obraze dané kamery pro kterou se počítalo mapování.

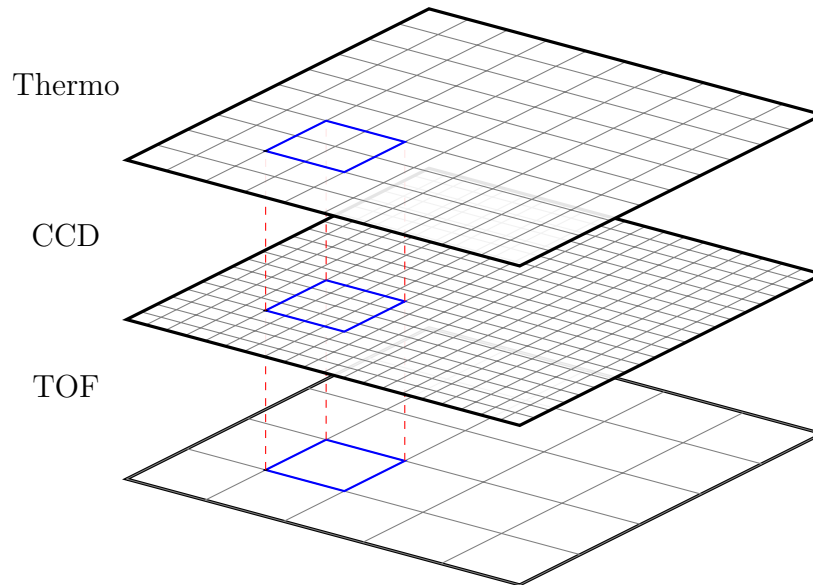
Pro další výpočty a fúzi se uloží výsledné souřadnice  $p_{ci}$  společně se vzdáleností bodu po homogenní transformaci  $z_i/w_i$ .

Jelikož celý postup je platný pro každý bod obrazu, byl celý algoritmus výpočtu optimalizován za pomoci vícevláknového zpracování. Touto optimalizací se dosáhlo 56% zrychlení běhu algoritmu při testování na čtyřjádrové počítači PC1 a 75% zrychlení na osmijádrové počítači PC2.

### 6.1.3 Výsledná fúze dat termokamery a CCD kamery (FUSION)

Vstupem této funkce jsou vektory odpovídajících pozic pixelů a vzdáleností spočítané předchozími metodami. Z poměrů rozlišení jednotlivých zdrojů dat a srovnání jejich zorného pole je zřejmé, že na každý paprsek TOF kamery připadá plocha cca 2x2 pixelů termokamery a cca 8x8 pixelů CCD kamery. Vše ilustruje obrázek 6.3

Výsledná fúze probíhá tak, že pro každý z paprsků TOF kamery jsou vyčteny příslušné čtyři teploty v okolí spočítaného namapovaného vektoru, a tyto čtyři teploty po přemapování na barvy jsou za pomoci metod AlphaBlendingu spojeny s příslušnými 16 pixely v okolí namapovaného vektoru CCD kamery.



Obr. 6.3: Ideové schéma fúze dat

Mapování teploty na barvu slouží k programovému zvýraznění zajímavých částí obrazu či jejich úplného potlačení. Například pro označení osob, které mají teplotu v okolí  $37\text{ }^{\circ}\text{C}$  je vytvořen profil alfakanálu, kdy kolem této teploty je barva termokamery neprůsvitná, obarvená výraznou barvou, a ostatní teploty jsou průsvitné, zobrazující podklad z obyčejné kamery.

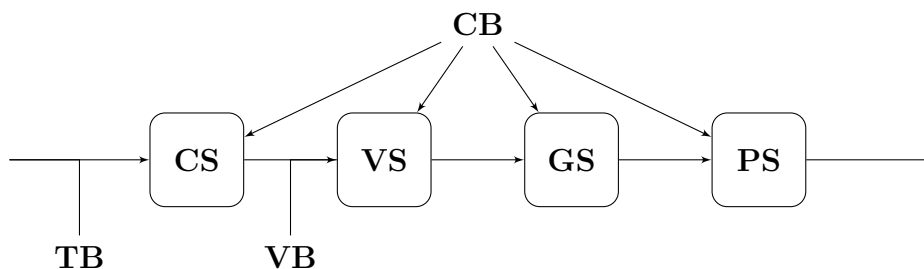
Mapovaná barva z teplotního profilu je nezávisle potlačitelná, a uživatel má možnost vybírat zdali chce vidět pouze obraz termokamery, ccd kamery, nebo fúzi obou, nebo lineární mix všech možností.

## 6.2 Vykreslování mapy

Pro vykreslování bylo použito karty nVidia plně kompatibilní s DirectX 11. Díky použití rozhraní DirectX 11 bylo možné spouštět určité části vykreslovacích algoritmů přímo na grafické kartě, čímž se dosáhlo vysokých výkonů při vykreslování.

Jednotlivé fáze grafické pipeline jsou popsány v následujících kapitolách:

- CS** Compute shader (filtrace a normalizace dat)
- VS** Vertex shader (Transformace souřadnic)
- GS** Geometry shader (Tvorba zobrazovacích primitiv)
- PS** Pixel shader (Osvětlení scény)
- TB** Texture buffer (Obsahuje mřížku obsazenosti)
- VB** Vertex buffer (Obsahuje souřadnice obsazených bodů)
- CB** Constant buffer (Obsahuje předpočítané mezivýsledky)



Obr. 6.4: DirectX 11.0 pipeline

### 6.2.1 Compute shader (CS)

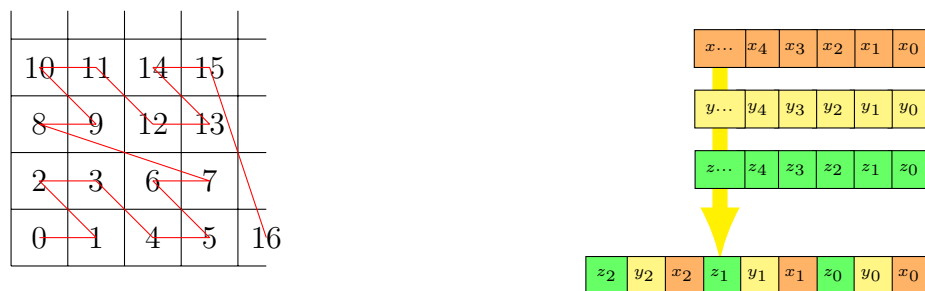
Účelem tohoto mikrokódu, běžícího na grafické kartě je filtrace a normalizace vstupních dat.

Vstupními daty tohoto mikrobloku je matice pravděpodobností obsazenosti mapy a teploty, uložená jako jednoduchý Texture3D buffer, obsahující hodnotu pravděpodobnosti i teploty ve dvou hodnotách typu float pro každou buňku. Tento formát umožňuje grafické kartě velmi rychlý přístup k obsahu na základě obvyčejného smplování textury. Jakékoliv kompresované formáty typu octree je nutné v současnosti pro zobrazení nejprve rozbalit do paměti. V současnosti je vyvíjen nový Compute Shader, který dokáže číst přímo komprimovaná data ve formátu Octree a generovat z něj data pro Vertex Shader. V okamžiku psaní práce však není stabilní a proto není součástí práce.

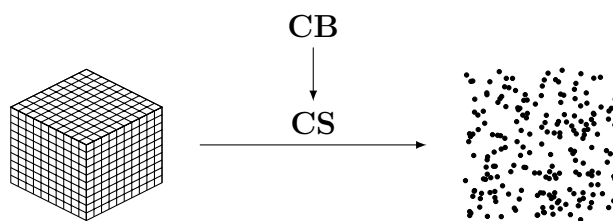
Během vývoje byl testován i systém fraktálového uložení mapy. Celý systém fraktálového uložení vychází z principu jednoduchého přeskládání bitů adresních souřadnic tří os  $(x_n..x_0, y_n..y_0, z_n..z_0)$  na lineární adresu buňky  $x_n y_n z_n .. x_0 y_0 z_0$ . Princip bitového mixování i výsledný Mortonův fraktál je uveden na obrázku 6.5. Vzhledem k použití jazyka C#, který v současné době neumožňuje využít instrukce SIMD však nebylo dosaženo lepších výsledků, protože algoritmus výpočtu adresy s pomocí obvyčejných instrukcí je výpočetně a časově náročnější než náhodný přístup do grafické paměti. V práci tudíž není použit, ale je připraven pro jeho použití.

Program Compute Shaderu je programově rozdělen do 8x8x8 vláken, kdy každé z vláken obsluhuje jeden bod zdrojové mapy. Touto masivní paralelizací se docílilo masivního zrychlení výpočtů nad mapou. Každé vlákno nejprve načte zdrojová data a zkontroluje zdali je daný bod obsazen. Pokud není obsazen, vlákno končí. Touto filtrací se opět docílí obrovského nárůstu výpočetního výkonu, protože ve většině případů je mapa prázdná.

Posléze mikrokód za pomoci indexace v textuře teplotní škály zjistí barvu odpovídající dané teplotě bodu a na základě algoritmu "Premultiplied alpha" a uživatelem



Obr. 6.5: Fraktálové uložení mapy

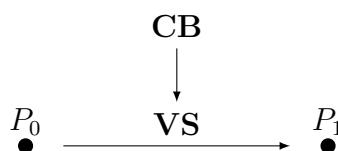


Obr. 6.6: Činnost Compute Shaderu

požadované průhlednosti teplotního profilu vypočítá výslednou barvu naměřeného bodu. Dále mikrokód prozkoumá sousedící body mapy a zkontroluje jejich obsazenost. Získaná hodnota se dále používá pro optimalizaci zobrazení v Geometry Shaderu.

Spočítanou barvu bodu společně se souřadnicemi  $x, y, z$  a informací o obsazenosti sousedních bodů mikrokód ukládá do VertexBufferu pro použití dalšími shadery v grafickém řetězci.

### 6.2.2 Vertex Shader (VS)



Obr. 6.7: Činnost Vertex Shaderu

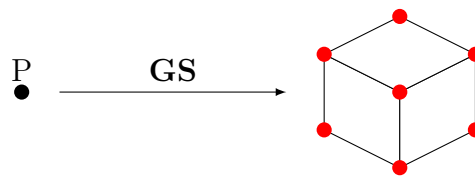
Tento shader slouží k načtení hodnot z VertexBufferu a jejich transformaci do souřadnic obrazovky. Transformace souřadnic se provádí za pomoci standardní ma-

tice homogenní transformace s obsahem matic Model a View. Projektivní transformace není v tomto kroku použita.

$$P_1 = MP_0 \quad (6.6)$$

### 6.2.3 Geometry Shader (GS)

Tento shader každý bod získaný z VS rozšíří a vytvoří z něj 6 stěnnou krychli. Shader výslednou krychli optimalizuje dle informací z CS o sousedících obsazených bodech, a nezobrazuje stěny krychle těsně sousedící s další obsazenou krychlí. Výsledkem je tedy generování 0 až 24 vertexů pro každý vstupní vertex.



Obr. 6.8: Činnost Geometry Shaderu

GS dále spočítá pro každou stěnu krychle korektní normálové vektory důležité pro správný výpočet osvětlení. Nakonec výslednou krychli transformuje projektivní transformací do souřadného systému obrazovky. S výhodou lze použít distributivní pravidlo maticového násobení ( $M(x + n) = Mx + Mn$ ) pro výpočet jednotlivých bodů krychle i pro výpočet normál. Transformované normály jsou pro zrychlení výpočtů předpřipraveny v ConstantBufferu (CB).

### 6.2.4 Pixel Shader (PS)

PS přejímá po rasterizaci informace z GS, a na jejich základě, a na základě informací o světlech, pozici pozorovatele a objektu předané v ConstantBufferu (CB) spočítá phongův model pro jas objektu.

Za pomoci informace o osvětlení a barvě bodu tento bod obarví požadovanou nasvětlenou barvou a výslednou barvu pixelu odešle na výstup.

## 7 DOSAŽENÉ VÝSLEDKY

Výsledky, uvedené v této kapitole jsou změřeny na sadě dat, která byla kompletně vytvořena robotem Orpheus-X3. Všechna zpracovávaná data od fáze skenování do mračna bodů a ukládání snímků kamer, přes následnou integraci změřených dat do mřížky Evidence Grid, kompresi této mřížky do formátu Octree, přenos Octree mapy mezi počítači, ukládání na mapovém serveru, rendering získané mapy z pohledu virtuálního robotu, mapování kamer na sebe a volitelné přepínání vizualizace mezi reálným robotem a virtuálním robotem v mapě jsou plně pod naší kontrolou.

Dříve uvedené výpočty byly vytvořeny a optimalizovány v jazyce C# autorem práce bez použití knihoven třetích stran s výjimkou knihovny SlimDX [12] pro práci s DirectX rozhraním a knihovny pro příjem dat z vizuálních snímačů [13]. Použité knihovny však byly pro použití s tzv. Managed kódem neefektivní, proto autor některé z nich pro použití v systému upravil.

Pro srovnávání výsledků bylo použito výpočetní techniky uvedené v tab. 7.1. Parametry komponent byly získány za pomoci programů GPU-Z, CPU-Z a příslušných katalogových listů příslušných výrobců.

Tab. 7.1: Parametry počítačů použitých pro dynamické měření

	PC1	PC2
<i>Grafická karta</i>		
Rodina	<i>nVidia® GeForce™</i>	
Typ	GTX580	GTX570
Paměť	1536 MiB	1280 MiB
Frekvence jádra	795 MHz	780 MHz
Frekvence shaderů	1590 MHz	950 MHz
Frekvence paměti	2004 MHz	1560 MHz
Propustnost paměti	192.4 GiB/s	152.4 GiB/s
<i>Počítač</i>		
Rodina	<i>Intel®</i>	
Typ	Core™ i7 950	Core™ i7 4930K
Počet jader	4 (× 2 HT)	8 (× 2 HT)
Paměť	24 GiB	16 GiB
Frekvence jádra	3.1 GHz	3.4 GHz
Frekvence sběrnice	533 MHz	633 MHz
Operační systém	Windows 7 Pro	Windows 8.1
<i>Úložiště č. 1 (data/data+operační systém)</i>		
Typ	WDC WD2002FAEX	WDC WD10EALX
Velikost	2 TB	1 TB
Připojení	SATA III 6 GiB/s	SATA III 6 GiB/s
Vyrovnávací paměť	64 MiB	32 MiB
Latence	4.2 ms	5.5 ms
Rychlost zápisu	138 MiB/s	126 MiB/s
<i>Úložiště č. 2 (operační systém)</i>		
Typ	-	Samsung SSD 840 Pro
Velikost	-	256 GiB
Připojení	-	SATA III 6 GiB/s
Vyrovnávací paměť	-	512 MiB
Latence	-	0.8 ms
Rychlost zápisu	-	450 MiB/s

## 7.1 Experiment

Mapa byla tvořena v laboratořích a chodbách budovy Technická 12, Brno (viz obrázky 7.1 nasnímané v průběhu měření robotem Orpheus-X3.). Celá mapa byla vytvořena během jednoho cyklu měření ve večerních a nočních hodinách vzhledem ke konstantnosti osvětlení.



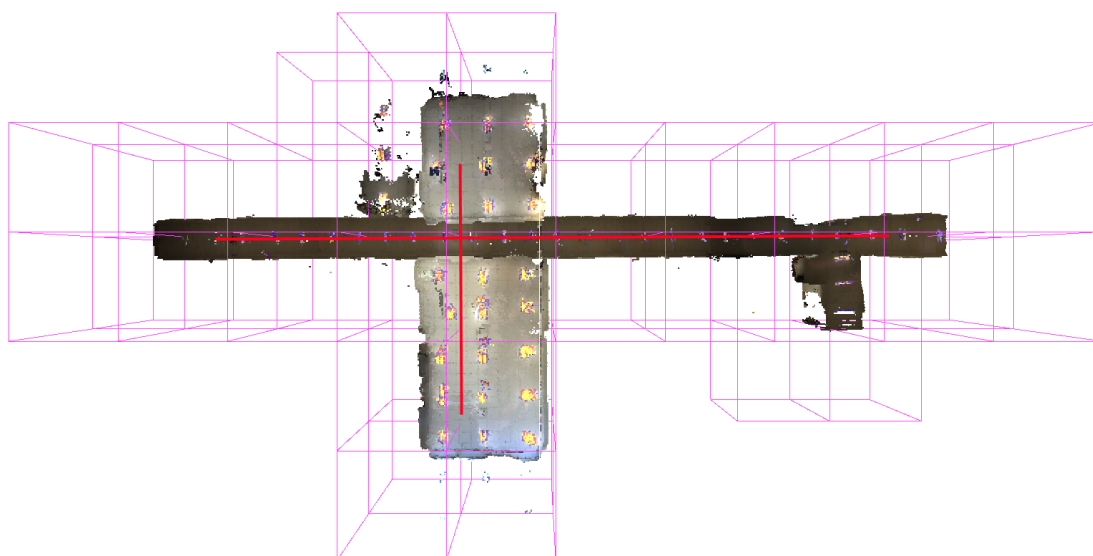
Obr. 7.1: Měření mapy

Měření bylo rozděleno do dvou vzájemně kolmých úseků (vyznačeny červeně v obr. 7.2) po kterých se robot Orpheus-X3 pohyboval po přímce a měřil vzdálenost od konce chodby pro každý naměřený scan. Robot byl připojen dvěma gigabitovými ethernetovými kabely, které jsou na některých snímcích mapy viditelné.

Při pohledu dovnitř místnosti v mapě na obrázku 7.3 je názorně vidět oblast, ve které paprsky použité TOF kamery nasnímalý prázdný prostor a oblasti, kde nedošlo k dotyku s žádným objektem, a vznikla tam tedy bílá neobarvená obast. Z obrázku 7.3 je též patrná trajektorie robotu během měření (mírně obarvená čára na podlaze způsobená nedokonalostí ořezu měřených dat).

Během měřicího experimentu bylo nasnímáno více než 16 GiB surových dat, uložených zejména jako snímky obrazu videokamer a matice hodnot ostatních snímačů v separátních souborech. Tato data byla následně zpracovávána pro vytvoření mapy a její kreslení.

Během experimentu byly též v reálném čase snímány data z obrazových snímačů a prezentovány operátorovi do stereovizních brýlí virtuální reality.



Obr. 7.2: Pohled na měřenou scénu shora s červeně vyznačenou trajektorií robotu



Obr. 7.3: Mapa - pohled do jedné z měřených laboratoří

## 7.2 Tvorba Octree mapových podkladů

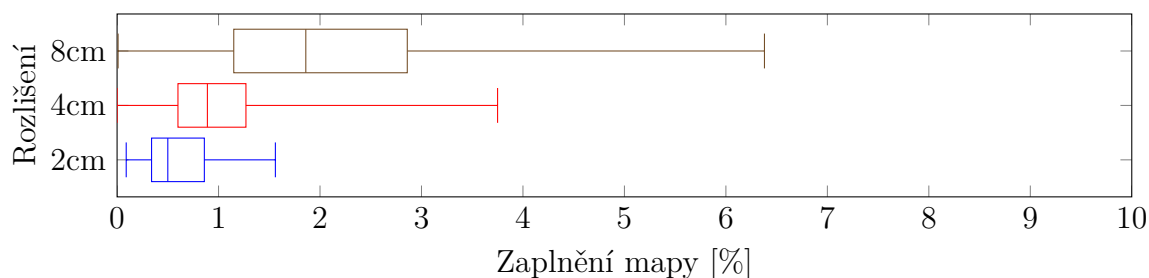
Všechna surová data naměřená v experimentu sloužila jako zdroj pro tvorbu mapy. Mapa byla sestavena z oblastí o velikosti 5.12m. Měření bylo sestaveno tak, že hrany buněk se nacházejí uprostřed místností. Měření obsahuje data místností (2x2x5 buněk), data chodby (2x2x7 buněk) a odbočky chodby (2x2x2 buněk). Buněk bylo tedy ve všech měřeních 56, rozprostřených ve dvou patrech z nichž každé patro mělo 28 buněk.

Buňky, které obsahovaly extrémně málo informace byly pro účely srovnání zahrzeny. Po eliminaci těchto buněk byla vybrána množina 52 shodných oblastí pro různé úrovně Octree stromu. Jejich srovnání udává tabulka 7.2

Tab. 7.2: Statické výsledky

<i>Buňka</i>			
Délka hrany	5.12 m		
Dělení hrany	256 voxelů	128 voxelů	64 voxelů
Hloubka octree	8 bitů	7 bitů	6 bitů
Rozlišení	2 cm/voxel	4 cm/voxel	8 cm/voxel
Zábor paměti	64 MiB/buňku	8 MiB/buňku	1 MiB/buňku
<i>Mapa</i>			
Pokrytá plocha	~13421 m <sup>2</sup> (25.6 m × 51.2 m × 10.2 m [š × d × v])		
Počet buněk	52		
Celkový zábor paměti	3328 MiB	416 MiB	52 MiB
Neprázdné voxely	4389419 voxelů	1493079 voxelů	350574 voxelů
<i>Octree mapa</i>			
Celkový zábor paměti	43 MiB	14 MiB	2 MiB
Kompresní poměr	1:77	1:29	1:26

Z provedených měření vyplývá, že vyplněná oblast daty roste se snižujícím se rozlišením mapy. Body odrazu se zvětšují a zabírají tak větší objem. To je vidět na grafu 7.4.



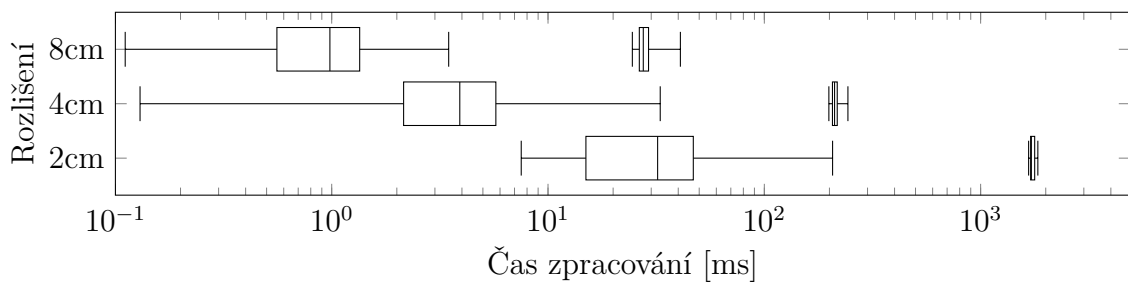
Obr. 7.4: Závislost zaplnění mapy v závislosti na rozlišení mapy pro všechny buňky měření

Dále byla důsledně zkoumána rychlost práce s octree mapou, jmenovitě její komprese a dekomprese. Pro testy byl použit počítač PC1 z důvodu velkého množství paměti RAM a velkého disku. Měřil se čas načítání a ukládání každé buňky mapy zvlášť a z těchto vstupních dat vznikl statistický přehled zobrazený na obrázku 7.5. V

grafu hodnoty více vpravo odpovídají ukládání do octree, hodnoty vlevo odpovídají načítání z octree. Statistika byla provedena ze všech dat.

Z grafu na obrázku 7.5 plyne, že časová náročnost komprese je o více než řád vyšší než dekomprese. To je dáno algoritmem procházení mapy, kdy pro kompresi je nutné projít a otestovat všechny body, ale pro dekompresi stačí projít pouze body, které byly předtím obsazeny. Poměr doby ukládání a načítání je blízký zaplněnosti mapy.

Z rozptylu hodnot napříč celou mapou lze soudit na vysokou závislost doby načítání na zaplněnosti mapy a naopak malou závislost doby ukládání na zaplněnosti mapy.



Obr. 7.5: Časová náročnost konverze mapy mezi Octree a Evidence Grids v závislosti na rozlišení.

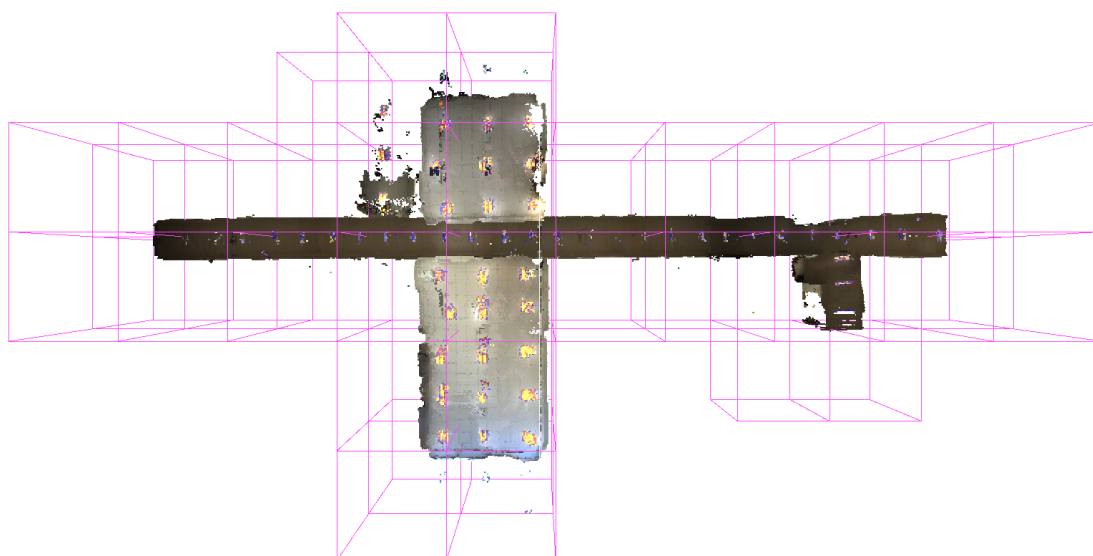
Úroveň komprese souborů odpovídá teoretickým předpokladům odvozeným v kapitole 2.4, kdy velikost souboru je cca o 7% větší než součet velikostí paměti všech obsazených bodů.

### 7.2.1 Vykreslování mapy

Výsledná mapa (na obrázku 7.6) byla následně podrobena zkoumání rychlosti vykreslování navrženým algoritmem.

Měření doby vykreslování obrazu probíhalo na sekvenci pohledů, které obsahovaly různé části mapy, s různým pokrytím, pohled na celou mapu, jakož i pohledy na detailní části mapy. Měření probíhalo s přesností 1 ms. Srovnání s mapou bez optimalizace je pouze orientační, určité optimalizované algoritmy není možno v současné verzi vypnout a jsou přítomny v obou měřeních.

Z tabulky 7.3 je zřejmé, že došlo k optimalizaci zejména průměrných a minimálních časů kreslení, tedy že výsledná optimalizace pokrývá pouze část zobrazení. Pro optimalizaci těžko optimalizovatelných procesů, jako například kreslení celé scény,



Obr. 7.6: Pohled na celou vytvořenou mapu shora

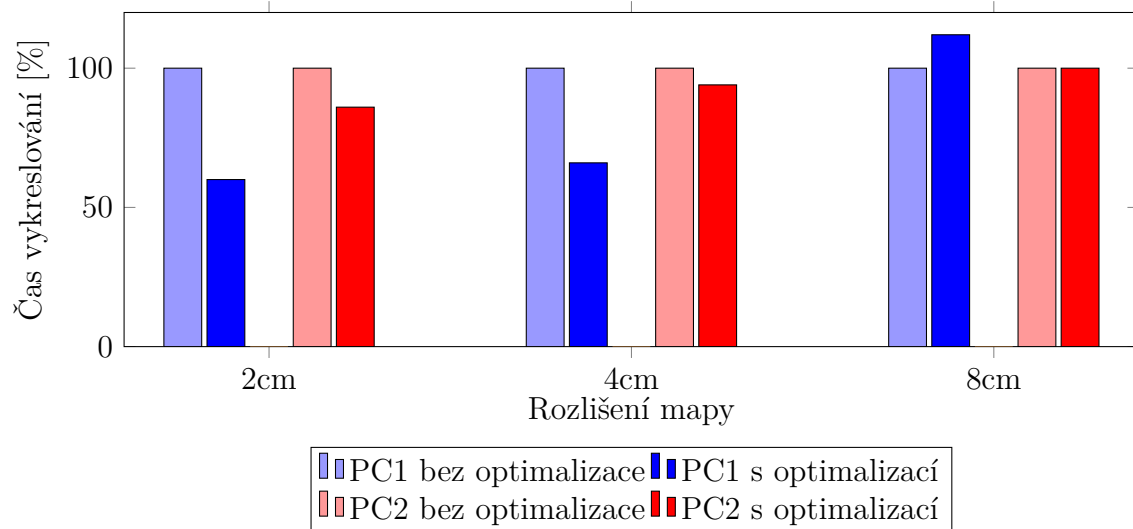
Tab. 7.3: Vliv optimalizace vykreslování celé mapy

	2 cm/voxel		4 cm/voxel		8 cm/voxel	
	PC1	PC2	PC1	PC2	PC1	PC2
<i>Vykreslování úplné mapy před optimalizací</i>						
Využitá paměť grafiky	1464 MiB	1160 MiB	592 MiB	745 MiB	235 MiB	381 MiB
Vytížení GPU procesoru	92 %	92 %	86 %	92 %	80 %	80 %
Vytížení GPU paměti	1 %	2 %	12 %	15 %	10 %	15 %
Čas na vykreslení(min)	3.0 s	1.8 s	16 ms	11 ms	7 ms	6 ms
Čas na vykreslení(průměrně)	5.0 s	2.8 s	18 ms	17 ms	8 ms	7 ms
Čas na vykreslení(max)	11.0 s	4.5 s	39 ms	24 ms	17 ms	10 ms
<i>Vykreslování úplné mapy po optimalizaci</i>						
Využitá paměť grafiky	1464 MiB	1197 MiB	601 MiB	721 MiB	237 MiB	357 MiB
Vytížení GPU procesoru	89 %	92 %	87 %	92 %	78 %	78 %
Vytížení GPU paměti	4%	2 %	11 %	14 %	11 %	12 %
Čas na vykreslení(min)	1.0 s	1.9 s	8 ms	14 ms	8 ms	6 ms
Čas na vykreslení(průměrně)	3.0 s	2.4 s	12 ms	16 ms	10 ms	7 ms
Čas na vykreslení(max)	11.0 s	4.8 s	31 ms	34 ms	12 ms	13 ms

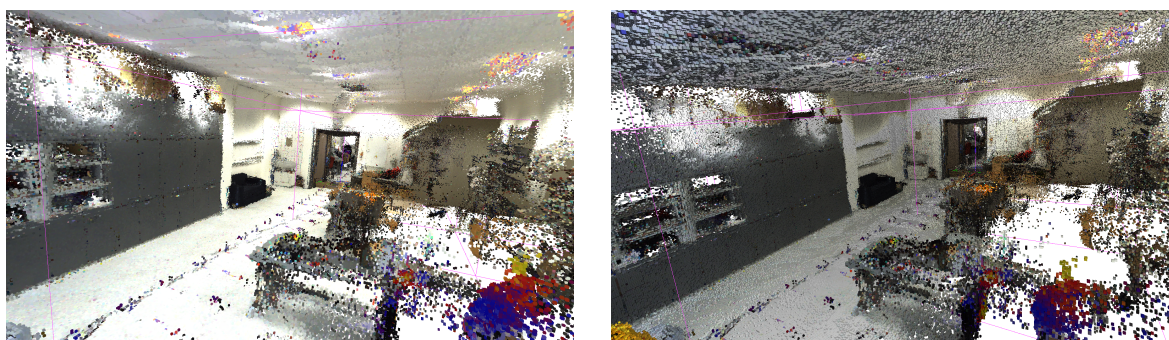
by bylo zapotřebí přistoupit k úplně jinému způsobu vykreslování, jakým je například raytracing, popisovaný v [22]. Výsledky optimalizace vykreslování vyjádřené v procentech oproti průměrné době vykreslování jsou znázorněny na obrázku 7.7:

Na obrázku 7.8 je zobrazen výsledný obraz scény bez optimalizace a s optimalizací. Je vidět, že stínování bočních ploch pouze zvyšuje členitost terénu a odvádí pozornost pozorovatele od prohlížené scény.

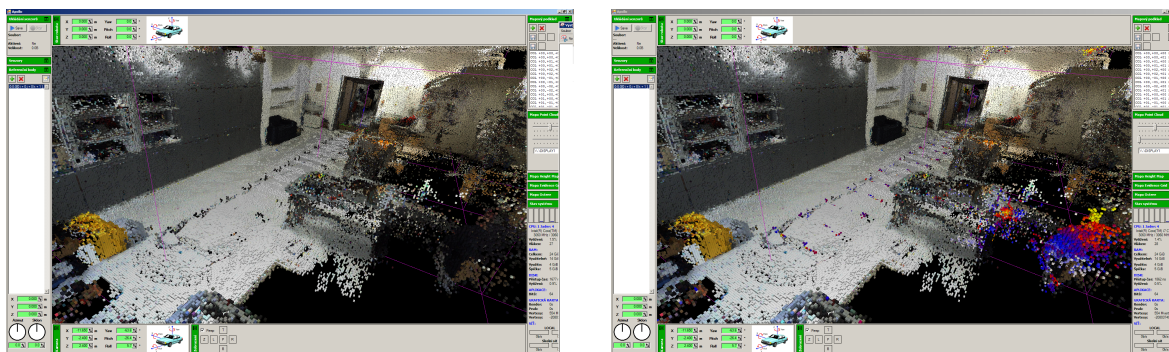
Na obrázku 7.9 je vidět způsob zobrazení teplot okolí při uživatelem nastavené nízké průhlednosti teplot (vpravo) a vysoké průhlednosti teplot (vlevo). Teplotní mapa byla vybrána taková, aby zvýrazňovala teploty blízké teplotě lidského těla. Je vidět zvýraznění místa, kde se při experimentu nacházel člověk.



Obr. 7.7: Výsledky optimalizace vykreslování



Obr. 7.8: Různé způsoby osvětlení

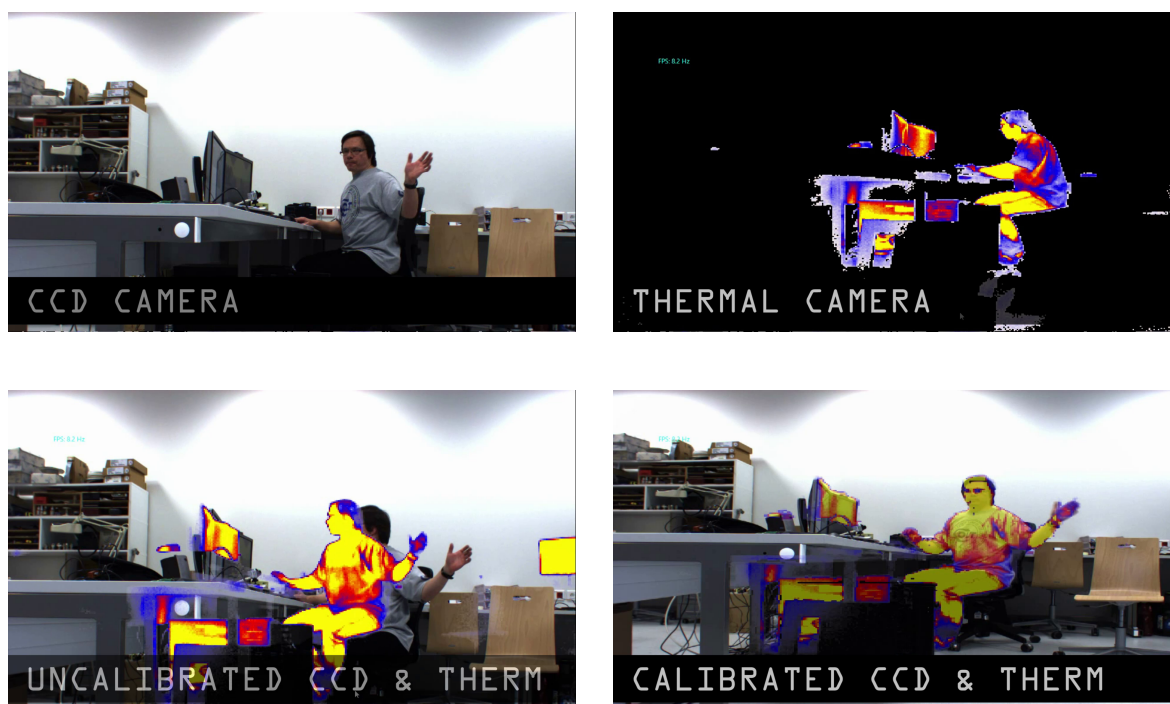


Obr. 7.9: Mapování thermo

## 7.2.2 Výsledky fúze dat

Pro testování fúze dat v reálném čase byl použit PC2, protože jako jediný má dostatečnou síťovou konektivitu (2x 1Gbps síťová karta). Všechny kamery jsou do algoritmů přenášeny v plné rychlosti 30 snímků za sekundu. Limitujícím faktorem pro rychlost zpracování je rychlost TOF kamery, která se mění dle parametrů osvětlení snímané scény a momentálního nastavení snímače mezi 6 snímků za sekundu až 15 snímků za sekundu.

Výsledky fúze dat nejlépe prokazuje video, zachycené z reálného běhu aplikace, přiložené na datovém nosiči. V následujících obrázcích jsou uloženy snímky z tohoto videa.



Obr. 7.10: Fúze dat (snímky z videa na přiloženém datovém nosiči)

Nižší snímková rychlost kolem 8 snímků za sekundu je způsobená vyšší latencí TOF kamery při daném osvětlení.

## 8 ZHODNOCENÍ A ZÁVĚR

Jak již bylo řečeno v úvodu, hlavním cílem mé dizertační práce bylo vytvořit systém pro multispektrální datovou fúzi, kterou by bylo možné využít pro mapování a teleprezenční režim v multirobotickém systému – v našem případě v systému CASSANDRA, který společně vyvíjíme na našem pracovišti. Ačkoli je třeba upozornit, že současný stav jak mé práce, tak celého systému CASSANDRA, rozhodně není možné považovat za konečný, jsou zde prezentované algoritmy funkční a použitelné.

Navržený způsob reprezentace fúzovaných dat pro mapování se osvědčil, jeho výhodou je i možnost rozšiřitelnosti v budoucnu.

Pro další vývoj je zásadní možnost kombinace několika typů map (především Octree a Evidence Grid) a z toho vyplývající šetření výpočetních a paměťových prostředků a optimalizace pro bezdrátový přenos dat mezi roboty a operátorskou stanicí. Pozitivní vliv „bezeztrátové komprese dat“ pomocí Octree je dobře patrný z výsledků v předchozí kapitole. Přitom je však poměrně dobře vyřešen i problém s distribucí dat i z robotů, které nemají dostatečný výpočetní výkon na to, aby převáděly mapy do Octree.

Tento způsob reprezentace dat, uložení map a jejich správy v rámci multirobotického systému považuji za unikátní a doposud jsem se v literatuře nesetkal s podobným řešením. Na základě výsledků předchozí kapitoly je přitom možno říci, že jde o systém funkční a za běžných podmínek velmi efektivní.

Dále jsem se zabýval optimalizací zobrazení map pro operátora, což je další velmi důležitá součást celku, bez které by celý systém nebyl kompletní a konečný uživatelský zážitek by nebyl dobrý. Na základě analýzy je opět možno říci, že díky masivní paralelizaci a optimalizaci se podařilo vytvořit systém, který může i na dnešním hardware v reálném čase zobrazovat extrémní množství dat, které multispektrální mapy s vysokým rozlišením jednoznačně představují. Ačkoli při vysokém rozlišení mapy ještě obecně nedosahujeme zcela plynulého zobrazení (60Hz a více), je současný stav pro práci přijatelný s tím, že předpokládám další optimalizace a zrychlení při použití nadcházející generace grafických karet.

Jak již bylo řečeno, práce na systému nepovažuji za dokončené. V nejbližší době se budu zabývat následujícími činnostmi:

- Paralelizace přepočtů kalibrace z jednotlivých snímačů i vzájemný prostorový přepočet pomocí homogenních transformací pomocí výpočetních prostředků grafické karty.
- Časová synchronizace získaných dat, což jednak podstatně vylepší uživatelský zážitek, ale také podstatně zrychlí proces získávání dat při automatickém mapování.

- Testování distribuce mapy mezi dvěma zcela odlišnými roboty (Orpheus-X3 a envMap) se zcela rozdílným sensorickým vybavením.

## Poděkování

Tato práce vznikla v rámci CEITEC - Středoevropského technologického institutu s pomocí výzkumné infrastruktury financované projektem CZ.1.05/1.1.00/02.0068 z Evropského fondu regionálního rozvoje, za podpory Technologické Agentury České Republiky projektem č. TE01020197 „Centrum Aplikované Kybernetiky 3“ a grantu VG 2012 2015 096 „Kooperativní robotický průzkum nebezpečných oblastí“ Ministerstva Vnitra České Republiky, v rámci programu „Programu bezpečnostního výzkumu České republiky v letech 2010 až 2015“ (BV II/2-VS).

## LITERATURA

- [1] Bartsch, H. J.; Tichý, Z.: *Matematické vzorce*. SNTL, 1987.
- [2] Berler, A.; Shimony, S. E.: Bayes networks for sensor fusion in occupancy grids. In *Proceedings of the Thirteenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-97)*, ročník 13, San Francisco, CA, USA: Morgan Kaufmann, Srpen 1997, ISBN 1-55860-485-5, s. 14–21.
- [3] Burian, F.; Kocmanova, P.; Zalud, L.: Robot Mapping with Range camera, CCD Cameras and Thermal Imagers. In *In Proc. of the 19th International Conference on Methods and Models in Automation and Robotics*, 2014.
- [4] Chacon, S.: *Pro Git*. Berkely, CA, USA: Apress, první vydání, 2009, ISBN 1430218339, 9781430218333.
- [5] Elfes, A.: *Occupancy Grids: A probabilistic framework for robot perception and navigation*. Dizertační práce, Carnegie-Mellon University, Pittsburgh, PA, USA, 1989.
- [6] Fielding, R.; Gettys, J.; Mogul, J.; aj.: Hypertext Transfer Protocol – HTTP/1.1. RFC 2068 (Proposed Standard), Leden 1997, obsoleted by RFC 2616.  
URL <http://www.ietf.org/rfc/rfc2068.txt>
- [7] Fielding, R.; Gettys, J.; Mogul, J.; aj.: Hypertext Transfer Protocol – HTTP/1.1. RFC 2616 (Draft Standard), Červen 1999, obsoleted by RFCs 7230, 7231, 7232, 7233, 7234, 7235, updated by RFCs 2817, 5785, 6266, 6585.  
URL <http://www.ietf.org/rfc/rfc2616.txt>
- [8] Fielding, R.; Reschke, J.: Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing. RFC 7230 (Proposed Standard), Červen 2014.  
URL <http://www.ietf.org/rfc/rfc7230.txt>
- [9] Fielding, R.; Reschke, J.: Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. RFC 7231 (Proposed Standard), Červen 2014.  
URL <http://www.ietf.org/rfc/rfc7231.txt>
- [10] Glennie, C.; Lichti, D. D.: Static Calibration and Analysis of the Velodyne HDL-64E S2 for High Accuracy Mobile Scanning. In *Remote Sensing*, 2010, ISSN 2072-4292, s. 1610–1624.

- [11] Glennie, C.; Lichti, D. D.: Static Calibration and Analysis of the Velodyne HDL-64E S2 for High Accuracy Mobile Scanning. *Remote Sensing*, ročník 2, č. 6, 2010: s. 1610–1624, ISSN 2072-4292.
- [12] Group, S.: SlimDX DirectX wrapper library. 2014.  
URL <http://slimdx.org/>
- [13] ImagingSource: IC Imaging Control .NET. 2014.  
URL [http://www.theimagingsource.com/en\\_US/support/documentation/icimagingcontrol-dotnet/](http://www.theimagingsource.com/en_US/support/documentation/icimagingcontrol-dotnet/)
- [14] Jež, O.: 3D Mapping and Localization Using Leveled Map Accelerated ICP. In *European Robotics Symposium 2008*, 2008, ISBN 978-3-540-78315-2, s. 343–353.
- [15] Kilpela, A.: *Pulsed time-of-flight laser range finder techniques for fast, high precision measurement applications*. Dizertační práce, University of Oulu, Oulu, Finland, 2004.  
URL <http://herkules.oulu.fi/isbn9514272625/?lang=en>
- [16] Kocmanova, P.: Kalibrace a ověření přesnosti hybridního skeneru Velodyne. In *Sborník anotací konference JUNIORSTAV 2014*, 2014, ISBN 978-80-214-4851-3, s. 1–6.
- [17] Kocmanova, P.; Zalud, L.: Proximity Laser Scanner Calibration for Rescue Robotics. In *USB Proceedings IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, 2013, s. 2458–2463.
- [18] Kocmanova, P.; Zalud, L.: Spatial Calibration of TOF Camera, Thermal Imager and CCD Camera. In *Mendel 2013: 19th International Conference on Soft Computing*. Brno: Brno University of Technology, Faculty of Mechanical Engineering, 2013, ISBN 978-80-214-4755-4, s. 343–348.
- [19] Kocmanova, P.; Zalud, L.; Burian, F.: Multispectral Texture Mapping for Telepresence and Autonomous Mobile Robotics. In *Proceedings of the 3rd International Conference on Mechanical Engineering and Mechatronics*. Prague, Czech Republic, 2014, s. 158–1–158–9.
- [20] Kocmanova, P.; Zalud, L.; Chromy, A.: 3D Proximity Laser Scanner Calibration. In *18th International Conference on Methods and Models in automation and Robotics*, 2013, ISBN 978-1-4673-5507-0, s. 742–747.
- [21] Orechovesky, J. R.: *Single Source Error Ellipse Combination*. Diplomová práce, Naval Postgraduate School Monterey CA, Monterey, Canada, 1996.

- [22] Romisch, K.; Moller-Nielsen, P.: *Sparse Voxel octree Ray Tracing on the GPU*. Dizertační práce, Aarhus University, Aarhus, Denmark, 2009.
- [23] Rusu, R. B.; Cousins, S.: 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, Květen 2011.  
URL <http://www.pointclouds.org>
- [24] Sick AG: *Telegrams for Operating / Configuring the LMS2xx Laser Measurement Systems*. 2003.
- [25] Smith, R. C.; Cheeseman, P.: On the representation and estimation of spatial uncertainty. *International Journal of Robotic Research*, ročník 5, č. 4, Prosinec 1986: s. 56–68, ISSN 0278-3649.
- [26] Thrun, S.; Burgard, W.; Fox, D.: *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005, ISBN 0262201623.
- [27] Torvalds, L.: Git distributed version control system. 2014.  
URL <http://git-scm.com/>
- [28] Štěpán, P.: *Vnitřní reprezentace prostředí pro autonomní mobilní roboty*. Dizertační práce, České Vysoké Učení Technické v Praze, Praha, Česká Republika, 2001.
- [29] Velodyne: *HDL-64E S2 and S2.1*. 2013.
- [30] Ye, C.: Mixed Pixels Removal of a Laser Rangefinder for Mobile Robot 3-D Terrain Mapping. 2004.
- [31] Ye, C.; Borenstein, J.: A new terrain mapping method for mobile robots obstacle negotiation. In *In Proc. of the UGV Technology Conference at the 2002 SPIE AeroSense Symposium*, 2003, s. 21–25.
- [32] Ye, C.; Borenstein, J.: A Novel Filter for Terrain Mapping with Laser Rangefinders. In *in the IEEE Transactions on Robotics and Automation*, 2003, s. 913–921.

# SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

$P(A)$	<i>PMF</i> (Probability Mass Function) Pravděpodobnost (diskrétní)
$p(A)$	<i>PDF</i> (Probability Density Function) Hustota pravděpodobnosti (spojitá)
OG	(Occupancy Grid) Mřížka obsazenosti
$\mathcal{M}$	Mřížka obsazenosti
$\mathcal{M}_i$	$i$ -tá buňka mřížky obsazenosti
$\mathcal{M}_i^O$	míra obsazenosti $i$ -té buňky mřížky obsazenosti
$\mathcal{M}_i^E$	míra volnosti $i$ -té buňky mřížky obsazenosti
$\mathcal{W}$	Svět
$\{\mathcal{W}\}$	Množina všech světů
$\mathcal{W}_i$	$i$ -tá buňka světa
$\mathcal{W}_i^O$	Předpoklad obsazené $i$ -té buňky světa
$\mathcal{W}_i^E$	Předpoklad volné $i$ -té buňky světa
$\{\mathcal{W}_i^O\}$	Množina všech světů, kde je $i$ -tá buňka světa obsazená
$\{\mathcal{W}_i^O\}_j$	$j$ -tý svět, z množiny všech světů, kde je $i$ -tá buňka světa obsazená
$\mathcal{R}$	Měření
$\mathcal{R}_j$	Měření dosahuje na buňku $j$
$\{\mathcal{R}\}$	Množina měření
$\hat{X}$	Odhad veličiny $X$

# SEZNAM PŘÍLOH

<b>A</b>	<b>Matematické pomůcky</b>	<b>82</b>
A.1	Odvození inverze symetrické matice 3x3 . . . . .	82
A.2	Metoda nejmenších čtverců . . . . .	83
A.2.1	Příklad . . . . .	84
A.3	Rekurzivní metoda nejmenších čtverců . . . . .	84
<b>B</b>	<b>Statistické pomůcky</b>	<b>85</b>
B.1	N-rozměrné normální rozdělení . . . . .	85
B.1.1	Dvourozměrné normální rozdělení [25] [21] . . . . .	85
B.1.2	Trojrozměrné normální rozložení . . . . .	86
B.2	Zjištění parametrů elipsy . . . . .	86

# A MATEMATICKÉ POMŮCKY

## A.1 Odvození inverze symetrické matice 3x3

Nechť je definována matice o třech řádcích a třech sloupcích následovně:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \quad (\text{A.1})$$

Tato matice je symetrická (splňuje podmínku  $A^T = A$ ). Inverzi matice je možné spočítat dle obecného vztahu:

$$A^{-1} = \frac{1}{|A|} A^* \quad (\text{A.2})$$

kde  $A^*$  označuje adjungovanou matici. Tento vztah lze rozepsat následovně:

$$A^{-1} = \frac{1}{|A|} \begin{pmatrix} a_{22}a_{33} - a_{32}a_{23} & a_{31}a_{23} - a_{21}a_{33} & a_{21}a_{32} - a_{31}a_{22} \\ a_{32}a_{13} - a_{12}a_{33} & a_{11}a_{33} - a_{31}a_{13} & a_{31}a_{12} - a_{11}a_{32} \\ a_{12}a_{23} - a_{22}a_{13} & a_{21}a_{13} - a_{11}a_{23} & a_{11}a_{22} - a_{21}a_{12} \end{pmatrix} \quad (\text{A.3})$$

Pokud je matice  $A$  symetrická ( $A^T = A$ ), platí pro všechny indexy  $i, j \in \{1, 2, 3\}$ :

$$a_{i,j} = a_{j,i} \quad (\text{A.4})$$

Výsledná inverze se poté zjednoduší na:

$$A^{-1} = \frac{1}{|A|} \begin{pmatrix} a_{22}a_{33} - a_{23}^2 & a_{13}a_{23} - a_{12}a_{33} & a_{12}a_{23} - a_{13}a_{22} \\ a_{13}a_{23} - a_{12}a_{33} & a_{11}a_{33} - a_{13}^2 & a_{12}a_{13} - a_{11}a_{23} \\ a_{12}a_{23} - a_{13}a_{22} & a_{12}a_{13} - a_{11}a_{23} & a_{11}a_{22} - a_{12}^2 \end{pmatrix} \quad (\text{A.5})$$

Což je opět matice symetrická. Pro úplný výpočet všech hodnot matice stačí spočítat pouze následujících 6 koeficientů:

$$\begin{aligned} d &= a_{11}a_{22}a_{33} + 2a_{12}a_{23}a_{13} - a_{12}^2a_{33} - a_{23}^2a_{11} - a_{13}^2a_{22} \\ m_1 &= \frac{a_{22}a_{33} - a_{23}^2}{d} & m_{12} &= \frac{a_{13}a_{23} - a_{12}a_{33}}{d} \\ m_2 &= \frac{a_{11}a_{33} - a_{13}^2}{d} & m_{13} &= \frac{a_{12}a_{23} - a_{13}a_{22}}{d} \\ m_3 &= \frac{a_{11}a_{22} - a_{12}^2}{d} & m_{23} &= \frac{a_{12}a_{13} - a_{11}a_{23}}{d} \end{aligned} \quad (\text{A.6})$$

Zapsaných do původní matice takto:

$$A^{-1} = \begin{pmatrix} m_1 & m_{12} & m_{13} \\ m_{12} & m_2 & m_{23} \\ m_{13} & m_{23} & m_3 \end{pmatrix} \quad (\text{A.7})$$

Při počítání se symetrickými maticemi je možné využít symetričnosti a přenášet i ukládat pouze těchto 6 koeficientů, které plně charakterizují celou matici.

## A.2 Metoda nejmenších čtverců

Tato metoda slouží k estimaci neznámých parametrů modelu ze známých změřených hodnot na reálném systému. Nechť je model zapsán v odchytkovém tvaru následovně:

$$\varphi^T(k)\theta = \zeta(k) + \varepsilon(k) \quad (\text{A.8})$$

kde

$\varphi^T(k)$  ... Vektor měřených veličin

$\zeta(k)$  ... Pravá strana rovnice modelu

$\varepsilon(k)$  ... Aditivní bílý šum působící na pravou stranu modelu

$\theta$  ... Vektor parametrů modelu. Konstanta nezávislá na  $k$ .

Nechť je matice obsahující historii  $N$  měření označena  $\Phi$ , a  $k$  nim matice příslušejících pravých stran označena  $R$  následovně:

$$\Phi(k)\theta = R(k) \quad (\text{A.9})$$

kde:

$$R(k) = \begin{bmatrix} y(k) \\ y(k-1) \\ \vdots \\ y(k-N) \end{bmatrix} \quad \Phi(k) = \begin{bmatrix} \varphi_x^T(k) \\ \varphi_x^T(k-1) \\ \vdots \\ \varphi_x^T(k-N) \end{bmatrix} \quad (\text{A.10})$$

Tuto soustavu rovnic je možné řešit vůči neznámým parametřům  $\theta$  metodou nejmenších čtverců.

$$\theta = (\Phi^T \Phi)^{-1} \Phi^T R \quad (\text{A.11})$$

Je možné vyjádřit i nejistotu identifikovaných parametrů ze vztahu:

$$u_\theta = (\Phi^T \Phi)^{-1} u_\varepsilon \quad (\text{A.12})$$

kde

$$u_\varepsilon = k_\alpha \sigma_\varepsilon \quad (\text{A.13})$$

koeficient  $k_\alpha$  je volen tak, aby výsledná nejistota byla dostatečně velká. V této práci byla zvolena konstanta  $k_\alpha = 1.96$  pokrývající 95% měření.

Matice  $\Phi^T \Phi$  je vždy symetrická podle hlavní diagonály. Je tedy možné počítat pouze šest koeficientů místo devíti, a tím dosáhnout zvýšení rychlosti výpočtů bez ztráty přesnosti.

Inverzní matici z těchto šesti koeficientů je možné spočítat dle odvození A.1

### A.2.1 Příklad

$$\begin{aligned} \Phi^T \Phi &= \begin{bmatrix} x(k) & x(k-1) & \cdots & x(0) \\ y(k) & y(k-1) & \cdots & y(0) \\ 1 & 1 & \cdots & 1 \end{bmatrix} \times \begin{bmatrix} x(k) & y(k) & 1 \\ x(k-1) & y(k-1) & 1 \\ \vdots & \vdots & \vdots \\ x(0) & y(0) & 1 \end{bmatrix} = \\ &= \begin{bmatrix} \sum x^2 & \sum xy & \sum x \\ \sum xy & \sum y^2 & \sum y \\ \sum x & \sum y & \sum 1 \end{bmatrix} \end{aligned} \quad (\text{A.14})$$

$$\Phi^T R = \begin{bmatrix} x(k) & x(k-1) & \cdots & x(0) \\ y(k) & y(k-1) & \cdots & y(0) \\ 1 & 1 & \cdots & 1 \end{bmatrix} \times \begin{bmatrix} -z(k) \\ -z(k-1) \\ \vdots \\ -z(0) \end{bmatrix} = \begin{bmatrix} -\sum xz \\ -\sum yz \\ -\sum z \end{bmatrix} \quad (\text{A.15})$$

## A.3 Rekurzivní metoda nejmenších čtverců

Slouží k průběžné identifikaci parametrů soustavy

$$e(k) = \zeta(k) - \varphi(k)^T \theta(k-1) \quad (\text{A.16a})$$

$$K(k) = \frac{P(k-1)\varphi(k)}{1 + \varphi(k)^T P(k-1)\varphi(k)} \quad (\text{A.16b})$$

$$P(k) = P(k-1) - K(k)\varphi(k)^T P(k-1) \quad (\text{A.16c})$$

$$\theta(k) = \theta(k-1) + K(k)e(k) \quad (\text{A.16d})$$

## B STATISTICKÉ POMŮCKY

### B.1 N-rozměrné normální rozdělení

pravděpodobnost výskytu robotu na konkrétní pozici  $L$  lze vyjádřit jako

$$P(L) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{1}{2} \Delta^T \Sigma^{-1} \Delta} \quad (\text{B.1})$$

Kde  $L$  je vektor pozice vyšetřovaného bodu,  $\Delta = L - \mu$  je vektor odchylek bodu  $L$  od vektoru středních hodnot  $\mu$ . Všechny vektory jsou délky  $n$ .

$$L = \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_n \end{bmatrix} \quad \Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nn} \end{bmatrix} \quad \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix} \quad (\text{B.2})$$

$\Sigma$  je kovarianční matice náhodné veličiny. Pro experimentální zjištění jednotlivých hodnot z  $N$  měření lze psát:

$$\Sigma_{ij} = \sigma_{ij} = E(\Delta_i \Delta_j) = \frac{1}{N} \sum_{k=0}^N \Delta_i^{(k)} \Delta_j^{(k)} \quad (\text{B.3})$$

#### B.1.1 Dvourozměrné normální rozdělení [25] [21]

$$L = \begin{bmatrix} L_x \\ L_y \end{bmatrix} \quad \Sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix} \quad \mu = \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix} \quad (\text{B.4})$$

$$|\Sigma| = \sigma_{xx} \sigma_{yy} - \sigma_{xy}^2 \quad (\text{B.5})$$

Dle [25] je možné vyjádřit inverzi matice jako

$$\Sigma^{-1} = \frac{1}{\sigma_{xx} \sigma_{yy} - \sigma_{xy}^2} \begin{bmatrix} \sigma_{yy} & -\sigma_{xy} \\ -\sigma_{xy} & \sigma_{xx} \end{bmatrix} \quad (\text{B.6})$$

Množina všech bodů s konstantní pravděpodobností (elipsa):

$$\frac{1}{2(\sigma_{xx} \sigma_{yy} - \sigma_{xy}^2)} \begin{bmatrix} \Delta_x \\ \Delta_y \end{bmatrix}^T \begin{bmatrix} \sigma_{yy} & -\sigma_{xy} \\ -\sigma_{xy} & \sigma_{xx} \end{bmatrix} \begin{bmatrix} \Delta_x \\ \Delta_y \end{bmatrix} = r^2 \quad (\text{B.7})$$

Po úpravě

$$\frac{\sigma_{xx} \sigma_{yy}}{2(\sigma_{xx} \sigma_{yy} - \sigma_{xy}^2)} \left( \frac{\Delta_x^2}{\sigma_{xx}} + \frac{2\sigma_{xy} \Delta_x \Delta_y}{\sigma_{xx} \sigma_{yy}} + \frac{\Delta_y^2}{\sigma_{yy}} \right) = r^2 \quad (\text{B.8})$$

## B.1.2 Trojrozměrné normální rozložení

$$x = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad \Sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_{zz} \end{bmatrix} \quad \mu = \begin{bmatrix} \mu_x \\ \mu_y \\ \mu_z \end{bmatrix} \quad (\text{B.9})$$

$$|\Sigma| = \sigma_{xx}\sigma_{yy}\sigma_{zz} + 2\sigma_{xy}\sigma_{yz}\sigma_{xz} - \sigma_{xx}\sigma_{yz}^2 - \sigma_{yy}\sigma_{xz}^2 - \sigma_{zz}\sigma_{xy}^2 \quad (\text{B.10})$$

$$\Sigma^{-1} = \frac{1}{|\Sigma|} \begin{bmatrix} \sigma_{yy}\sigma_{zz} - \sigma_{yz}^2 & \sigma_{xz}\sigma_{yz} - \sigma_{xy}\sigma_{zz} & \sigma_{xy}\sigma_{yz} - \sigma_{xz}\sigma_{yy} \\ \sigma_{yz}\sigma_{xz} - \sigma_{xy}\sigma_{zz} & \sigma_{xx}\sigma_{zz} - \sigma_{xz}^2 & \sigma_{xz}\sigma_{xy} - \sigma_{xx}\sigma_{yz} \\ \sigma_{xy}\sigma_{yz} - \sigma_{yy}\sigma_{xz} & \sigma_{xz}\sigma_{xy} - \sigma_{xx}\sigma_{yz} & \sigma_{xx}\sigma_{yy} - \sigma_{xy}^2 \end{bmatrix} \quad (\text{B.11})$$

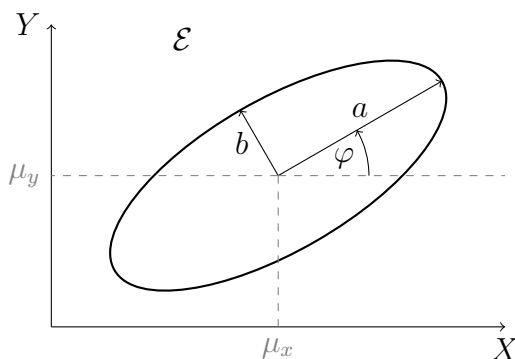
osy je možné vytvořit za pomoci algoritmu SVD

## B.2 Zjištění parametrů elipsy

Jeden z problémů který je třeba v této disertační práci řešit je operace, která ze známých koeficientů kovarianční matice vykreslí elipsu  $\mathcal{E}$ , která ohraničuje oblast, ve které se nachází statistické procento všech pokusů náhodného jevu. (viz obrázek B.1).

Dostupná literatura se velmi liší ve způsobu vyjadřování jednotlivých koeficientů a s tím souvisejících rovnic. Velmi často proto dochází u autorů cizích i domácích k záměnám funkcí mezi různými popisy.

Pro exaktní vyjádření modelu, se kterým pracuje tato disertační práce je tento postup rozebrán i s postupem odvození, ze kterého plynou výsledné vztahy B.28 a B.29.



Obr. B.1: Parametry elipsy

Pro jeho řešení je třeba začít vyjádřením rotační transformace základní elipsy, a zpětném vyjádření neznámých koeficientů původní elipsy z koeficientů transformované elipsy.

Je dána parametrická rovnice základní (netransformované) elipsy  $\mathcal{E}$  v kanonickém tvaru

$$\mathcal{E} : \quad A'x'^2 + B'y'^2 = r^2 \quad (\text{B.12})$$

Parametr  $r^2$  definuje množinu bodů, které jsou na konstantní úrovni pravděpodobnosti. Pro jeho vyjádření z pravděpodobnosti:

$$P(x, y \in \mathcal{E}) = 1 - e^{-\frac{r^2}{2}} \quad (\text{B.13})$$

Zlogaritmováním vyjde výsledek:

$$r^2 = -2 \ln(1 - P(x, y \in \mathcal{E})) \quad (\text{B.14})$$

Tuto elipsu z rovnice B.12 je možné transformovat rotací kolem středu elipsy.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{bmatrix} \times \begin{bmatrix} x' \\ y' \end{bmatrix} \quad (\text{B.15})$$

Pro zjištění úhlu je potřeba převod z pootočené soustavy na nepootočenou, tedy inverzní transformaci k předchozí.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{bmatrix}^{-1} \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \varphi & \sin \varphi \\ -\sin \varphi & \cos \varphi \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} \quad (\text{B.16})$$

Po dosazení transformace B.16 do B.12 je možné spočítat

$$A'(x \cos \varphi + y \sin \varphi)^2 + B'(-x \sin \varphi + y \cos \varphi)^2 = r^2 \quad (\text{B.17})$$

Rovnici B.17 je nutno formálně přeznačit do tvaru:

$$Ax^2 + Bxy + Cy^2 = r^2 \quad (\text{B.18})$$

Kde jednotlivé proměnné jsou

$$A = A' \cos^2 \varphi + B' \sin^2 \varphi \quad (\text{B.19a})$$

$$B = 2(A' - B') \sin \varphi \cos \varphi \quad (\text{B.19b})$$

$$C = A' \sin^2 \varphi + B' \cos^2 \varphi \quad (\text{B.19c})$$

Rovnice B.18 odpovídá maticovému zápisu statistiky

$$\Sigma^{-1} = \begin{bmatrix} A & \frac{B}{2} \\ \frac{B}{2} & C \end{bmatrix} \quad (\text{B.20})$$

Aplikací goniometrických vzorců [1] je možné dojít ke vztahu

$$\tan 2\alpha = \frac{2 \tan \alpha}{1 - \tan^2 \alpha} = \frac{2 \sin \alpha \cos \alpha}{\cos^2 \alpha - \sin^2 \alpha} \quad (\text{B.21})$$

Formálním přepisem proměnných ve vztazích B.19a až B.19c je možné dojít k následujícím vztahům

$$\cos^2 \varphi = \frac{A - B'}{A' - B'} \quad (\text{B.22a})$$

$$2 \sin \varphi \cos \varphi = \frac{B}{A' - B'} \quad (\text{B.22b})$$

$$\sin^2 \varphi = \frac{C - B'}{A' - B'} \quad (\text{B.22c})$$

Aplikací vzorců B.22a B.22b a B.22c na B.21 je možné odhalit jednoduchý vztah pro úhel, který svírá hlavní poloosa elipsy se souřadným systémem

$$\tan 2\alpha = \frac{2 \sin \alpha \cos \alpha}{\cos^2 \alpha - \sin^2 \alpha} = \frac{B}{A - C} \quad (\text{B.23})$$

Delky jednotlivých poloos elipsy je možné získat z vlastních čísel matice  $\Sigma^{-1}$ . Vlastní čísla matice je možné vypočítat řešením charakteristické rovnice matice.

$$|\Sigma^{-1} - \lambda \mathbf{I}| = 0 \quad (\text{B.24})$$

Rozepsáním jednotlivých složek:

$$\begin{vmatrix} A - \lambda & \frac{B}{2} \\ \frac{B}{2} & C - \lambda \end{vmatrix} = 0 \quad (\text{B.25})$$

Řešení charakteristické rovnice jsou hledaná vlastní čísla  $\lambda_1$  pro hlavní poloosu a  $\lambda_2$  pro vedlejší poloosu

$$\lambda_{1,2} = \frac{A + C \pm \sqrt{(A - C)^2 + B^2}}{2} \quad (\text{B.26})$$

Kvadrát délek poloos elipsy je možné spočítat použitím následujících rovnic:

$$\begin{bmatrix} a^2 \\ b^2 \end{bmatrix} = \begin{bmatrix} r^2 / \lambda_1 \\ r^2 / \lambda_2 \end{bmatrix} = \begin{bmatrix} \frac{2r^2}{A + C + \sqrt{(A - C)^2 + B^2}} \\ \frac{2r^2}{A + C - \sqrt{(A - C)^2 + B^2}} \end{bmatrix} \quad (\text{B.27})$$

Rozepsáním rovnic B.27 a dosazením do nich dle předpisů B.14 a B.6 je možné dojít k závěrným vztahům pro kvadráty hlavních poloos elipsy a jejich otočení kolem hlavní souřadné soustavy:

$$\begin{bmatrix} a^2 \\ b^2 \end{bmatrix} = \begin{bmatrix} 4 \frac{-\ln(1 - P(x, y \in \mathcal{E}))(\sigma_{xx}\sigma_{yy} - \sigma_{xy}^2)}{\sigma_{xx} + \sigma_{yy} + \sqrt{(\sigma_{yy} - \sigma_{xx})^2 + 4\sigma_{xy}^2}} \\ 4 \frac{-\ln(1 - P(x, y \in \mathcal{E}))(\sigma_{xx}\sigma_{yy} - \sigma_{xy}^2)}{\sigma_{xx} + \sigma_{yy} - \sqrt{(\sigma_{yy} - \sigma_{xx})^2 + 4\sigma_{xy}^2}} \end{bmatrix}. \quad (\text{B.28})$$

A pro úhel elipsy s kladným souřadným systémem  $x$ :

$$\varphi = \frac{1}{2} \tan^{-1} \frac{-2\sigma_{xy}}{\sigma_{yy} - \sigma_{xx}} \quad (\text{B.29})$$

Tyto rovnice s předloženým odvozením jsou shodné, jako ty, uvedené v [25] strana 67, i na jiný zápis v [5] strana 118. V práci [21] jsou uvedena jiná odvození, která však mají obdobný matematický základ a výsledky jsou číselně shodné.