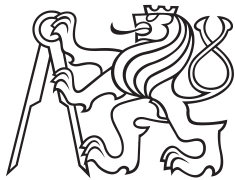


Diplomová práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra kybernetiky

Lokalizace mobilního robotu sledováním stropu

Jiří Koktan

Vedoucí: Ing. Karel Košnar, Ph.D
Obor: Kybernetika a robotika
Studijní program: Kybernetika a robotika
Květen 2019

Poděkování

Chtěl bych poděkovat vedoucímu práce Ing. Karlovi Košnarovi, Ph.D za neochvějnou podporu při celém průběhu tohoto projektu.

Prohlášení

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

V Praze, dne

.....
podpis autora práce

Abstrakt

Tato práce se zabývá lokalizací mobilního robotu sledováním stropu místnosti. Práce nejdříve pojednává o používaných metodách vizuální lokalizace a obrazových příznamech. Dále se zabývá výběrem vhodných obrazových příznaků a metody. Následně je pomocí těchto příznaků vytvořena mapa pro lokalizaci mobilního robotu. Vybraná metoda je implementována a testována na reálném robotu v kancelářském prostředí. Implementace metody je realizována v jazyce C++.

Klíčová slova: mobilní robot, strop, obrazové příznaky, lokalizace

Vedoucí: Ing. Karel Košnar, Ph.D

Abstract

This work deals with mobile robot localization using ceiling features. The paper first presents the world's methods used for robot visual localization and feature detection. In the next part the suitable method and detector are chosen and described. After the creation of the map, curtain method is implemented. This method is then subjected to testing on real mobile robot in the laboratory environment. The implementation of the method is realized in C++.

Keywords: mobile robot, ceiling, feature, localization

Title translation: Localization of a mobile robot by ceiling observation

Obsah

1 Úvod	1	4.1.1 Turtlebot 2.....	32
1.1 Definice úlohy	2	4.1.2 Hardware	35
2 Metody	5	4.2 Implementované metody	37
2.1 Metody používané pro lokalizaci sledováním stropu	5	4.2.1 Open CV	37
2.1.1 Přirozené landmarky	6	4.2.2 Příprava	38
2.1.2 Umělé landmarky	6	4.2.3 Porovnávání obrazových příznaků	39
2.1.3 Lokalizace	7	4.2.4 Tvorba mapy	40
3 Teoretický postup řešení	11	4.2.5 Lokalizace	41
3.1 Obrazové příznaky	11	4.2.6 Použitý software	42
3.1.1 Vybírání vhodného detektoru	13	5 Experimenty	43
3.2 Mapa	18	5.1 Příprava	43
3.2.1 Tvorba mapy	19	5.1.1 Rosbag pro výběr detektoru obrazových příznaků	44
3.3 Lokalizace	24	5.1.2 Rosbag pro tvorbu mapy....	44
3.3.1 Částicový filtr	25	5.2 Tvorba mapy.....	45
4 Implementace	31	5.3 Lokalizace	47
4.1 Robot	31	5.3.1 SURF	47
		5.3.2 KAZE	49

5.3.3 AKAZE	50
5.3.4 Shrnutí	51
6 Závěr	53
Literatura	55
Zadání práce	61

Obrázky

2.1 Ukázka obrazových příznaků a přímek detekovaných na stropu	6	3.6 Procentuální shoda mezi všemi obrazovými příznaky, tedy každý s každým pro daný detektor, při statickém obraze s thresholdem 0.6	16
2.2 Ukázka uměle přidaných landmarků	7	3.7 Shoda obrazových příznaků pro jedoucí robot s thresholdem filtru 0.8 a trasou na 3m	16
2.3 Ukázka map vytvořených algoritmem SLAM	8	3.8 Shoda obrazových příznaků pro jedoucí robot s trasou na 6 m	17
2.4 Obrázky detekce objektů	9	3.9 Párování obrazových příznaků [6]	20
3.1 Detekované obrazové příznaky v místnosti laboratoře	12	3.10 Ukázka systému Vicon [27]. . . .	22
3.2 Porovnávání dvou fotografií stejného objektu. Obrazové příznaky určené jako shodné jsou propojeny barevnou přímkou. Zelené jsou správně určeny jako shodné a červené jsou jako shodné označeny nesprávně. [47]	12	3.11 Komponen systému Vicon [28]	22
3.3 Laboratoř a výrazné prvky na stropu	14	3.12 Mapa vytvořená detektorem AKAZE	23
3.4 Množství obrazových příznaků nalezených jednotlivými detektory ve statickém obraze	14	3.13 Obrazové příznaky detekované na stropu laboratoře	24
3.5 Množství obrazových příznaků nalezených jednotlivými detektory při průjezdu částí laboratoře	15	3.14 Znázornění pohybu diferenciálního vozidla a měřené veličiny [33].	27
		3.15 Znázornění parametrů pohybového modelu [48]	28
		3.16 Vzorkování Částicového filtru [49]	29
		4.1 Ukázka konfigurace robotu použitého při experimentu a implementaci	32
		4.2 Ukázka verzí Turtlebotů [34] . .	33

4.3 Ukázka robotů typu Turtlebot .	33	5.8 Lokalizace v mapě KAZE	49
4.4 Rozměr robotu Turtlebot 2 [34]	34	5.9 Lokalizace v mapě KAZE	49
4.5 Obrazce použité pro kalibraci kamery [35]	35	5.10 Lokalizace v mapě AKAZE . . .	50
4.6 Ukázka snímacího systému robotu	36	5.11 Lokalizace v mapě AKAZE . . .	50
4.7 Ukázka počítače NUC5i5RYK .	36	5.12 Robot lokalizovaný v mapě AKAZE	51
4.8 Ukázka schopností knihoven OpenCV	37		
4.9 Pomocné obrazce pro kalibraci kamery	38		
4.10 Schéma struktury kódu v lokalizačním vlákně	42		
5.1 Trajektorie robotu při nahrávání prvotních záznamů	44		
5.2 Trajektorie robotu při skenování celého stropu pracovního prostoru	45		
5.3 Mapa SURF	46		
5.4 Mapa KAZE	46		
5.5 Mapa AKAZE	47		
5.6 Lokalizace v mapě SURF	48		
5.7 Lokalizace v mapě SURF	48		

Tabulky

4.1 Parametry počítače Intel NUC5i5RYK	36
---	----



Kapitola 1

Úvod

V současném období robotického růstu zefektivňujeme veškerou činnost pomocí robotických pomocníků. Nejčastěji zastávají těžkou a monotonní práci, kterou vykonávají s větším výkonem, než lidští pracovníci. Zejména se jedná o dopravníky, manipulátory a mechanická ramena. Tyto roboty jsou navrženy pro specifickou opakující se činnost a rozsah jejich působnosti je značně omezený.

Nemalou množinu robotů usnadňujících lidem život zastávají také autonomní mobilní roboty. Jsou to například autonomní sekačky na trávu, robotické vysavače nebo dokonce průvodci v muzeích. Řadí se mezi ně i průzkumné nebo bojové drony s velice složitými řídicími algoritmy. Naší neustálou snahou je zdokonalování těchto robotických strojů. Snažíme se vyvíjet stále lepší verze, ať už zrychlením a zefektivněním průmyslových robotů nebo lepším rozhodováním a robustností autonomních systémů.

Jednou z nejdůležitějších schopností autonomního mobilního robotu je lokalizace. Pro bezkolizní a bezpečný pohyb v pracovním prostředí musí robot znát nejbližší okolí a musí být schopen se v něm lokalizovat a to co nejpřesněji. Lokalizace robotu v pracovním prostředí je jedním ze základních prvků mobilní robotiky. Často jsou k tomu používány různé typy map popisující okolní prostředí. Robot se může v takovéto mapě lokalizovat pomocí velké škály různých senzorů a metod, které budou detailně popsány níže.

Senzory jsou děleny na aktivní/pasivní a vnitřní/vnější. Aktivní senzory interagují s okolním prostředím a měří reakce prostředí. Pasivní senzory měří pouze hodnoty vstupující do senzoru z okolního prostředí. Vnitřní senzory

měří výhradně hodnoty vnitřního systému, tedy robotu. Vnější senzory dávají informace z pracovního prostředí robotu. Pro získávání informací o svém okolí jsou nejvhodnější vnější senzory, to jsou například ultrazvukové a laserové dálkoměry a kamery. Data z těchto senzorů mohou být použita pro určení polohy robotu nebo pro sestavování mapy.

Mapa je struktura reprezentující okolní prostředí ve srozumitelné formě pro robot. Mapu si může robot vytvářet sám během svého pohybu a to ukládáním dat ze senzorů do této struktury nebo mu může být přidělena. Přesnost a kvalita by měla odpovídat přesnosti, s kterou robot musí postupovat, aby dosáhl požadovaných výsledků. Důležitý je také typ reprezentace, ten může mít dopad na složitost výpočtů a funkcí s ní souvisejících. Nejznámější typy map jsou sensorická, mřížka obsazenosti, topologická, geometrická.

Sensorická mapa obsahuje přímá data ze senzorů přidělaná do mřížky obsazenosti. Pracovní prostor je rozdělen do jednotlivých buněk, jejichž hodnota určuje informaci o terénu. Velikostí mřížky je určen rozměr a rozlišení vytvořené mapy. Sensorické mapy jsou velice oblíbené díky jejich jednoduchosti, snadnému přístupu a práci s uloženými daty. Topologická mapa reprezentuje pracovní prostředí robotu jako logický graf. Jednotlivá místa (místnosti, pozice) jsou znázorněna jako uzly grafu a mohou obsahovat informace o daném místě. Tato jednotlivá místa jsou spojena cestami, které popisují způsob pohybu mezi jednotlivými uzly. Geometrická mapa je složena pomocí geometrických primitiv, neboli úseček a polygonů. Jednotlivá tato primitiva jsou spojována do samostatných objektů. Taková reprezentace mapy je vhodná při zadávání známého prostředí robotu, například elektronické plány budov. Nevýhodou této mapy bývá její problematická tvorba, úprava a přiřazení sensorových dat.

1.1 Definice úlohy

Cílem této práce je návrh a implementace metody pro lokalizaci mobilního robotu ve známé mapě pomocí sledování stropu. Nejprve je nutné seznámit se s metodami vizuální lokalizace mobilního robotu. Dále se obeznámit s obrazovými příznaky a jejich popisem a vybrat vhodné detektory obrazových příznaků pro průmyslové a kancelářské prostředí. Vytvořit pomocí vybraných detektorů mapu prostoru, ve kterém se musí robot lokalizovat. Následně vybrat a naimplementovat nejvhodnější metodu pro lokalizaci v dané mapě. Na závěr experimentálně ověřit funkčnost zvolené metody.

Tato metoda by měla být využita pro lokalizaci autonomního mobilního průmyslového vozíku v továrním prostředí.

Pokyny pro vypracování jsou následující, viz. zadání :

- Seznamte se s metodami vizuální lokalizace mobilního robotu.
- Seznamte se s obrazovými příznaky a jejich popisem.
- Vyberte nejvhodnější příznaky pro popis stropu v průmyslovém a kancelářském prostředí.
- Navrhněte a naimplementujte metodu lokalizace mobilního robotu ve známé mapě.
- Vyhodnoťte vlastnosti implementované metody.

Kapitola 2

Metody

Často jsou pro lokalizační metody používány aktivní dálkové senzory jako sonar nebo laserový dálkoměr. Mají ovšem svoje nedostatky, jak v použitelnosti, tak v ceně. Dále tu jsou obrazové senzory nebo-li kamery, v některých situacích je jejich použití vhodnější a cenově úspornější. Proto se objevuje stále více metod založených na těchto senzorech. Metody mohou pracovat s jednou nebo více kamerami s různým umístěním a překryvem.

Tato práce se zabývá metodou, kde je pouze jedna kamera snímající strop. Kamera je umístěna na horní části robotu a detekuje význačné body na stropu, podle kterých je možné určení pozice robotu. Můžeme se setkat i s opačným přístupem k problému lokalizace v uzavřených prostorách, kdy je kamera umístěna na stropu a snímá marker na robotu. Tento přístup je velice oblíbený, ale vyžaduje pokrytí kamerami v celém pracovním prostoru [1], [2].

2.1 Metody používané pro lokalizaci sledováním stropu

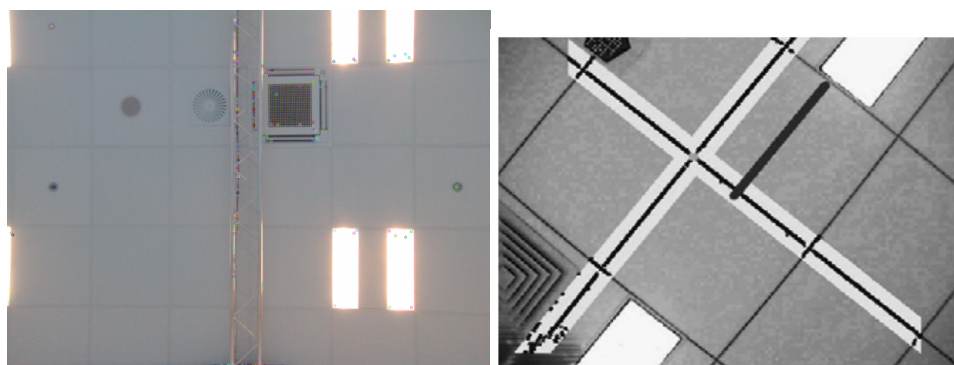
Metody pro lokalizaci potřebují orientační body, pomocí kterých se dá určit pozice a orientace robotu. Mohou to být například obrazové příznaky, landmarky nebo tagy. Tyto orientační body mohou být přirozené nebo umělé.

2.1.1 Přirozené landmarky

Lokalizace s využitím přirozených landmarků nepotřebuje na rozdíl od umělých landmarků přípravu pracovního prostředí. Zato jsou přirozené landmarky náchylnější k chybnému rozpoznávání a porovnávání. Přirozené landmarky se hledají v okolním prostředí robotu s využitím složitých algoritmů, například SIFT, SURF nebo detekcí rohů a přímek.

Lokalizace je složitá v dynamickém prostředí, kdy může landmark měnit pozici nebo úplně zmizet. Potom může docházet k potížím s jejich rozpoznáváním a porovnáváním. Jedno z míst, kde dochází k minimálním změnám, je právě například strop místnosti.

Kancelářské stropy mají často opakující se vzory, například čtvercové obložení a toho je právě využito v práci [3]. Na stropu jsou detekovány rohy a přímky, které slouží ke zjištění pozice robotu. Natočení je vypočítáno z ortogonálních přímek, které jsou tvořeny obložním s čtvercovým vzorem. Pokud nejsou ortogonální přímky k dispozici, je orientace a pozice dopočítána pomocí párování obrazových příznaků.



(a) : Obrazové příznaky nalezené pomocí algoritmu SURF na stropu laboratoře CIIRC

(b) : Detekované čtvercové obložení stropu v projektu [19]

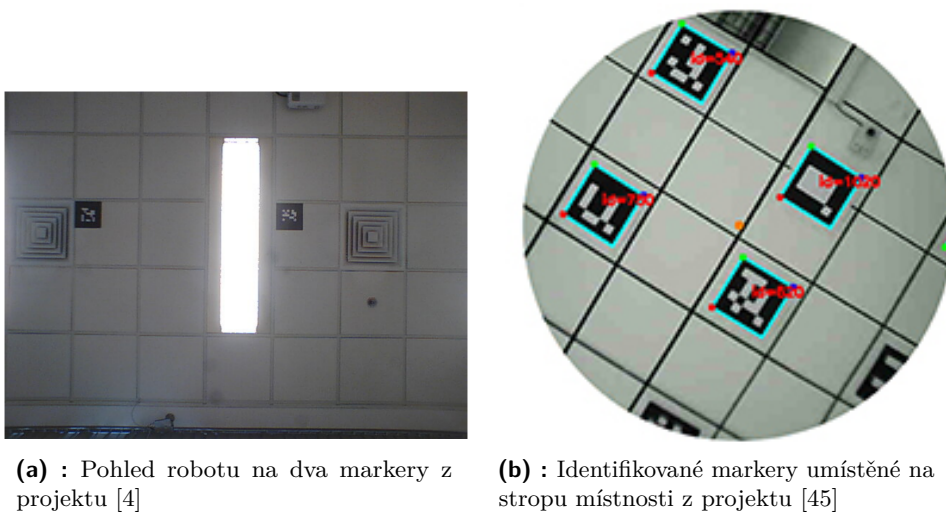
Obrázek 2.1: Ukázka obrazových příznaků a přímek detekovaných na stropu

2.1.2 Umělé landmarky

Používání umělých značek (landmarků) je velice oblíbený způsob zajištění jasných a snadno porovnatelných význačných bodů. Tyto uměle vytvořené orientační body jsou předem vhodně rozmístěny v pracovním prostředí podle

potřeby. Jejich hlavní výhodou je snazší detekce a je možné z nich vyčíst další informace, podle kterých je možné určit pozici a orientaci robotu.[4]

Například v [5] vychází z markerů rovnoměrně rozprostřených po stropu místnosti a obsahují informace o svém ID. Pozice každého markeru a výška stropu je známá. Jakmile robot zaznamená marker, je schopen dopočítat svojí relativní pozici a natočení vůči markeru. Ta je následně přepočítána do globálních souřadnic místnosti. Pokud je detekováno více markerů najednou, jsou vypočtené pozice průměrovány.



Obrázek 2.2: Ukázka uměle přidaných landmarků

Špatné rozpoznání nebo dokonce záměna markerů vede k zásadním chybám v lokalizaci robotu. Proto velké množství projektů používá tyto uměle přidané landmarky, které je snazší detekovat a správně identifikovat.

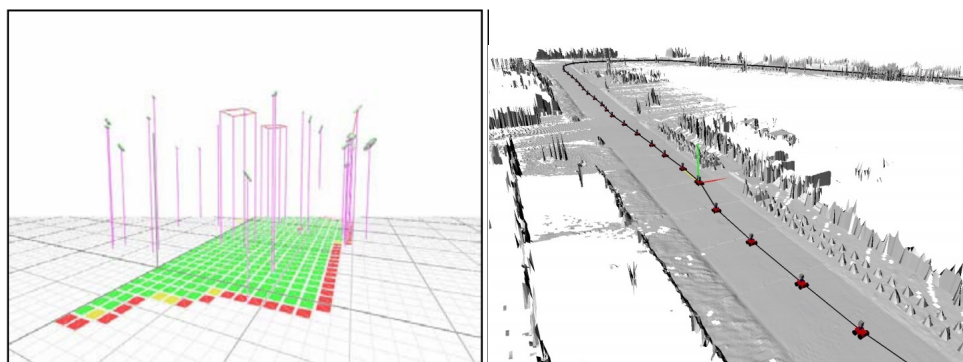
2.1.3 Lokalizace

Používaných způsobů pro lokalizaci robotu je několik. Jedná se například o metody SLAM [8], [9], [10], Kalmanův filtr[14], [15], Particle filtr [16], [17], [18], Markovova lokalizace [11], [12], [13], párování obrazových příznaků [6], lokalizace podle specifických tvarů[21] a objektů jako obložení[19], [20] nebo pouze dopočítávání pozice podle umělých landmarků [4], [5].

■ SLAM

Velice známý je přístup SLAM (simultaneous localization and mapping) pro současnou lokalizaci a mapování. SLAM dokáže rychle a spolehlivě vytvářet mapu a zároveň v ní lokalizovat robot. Většinou je tento algoritmus spojený s používáním laserového dálkoměru nebo sonaru, ale i ty mají své limity a nejsou vždy nejvhodnější. Existují případy použití Indoor SLAMu, pro lokalizaci a mapování pomocí kamery, často s použitím dronu, například v projektu [22].

SLAM algoritmus upravený pro simultánní lokalizaci a mapování stropu kamerou CV-SLAM (ceiling vision-based simultaneous localization and mapping) byl použit v práci [7]. Na stropu jsou detekovány landmarky pomocí Harrisonova rohového detektoru. Pomocí porovnávání aktuálních obrazových příznaků a 3D landmarků se dopočítává pozice robotu.



(a) : Mapa vytvořená algoritmem CV-SLAM v projektu [7]

(b) : Ukázka mapy vytvořené algoritmem SLAM při jízdě po silnici z projektu [46]

Obrázek 2.3: Ukázka map vytvořených algoritmem SLAM

■ Částicový filtr

Částicový filtr neboli Monte Carlo Localization (MCL) je algoritmus odhadující pozici robotu v předem známé mapě s využitím sensorového a pohybového modelu. Tato oblíbená metoda je často používána s laserovými dálkoměry, ale ani obrazové senzory nejsou výjimkou.

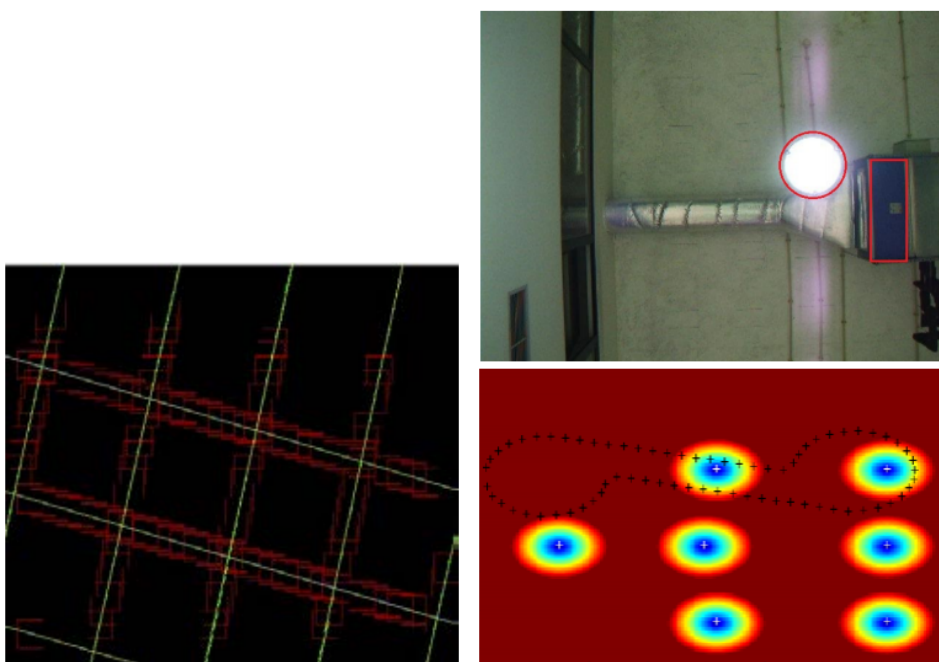
Pozice je určována částicemi, které jsou rozprostřeny po mapě. Každá částice reprezentuje robot s určitou pozicí a orientací. Velice důležitým parametrem každé částice je váha w určující pravděpodobnost, že se robot na dané pozici nachází. Tato pravděpodobnost je ovlivňována sensorovým modelem. Změny

polohy robotu jsou na jednotlivé částice aplikovány pomocí pohybového modelu, který simuluje jeho pohyb.

■ Specifické tvary a objekty

Některé projekty například [19], [20] využívají podobnosti kancelářských stropů a rozpoznávají jednotlivé čtverce obložení. Následně určují relativní pozici robotu podle změny stropu.

Také je možné detekovat jiné objekty, například světla, jako v práci [21]. Pokud je podobně jako u umělých landmarků předem známa jejich pozice, je následně robot lokalizován pomocí Markovovy lokalizace.



(a) : Detekce jednotlivých bloků obložení na stropu místnosti v projektu [19]

(b) : Detekování světla na stropu a mapa z projektu [21]

Obrázek 2.4: Obrázky detekce objektů

Kapitola 3

Teoretický postup řešení

Nejdříve je nutné určit konfiguraci robotu. Podle nastudovaných metod je nejvhodnější a nejpoužívanější konfigurací jedna kamera na horní části mobilního robotu sledující strop. Je možné využít více kamer, buď pro pokrytí větší plochy pro snímání, nebo při překryvu kamer pro získávání hloubkového obrazu. Pro tento projekt bude dostačující pouze jedna kamera, přes kterou budou hledány obrazové příznaky.

Dále by bylo možné kameru umístit pod náklonem a dosáhnout tak zvětšení záběru, ale přínos takového umístění by nejspíše nebyl zásadní. V takovém případě by byla nutná perspektivní transformace a bylo by možné použít pouze část záběru. Nenašloněná kamera směřující přímo na strop je jednoduchým a ideálním řešením. Konfigurace senzorů v této práci je tedy obdobná jako ve většině podobných metod [6], [19], [7].

3.1 Obrazové příznaky

Obraz z kamery musí být zpracován a musí z něj být získáno maximum potřebných dat. Z obrazu jsou tedy následně získány obrazové příznaky nebo také featury. Jedná se o význačné body s určitým rysem nacházejícím se v obraze. Tyto obrazové příznaky jsou detekovány pomocí různých detektorů, což jsou algoritmy fungující na rozličných principech. Velmi oblíbený je SIFT detektor, který byl použit například v projektu [7], ale vždy nemusí vyhovovat danému prostředí a podmínkám. Bude tedy nutné vybrat vhodný detektor

pro pracovní prostředí tohoto projektu.

Nalezené význačné body, tedy příznaky, jsou algoritmem popsány pomocí deskriptorů. Podle těchto deskriptorů je možné konkrétní příznak rozpoznat v jiném obraze, i když třeba došlo k translaci, rotaci, deformaci nebo změně osvětlení v daném obraze. Samozřejmě čím jsou tyto změny radikálnější, tím je těžší stejný obrazový příznak identifikovat se vzorem.[23]



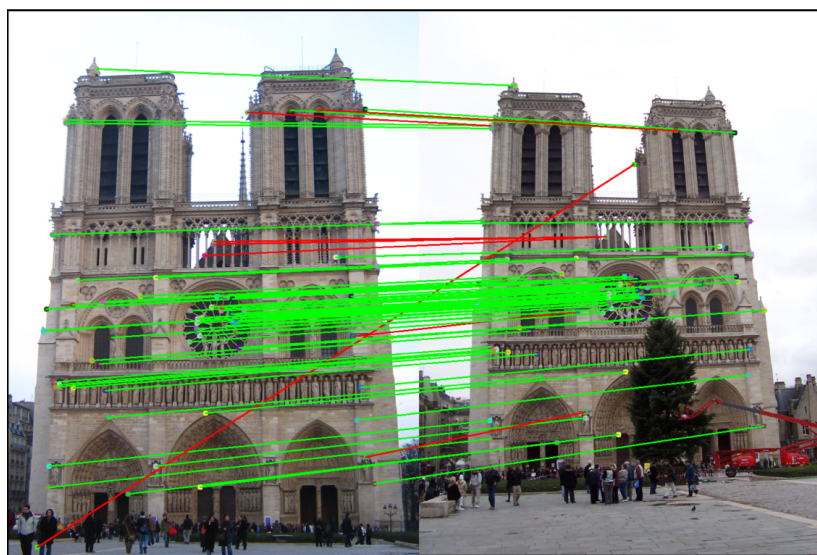
(a) : Význačné body nalezené algoritmem SURF



(b) : Význačné body nalezené algoritmem FAST

Obrázek 3.1: Detekované obrazové příznaky v místnosti laboratoře

Pro samotné porovnání obrazových příznaků se používají takzvané matchery[24]. Jedná se o algoritmy, které porovnávají deskriptory příznaků a určí, zda se jedná o tentýž obrazový příznak. Je dokonce možné určit, jak moc se dané příznaky shodují a každému příznaku přiřadit i několik nejlepších výsledků.



Obrázek 3.2: Porovnávání dvou fotografií stejného objektu. Obrazové příznaky určené jako shodné jsou propojeny barevnou přímkou. Zelené jsou správně určeny jako shodné a červené jsou jako shodné označeny nesprávně. [47]

V této práci budeme uvažovat pouze následující detektory:

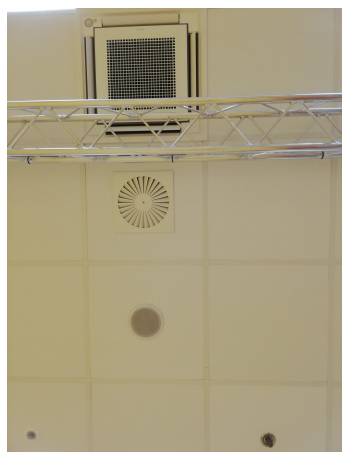
- STAR
- FAST
- SIFT
- SURF
- MSER
- GFTT
- AFD
- SBD
- ORB
- BRISK
- KAZE
- AKAZE

■ 3.1.1 Vybírání vhodného detektoru

Vhodný detektor do kancelářského prostředí laboratoře, v níž se tento projekt vyvíjí, bude nejlépe vybrat na základě několika jednoduchých experimentů. Od detektoru je vyžadováno co největší množství nejkvalitnějších obrazových příznaků.

Laboratoř je umístěna uprostřed budovy a je téměř úplně odstíněna od přirozeného venkovního zdroje světla. Díky tomu nebude mít počasí ani část dne vliv na změnu osvětlení v místnosti. Strop laboratoře je tvořen monotónním bílým čtvercovým obložením typickým pro kancelářské prostory. Na první pohled jsou na stropu hlavními význačnými objekty výfukové průduchy ventilace, reproduktory, detektory požáru a sprinklerové hasící zařízení.

Zjistíme kolik příznaků je každý detektor schopen nalézt. Nejdříve ve statickém obrazu a následně při translaci a rotaci robotu. Potom porovnáme deskriptory všech obrazových příznaků a to zase jak pro statický obraz, tak při translaci a rotaci robotu. Pro detektory STAR, FAST, MSER, GFTT, AFD a SBD je jako deskriptor použit SIFT.



(a) : Všechny důležité prvky na stropu, tedy ventilační výstup, traverza, reproduktor, hasicí zařízení, pohybový senzor.



(b) : Celkový pohled na laboratoř

Obrázek 3.3: Laboratoř a výrazné prvky na stropu

■ Hledání obrazových příznaků

Nejdříve zjistíme, kolik obrazových příznaků dokáže jaký detektor v pracovním prostředí laboratoře vůbec nalézt a to pro statický obraz zatížený pouze šumem, tak i při pohybu robotu.

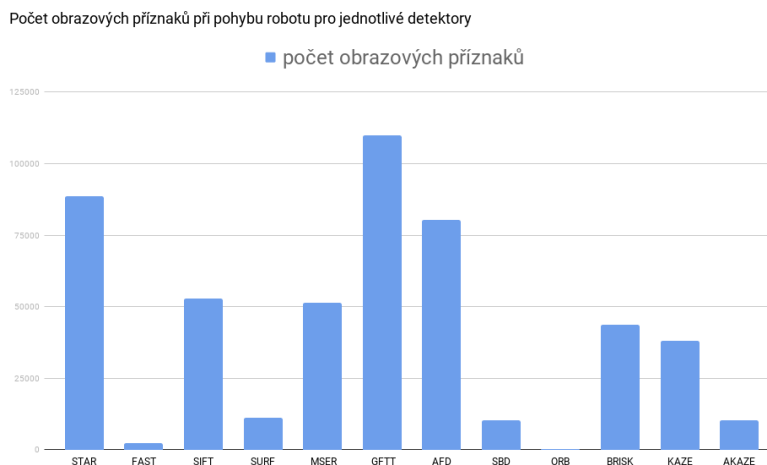
Počet obrazových příznaků při statickém obrazu pro jednotlivé detektory



Obrázek 3.4: Množství obrazových příznaků nalezených jednotlivými detektory ve statickém obrazu

Pokud robot stojí na místě, snímá strop po dobu 18 sekund a detekuje v

něm obrazové příznaky, získáme výsledky na obrázku 3.4. Algoritmus STAR rapidně převýšil ostatní detektory, následuje skupina detektorů AFD, GFTT, SIFT a KAZE. Nejhorší výsledky má na první pohled detektor ORB a FAST.



Obrázek 3.5: Množství obrazových příznaků nalezených jednotlivými detektory při průjezdu částí laboratoře

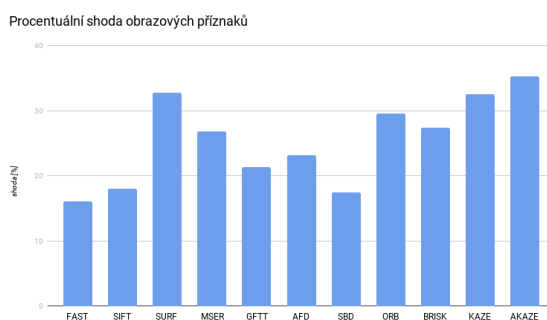
Při snímání stropu v případě pohybujícího se robotu jsou výsledky mírně odlišné. Robot urazil s mírnou rotací dráhu dlouhou přibližně 4 metry. V tomto testu, viz obrázek 3.5, nejvíce detekovaných obrazových příznaků má algoritmus GFTT, následovaný detektorem STAR. Méně obrazových příznaků detekovala skupina AFD, SIFT a MSER. Nejhorší výsledky mají opět detektory ORB a FAST.

Na první pohled je množství detekovaných obrazových příznaků pro nepohybující se robot větší než pro robot jedoucí kratší dráhu. Zatímco v první části dominoval STAR detektor, v druhé byl překonán GFTT detektorem a téměř se mu vyrovnal algoritmus AFD. Tyto tři detektory jsou dobrými kandidáty pro nejvhodnější algoritmus do kancelářského prostředí laboratoře.

■ Porovnávání obrazových příznaků

Další důležitou skutečností je schopnost porovnat deskriptory jednotlivých obrazových příznaků z různých obrazů a správně určit jejich shodu. Jedná se vlastně o kvalitu daného příznaku. Tu ověříme pomocí porovnávacího algoritmu FLANN a to opět pro statický obraz, tak i pro jedoucí robot. Také použijeme filtr zakládající se na Lowe's ratio testu [25], [26], pro lepší porovnávání výsledků.

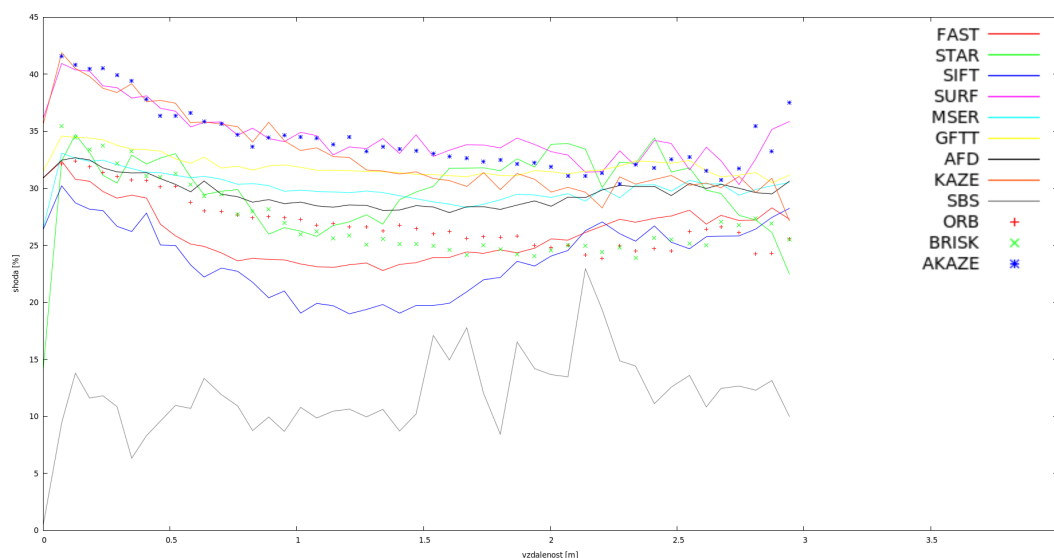
3. Teoretický postup řešení



Obrázek 3.6: Procentuální shoda mezi všemi obrazovými příznaky, tedy každý s každým pro daný detektor, při statickém obraze s thresholdem 0.6

Při porovnávání obrazových příznaků nalezených v téměř identickém obraze, by měli být příznaky většinou stejné a tedy i logicky shodné. Přesto, když nastavíme threshold filtru na hodnotu 0.6, nejlepší výsledky nedosahují ani 40% shody, viz obrázek 3.6. Přes 30% se dostala pouze skupina detektorů SURF, KAZE a AKAZE. Při pokusech s jinou hodnotou thresholdu filtru se procentuální shoda obrazových příznaků zvyšuje, ale poměry mezi detektory se výrazněji nemění.

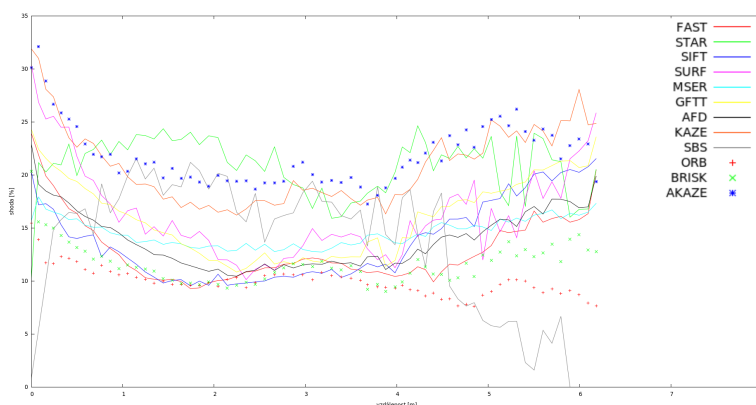
Pro testování s jedoucím robotem budeme měřit shodu všech obrazových příznaků daného detektoru v závislosti na ujeté vzdálenosti. Jinými slovy je měřena shoda stejných obrazů, které se s ujetou vzdáleností robotu stále méně překrývají a jejich shoda by měla klesat. Pro měření přesné pozice robotu je využit systém Vicon a bude opět použitý filtr s různým nastavením thresholdu.



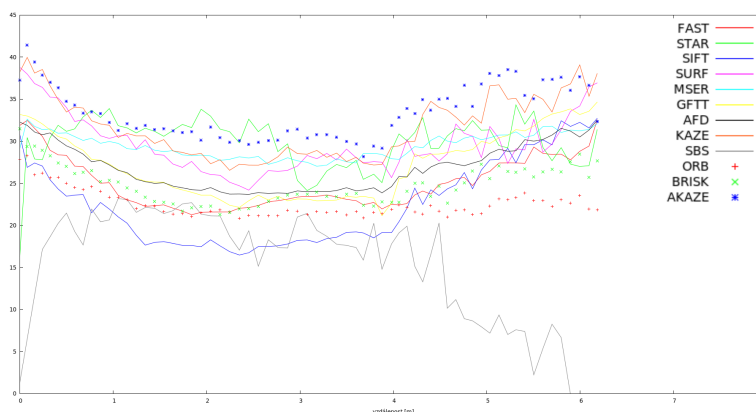
Obrázek 3.7: Shoda obrazových příznaků pro jedoucí robot s thresholdem filtru 0.8 a trasou na 3m

V grafu na obrázku 3.7 má viditelně nejvyšší hodnotu v nulové vzdálenosti a blízkém okolí skupina tří detektorů SURF, KAZE a AKAZE. Jedná se o stejnou skupinu, která dosáhla nejlepších výsledků při porovnávání obrazových příznaků se statickým obrazem, viz 3.6.

Tato trojice má nejvyšší hodnotu shody téměř v celém průběhu jízdy robotu. Podobný výsledek je vidět v grafech na obrázku 3.8, kde trojice dosahuje nejlepších výsledků na začátku a také téměř v celém průběhu. Počáteční hodnoty v obrázku 3.8a s thresholdem 0.6 se dokonce téměř shodují s grafem 3.6 pro statický obraz.



(a) : Shoda obrazových příznaků pro jedoucí robot s thresholdem filtru 0.6



(b) : Shoda obrazových příznaků pro jedoucí robot s thresholdem filtru 0.8

Obrázek 3.8: Shoda obrazových příznaků pro jedoucí robot s trasou na 6 m

Důvod proč se na obrázcích 3.8 s narůstající vzdáleností nesnižuje procentuální shoda je ten, že na třech metrech ujeté vzdálenosti robot detekuje téměř identický výfuk ventilace jako na své startovní pozici. To poukazuje na fakt, že monotónní a opakující se prvky na stropu místnosti značně ztěžují rozpoznávání a tedy i lokalizaci robotu.

■ Shrnutí

Z výsledných grafů a dat lze říci, že pro největší detekovaný počet obrazových příznaků je nejvhodnější STAR, GFFT nebo AFD detektor. Z hlediska kvality nalezených příznaků a možnosti správné shody nejlépe vyšla skupina detektorů SURF, KAZE a AKAZE. Pro tuto práci bude vybrána druhá skupina, protože je nutné co nejlépe porovnávat snímané okolí se známou mapou.

V tomto experimentu byly detektory, deskriptory i matchery použity v základním nastavení. Nebyly jim tedy nastaveny žádné parametry ani proměnné, které by mohli výrazně ovlivnit chování některých algoritmů a naprosto změnit výsledky všech testů.

Toto nastavení parametrů by muselo být dlouhodobě upravováno a opakovaně testováno pro každý algoritmus, proto bylo rozhodnuto použít algoritmy v základním nastavení. Také nebyl pro určité detektory vyzkoušen jiný deskriptor než SIFT deskriptor, což je zmíněno výše. A to by mohlo mít vliv na výsledky, i když nejspíše málo výrazný.

Za nejvhodnější volbu můžeme považovat tedy detektory SURF, KAZE a AKAZE. V další sekci při vytváření mapy budou zkusmo použity všechny tři algoritmy pro detekci obrazových příznaků.

■ 3.2 Mapa

Jakmile jsou detekovány příznaky v obrazovém vstupu, je možné je porovnat s obrazovými příznaky v mapě a lokalizační metodou zjistit pozici a orientaci robotu. Nejdříve je nutné vytvořit samotnou mapu a v ní je teprve možné lokalizovat robot. V mapě musí být zaznamenány jednotlivé obrazové příznaky a jejich pozice.

Samotná mapa musí být vhodně reprezentovaná. Přestože je mřížka obsazenosti v robotice velmi často používaná, pro toto uspořádání bodů je nevhodná. Body jsou v určitých oblastech velmi blízko sebe a jiné oblasti jsou zase úplně prázdné. Proto byla původně vybrána jednoduchá hash mapa, kde jednotlivé prvky obsahovaly popis obrazového příznaku a jeho pozici. Postupem času se hash mapa ukázala být v kombinaci s lokalizační metodou výpočetně neúspěšná a byla nahrazena reprezentací typu KD-Tree .

■ 3.2.1 Tvorba mapy

Pozice každého obrazového příznaku musí být nejdříve spočítána relativně k pozici robotu a následně převedena do globálních souřadnic daných pracovním prostorem v místnosti. Pro snadné a poměrně přesné určení pozice robotu v pracovním prostoru při vytváření mapy použijeme systém Vicon, který bude detailněji popsán níže.

Systém Vicon je zde použit jako jednorázová pomoc při lokalizaci robotu pouze při první tvorbě mapy. Určování pozice robotu by při prvním tvoření mapy mohla být realizována jinou metodou, například drahými laserovými dálkoměry, které budou po vytvoření mapy nahrazeny pouze levnou kamerou.

Pozici obrazového příznaku vůči robotu vypočítáme párováním nebo triangulací. Všechny popisy obrazových příznaků s jejich pozicí jsou následně uloženy do textového dokumentu a připraveny pro načtení lokalizační metodou do mapy.

Aby bylo možné použít párování obrazových příznaků, je nezbytné najít pomocí matcheru totožný příznak ve dvou obrázcích z různých pozic robotu a teprve pro tento příznak vypočítat pozici. Pro porovnávání příznaků je použit Brute-force matcher, vyhledávající nejlepší páry dvou deskriptorů daných obrazových příznaků.

Robot ujede určitou vzdálenost, porovná obrazové příznaky z aktuálního snímku a ze snímku na předchozí pozici. Snímky musí mít překryv pro nalezení dvojic shodných příznaků, následně se pro každou shodu dopočítá pozice vůči robotu.

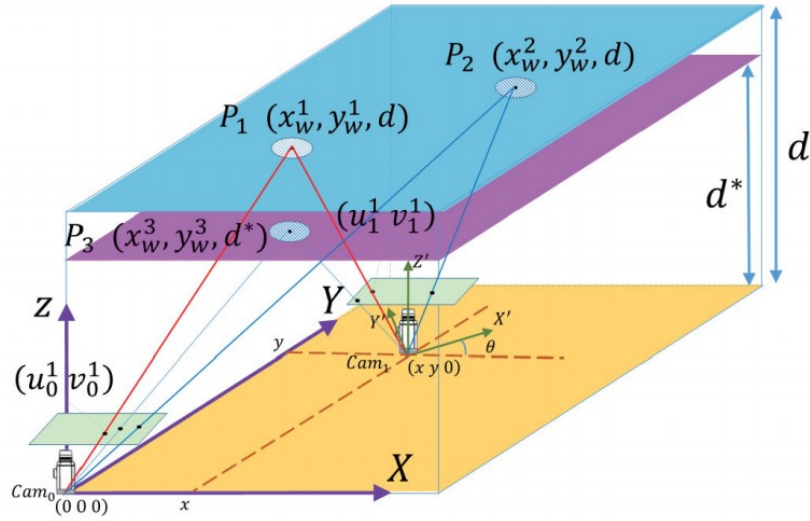
■ Výpočet

Při výpočtu vycházíme ze znalosti výšky stropu, která je 3.5 metru od robotu, ale některé prvky, jako například nosníky a kamery systému Vicon, se nacházejí níže. Pro minimální výšku těchto objektů je v této práci uvažována hodnota 2.5 metru. Všechny obrazové příznaky detekované mimo tento interval jsou automaticky klasifikovány jako nevhodné a není s nimi dále počítáno.

Následně je nutné vybrat skenovací krok, ujetou vzdálenost robotu pro porovnání dvou obrazů, kterým bude mapa tvořena. Kamera snímá povrch

stropu o rozměru přibližně 3 m x 4 m a skenovací krok musí být takový, aby vznikl dostatečný překryv. Mapa byla nejdříve vytvořena se skenovacím krokem o velikosti jednoho metru, ale obsahovala malé množství správných obrazových příznaků.

Dále byl použit krok 0.5 metru a 0.25 metru. Nakonec byl zvolen krok o velikosti 0.1 metru, kdy jsou sice opakovaně nacházeny stejné obrazové příznaky, ale překryv obrazů je veliký a nepřesnost pozice příznaku není tak zásadní.



Obrázek 3.9: Párování obrazových příznaků [6]

Výpočet podle obrázku 3.9 provedeme nejdříve v rovině x , tím získáváme trojúhelník. Pro konkrétní obrazový příznak, například P_1 v obrázku, vypočítáme jeho úhel φ_1 od osy Z pro první pozici robotu a φ_2 od osy Z' pro druhou pozici robotu,

$$\varphi_1 = \frac{\pi}{2} - \text{atan2}(x_1 - s_x, o_x) \quad (3.1)$$

$$\varphi_2 = \frac{\pi}{2} + \text{atan2}(x_2 - s_x, o_x) \quad (3.2)$$

kde x_1 je pozice x obrazového příznaku v obrazu kamery na první pozici robotu, x_2 je pozice x obrazového příznaku v obrazu kamery na druhé pozici robotu, s_x je pozice x optického středu kamery a o_x je ohnisková vzdálenost kamery pro osu x .

Dále dopočítáme třetí úhel v trojúhelníku θ a sinovou větou zjistíme délku strany \mathbf{b} ležící mezi druhou pozicí robotu a vrcholem na stropu obsahující obrazový příznak,

$$\theta = \pi - \varphi_1 - \varphi_2 \quad (3.3)$$

$$b = \frac{\sin(\varphi_1)}{\sin(\theta)} \cdot x_l \quad (3.4)$$

kde x_l je vzdálenost ujetá robotem ve směru osy x . Od úhlu φ_2 odečteme $\frac{\pi}{2}$, používané pro případ bodu P_2 a vypočítáme vzdálenost d , výšku příznaku na stropu, a pozici x_R ,

$$\varphi'_2 = \text{abs}(\varphi_2 - \frac{\pi}{2}) \quad (3.5)$$

$$d = \cos(\varphi'_2) \cdot b \quad (3.6)$$

$$x_R = \sin(\varphi'_2) \cdot b \quad (3.7)$$

kde x_R je pozice obrazového příznaku vůči robotu v ose x .

Totožný postup se provede v rovině y , kde dostaneme stejné výsledky a místo x_R dostaneme y_R .

Převod do globálních souřadnic x'_R a y'_R uděláme pomocí transformační matice,

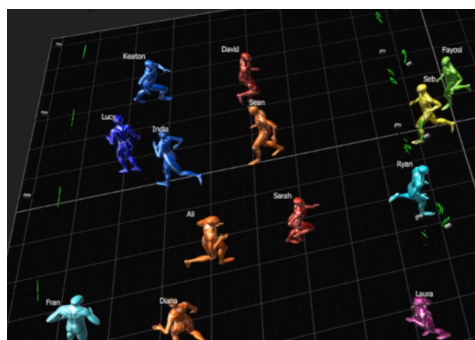
$$\begin{bmatrix} x'_R \\ y'_R \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & x_v \\ \sin(\phi) & \cos(\phi) & y_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_R \\ y_R \\ 1 \end{bmatrix} \quad (3.8)$$

,kde x_v a y_v jsou pozice a ϕ natočení ze systému Vicon.

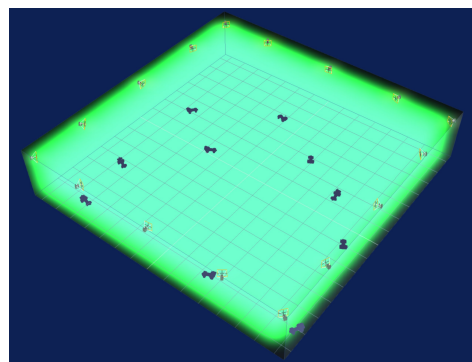
■ Vicon

Systém Vicon[27] je jednou z metod pro záznam a analýzu pohybu, patří mezi takzvané motion capture systems. Tyto systémy se používají nejen v robotice, filmovém a herním průmyslu, ale také například ve zdravotnictví. Cílem je získat přesný záznam pozice a pohybu jednotlivých význačných bodů.[28]

Princip metody spočívá ve vhodném rozmístění soustavy kamer snímající konkrétní scénu z několika pohledů. Kamery jsou navzájem synchronizovány a spolu se zdroji světla (jedná se o blízké infračervené spektrum) kolem objektivů vyzařují světlo, které reaguje s umístěnými reflexními značkami (markery) a vytváří obraz pohybujícího se předmětu. Snímání systémem Vicon má poměrně vysokou přesnost, tedy kolem desetin milimetru. [28]



(a) : Snímání lidského pohybu pro velkou skupinu pohybujících se objektů



(b) : Rozmístění kamer pro snímání pohybů robotů

Obrázek 3.10: Ukázka systému Vicon [27]

V této práci bude zmíněná metoda použita pouze pro přesnou lokalizaci robotu při tvorbě mapy. Takto vytvořená mapa je velikostně omezena plochou snímanou systémem Vicon a za její hranicí není možné obrazové příznaky do mapy přiřazovat. Proto bylo rozhodnuto souřadnicový systém převést do souřadnic daných systémem Vicon.



(a) : Kamery snímající prostor se zdrojem infračerveného světla



(b) : Markery detekované kamerami

Obrázek 3.11: Komponen systému Vicon [28]

Uložení a načtení mapy

Po nasnímání pracovního prostředí příslušnou metodou je daný vektor deskriptorů a pozic obrazových příznaků uložen do textového dokumentu. Každý

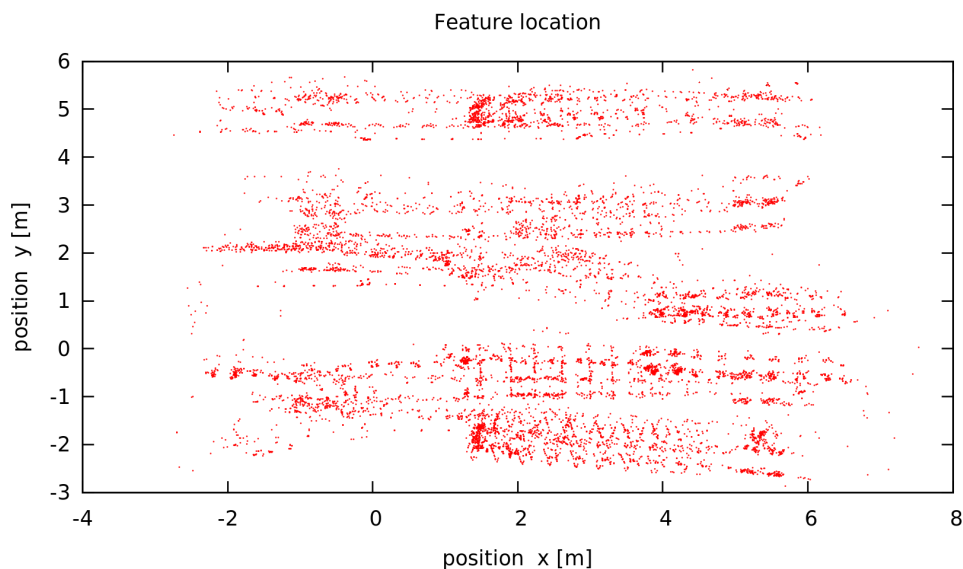
řádek v tomto souboru obsahuje popis konkrétního příznaku a jeho danou pozici. Pokud pro daný deskriptor existuje více pozic, jsou připojeny na konec řádku za předchozí pozici.

Následně lokalizační metoda načte data z textového dokumentu, pozice jsou zapsány do KD-Tree reprezentace a jsou k nim přiřazeny deskriptory jednotlivých obrazových příznaků.

■ Shrnutí

Byla vytvořena mapa okolí robotu, jedná se o oblast pracovní plochy systému Vicon. Tato pracovní plocha má rozměr 12 x 9 metrů a téměř uprostřed stojí opěrný sloup budovy.

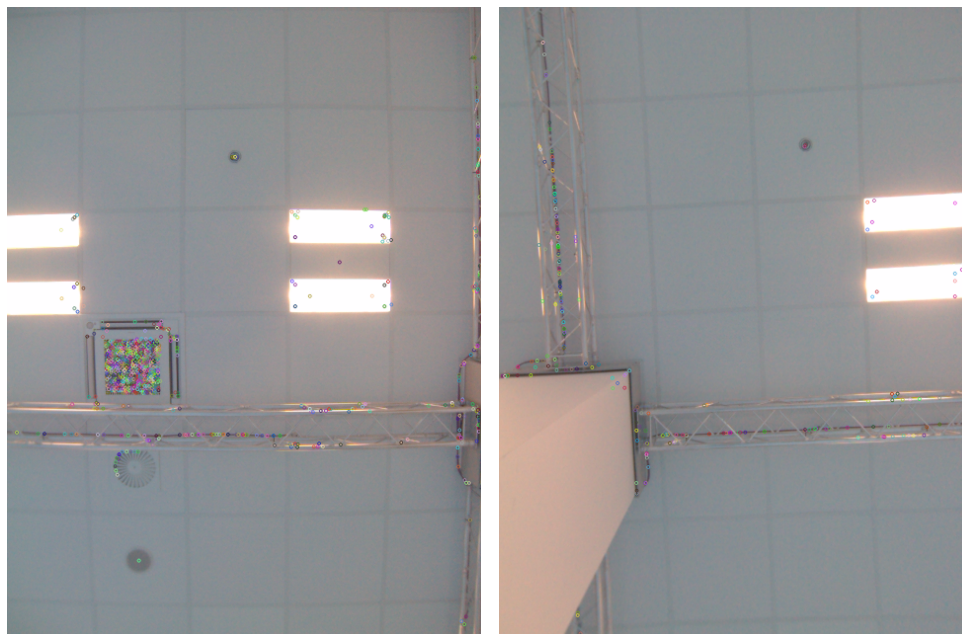
Mapa celého pracovního prostoru vytvořená pomocí algoritmu SURF obsahuje 9655 obrazových příznaků, mapa vytvořená KAZE má 15 465 obrazových příznaků a mapa vytvořená AKAZE má 10 641 obrazových příznaků. Tento výsledek odpovídá předešlému testu detektorů, kde algoritmus AKAZE, následovaný KAZE a SURF, dosáhl nejlepších výsledků v daných podmínkách laboratoře.



Obrázek 3.12: Mapa vytvořená detektorem AKAZE

Na obrázku 3.12 je možné si v mapě povšimnout několika význačných množin bodů na souřadnicích $[6,3]$, $[2,3]$, $[-1,3]$ a $[4,-0.5]$. Jedná se právě o

výstupy ventilace, které patří mezi nejlépe detekovatelné objekty na stropu laboratoře a proto je zde tolik nalezených obrazových příznaků. Už nyní je možné předpokládat, že právě v těchto místech se bude robot nejlépe lokalizovat.



(a) : Obrazové příznaky detekované na výstupu z ventilace

(b) : Obrazové příznaky detekované na opěrných sloupech budovy

Obrázek 3.13: Obrazové příznaky detekované na stropu laboratoře

Dalším dobrým orientačním objektem jsou traverzy pro umístění systému Vicon a rohy opěrných sloupů budovy, které je možné rozpoznat na souřadnicích $[6,5]$, $[2,5]$ a $[5,-2]$. Na základě procházení jednotlivých snímků je možné tvrdit, že i požární hlásiče a hasící zařízení jsou i přes svůj malý rozměr a nenápadnost zaznamenány v mapě. Nejméně významnými a nejspíše také nejméně hodnotnými orientačními body jsou přechody jednotlivých čtvercových dlaždic obložení stropu místnosti a světla.

3.3 Lokalizace

Po vytvoření mapy je nutné robot v mapě lokalizovat. V této práci bude použit pro určení pozice robotu Částicový filtr. Jednou z výhod této lokalizační metody je uchovávání více hypotéz pro pozici robotu. Tento fakt je vhodný pro lokalizaci v prostředí s podobnými vzory. Například, kdyby byl použit oblíbený Kalmanův filtr a došlo by ke špatnému odhadu první pozice, robot

by se nejspíš nedokázal lokalizovat správně. Další výhodou jsou již získané zkušenosti s touto metodou a její znalost.

■ 3.3.1 Částicový filtr

Částicový filtr[29], [30], [31] je metoda odhadující pozici a orientaci robotu pomocí částic a skládá se ze sensorového a pohybového modelu.

Částice jsou náhodně rozprostřeny po známé mapě. Každá částice reprezentuje robot, tedy jeho pozici a orientaci. Dále mají jako parametr váhu w s hodnotou $\langle 0,1 \rangle$ určující pravděpodobnost, že se jedná o pozici reálného robotu.

Následně je každá částice posunuta podle pohybu robotu. To zajišťuje pohybový model, kde je počítáno s náhodnou složkou pohybu a případným šumem. Pokud nějaká částice vyjede z mapy nebo se ocitne v překážce, je odstraněna.

Dále je na řadě ohodnocení částic. Každé částici je přiřazena váha podle sensorových dat a sensor modelu. Částice jsou potom normalizovány, aby jejich váha byla v hodnotě $\langle 0,1 \rangle$ a jsou náhodně vybírány. Následně jsou chybějící částice vygenerovány do mapy s náhodnou pozicí.

Celý algoritmus je možné popsat pseudo kódem:

Listing 3.1: Pseudo kód částicového filtru

```

1 for i=0 to M do
2     Xt= sample_motion_model(Ui,Xt-1)           //posun castic
3     Wt= sensor_model(Zt,Xt)                   //ovahovani castic
4     a=a+Wt                                     //pricteni normalizacniho faktoru
5     St=St+⟨Xt,Wt⟩                             //pridani castic
6 end for
7
8 for i=0 to M do                               //normalizace
9     Wt=Wt/a
10 end for
11
12 for i=0 to M do
13     Low_Variance_Sampled_Particles //nahodne vybrane castice
14     St=St+⟨Xt,Wt⟩                       //pridani castic
15 end for

```


■ Senzorový model

Senzorový model určuje důvěryhodnost naměřených dat okolního prostředí a přiřazuje váhu jednotlivým částicím. Pro dálkové senzory jako je laser nebo sonar je senzorový model složen ze čtyř pravděpodobnostních modelů. Jedná se o model šumu, nečekané překážky, náhodného měření a maximální vzdálenosti.

V případě tohoto projektu má senzorový model jiný tvar. Sensorová data vstupující do modelu jsou obrazové příznaky detekované v aktuálním obrazu z kamery. Následně pro každou částici je vypočítáno jakou oblast stropu by mohl robot na její pozici vidět a jaké příznaky se v této oblasti mapy nacházejí. Tyto obrazové příznaky z mapy jsou porovnávány s příznaky vstupujícími do modelu.

Porovnávání je stejně jako při tvorbě mapy realizováno Brute-force matchingem, vyhledávajícím nejlepší páry dvou deskriptorů daných obrazových příznaků. Každý pár příznaků vypočtených jako nejhodnější obsahuje informaci o odlišnosti, zvanou *distance*. Právě pomocí tohoto parametru sestavíme pravděpodobnostní funkci pro ohodnocení částic.

Vytvoříme sumu druhých mocnin všech vzdáleností párů obrazových příznaků pro danou částici a odmocníme.

$$D = \sqrt{\sum_{i=1}^M S_{dist}^2} \quad (3.9)$$

Potom přiřadíme dané částici váhu podle vzorce,

$$W = \frac{1}{D + 10^2} \cdot N \quad (3.10)$$

kde N je počet shod pro danou částici. V dalším kroku algoritmu je váha částice normalizována, aby váha každé částice nabývala hodnot v intervalu $\langle 0,1 \rangle$.

■ Odometrie

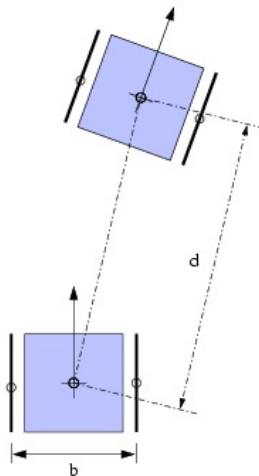
Jako základní způsob pro určování polohy je považována odometrie, která umožňuje měřit relativní změnu pozice mezi dvěma časovými okamžiky. Ve své podstatě odometrický polohovací systém převádí naměřenou rotaci kol mobilního robotu na jeho aktuální polohu a orientaci. Rotace kol může být zjišťována přímo z akčních zásahů na motory, nebo z optických, magnetických, či jiných obdobných čidel snímající přímo jejich otáčení.

První řešení je zpravidla konstrukčně jednodušší, ovšem je zatíženo přídatnou nesystematickou měřicí chybou v případě, že pohonná kola robotu se protáčejí vlivem povrchu nebo blokací těla robotu. Absolutní poloha se obecně získá integrací jednotlivých změn pozice:

$$x_1 = x_0 + \int_{t_0}^{t_1} \frac{dx}{dt} dt \quad (3.11)$$

kde $\frac{dx}{dt}$ je změna pozice neboli rychlost, x_0 je pozice v čase t_0 a x_1 je pozice v čase t_1 .

Vlastní poloha robotu je určována na základě znalosti kinematiky daného mobilního robotu, velikosti kola a počtu impulsů na jednu otáčku. Hlavně zvolený model robotu ovlivňuje typ řízení. Robot Turtlebot vybraný pro tuto práci je non-holonického typu a jedná se o diferenciální podvozek.



Obrázek 3.14: Znázornění pohybu diferenciálního vozidla a měřené veličiny [33]

Další typy robotických modelů jsou například auto a všesměrové roboty.[32] Základním rysem non-holonického modelu jsou dvě nezávisle poháněná kola nebo nápravy. Pokud se obě točí stejně rychle stejným směrem, robot jede rovně. Pokud se naopak točí opačným směrem, točí se celý robot na místě kolem středu mezi nápravami. To je také nejčastější referenční bod.

Bohužel každé měření kol obsahuje chybu. Tato chyba je většinou sama o sobě malá, ale při větší dráze může mít zásadní vliv. Čím více změn pozice robotu je měřeno, tím více se chyba akumuluje a tím více se vypočtená relativní pozice liší od skutečné [33], [32].

■ Motion model

Pohybový model zajišťuje posun jednotlivých částic podle odometrie a přidává nepravidelnou odchylku. Vstupními parametry je tedy již výše zmíněná odometrie robotu.

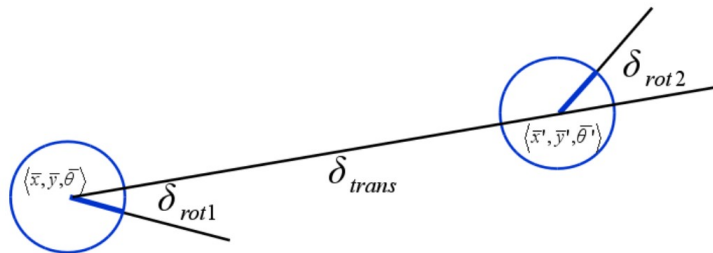
Odchylka je realizována pomocí 4 parametrů, α_1 , α_2 , α_3 a α_4 . Hodnota těchto parametrů pro robot Turtlebot byla experimentálně zjištěna na předmětu B3M33MKR. Jejich hodnoty jsou :

$$\alpha_1 = 0.1 \quad (3.12)$$

$$\alpha_2 = 0.18 \quad (3.13)$$

$$\alpha_3 = 0.03 \quad (3.14)$$

$$\alpha_4 = 0.008 \quad (3.15)$$



Obrázek 3.15: Znázornění parametrů pohybového modelu [48]

Náhodná odchylka je do odometrie přidána podle následujících vzorců,

$$\hat{\delta}_{rot1} = \delta_{rot1} + \text{sample}(\alpha_1 |\delta_{rot1}| + \alpha_2 |\delta_{trans}|) \quad (3.16)$$

$$\hat{\delta}_{trans} = \delta_{trans} + \text{sample}(\alpha_3 |\delta_{rot1}| + \alpha_4 (|\delta_{rot1}| + |\delta_{rot2}|)) \quad (3.17)$$

$$\widehat{\delta}_{rot2} = \delta_{rot2} + \text{sample}(\alpha_1|\delta_{rot2}| + \alpha_2|\delta_{trans}|) \quad (3.18)$$

,kde parametry $\delta_{rot1}, \delta_{rot2}, \delta_{trans}$ jsou zřetelné z obrázku 3.15.

Pozice částice je určena vzorcí

$$x' = x + \widehat{\delta}_{trans} \cdot \cos(\phi + \widehat{\delta}_{rot1}) \quad (3.19)$$

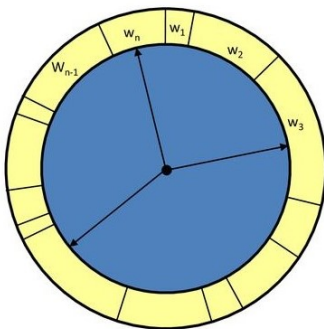
$$y' = y + \widehat{\delta}_{trans} \cdot \sin(\phi + \widehat{\delta}_{rot1}) \quad (3.20)$$

$$\phi' = \phi + \widehat{\delta}_{rot1} + \widehat{\delta}_{rot2} \quad (3.21)$$

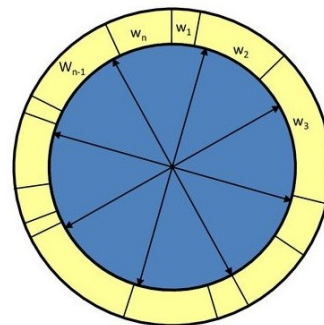
Shrnutí

Pozice robotu je v této metodě určována podle částic a to zejména podle shluků těchto částic na místech, kde se robot s největší pravděpodobností nachází. Shluky vznikají zejména zachováváním částic s vysokou pravděpodobností, tedy váhou w , pomocí ruletového kola nebo jako v této práci Low Variance vzorkováním.

V případě ruletového kola jsou náhodně losovány částice, které zůstanou a jsou brány v potaz. Nevylované jsou odstraněny. Princip je v tom, že čím větší váhu částice má, tím je větší pravděpodobnost že bude vyloována. Tímto způsobem částice s větší pravděpodobností zůstávají častěji než částice s malou pravděpodobností. Princip Low Variance vzorkování je obdobný, ale výpočetně úspornější.



(a) : Vysvětlivka ruletového kola



(b) : Low Variance vzorkování

Obrázek 3.16: Vzorkování Částicového filtru [49]

Už teď je možné předpokládat, že shluky částic se budou nejlépe vytvářet na jasných orientačních bodech v mapě, jako jsou například výfuky ventilace,

traverzy a sloupy. Zároveň to přináší otázku, jestli jsou popisy obrazových příznaků na těchto téměř stejných objektech dostatečně odlišné pro účinné rozpoznání. Pokud nebudou dostatečně odlišné, budou se shluky částic pravděpodobně tvořit vždy na místě všech podobných objektů, kterých je v celém pracovním prostředí velké množství.

Musíme si položit otázku, která ze tří vytvořených map bude pro lokalizaci robotu nejvhodnější. Jestli mapa vytvořená detektorem KAZE s největším počtem detekovaných obrazových příznaků, ale horším porovnáváním než detektor AKAZE, jak je vidět v grafu 3.6, jehož mapa je velice řídko obsazena. To vše objasní experiment.

Kapitola 4

Implementace

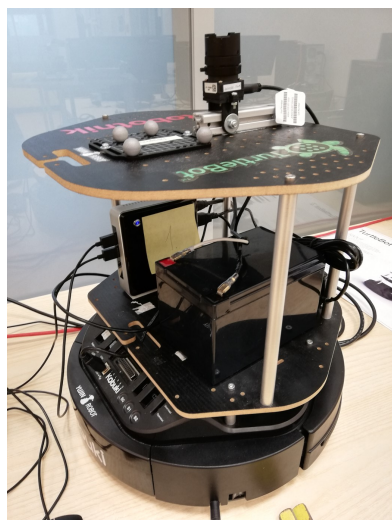
V této kapitole je popsána implementace hlavní lokalizační metody, metody porovnávání obrazových příznaků a tvorba mapy. Také je zde popsán použitý robotický nosič Turtlebot 2 s použitou kamerou a dalším vybavením. Nechybí ani popis ROSu (Robot Operating System), použitých knihoven OpenCV (Open Source Computer Vision Library) a ostatního softwaru.

4.1 Robot

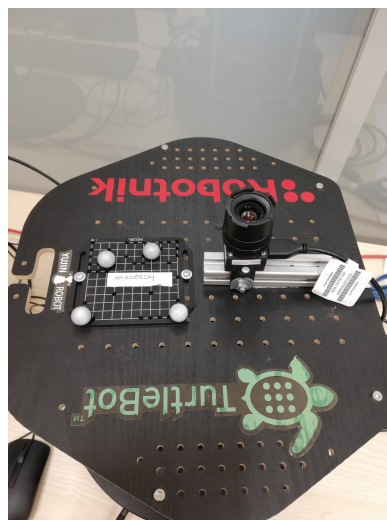
Celý robot se skládá z těchto komponent:

- Turtlebot2
- počítač Intel NUC
- kamera Basler daA1280-54um
- Vicon marker
- napájecí soustava

Základním stavebním prvkem je Turtlebot 2, jedná se o mobilní robot s více patrovou konstrukcí vhodnou pro umístění senzorů a čidel. Jeho detailní



(a) : Robot Turtlebot s umístěnou výbavou



(b) : Umístění kamery a markeru na horní části robotu

Obrázek 4.1: Ukázka konfigurace robotu použitého při experimentu a implementaci

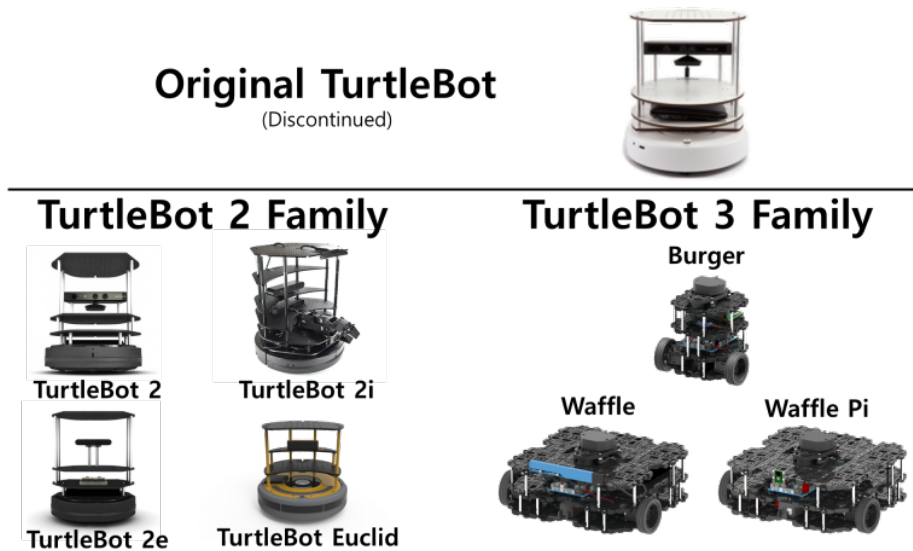
vlastnosti budou rozebrány v další kapitole. Uprostřed vrchní části Turtlebota je umístěna kamera snímající strop a markery systému Vicon pro lokalizaci.

V dolním patře konstrukce je umístěn osobní počítač typu NUC do kterého je zapojen Turtlebot i kamera. Na tomto počítači se systémem Ubuntu 14 a ROS Indigo, běží všechny procesy a je řízen celý robot. Dále je v dolním patře umístěna napájecí soustava počítače, skládající se ze standardního zdroje a baterie.

4.1.1 Turtlebot 2

TurtleBot[34] patří mezi levné osobní mobilní roboty s otevřeným softwarem. První TurtleBot vznikl v prosinci roku 2010 a od té doby bylo vytvořeno několik verzí. Primárně se jedná o tři verze, které mají různé modifikace. Ukázka na obrázku 4.2.

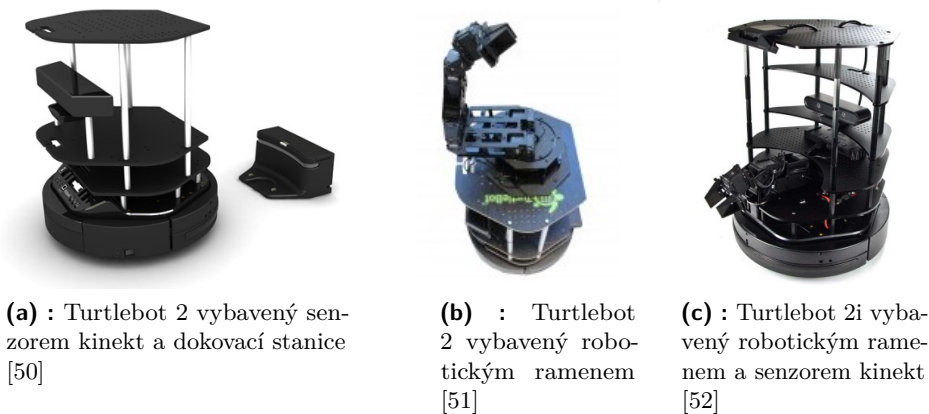
Turtlebot může být vybaven velkým množstvím různých senzorů s volným rozmístěním a dokonce robotickým ramenem, jako na obrázku 4.3b a 4.3c. Velice často je vybaven obvyklým senzorem kinect od firmy Microsoft. Robot ve spodní části obsahuje baterii s kapacitou 2200 mAh, kterou je možné dobít jak pomocí standardní kabelové nabíječky, tak také dokovací stanicí,



Obrázek 4.2: Ukázka verzí Turtlebotů [34]

jako na obrázku 4.3a.

Turtlebot je vyvíjen ve spolupráci s vývojáři systému ROS a dalších. Proto je tento obecně ve vývoji používaný robotický systém vhodnou volbou pro jeho řízení. Co se týče specifikací, samotný robot má plochý válcovitý tvar s průměrem 354 mm a výškou 89 mm. Jeho hmotnost je přibližně 6.3 kg a maximální nosnost je 5 kg.



(a) : Turtlebot 2 vybavený senzorem kinect a dokovací stanice [50]

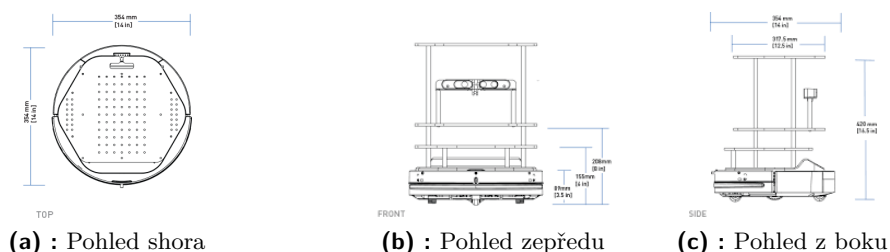
(b) : Turtlebot 2 vybavený robotickým ramenem [51]

(c) : Turtlebot 2i vybavený robotickým ramenem a senzorem kinect [52]

Obrázek 4.3: Ukázka robotů typu Turtlebot

Dále je robot vybaven motorem pro každé ze dvou kol a jedním kolem stabilizačním. Jedná se o diferenciální podvozek typu tank, takže Turtlebot je schopen otáčet se na místě a to až rychlostí 180° za sekundu. Jeho maximální rychlost jízdy je omezena hodnotou 0.65 metrů za sekundu. Také jsou v

těle Turtlebotu zabudovány tři senzory tlaku, tedy nárazníky, například pro snímání kontaktu s překážkou. Další senzory jsou umístěny na kolech, kde detekují sjetí z povrchu, což mohou způsobit negativní překážky nebo schody.



Obrázek 4.4: Rozměr robotu Turtlebot 2 [34]

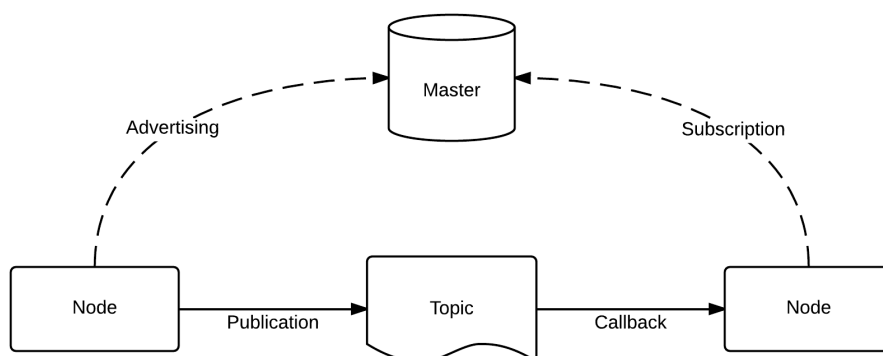
■ ROS

ROS(Robot Operating System)[35] je nástavba operačního systému od firmy Willow Garage pro tvorbu robotického softwaru. Je to framework obsahující nástroje, knihovny a strukturu pro snazší vytváření složitých robotických systémů. Umožňuje sdílení a využívání metod napříč pracovišti a institucemi.

ROS je otevřený software, který vznikl roku 2010 a nyní má několik verzí. Nejpoužívanější verze jsou Indigo, Kinetic, Lunar, Melodic. Tento software je možné používat zejména na unixových operačních systémech, jako je Ubuntu nebo Mac OS X. ROS umožňuje implementaci ve většině moderních programovacích jazycích, jako například Python, C++, Lisp, Java a Lua.

Hlavní souborovou jednotkou jsou balíčky (packages), ty obsahují knihovny, procesy a data metod a kódů v nich uložených. Komunikace je realizována grafem, kde každý senzor nebo proces (node) reprezentuje uzel. Hrany v tomto grafu jsou reprezentovány kanály(topic), které jednotlivé procesy vytvářejí a posílají s jejich pomocí data. Data jsou předávána ve zprávách(messages), jedná se o datové struktury s různým typem podle služby, která je posílá.

Dalším důležitým formátem jsou bagy (bags), jedná se o zálohy dat z ROSovských zpráv v určitém časovém intervalu. Jedná se o způsob ukládání dat, posílaných mezi procesy, a dat ze senzorů. Tyto balíčky jsou velmi důležité při testování algoritmů, protože je lze jednou nahrát a následně přehrávat a simulovat danou situaci.



Obrázek 4.5: Obrazce použité pro kalibraci kamery [35]

■ 4.1.2 Hardware

■ Kamera

Záznam obrazu je realizován kamerou Basler daA1280-54uc [36], vybavenou objektivem s ohniskovou vzdáleností 4 mm. Rozlišení snímaného obrazu je 1280 x 960 pixelů a jedná se o barevný obraz. Zorný úhel kamery je 60° a na stropu ve vzdálenosti 3.5 metrů od robotu snímá oblast o rozměru přibližně 4 x 3 metry. Kamera je napájena přímo USB portem připojeným rovnou do počítače.

Pro plnou funkčnost a ovladatelnost kamery byl použit software Pylon[37], který obsahuje ovladače a další nástroje. Tento software je vydáván přímo výrobcem kamery, firmou Basler, a jedná se o otevřený software s možností úpravy.

■ Počítač Intel NUC5i5RYK

Centrální řídicí jednotkou robotu je počítač Intel NUC5i5RYK [38]. Ovládá Turtlebot, přijímá obraz z kamery a provádí veškeré výpočty. Na počítači je nainstalovaný operační systém Ubuntu 14, ROS Indigo a další potřebný software. Hlavní výhodou tohoto počítače je jeho malý rozměr a hmotnost, takže je velice vhodný pro umístění na Turtlebot.



(a) : kamera Basler daA1280-54uc [36]



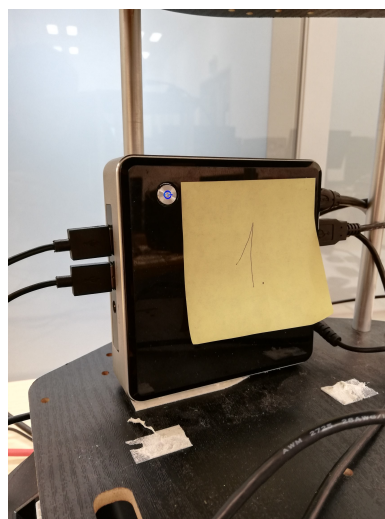
(b) : objektiv použit s kamerou Basler daA1280-54uc [41]

Obrázek 4.6: Ukázka snímacího systému robotu

Turtlebot je k počítači připojen a ovládán přes USB port, stejně jako kamera. Dále jsou k dispozici další dva USB porty a mini HDMI port pro připojení monitoru. Počítač je napájen zdroje s výstupním napětím 19 V.



(a) : Řídící počítač Intel NUC5i5RYK [53]



(b) : Řídící počítač Intel NUC5i5RYK se všemi periferemi

Obrázek 4.7: Ukázka počítače NUC5i5RYK

Procesor	Intel Core i5-5250U
Grafická karta	Intel HD Graphics 6000
Paměť RAM	16 GB
Uložiště	230 GB
Porty	4x USB 3.0, mini HDMI 1.4a, mini DisplayPort 1.2, 3,5mm Jack
Síť	Intel Wireless-AC 7265, Rj45 port

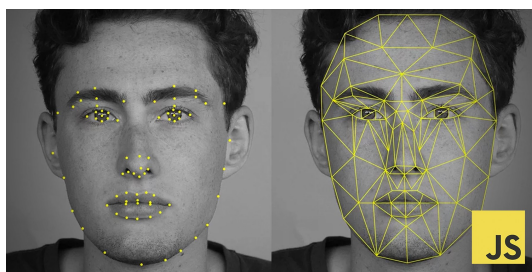
Tabulka 4.1: Parametry počítače Intel NUC5i5RYK

4.2 Implementované metody

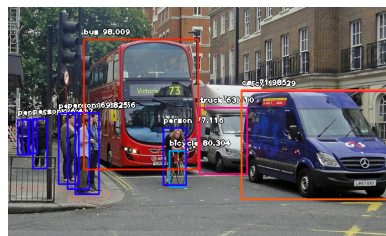
4.2.1 Open CV

OpenCV (Open Source Computer Vision Library) [39] je soubor knihoven pro počítačové vidění a strojové učení. Tyto knihovny jsou volně dostupné a upravitelné. Poskytují základní metody pro práci s obrazem a různé algoritmy pro jeho zpracování. OpenCV má více než 47 tisíc uživatelů a je používáno ve výzkumu, ale i v komerčním prostředí.

Knihovny OpenCV obsahují více než 2500 algoritmů a metod z oblasti počítačového vidění a strojového učení. Tyto algoritmy mohou být použity například na detekování a poznávání obličejů, identifikaci objektů, klasifikaci lidských gest ve videu, sledování objektu, extrakci 3D modelu, tvorbu 3D mraků bodů ze stereo kamer, spojování obrázků pro zvýšení rozlišení scény, hledání podobných obrazů, odstranění rudých očí z obrazů s využitím blesku, sledování pohybu očí a další. Interface podporuje jazyky C++, Python,



(a) : Rozpoznávání obličeje pomocí knihoven OpenCV [54]



(b) : Detekce objektů pomocí knihoven OpenCV [55]

Obrázek 4.8: Ukázka schopností knihoven OpenCV

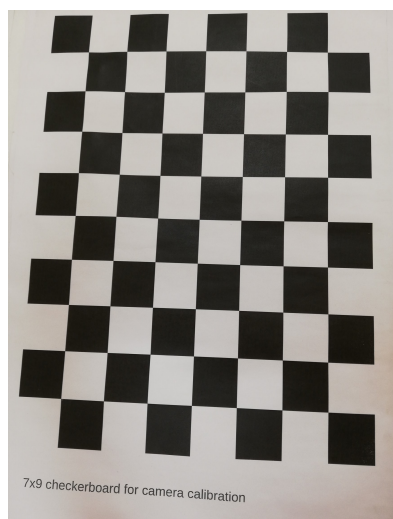
Java, Matlab a širokou škálu operačních systémů, například Windows, Linux, Android a Mac OS. Součástí algoritmů je detailní dokumentace a popis použité metody. Dále existuje aktivní komunitní fórum pro řešení chyb a problémů.

OpenCV se široce používá po celém světě, například pro sledování důlního zařízení v Číně, pro rychlé rozpoznávání obličejů v Japonsku nebo pro detekci topících se lidí v bazénech po celé Evropě. Dále je široce využíváno v robotickém vývoji a továrním průmyslu, třeba pro kontrolu etiket produktů.

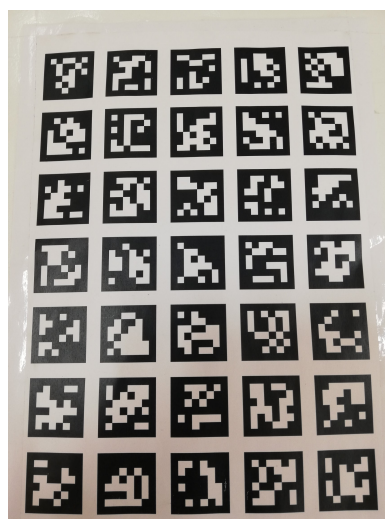
V této práci jsou použity z knihoven OpenCV pouze algoritmy na základní úpravu obrazu, kalibraci kamery, detekci obrazových příznaků v obraze a jejich porovnávání.

4.2.2 Příprava

Prvním krokem před implementací metod bylo nutné provést zaostření a kalibraci kamery. Kameru pevně umístěnou na vrchní části robotu zaostříme na vzdálenost snímaného stropu. Kalibraci nejdříve provedeme pomocí ROSovského balíčku `camera_calibration` využívající knihoven OpenCV a šachovnice. Bohužel hodnoty získané touto kalibrací se ukázaly být nesprávné. Tomu napovídal i fakt, že kolegové s podobným typem kamery a jiným detektorem měli diametrálně odlišné hodnoty.



(a) : šachovnice pro kalibraci balíčkem `camera_calibration`



(b) : Apriltagy pro kalibraci kamery

Obrázek 4.9: Pomocné obrazce pro kalibraci kamery

Následně byl použit balíček Apriltag camera calibration [40] a byly získány tyto hodnoty:

$$C_M = \begin{bmatrix} 1089.811186 & 0.000000 & 642.894805 \\ 0.000000 & 1093.843267 & 478.331197 \\ 0.000000 & 0.000000 & 1.000000 \end{bmatrix} \quad (4.1)$$

$$D_{coef} = [-0.205646 \quad 0.039258 \quad -0.003634 \quad 0.001089 \quad 0.000000] \quad (4.2)$$

kde C_M je matice kamery a D_{coef} jsou distorzní koeficienty.

Vytvoříme v ROSu workspace a použijeme balíček (package) **camera_node** z práce [41], [42], ve kterém budeme naši metodu implementovat. Dále použijeme v balíčku předpřipravenou třídu **camera_Node**, která bude přijímat obraz z kamery a následně ho bez jakékoliv úpravy posílat na kanál(topic) s názvem **/topCamera/image**. Také zajistíme přijímání informace o pozici robotu ze systému Vicon. K tomu je použit balíček **vrpn_client_ros**, je nutné se připojit na stejnou wi-fi síť jako systém Vicon a následně je pozice robotu přijímána ve formě zpráv z kanálu (topic).

Využijeme možností systému ROS a budeme vytvářet rosbag při průjezdu robotu laboratoří. Jedná se o ukládání veškerých naměřených dat a komunikace v systému ROS, tedy například kanálů (topic) **/cameraTop/image** s obrazem a **/Marker04** s pozicí robotu.

Při následném přehrávání rosbagu je simulována jízda robotu a zároveň mohou být detekovány obrazové příznaky a prováděny výpočty. Hlavní výhodou je opakovatelnost přehrávání při jakékoliv změně v metodě, ať už se jedná o výpis v kódu nebo změnu parametrů. Tato opakovatelnost výrazně zrychluje a usnadňuje práci. Následně vytvoříme několik záznamů o průjezdu robotu s různou trajektorií a startovním bodem, podle potřeby.

4.2.3 Porovnávání obrazových příznaků

První naimplementovaná metoda slouží pro porovnávání obrazových příznaků a to jak z hlediska jejich kvality, tak kvantity.

Statický obraz

Pro porovnávání a počítání obrazových příznaků při statickém obraze, ovlivněným jenom šumem, je použita mírně modifikovaná třída pro záznam obrazu. Tedy modifikujeme soubor **cameraNode.cpp** posílající obraz z kamery a vznikne **Detection.cpp**.

V této třídě detekujeme obraz rovnou z kamery. Ten je zbaven deformace s použitím funkce **undistort** z knihoven OpenCV a jsou na něm detekovány obrazové příznaky všemi dvanácti detektory. Z každé detekce dostaneme vektor *keypointů* a vektor *deskriptorů* nebo-li popisů, oba vektory jsou uloženy. Následně jsou porovnány vždy dva poslední deskriptory pro každý detektor

pomocí Flann matcheru, až na výjimku detektorů ORB BRISK a AKAZE, které jsou porovnávány Brutal-force matcherem. Tyto shody jsou uloženy do vektoru a následně při každém cyklu sečteny.

■ Dynamický obraz

Pro detekci a porovnávání obrazových příznaků při jízdě robotu byla vytvořena třída **featureCalculation.cpp**. V této třídě je realizace detekce a porovnávání příznaků totožné jako v souboru **Detection.cpp**. Zásadní odlišností je přehrávání dat z rosbagu, které je hojně použito i v následujících metodách. Aby bylo možné číst data z kanálů (topic) a zároveň provádět detekci a výpočty v obraze, bylo vytvořeno další vlákno(thread) s názvem **findFeature**.

Zatímco se v jednom vláknu načítají a ukládají obrazy a pozice robotu ze systému Vicon, v druhém vlákne jsou prováděny složité výpočty detekce, které by znemožňovaly plynulé čtení dat. Navíc jsou v této třídě porovnávány všechny deskriptory se všemi, ovšem pro daný detektor a to v závislosti na vzdálenosti. Po vypočítání všech vstupních dat, jsou výsledky vykresleny pomocí programu *gnuplot* [43] a uloženy do textového dokumentu.

■ 4.2.4 Tvorba mapy

Metoda pro tvorbu mapy je realizována v souboru **MapCreation.cpp** a je postavená na podobném dvouvláknovém principu jako metoda předchozí. Opět je zde čtený obraz a pozice poskytnutá systémem Vicon z rosbagu, ale prováděné výpočty jsou mírně odlišné. Detekce obrazových příznaků je zde prováděna pouze jedním ze tří detektorů, které vyšli nejlépe v předchozím testu, tedy SURF, KAZE a AKAZE.

Jakmile robot ujede nastavenou vzdálenost skenovacího kroku, porovná popisy obrazových příznaků z obou obrazů pomocí Brute-force matcheru. Následně je pro všechny shody vypočítána pozice daného obrazového příznaku podle vzorců v teoretické kapitole. Výška obrazového příznaku je vypočítána podle osy v níž robot ujel delší vzdálenost.

Popisy obrazového příznaku a jejich už známé pozice jsou nejdříve uloženy do vektoru a následně zapsány do textového dokumentu. Na každém řádku

se nachází jeden obrazový příznak s pozicemi v následujícím tvaru,

$$\left[N_0 \ N_1 \ \dots \ N_j \right] X_0 \ Y_0 \ Z_0 \ \dots \ X_k \ Y_k \ Z_k \quad (4.3)$$

, kde N je prvek popisu, který může být celé nebo desetinné číslo v závislosti na zvoleném popisovači, k je počet prvků popisů, X, Y, Z jsou globální pozice v mapě pro daný obrazový příznak a k je počet pozic tohoto příznaku.

Mapa, v podobě všech vypočtených bodů, je vykreslena nástrojem *gnuplot*.

4.2.5 Lokalizace

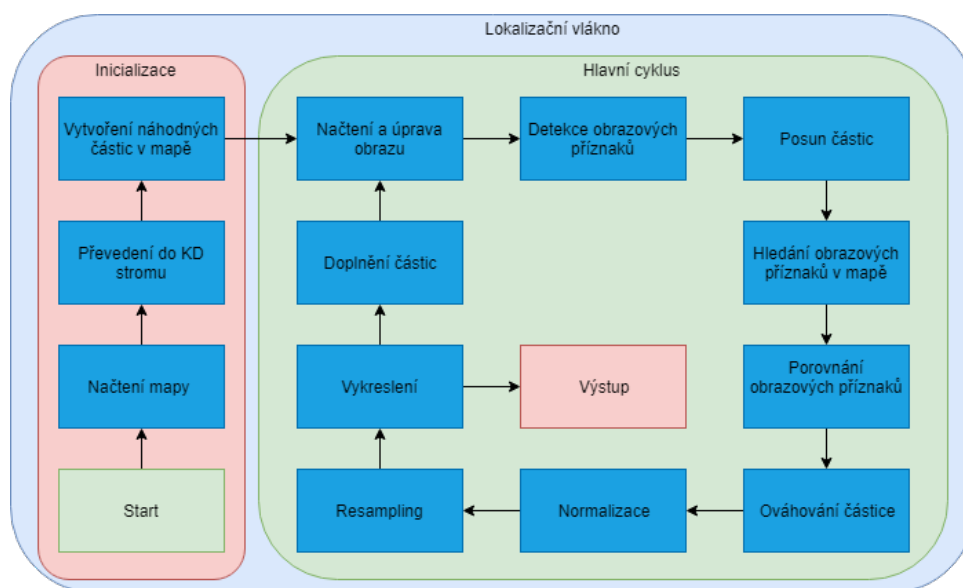
Lokalizační metoda se nachází v souboru **Localization.cpp** a má opět dvouvláknovou architekturu pro načítání dat z rosbagu a pro výpočet. Lokalizace je realizována pomocí částicového filtru, kde vstupní data jsou obrazy z kamery a odometrie robotu. V prvním vlákne tedy probíhá načítání a následné ukládání pozice robotu a obrazu z kamery. V druhém vlákne probíhá samotná lokalizace.

V lokalizační části nejdříve načteme mapu z textového souboru a uložíme ji do vhodné struktury. Pro zrychlení vyhledávání v mapě uložíme pozice do struktury KD-tree z balíčku [44]. A samotné popisy obrazových příznaků do vektoru, kde pořadové ID uložené pozice v KD stromu odpovídá pořadí příslušného popisu ve vektoru.

Dále jsou v mapě vytvořeny náhodné částice a je spuštěn hlavní cyklus. Hlavní cyklus je dán načtením nových sensorových dat ze sdíleného vektoru a následně je v něm prováděn výpočet obrazových příznaků, posun částic, ohodnocení částic, normalizace, resampling a přidávání částic. Aktuální částice a pozice robotu jsou ve formě markerů posílány na kanál (topic) **marker_array**.

Celý proces je zobrazen v grafu 4.10. Nejdříve je obraz zbaven deformace, potom jsou v něm zvoleným detektorem nalezeny obrazové příznaky a částice jsou posunuty pohybovým modelem podle odometrie. Dále jsou jednotlivé částice ohodnoceny sensorovým modelem, jehož vstupem jsou detekované obrazové příznaky z obrazu kamery.

V sensorovém modelu jsou pomocí mapy, tedy KD stromu, zjištěny pozice a následně popisy obrazových příznaků, které může konkrétní částice



Obrázek 4.10: Schéma struktury kódu v lokalizačním vlákne

ze své pozice vidět. Tyto popisy obrazových příznaků jsou pomocí Brute-force matcheru porovnány s popisy vstupujícími do metody. Matcher vrací vektor shod, kde každá shoda má parametr důvěryhodnosti (distance). Podle této důvěryhodnosti a počtu shod obrazových příznaků pro danou částici je vypočítána její váha podle vzorce 3.10 v teoretické části.

Následně jsou částice normalizovány, aby jejich váha byla v intervalu $<0,1>$, doplňovány, aby jejich počet byl konstantní, a resamplovány, aby byly duplikovány částice s vyšší hodnotou. Výsledkem této metody by měly být shluky částic na aktuální pozici robotu. To je možné sledovat vykreslením markerů z našeho kanálu programem *rviz*, který je součástí ROSu.

■ 4.2.6 Použitý software

Pro tuto práci bylo použito rozmanité množství balíčků, editorů a softwaru. Jedny z nich jsou ROS, OpenCV, gnuplot, rviz, Google Docs, Latex.

Kapitola 5

Experimenty

5.1 Příprava

Experiment proběhl s robotem Turtlebot 2 a vybavením popsáním v sekci výše. Kamera umístěna ve středu robotu na jeho horní plošině snímá strop ve výšce 3.8 m. Na vrchu robotu je také umístěn marker pro systém Vicon zajišťující přesnou lokalizaci. Drobné nepřesnosti v umístění markeru i kamery přesně ve středu robotu zanedbáme, jedná se o jednotlivé centimetry.

Během průběhu práce bylo vytvořeno několik záznamů průjezdu robotu pro různé účely s různou délkou, startovním bodem a trajektorií. Robot byl ručně řízen přes ssh tunel z jiného počítače s využitím balíčku **kobuki_keyop**. Hlavní zaznamenané kanály(topic) obsahovaly obraz, odometrii a pozice ze systému Vicon.

Tvorbu rosbagů můžeme rozdělit do dvou částí, na záznamy pro prvotní vybírání vhodného detektoru obrazových příznaků a na záznamy pro tvorbu mapy. Všechny rosbagy mají společnou vlastnost, začátek je vždy v pravém horním rohu mapy a jede ve směru osy x na druhý konec pracovního prostoru.

Hlavním důvodem tohoto startovního bodu bylo umístění pracoviště v laboratoři a zároveň řada výfuků ventilace, vhodných pro detekci obrazových příznaků, řazených se ve směru osy x .

5.1.1 Rosbag pro výběr detektoru obrazových příznaků

Hlavní rosbag použitý pro výběr detektoru obrazových příznaků začínal na startovní pozici [2.93752,4.22457] téměř pod výfukem ventilace. Robot ujel vzdálenost šesti metrů a skončil jízdu pod podobným výfukem ventilace a to se také projevilo při testování. Celý záznam trvá 57 sekund a má velikost téměř půl GB. Přibližnou trajektorii je možné vidět na obrázku 5.1.



Obrázek 5.1: Trajektorie robotu při nahrávání prvotních záznamů

Záznamů podobné trajektorie vzniklo několik, ale tento ideálně prověřil chování detektorů na dva velice podobné objekty. Také byl vytvořen a zpracován záznam rotace robotu na místě, který by ověřil robustnost jednotlivých detektorů při natočení obrazu. Ale bohužel na poslední chvíli bylo zjištěno, že záznam byl pořízen s rozostřenou kamerou a výsledky nejsou legitimní, a proto nejsou uvedeny v této práci.

5.1.2 Rosbag pro tvorbu mapy

Dále bylo nutné vytvořit rosbag, z jehož záznamu by bylo možné vytvořit kompletní mapu celého pracovního prostoru. Nejdříve ale vznikl menší záznam pokrývající pouze polovinu laboratoře, na kterém byla otestována funkčnost metody pro tvorbu mapy.

Ukázalo se, že robot pohybující na kraji pracovního prostoru systému Vicon, může být občas ztracen z dohledu kamer a na chvíli nemá informaci o své pozici. Tím ovšem v záznamu vzniknou snímky, ke kterým není možné přiřadit pozici. Tento případ neuměla metoda pro tvorbu mapy řešit a kumulovala se chyba.



Obrázek 5.2: Trajektorie robotu při skenování celého stropu pracovního prostoru

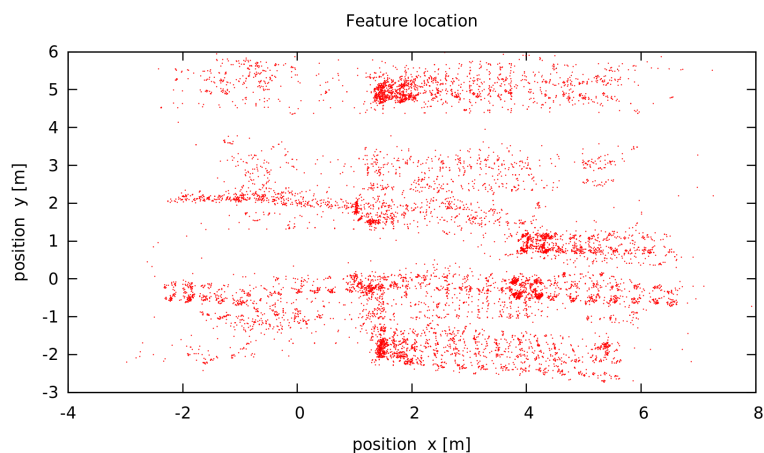
Následně byl vytvořen záznam průjezdu celým pracovním prostorem, ale s odstupem alespoň jednoho metru od krajů pracovního prostoru, aby nedošlo ke ztrátě markeru v systému Vicon. Po odstranění tohoto nedostatku v metodě pro tvorbu mapy byl vytvořen rosbag s průjezdem celého pracovního prostoru i s pozicemi těsně na hraně viditelnosti pro systém Vicon.

5.2 Tvorba mapy

Bylo vytvořeno několik map z různých záznamů průjezdu robotu laboratoří a různých detektorů. Presentovaným výsledkem jsou tři mapy vytvořené z finálního rosbagu pokrývajícího celou oblast pracovního prostoru systému Vicon. Mapy byly vytvořeny s využitím tří detektorů, SURF, KAZE a AKAZE, které se v prvotních testech ukázaly být nejvhodnější.

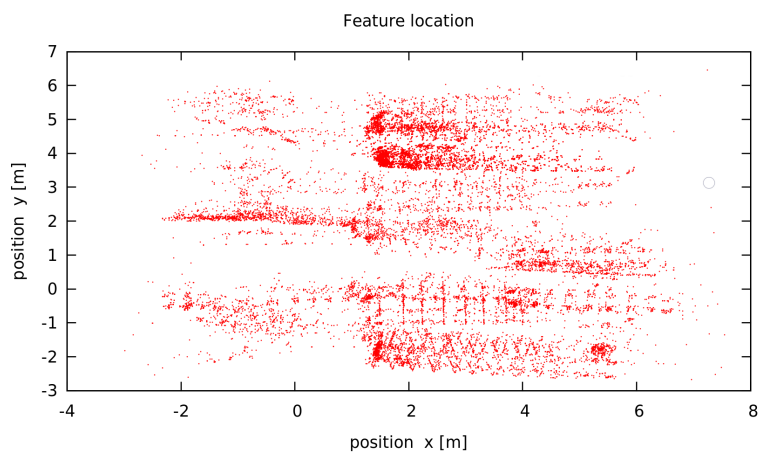
Nejdříve byla ovšem metoda pro tvorbu mapy testována na rosbagu primárně sloužícího pro porovnání detektorů obrazových příznaků, kdy robot

urazil trasu 6 metrů ve směru osy x pod ventilačními výfuky. Takto byla poprvé otestována funkčnost metody a správnost výpočtů. Protože se jednalo o malý prostor, mohly být jednotlivé obrazy s obrazovými příznaky jednoduše sledovány a výsledky jejich pozic kontrolovány. V tomto testu byl také zkoušen různý skenovací krok a nakonec byl zvolen 0.1 m.



Obrázek 5.3: Mapa SURF

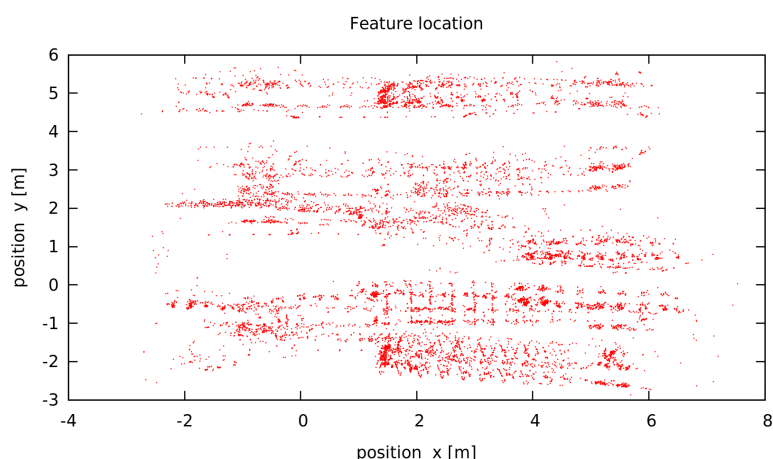
Jednotlivé výsledné mapy jsou na obrázku 5.3, 5.4 a 5.5, každá mapa má velikost 12 x 9 metrů a obsahuje různé množství obrazových příznaků. Mapa SURF obsahuje 9 456 příznaků, mapa KAZE obsahuje 15 554 příznaků a mapa AKAZE obsahuje 10 545 příznaků.



Obrázek 5.4: Mapa KAZE

Obecně můžeme říct, že hlavní detekované prvky jsou traverzy, sloupy a výstupy ventilace. Všechny tyto objekty jsou velice podobné, téměř identické. Pokud je vůbec dokáže lokalizační metoda účinně odlišit, můžeme předpokládat, že shluky se budou tvořit pod všemi podobnými prvky.

Nejvíce nalezených obrazových příznaků obsahuje mapa vytvořená pomocí



Obrázek 5.5: Mapa AKAZE

detektoru KAZE, ale jsou-li tyto detekované body dostatečně kvalitní pro spolehlivou lokalizaci, je otázkou. Možná dosáhne lepších výsledků SURF mapa s nejmenším množstvím obrazových příznaků, ale zato zřejmě více kvalitních a originálních. Jestli je nejlepší mapou AKAZE, která je někde uprostřed, zjistí až experiment v následující kapitole.

5.3 Lokalizace

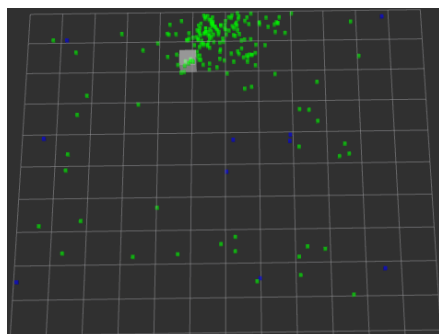
Testování lokalizační metody probíhalo formou přehrávaného záznamu z rosbagu, jako simulace reálné jízdy robotu. Pro jednoznačnost byl použit záznam pro tvorbu mapy, který projíždí celým pracovním prostorem. Zároveň jsou tedy vstupy do lokalizační metody totožné jako při tvorbě mapy a tím jsou zajištěny ideální podmínky pro lokalizaci.

Dále je použit prvotní záznam jízdy pro porovnání detektorů obrazových příznaků, obsahující jízdu pod ventilačními výfuky pod kterými by se měl robot lokalizovat. Experiment je prováděn postupně pro každou mapu zvlášť s detektorem odpovídajícím mapě a stejnými podmínkami.

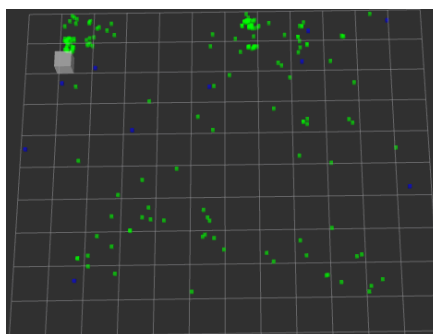
5.3.1 SURF

V mapě SURF se smysluplné shluky částic začnou tvořit po prvních 4 metrech jízdy, viz obrázek 5.6a, ale jsou ještě poměrně dost rozprostřeny na určení

pozice robotu. Teprve v rohu mapy, kde robot vidí jak ventilační výstup, tak sloup budovy, se tvoří soustředěnější shluky.



(a) : První nepřesné shluky

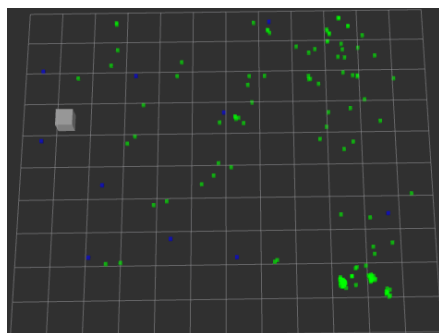


(b) : Shluky tvořící se na podobných místech

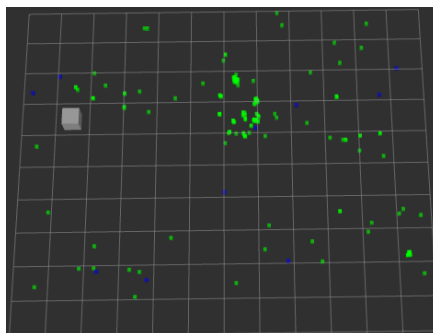
Obrázek 5.6: Lokalizace v mapě SURF

Na obrázku 5.6b je vidět, že hlavní shluk částic je od skutečné pozice robotu více než metr vzdálený. V právě horní části mapy jsou vidět menší shluky na místech, kde částice může vidět podobný výstup ventilace nebo kus opěrného sloupu.

V dalších částech cesty bohužel lokalizační metoda pozici robota úplně ztrácí a z metrové nepřesnosti se stává nesmyslná pozice. Na obrázku 5.7a jsou shluky úplně v jiné části mapy a o pár kroků později, viz obrázku 5.7b, je aktuálně viděný výstup ventilace poměrně přesvědčivě zaměněn za jiný o 4 metry dál. Bohužel ani v dalších částech trajektorie není robot schopen lokalizace na delší dobu.



(a) : Shluky částic v jiné části mapy než se nachází robot

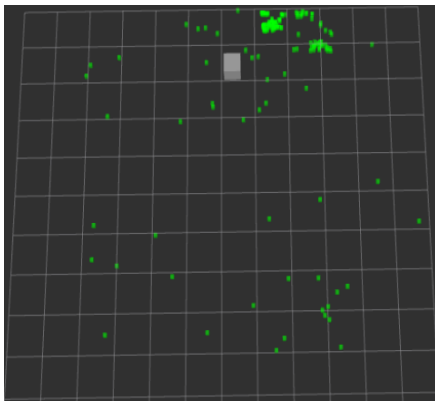


(b) : Shluky částic pod jiným orientačním prvkem

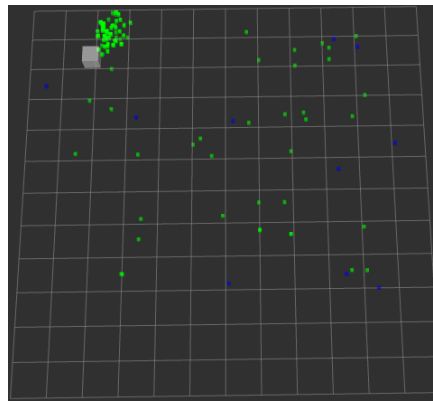
Obrázek 5.7: Lokalizace v mapě SURF

5.3.2 KAZE

Pro mapu KAZE se shluky začnou tvořit také po prvních 4 metrech jízdy a jsou viditelně méně rozprostřeny než při lokalizaci v mapě SURF, to je možné vidět na obrázku 5.8a. Shluk se sjednotí ještě před dojetím robotu do rohu mapy, viz obrázek 5.8b, tedy dříve než při SURF lokalizaci. Nejspíše stejně už vidí jak ventilační výstup, tak sloup budovy.



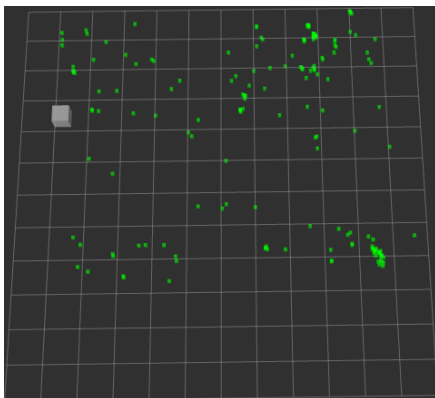
(a) : První nepřesné shluky



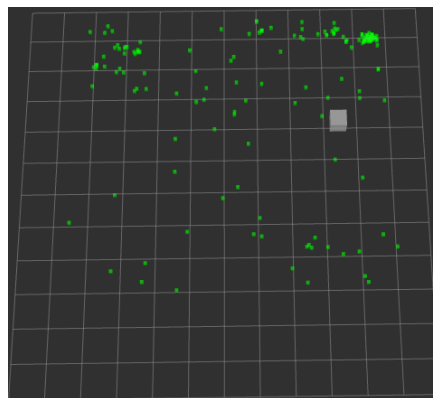
(b) : První přesnější lokalizace robotu

Obrázek 5.8: Lokalizace v mapě KAZE

Při dalším putování mapou se pozice robotu rychle vytratí, ale tvoří se shluky částic na více místech mapy pod orientačními prvky, obrázek 5.9a, jako jsou právě výstupy ventilace a traverzy. Shluky se tvoří většinou jinde, jako například na obrázku 5.9b, a pokud dojde k odhadu správné pozice, je stejně po několika metrech ztracena.



(a) : Pozice robotu je občas ztracena

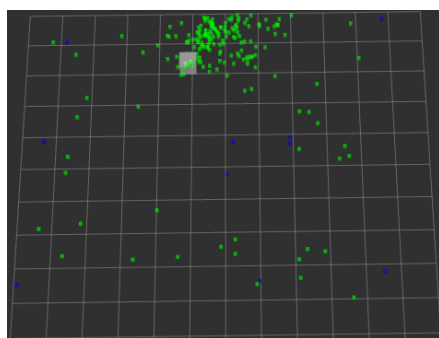


(b) : Tvoření shluků pod orientačními prvky

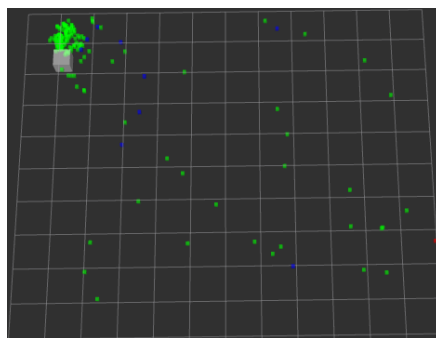
Obrázek 5.9: Lokalizace v mapě KAZE

5.3.3 AKAZE

Pro lokalizaci v mapě AKAZE vypadal průběh podobně jako u předchozích dvou map, velký oblak shluků za robotem kolem čtvrtého metru jízdy, viz obrázek 5.10a, a v levém horním rohu mapy přesnější určení pozice robotu. Bohužel i v této mapě jsou shluky kolem metru od skutečné pozice robotu a tvoří se spíše pod orientačními prvky, jak je možné vidět na obrázku 5.10b.



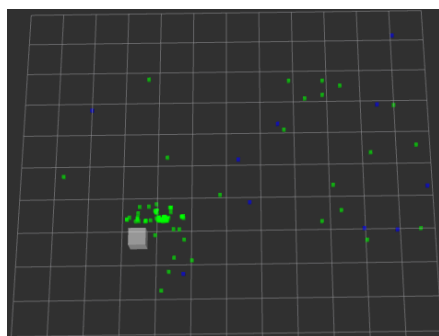
(a) : První nepřesné shluky



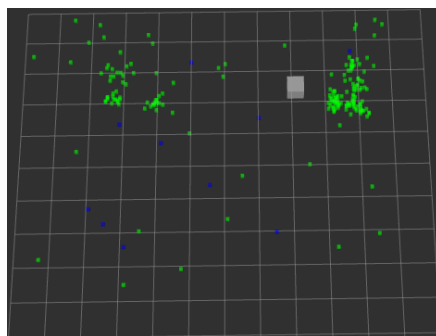
(b) : Nalezení pozice robotu

Obrázek 5.10: Lokalizace v mapě AKAZE

Pozitivním zjištěním je, že shluky se drží robotu po zbytek jízdy po celé mapě. Sice většinou ve vzdálenosti jednoho metru od skutečné pozice a občas se trochu rozprostřou, viz obrázek 5.11a, ale ani alternativní shluky částic pod podobnými prvky stropu nejsou zásadní, jak je možné vidět na obrázku 5.11b. A robot svojí pozici nikdy úplně neztratí jako u předchozích dvou map.



(a) : Mírně rozprostřené částice poblíž robotu

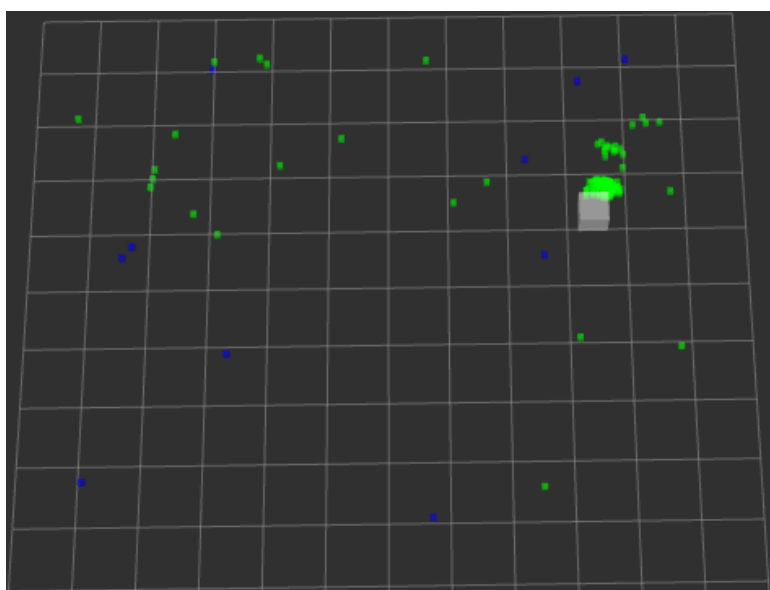


(b) : Alternativní shluky pod význačnými prvky

Obrázek 5.11: Lokalizace v mapě AKAZE

5.3.4 Shrnutí

Lokalizace robotu ve známé mapě pro tuto laboratoř je nevhodnější metoda zakládající se na obrazových příznacích získaných detektorem AKAZE. Jako jediná dokázala aspoň přibližně na vzdálenost kolem jednoho metru odhadovat pozici robotu souvisle při jízdě po celé mapě. V místech bez význačných orientačních bodů docházelo k mírným nepřesnostem, ale při detekci významných objektů byla pozice výrazně přesnější, jak je možné vidět na obrázku 5.12.



Obrázek 5.12: Robot lokalizovaný v mapě AKAZE

Lokalizace pro druhý kratší rosbag probíhala podobně, ve všech případech byl robot lokalizován téměř na konci trajektorie v místě ventilačního průduchu. Tento záznam nebyl dostatečně dlouhý pro větší srovnání různých verzí metody a proto zde není detailně rozebrán.

Samozřejmě podle předpokladu se shluky částic tvořily pod většinou význačných prvků, podle pozic zvláště pod výfuky ventilace. V některých mapách, například právě SURF mapě, byly prvky na stropě nesprávně vyhodnoceny a odhadovaná pozice byla nepřesná, viz obrázek 5.7b.

Další skutečností je nedostatečná přesnost aktuální metody. Přesnost pozice robotu se pohybovala kolem jednoho metru. To je nejspíše způsobeno vytvářením shluků částic přesně v místě významných orientačních prvků, místo na pozici robotu. Řešením problému by bylo přidat do váhovací funkce částic informaci o pozici příznaku v obrazu kamery.

Kapitola 6

Závěr

V této práci byly popsány základní principy a metody vizuální lokalizace robotu. Metody využívající přirozené landmarky i umělé značky, rozpoznávání rohů, světél a dalších objektů. Byly prostudovány metody pro lokalizaci, jako například SLAM, Kalmanův filtr, Částicový filtr, Markovova lokalizace nebo párování obrazových příznaků.

Dále byly nastudovány základní vlastnosti obrazových příznaků a jejich použití. Bylo využito volných knihoven OpenCV s kvalitní dokumentací a teoretickým popisem funkcionality. Nejvhodnější obrazový příznak byl vybrán cestou experimentu v kancelářském prostředí laboratoře, která je dobře odstíněna od okolního prostředí a zachovává téměř neměnné podmínky. Ze série experimentů vyšly vítězně detektory SURF, KAZE, AKAZE.

Následně se přešlo k vytvoření mapy s každým ze tří detektorů, reprezentující kancelářské prostředí laboratoře. Každá z těchto map vykazovala na první pohled v jinak monotonním prostředí význačné orientační prvky jako jsou výfuky ventilace, traverzy a sloupy. Mapy se od sebe lišily počtem, typem i kvalitou zaznamenaných obrazových příznaků.

Předposlední krokem byl návrh a tvorba lokalizační metody pro mobilní robot. Byl zvolen Částicový filtr kvůli vlastnosti uchovávání více hypotéz pro pozici robotu, která je v prostředí téměř identických orientačních prvků velmi důležitá. Implementace lokalizační metody, stejně jako všech ostatních částí této práce, je realizována v jazyku C++ v prostředí ROS.

V poslední kroku je tato metoda podrobena několika experimentům, které prověří její funkčnost a kvalitu. Zároveň bude rozhodnuto o nejvhodnějším obrazovém příznaku v prostředí laboratoře v níž vývoj probíhá. Podle výsledků experimentu je možné říci, že nejvhodnější obrazové příznaky jsou získány algoritmem AKAZE. Samotná metoda pro lokalizaci robotu je sice funkční, ale její přesnost je neuspokojivá.

Pro zpřesnění metody by mohly být do váhovací funkce částic přidány další proměnné zahrnující hodnotu polohy obrazového příznaku v obraze. Také by bylo vhodné provést komplexnější testování jak samotné metody, tak i obrazových příznaků. Případně pro každý detektor využít širokou škálu nastavení pro maximalizaci jeho efektivnosti.

Dále by mohla být realizována real-time lokalizace mobilního robotu, místo přehrávání záznamu jízdy. To by ovšem vyžadovalo zásadní výpočetní zrychlení aktuální lokalizační metody. Nejspíše i přechod na výpočetně výkonnější řídicí počítač.



Literatura

- [1] Prachya Kamol, Stefanos Nikolaidis, Ryuichi Ueda, Tamio Arai, *RFID Based Object Localization System using Ceiling Cameras with Particle Filter*, Future Generation Communication and Networking (FGCN 2007)
- [2] Philipp Michel, Joel Chestnutt, James Kuffner and Takeo Kanade, *Vision-Guided Humanoid Footstep Planning for Dynamic Environments*, 5th IEEE-RAS International Conference on Humanoid Robots, 2005
- [3] Liwei Han, Min Tan, and You Fu Li, *Ceiling-Based Visual Positioning for an Indoor Mobile Robot With Monocular Vision*, IEEE Transactions on Industrial Electronics 2009
- [4] Borislav Dzodzo, Long Han, Xu Chen, Huihuan Qian, Yangsheng Xu, *Realtime 2D Code Based Localization for Indoor Robot Navigation*, IEEE International Conference on Robotics and Biomimetics (ROBIO) 2013
- [5] Wen-Tsai Huang, Chun-Lung Tsai, Huei-Yung Lin, *Mobile Robot Localization using Ceiling Landmarks and Images Captured from an RGB-D Camera*, IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM) 2012
- [6] Pengjin Chen, Zhaopeng Gu*, Guodong Zhang, Hong Liu *Ceiling Vision Localization with Feature Pairs for Home Service Robots*, 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)
- [7] WooYeon Jeong, Kyoung Mu Lee, *CV-SLAM: a new ceiling vision-based SLAM technique*, 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems

- [8] Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem*, Aaai/iaai, 2002, 593598.
- [9] Zbyněk Winkler *Simultaneous Localization and Mapping*, 2003-12-08, Dostupné z: <https://robotika.cz/guide/slam/cs>, [cit. 14.05.2019]
- [10] Georg Klein, David Murray *Parallel Tracking and Mapping for Small AR Workspaces*, 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality
- [11] Lawrence R. Rabiner *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, 1989 Proceedings of the IEEE
- [12] ZENG-SHUN ZHAO *DECISION FUSION OF GLOBAL AND LOCAL IMAGE FEATURES FOR MARKOV LOCALIZATION*, 2009 International Conference on Wavelet Analysis and Pattern Recognition
- [13] Jana Košecká, Fayin Li *Vision Based Topological Markov Localization*, IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004
- [14] Pi Yuzhen, Yuan Quande, Zhang Benfa, *The Application of Adaptive Extended Kalman Filter in Mobile Robot Localization*, 2016 Chinese Control and Decision Conference (CCDC)
- [15] Rongbao Chen, He Zhao, Benxian Xiao, *Self-localization of Mobile Robot Based on Monocular and Extended Kalman Filter*, 2009 9th International Conference on Electronic Measurement & Instruments
- [16] Oliver Wulf, Daniel Lecking, Bernardo Wagner, *Robust Self-Localization in Industrial Environments based on 3D Ceiling Structures*, 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems
- [17] Joaquín López, Diego Pérez, Eduardo Zalama, Jaime Gómez-García-Bermejo, *Low Cost indoor mobile robot localization system*, 2011 11th International Conference on Intelligent Systems Design and Applications
- [18] Tae Gyun Kim, Hyun-Taek Choi, and Nak Yong Ko, *Localization of a Robot Using Particle Filter with Range and Bearing Information*, 2013 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)
- [19] Zhao Jun, Liu Guo-ping, *A Novel Localization Method for Indoor Mobile Robot Based on Odometry and Ceiling Visual Features*, 2015 34th Chinese Control Conference (CCC)
- [20] Liwei Han, Min Tan, You Fu Li, *Ceiling-Based Visual Positioning for an Indoor Mobile Robot With Monocular Vision*, IEEE Transactions on Industrial Electronics 2009

- [21] Francisco Dias, Hanna Schafer, Leonardo Natal, Carlos Carneira, *Mobile robot localisation for indoor environments based on ceiling pattern recognition*, 2015 IEEE International Conference on Autonomous Robot Systems and Competitions
- [22] Koray Celik, Soon-Jo Chung, Matthew Clausman, Arun K. Somani, *Monocular Vision SLAM for Indoor Aerial Vehicles*, 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems
- [23] Petr Barborka *Analýza metod pro detekci příznaků v digitalizovaném obraze*, Západočeská univerzita v Plzni Fakulta aplikovaných věd Katedra kybernetiky 2016
- [24] Frazer K. Noble *Comparison of OpenCV's Feature Detectors and Feature Matchers*, 2016 23rd International Conference on Mechatronics and Machine Vision in Practice (M2VIP)
- [25] David G. Lowe, *Distinctive Image Features from Scale-Invariant Keypoints*, International journal of computer vision, 2004, 60.2: 91-110.
- [26] Avi Kaplan, Tamar Avraham, Michael Lindenbaum, *Interpreting the Ratio Criterion for Matching SIFT Descriptors*, European Conference on Computer Vision 2016
- [27] Vicon motion capture system *About Vicon Motion Systems | VICON. Motion Capture Systems | VICON [online]. Dostupné z: <https://www.vicon.com>*, [cit. 14.05.2019]
- [28] J. HUDEC, *Analýza chůze*, Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013
- [29] N. Ganganath, H. Leung, *MOBILE ROBOT LOCALIZATION USING ODOMETRY AND KINECT SENSOR*, 2012 IEEE International Conference on Emerging Signal Processing Applications
- [30] Dairong Li, Qijun Chen, Zhiying Zeng, *Self-localization Algorithm of Mobile Robot Based on Unscented Particle Filter*, 2018 37th Chinese Control Conference (CCC)
- [31] Fengbing Luo, Bianjing Du, Zhen Fan, *Mobile Robot Localization Based on Particle Filter*, Proceeding of the 11th World Congress on Intelligent Control and Automation 2014
- [32] Ing. Roman Mázl, *Lokalizace pro autonomní systémy*, Disertační práce. ČVUT, Elektrotechnická fakulta, katedra kybernetiky Praha 2007
- [33] Jan Bjelka, *Návrh a realizace metody mapování okolí pro mobilní robot*, Diplomová práce na fakultě strojního inženýrství VUT na Ústavu automatizace a informatiky. 2007

- [34] TurtleBot., *TurtleBot [online]*. Copyright © Open Source Robotics Foundation, Inc. . Dostupné z: <https://www.turtlebot.com/> , [cit. 14.05.2019]
- [35] ROS(Robot operating system) *ROS.org | Powering the world's robots*. *ROS.org | Powering the world's robots [online]*. Dostupné z: <https://www.ros.org/> , [cit. 14.05.2019]
- [36] Basler dart daA1280-54uc *Basler dart daA1280-54uc (S-Mount) - Area Scan Camera. 302 Found [online]*. Copyright © 2019 Basler AG. Dostupné z: <https://www.baslerweb.com/en/products/cameras/area-scan-cameras/dart/daa1280-54uc-s-mount/> , [cit. 14.05.2019]
- [37] Pylon software *pylon Camera Software Suite | Basler. 302 Found [online]*. Copyright © 2019 Basler AG. Dostupné z: www.baslerweb.com/en/products/software/basler-pylon-camera-software-suite/ , [cit. 14.05.2019]
- [38] Intel NUC5i5RYK *Intel® NUC Kit NUC5i5RYK Product Specifications. Document Moved [online]*. Copyright © Intel Corporation. Dostupné z: <https://ark.intel.com/content/www/us/en/ark/products/83254/intel-nuc-kit-nuc5i5ryk.html> , [cit. 14.05.2019]
- [39] OpenCV knihovny *OpenCV. OpenCV [online]*. Copyright © Copyright 2019, OpenCV team. Dostupné z: <https://opencv.org/> , [cit. 14.05.2019]
- [40] April camera calibration *Camera suite - April. APRIL Laboratory APRIL Laboratory : Autonomy * Perception * Robotics * Interfaces * Learning [online]*. Dostupné z: https://april.eecs.umich.edu/wiki/Camera_suite , [cit. 14.05.2019]
- [41] Matěj Beránek *Experimentální prostředí pro skupinovou robotiku*, Bakalářská práce. ČVUT, Elektrotechnická fakulta, katedra kybernetiky Praha 2017
- [42] Martin Novotný *Robustní postupy řízení formací robotů*, Bakalářská práce. ČVUT, Elektrotechnická fakulta, katedra kybernetiky Praha 2017
- [43] Gnuplto graphing tools *gnuplot homepage. gnuplot homepage [online]*. Dostupné z: <http://www.gnuplot.info/> , [cit. 14.05.2019]
- [44] J. Frederico Carvalho *GitHub - crvs/KDTree: Simple C++ KD-Tree implementation. GitHub [online]*. Copyright © 2019. Dostupné z: <https://github.com/crvs/KDTree> , [cit. 14.05.2019]
- [45] Marcello Mulas, Nicolai Waniek, Jorg Conradt, *Hebbian Plasticity Realigns Grid Cell Activity with External Sensory Cues in Continuous Attractor Models*, Frontiers in computational neuroscience 2016

- [46] Dirk Hahnel, Wolfram Burgard, Dieter Fox, Sebastian Thrun, *An Efficient FastSLAM Algorithm for Generating Maps of Large-Scale Cyclic Environments from Raw Laser Range Measurements*, Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)
- [47] James Hays, Samarth Brahmabhatt, John Lamber, *Local Feature Matching CS 6476: Computer Vision 2018*, Dostupné z: <https://www.cc.gatech.edu/~hays/compvision/proj2/>, [cit. 14.05.2019]
- [48] Miroslav Kulich, Gael Ecorchard, *Intelligent and Mobile Robotics Group Czech Technical University in Prague, B3M33MKR*, Winter semester 2017/2018
- [49] Marie-Agnès Martineau, *Particle filters for Robot Localization*, Dostupné z: <https://slideplayer.com/slide/14966926/>, [cit. 14.05.2019]
- [50] TurtleBot2 Docking Station, *TurtleBot2 Docking Station - TURTLEBOT . Robots jouets, robots éducatifs et robots ménagers - Robot Advance [online]*. Dostupné z: <https://www.robot-advance.com/EN/art-turtlebot2-docking-station-1190.htm>, [cit. 14.05.2019]
- [51] Roscomponents - TurtleBot2 with Arm, *Roscomponents [online]*. Copyright ©n Dostupné z: https://www.roscomponents.com/361-thickbox_default/turtlebot-2.jpg, [cit. 14.05.2019]
- [52] Trossen Robotics, *Trossen Robotics - Robotic Arms, Crawlers Turrets and more! [online]*. Copyright © . Dostupné z: https://www.trossenrobotics.com/Shared/Images/Product/Interbotix-Turtlebot-2i-Mobile-ROS-Platform/Img3077_isolated.jpg, [cit. 14.05.2019]
- [53] Intel Nuc NUC5I5RYK - Heureka.cz, *Stolní počítače - Heureka.cz [online]*. Copyright © 2007. Dostupné z: <https://stolni-pocitace.heureka.cz/intel-nuc-nuc5i5ryk/>, [cit. 14.05.2019]
- [54] Antonio Russo, *Face detection & recognition with Javascript*, Dostupné z: <https://blog.beautifulinteractions.com/face-detection-and-recognition-with-javascript-9e0b51b1c012>, [cit. 14.05.2019]
- [55] Moses Olafenwa, *Object Detection with 10 lines of code*, Dostupné z: <https://towardsdatascience.com/object-detection-with-10-lines-of-code-d6cb4d86f606>, [cit. 14.05.2019]

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Koktan** Jméno: **Jiří** Osobní číslo: **434872**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra řídicí techniky**
Studijní program: **Kybernetika a robotika**
Studijní obor: **Kybernetika a robotika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Lokalizace mobilního robotu sledováním stropu

Název diplomové práce anglicky:

Localization of a mobile robot by ceiling observation

Pokyny pro vypracování:

1. Seznamte se s metodami vizuální lokalizace mobilního robotu
2. Seznamte se s obrazovými příznaky a jejich popisem
3. Vyberte nejvhodnější příznaky pro popis stropu v průmyslovém a kancelářském prostředí
4. Navrhněte a naimplementujte metodu lokalizace mobilního robotu ve známé mapě
5. Vyhodnoťte vlastnosti implementované metody

Seznam doporučené literatury:

1. Mobile Robot Localization using Ceiling Landmarks and Images Captured from an RGB-D Camera, Wen-Tsai Huang, Chun-Lung Tsai and Huei-Yung Lin, International Conference on Advanced Intelligent Mechatronics, 2012
2. Ceiling-Based Visual Positioning for an Indoor Mobile Robot With Monocular Vision, De Xu, Liwei Han, Min Tan, and You Fu Li, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 56, NO. 5, MAY 2009
3. CV-SLAM: a new ceiling vision-based SLAM technique, WooYeon Jeong, IROS 2005

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Karel Košnar, Ph.D., inteligentní a mobilní robotika CIIRC

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

Datum zadání diplomové práce: **06.02.2019**

Termín odevzdání diplomové práce: **24.05.2019**

Platnost zadání diplomové práce:

do konce letního semestru 2019/2020

Ing. Karel Košnar, Ph.D.
podpis vedoucí(ho) práce

prof. Ing. Michael Šebek, DrSc.
podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

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

Datum převzetí zadání

Podpis studenta