

České vysoké učení technické v Praze  
Fakulta elektrotechnická  
Katedra kybernetiky



Bakalářská práce

## **Detekce pohyblivých objektů v mobilní robotice**

*Aleš Ondráček*

Vedoucí práce: Ing. Jan Chudoba

Studijní program: Elektrotechnika a informatika, strukturovaný, Bakalářský

Obor: Kybernetika a měření

26. května 2010

## Poděkování

Za ochotu, množství cenných rad a věnovaný čas bych rád poděkoval vedoucímu práce ing. Janu Chudobovi.

Dále bych poděkoval ing. Vojtěchu Vonáskovi za pomoc a nástroje poskytnuté pro bakalářskou práci.

## Prohlášení

Prohlašuji, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

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

V Praze dne 20. 1. 2010

.....

### *Abstrakt*

Bakalářská práce se zabývá problémem nalezení a implementace vhodného algoritmu simultánní lokalizace, mapování a detekce pohyblivých objektů. Motivací je nalezení polohy pohyblivých objektů a jejich filtrace ze vstupních dat jiného mapovacího algoritmu. Zdrojem dat je laserový dálkoměr, pro určení polohy robotu je použit odometrický systém s upřesněním transformací získanou porovnáním měření dálkoměru algoritmem *ICP*.

V úvodu je úloha popsána zvláště pro část simultánní lokalizace a mapování a část detekce pohyblivých objektů, také jsou prezentovány výhody spojení obou úloh v jeden celek. Dále je přiblíženo zadání a cíle bakalářské práce.

Druhá kapitola se zabývá současným stavem problematiky ve světě a směry, kterými se výzkum ubírá. Podrobněji jsou popsány různé části úlohy při použití laserových dálkoměrů.

Třetí kapitola se zaměřuje na vlastní návrh algoritmu, důvody volby implementovaných postupů a podrobnosti řešení vzniklých problémů.

### *Abstract*

The bachelor thesis deals with the problem of finding and implementation of suitable algorithm for simultaneous localization, mapping and detection of moving objects. The motivation for the work is to point out the location of moving objects and filter them out from input data for another mapping algorithm. The source of data is laser range finder and odometrical system, which is used to determine the robot position. To get more precise position data, transformation resulting from comparison of range finder measurements is used.

The task is described separately for simultaneous localization and mapping and for detection and tracking of moving objects in the introduction. The advantages of connecting these parts together and goals of the bachelor thesis are presented here as well.

Second chapter deals with the state of the art of the scientific field and trends in the research. Some parts of the task using laser range finders are described more precisely.

The actual design of the algorithm, reasons why some procedures were chosen and solutions to problems that had occurred are described in third chapter.

# Obsah

<b>1 Úvod</b>	<b>1</b>
1.1 Simultánní lokalizace a mapování s detekcí a sledováním pohyblivých objektů	1
1.1.1 Simultánní lokalizace a mapování	2
1.1.2 Detekce a sledování pohyblivých objektů	2
1.2 Cíle bakalářské práce	3
<b>2 Současný stav problematiky</b>	<b>4</b>
2.1 Proximitní senzory	4
2.1.1 Sonar	4
2.1.2 IR proximitní senzor	4
2.1.3 Laserový dálkoměr	5
2.2 Detekce pohyblivých objektů	5
2.2.1 Segmentace nového skenu	5
2.2.2 Rozlišení pohyblivých segmentů	6
2.2.2.1 Konzistenční detekce	6
2.2.2.2 Detekce na základě porovnávání s mapou pohyblivých objektů	6
2.2.2.3 Detekce s informací o volném prostoru	7
2.3 Reprezentace prostředí	7
2.3.1 Mřížka obsazenosti	8
2.3.2 Bodová mapa	9
2.4 Lokalizace polohy	9
2.4.1 Pravděpodobnostní přístupy	9
2.4.1.1 Částicové filtry	9
2.4.1.2 Kalmanovy filtry	10
2.4.2 Přístupy bez pravděpodobnosti	11
2.4.2.1 Odometrie	11
2.4.2.2 Scan matching	12
2.4.2.3 ICP algoritmus	12
<b>3 Implementace SLAM &amp; DATMO</b>	<b>15</b>
3.1 Přehled jednotlivých kroků	15
3.1.1 Získání a segmentace nových dat	15
3.1.2 Asociace segmentů se stávajícími pohyblivými objekty	16
3.1.3 Upřesnění polohy nového skenu algoritmem ICP a detekce nových pohyblivých objektů	17

3.1.4	Aktualizace statické mapy a pozic sledovaných pohyblivých objektů . . . . .	17
3.2	Použité datové struktury . . . . .	18
3.2.1	Bodová mapa . . . . .	18
3.2.2	KD strom . . . . .	19
<b>4</b>	<b>Experimenty</b>	<b>21</b>
4.1	Laserový dálkoměr . . . . .	21
4.2	Odometrie . . . . .	22
4.3	Popis jednotlivých experimentů . . . . .	22
<b>5</b>	<b>Závěr</b>	<b>30</b>
	<b>Literatura</b>	<b>32</b>
<b>A</b>	<b>Seznam použitých zkratk</b>	<b>34</b>
<b>B</b>	<b>Obsah příloženého CD</b>	<b>35</b>

# Seznam obrázků

2.1	Detekce pohyblivých objektů . . . . .	7
2.2	Bodové páry vytvořené (a) ICP (b) IMRP . . . . .	14
3.1	Postup při zpracování nového měření . . . . .	16
3.2	Ukázka KD stromu dimenze dva . . . . .	20
4.1	Počáteční stav . . . . .	24
4.2	Pomalé uvedení statického objektu do pohybu . . . . .	25
4.3	Detekovaná krabice tlačaná člověkem . . . . .	25
4.4	Chybné přidání bodů do mapy při pohybu člověka blízko zdi . . . . .	25
4.5	Důsledek zastavení člověka těsně u zdi . . . . .	26
4.6	Člověk směřující do výklenku (1) . . . . .	26
4.7	Člověk směřující do výklenku (2) . . . . .	26
4.8	Člověk směřující do výklenku (3) . . . . .	27
4.9	Náhlá změna směru pohybu (1) . . . . .	27
4.10	Náhlá změna směru pohybu (2) . . . . .	27
4.11	Náhlá změna směru pohybu (3) . . . . .	28
4.12	Lokalizace pouze z odometrie . . . . .	28
4.13	Lokalizace z odometrie s upřesněním ICP algoritmem . . . . .	28
4.14	chyba ICP při neodstraněných pohyblivých objektech . . . . .	29

# Kapitola 1

## Úvod

Lokalizační problém roboty řeší při úlohách jako je průzkum neznámého prostředí, mapování a navigace v prostředí známém. Vstupní informací jsou data ze senzorů, které měří jiné veličiny jako je vzdálenost k překážkám, pootočení kol nebo například zrychlení. Sensory mají různou přesnost v různých prostředích, ve kterých se robot pohybuje, někdy může selhat vyhodnocení získaných dat. Lokalizační algoritmus z takových informací určuje polohu, kde se robot nachází a může získat i další informace, které se využívají například pro stavbu mapy.

### 1.1 Simultánní lokalizace a mapování s detekcí a sledováním pohyblivých objektů

V anglickém originále je problematika známa jako *Simultaneous Localization And Mapping with Detection And Tracking of Moving Objects* (nebo zkráceně *SLAM & DATMO*). Jednou částí úlohy je simultánní lokalizace a mapování (*SLAM*), kde dynamické objekty představují rušivou složku pro lokalizační algoritmus a zhoršují kvalitu výsledné mapy. Druhou součástí je detekce a sledování pohyblivých objektů (*DATMO*). V tomto případě naopak úlohu komplikují statické objekty, s nimiž mohou sledované pohyblivé objekty splývat nebo se za ně skrývat. Je proto výhodné oba algoritmy spojit v jeden, kde se vzájemně doplňují. *DATMO* zajišťuje filtraci pohyblivých objektů a napomáhá tak přesnější práci lokalizačního a mapovacího algoritmu. *SLAM* poskytuje přesný údaj o vlastní poloze robota, který usnadní detekci a sledování pohyblivých objektů.

*SLAM & DATMO* je velmi důležitá pro samostatnou práci robotu v prostředí, kde se vyskytují lidé nebo jiné pohyblivé objekty. Algoritmus je využíván v automobilové dopravě – především jako pomocný systém pro prevenci kolizí v běžném provozu. Plně automatická řízení stále nejsou pro samostatný provoz dostatečně spolehlivá. Prozatím se uplatňují převážně v průmyslu – pro přepravu zboží v dobře známých prostředích nebo jsou vidět v populárních soutěžích automatických vozidel. V mobilní robotice obecně je uplatnění *SLAM & DATMO* velmi široké, uvedeny byly pouze některé z mnoha aplikací. Úloha je dále rozdělena do dvou podskupin podle výše uvedeného rozdělení a popisována odděleně.



### 1.1.1 Simultánní lokalizace a mapování

Cílem lokalizačního algoritmu je určení pozice robotu ze sensorických dat. Jednou z možných metod je odometrie – jednoduchý a často používaný způsob určování ujeté trajektorie měřením pootočení kol (viz kapitola 2.4.2.1). Dále jsou používány inerciální senzory – například akcelerometry, které určují polohu z měření zrychlení a úhlových zrychlení v různých směrech. Oba způsoby jsou však zatíženy časově neomezenou chybou a jsou proto vhodné pro přesnější určení polohy pouze na krátké vzdálenosti. Pro zmenšení chyby lokalizace se využívá dalších upřesňujících postupů – především takových, které zpracovávají naměřené vzdálenosti k překážkám v okolí robotu.

Nejvhodnějším senzorem je laserový dálkoměr, důvodem je jeho schopnost velmi přesně měřit vzdálenost. Laserový paprsek dálkoměru je rozmítán v jedné rovině, navracena je série měření, nejčastěji pro úhel  $180^\circ$  před skenerem (podrobněji viz 2.1.3).

V lokalizaci jsou dominantní dva přístupy. Prvním z nich je *pravděpodobnostní*, kdy se počítá s neurčitostmi v datech a poloha je zjišťována určováním maximální pravděpodobnosti výskytu robotu v prostředí (podrobněji viz. 2.4.1). Druhým přístupem je *scan-matching*, neboli porovnávání skenů. Je hledána transformace mezi novým skenem a některým ze starších nebo mapou. Nalezená transformace současně určuje změnu polohy robotu. Metoda *scan-matching* je implementována v této práci a je podrobněji popsána v kapitole 2.4.2.

Pro mapování je velmi důležitá zvolená struktura ukládání dat. Pokud jsou data zpracovávána a ukládána v reálném čase, je vhodné vzít v úvahu i výpočetní (časovou) náročnost operací prováděných při stavbě mapy a vyhledávání v ní. Při práci se surovými sensorickými měřeními je dat zpravidla velké množství – důležitá je paměťová úspornost uložení a rychlost algoritmů pro vyhledávání informace uložené v mapě. Oba požadavky směřují k minimalizaci objemu zpracovávaných dat, to ale vede ke snížení kvality výsledné mapy. Požadavky na kvalitu výsledné mapy a výpočetní a paměťovou náročnost zde stojí proti sobě a je potřeba najít vhodný kompromis. V prostorách bohatých na geometricky dobře rozlišitelné tvary může být výhodnější použít aproximaci sensorických dat geometrickými primitivy jako jsou úsečky nebo různé mnohoúhelníky. Výsledek jejich extrakce z prostředí není vždy kvalitní a může být složité mapu vytvářet v reálném čase. Reprezentace geometrickými primitivy řeší problém zpracovávání velkých objemů dat.

Pro zjednodušení úlohy se přijímají různá omezení na okolní prostředí – například pokud je mapováno prostředí uvnitř budov, předpokládají se v mapované části rovné podlahy. Terénní nerovnosti jako jsou vysoké prahy – pokud je robot schopen je překonat – způsobují chybné výsledky odometrie. Pokud se robot více nakloní, rovina měření laserového skeneru se protne s rovinou země a ve výsledném měření opět vzniká falešná překážka. Při použití laserového skeneru dále předpokládáme, že překážky v okolním prostředí nemají zrcadlivý povrch. V opačném případě by nebyly měřitelné.

### 1.1.2 Detekce a sledování pohyblivých objektů

Pokud se robot pohybuje v prostředí, kde se vyskytují lidé, nebo obecněji pohyblivé objekty, je velmi důležité mít informaci o jejich poloze, odhadovaném směru a rychlosti pohybu. Robot potřebuje při plánování upravovat trajektorii tak, aby zabránil srážce (omezil její pravděpodobnost) nebo může upravit svou rychlost na úroveň kolemjdoucích.

Pro sledování pohyblivých objektů se dnes stále více používají kamery. Kamera je poměrně levný senzor, který zároveň poskytuje velké množství informací o okolním prostředí. U člověka získává zrakové vnímání až 80% informací o okolí, ve zpracování kamerového obrazu je proto do budoucna stále mnoho možností. Přes intenzivní výzkum v tomto oboru v posledních letech zatím lze jen v některých případech spolehnout pouze na data z kamery. Zpracování obrazu je výpočetně náročné – zejména pokud jsou počítány vzdálenosti s využitím stereometrie. V [4] je zvolena kombinace kamery a laserového dálkoměru, který pomáhá částečně řešit problém s určováním vzdálenosti v obraze. Oba senzory mají různá omezení, jejich kombinace ale může některé odstranit. Kamera dokáže dobře sledovat objekty, které jsou barevně výrazné, také není omezena při snímání pouze na jednu rovinu jako laserový skener. Znalost barvy objektů a dalších prostorových parametrů jako výška, které je možné alespoň přibližně z obrazu odečíst, pomáhá při jejich rozlišování – převážně tam, kde samotné měření vzdálenosti nestačí. Laserový skener měří pouze v jedné rovině a z detekovaných objektů se tak špatně získává více informace. Rozlišení objektů jen na základě rozložení a počtu bodů je proto složité. Hlavní výhodou laserového skeneru je schopnost přesně určit vzdálenost a detekovat tak pohyb.

## 1.2 Cíle bakalářské práce

Tématem je *Detekce pohyblivých objektů v mobilní robotice*, cílem práce je implementace algoritmu pro detekci pohyblivých objektů z jedoucího robotu a ověření jeho funkčnosti. Robot je vybaven odometrickým systémem a laserovým skenerem. Motivací je využití pro mapování, kde implementovaný algoritmus *SLAM & DATMO* má zajišťovat filtraci pohyblivých částí skenu (pravděpodobně kolemjdoucích) a tedy i kvalitnější výslednou mapu. Výhodou je kromě odstranění rušivých pohyblivých objektů z výsledné mapy také zpřesnění souřadnic bodů v mapě při lokalizaci s využitím *ICP*. Algoritmus *ICP* (viz. 2.4.2.3) byl pro práci poskytnut, jeho součástí je knihovna pro ukládání dat do struktur *KD stromu*. Knihovna je také využita samostatně pro ukládání dat do mapy.

## Kapitola 2

# Současný stav problematiky

### 2.1 Proximitní senzory

Pro mobilní i statické roboty je pro reprezentaci okolního prostředí klíčový senzor pro měření vzdálenosti. Informace o vzdálenosti k překážkám nebo objektům, se kterými má robot pracovat je nezbytná, pokud má robot jednat autonomně. Mezi tyto senzory patří:

- Sonar
- IR proximitní senzor
- Laserový dálkoměr

#### 2.1.1 Sonar

Sonary určují vzdálenost k nejbližší překážce měřením času mezi vysláním a detekcí ultrazvukového pulsu. S principem jsou spojeny problémy s teplotní závislostí rychlosti zvuku ve vzduchu, Dopplerovým efektem nebo vzdušným prouděním. Některé povrchy zvukové vlny pohlcují a jsou proto hůře detekovatelné, u jiných nastává zrcadlový odraz a naměřená vzdálenost je tak chybná. Sonar se využívá především pro kolizní detekci, kde je výhodný jeho široký měřicí úhel. Podobně jako IR proximitní senzor je sonar velmi dostupným a v mobilní robotice často používaným senzorem.

#### 2.1.2 IR proximitní senzor

IR proximitní senzor vysílá infračervené světlo, ve svých jednodušších provedeních má pouze dvojhodnotový výstup – vyslaný paprsek byl/nebyl detekován. Před senzorem se tedy do nastavené vzdálenosti nachází resp. nenachází překážka. Složitější varianty fungují na triangulačním principu a měří vzdálenost k překážce, dosah takového senzoru je desítky cm.

### 2.1.3 Laserový dálkoměr

Laserový dálkoměr je pro lokalizaci nejpoužívanějším senzorem, na kterém je založena i tato práce. Vzdálenost k překážce je vypočtena z času naměřeného mezi vysláním a detekcí laserového paprsku nebo z jeho fázového posuvu. Levnější modely měří vzdálenost triangulací. Pro správnou detekci vyslaného paprsku je nutné, aby měřený povrch měl správnou odrazivost. Povrch nesmí pohltit příliš mnoho energie záření – odražené světlo je v takovém případě pro detekci příliš slabé. Vhodné ale nejsou ani zrcadlovité povrchy, kdy nedojde k rozptylu záření na povrchu a jeho návratu k detektoru, ale paprsek je odražen dále. Zrcadlový odraz způsobuje falešná měření, případně není laserový paprsek detekován vůbec.

V mobilní robotice jsou velmi populární dálkoměry, jejichž paprsek je rozmítán v jedné rovině. Dálkoměr (laserový skener) vrací sérii měření s nastaveným úhlovým rozlišením. Například v této práci je použit laserový scanner *SICK* (viz. 4.1) s nastavením poskytujícím 361 měření pod úhlem  $180^\circ$  (úhlové rozlišení  $0.5^\circ$ ). Jedním z nedostatků laserových scannerů je jev „mixed pixel“, při kterém vznikají falešná měření – na rozhraní dvou různě vzdálených překážek. Stopa pohybujícího se laserového paprsku odpovídající měření jednoho bodu je v podstatě krátká úsečka, proto se na hranici dvou různě vzdálených objektů část paprsku odrazí od vzdálenějšího předmětu a část od bližšího. Vyhodnocovací elektronika následně vzdálenost určí v intervalu mezi oběma hodnotami. Problematikou „mixed pixel“ se hlouběji zabývá [8], kde je navržena filtrace na základě příslušnosti bodů k úsečkovým segmentům a vzdálenosti k nejbližším sousedům.

Existují i scannery snímající trojrozměrný obraz okolí rozmítáním paprsku ve dvou rovinách současně, bývají velmi drahé a v důsledku měření vzdálenosti velkého počtu bodů i pomalé. Pro získání trojdimenzionálního obrazu předmětu nebo trojdimenzionální mapy je často používán naklápěný 2D scanner.

## 2.2 Detekce pohyblivých objektů

### 2.2.1 Segmentace nového skenu

Naměřený sken má formu vektoru vzdáleností uspořádaných podle úhlu měření. Prvním krokem je rozdělení skenu na objekty. Rozdělení lze provést na základě rozdílu hodnot sousledných měření. U měření patřících jednomu objektu je rozdíl mezi souslednými hodnotami malý, pokud se před objektem nachází jiný objekt, bude rozdíl na rozhraní mezi nimi podstatně větší. Práh pro rozdělení scanu do objektů je možné určit na základě experimentu s různými hodnotami.

Segmentace se provádí i pro měření v kartézských souřadnicích přímým výpočtem vzdálenosti (popř. kvadrátu vzdálenosti – pro zjednodušení výpočtu) mezi souslednými body ve skenu, postup byl implementován v [3].

Rozlišení odděleného segmentu je důležité zejména pro pohyblivé objekty. Postup ale z principu nefunguje dobře pro objekty pohybující se blízko zdi nebo shlukující se k sobě. V [5] implementují autoři metodu založenou na předpokladu, že objekty v okolí robotu jsou konvexní nebo je lze rozložit na „přibližně konvexní“ útvary. Pro každý segment získaný postupem porovnávání sousledných naměřených hodnot je spočtena viditelná část konvexního obalu. Vzdálenosti každého bodu segmentu od spočteného konvexního obalu jsou porovnány

se stanoveným prahem. Pokud hodnota vzdálenosti práh překročí, objekt nelze považovat ani za "přibližně konvexní". V místě, kde je vzdálenost bodu od obalu maximální je objekt rozdělen, postup je aplikován znovu na oba nově získané segmenty.

## 2.2.2 Rozlišení pohyblivých segmentů

Zde jsou uvedeny dva způsoby detekce pohyblivých objektů:

- Detekce založená na konzistenci mezi statickou mapou a novým skenem
- Detekce založená na porovnávání s mapou pohyblivých objektů.

### 2.2.2.1 Konzistenční detekce

Algoritmus se skládá ze dvou kroků – detekce pohyblivých bodů v novém scanu a kombinace výsledků ze segmentace a výsledků prvního kroku. Postup byl implementován v [11], kde je použita metoda založená na Kalmanově filtru pro prvotní odhad transformace statické mapy do souřadného systému nově získaného scanu. Transformace je dále upřesněna porovnáním obou skenů algoritmem *ICP* a mapa je převedena z kartézských do polárních souřadnic. V této souřadné soustavě je již snadné rozpoznat pohyblivé body porovnáním hodnot pro jednotlivé úhly. Segment je identifikován jako potenciální pohyblivý objekt pokud je pohyblivých bodů v segmentu více než empiricky určených 50%. Postup má nevýhody – je založen na pohybu, dočasně statické objekty (stojící člověk) proto nebudou rozpoznány. Pokud je perioda mezi jednotlivými skeny příliš krátká, rozdíly poloh pomalu se pohybujících objektů budou nepatrné a objekt také nebude detekován. Pro velikost periody snímání laserového skeneru nemusí být menší hodnota nutně lepší a je možné nalézt optimum pro konkrétní aplikace.

### 2.2.2.2 Detekce na základě porovnávání s mapou pohyblivých objektů

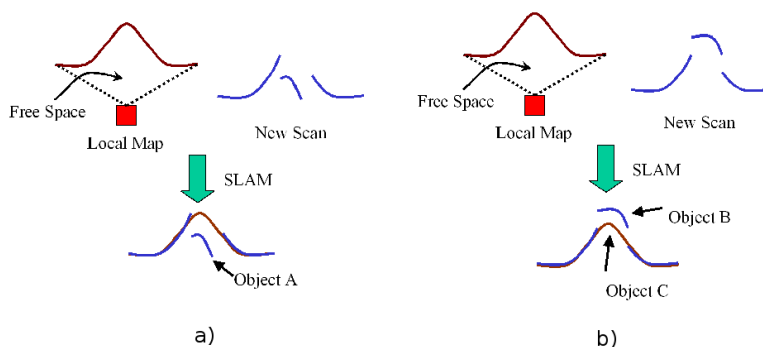
„Mapa pohyblivých objektů“ poskytuje informaci o lokacích předchozích výskytů pohyblivých objektů (mapou jsou nazývány souhrnné údaje z jednotlivých instancí sledovacího algoritmu). Pokud se nějaký segment z nového skenu nachází v oblasti, kde se v minulém cyklu algoritmu nacházel pohyblivý objekt, reprezentuje s vysokou pravděpodobností pohyblivý objekt. Metoda není schopna odhalit nové pohyblivé objekty, využívá se pouze pro sledování již detekovaných.

V [3] je popsána metoda pro redukci prohledávaného prostoru při hledání nového segmentu nacházejícího se v oblasti, kde se předtím vyskytoval pohyblivý objekt. Kolem každého nového segmentu i známého sledovaného objektu (odhadu jeho pozice v aktuálním čase) je „opsán“ myšlený obdélník. Vzhledem k případné náhlé akceleraci objektu (pro průměrnou hodnotu  $10m/s^2$ ) je obdélník sledovaného objektu rozšířen na každou stranu o  $50mm$ . Korrespondence mezi novým segmentem a sledovaným objektem jsou hledány pouze v případě, kdy se překrývají jejich obdélníky.

### 2.2.2.3 Detekce s informací o volném prostoru

Informace o volném prostoru je obsažena např. v mřížce obsazenosti (viz. 2.3.1). Detekce pohyblivých objektů je takto řešena v [12]. Druhým případem je postup pro statické detektory, kdy je ukládána největší naměřená vzdálenost jako hranice prostředí a jakékoli bližší měření proto musí patřit pohyblivému objektu (implementováno v [7]). Konkrétní postup má dvě pravidla:

- **Pravidlo 1:** Pokud je z předchozích měření známo, že je konkrétní místo v mapě neobsazené a v novém skenu je na místě nalezen objekt, potom se s velkou pravděpodobností jedná o pohyblivý objekt.
- **Pravidlo 2:** Pro objekt  $B$  (viz. 2.1) platí, že o něm nemůžeme pouze z porovnání dvou skenů s jistotou nic říci, neboť mohl být pouze zakryt objektem  $C$ , o tom ale víme, že je pohyblivý.



Obrázek 2.1: Detekce pohyblivých objektů (převzato z [11]): (a) Pravidlo 1, (b) Pravidlo 2

## 2.3 Reprezentace prostředí

Způsob reprezentace prostředí z velké části závisí na účelu, ke kterému je mapa využita. Některé typy map jsou vhodnější pro řízení pohybu robota, jiné pro plánování trajektorie. Možné dělení map:

- Senzorické
- Geometrické
- Topologické
- Symbolické

**Senzorické mapy** pracují přímo s naměřenými daty, tato reprezentace je poměrně paměťově náročná. Tvorba mapy je ale jednodušší, protože vstupní data nevyžadují příliš rozsáhlé zpracování. Bez některých pomocných struktur je získávání informace v mapě uložené složité. Mapa je výhodná zejména pro řízení pohybu robota.

**Geometrické mapy** popisují prostor pomocí geometrických primitiv – úseček nebo různých polygonů. Tento druh map je paměťově úsporný, náročnější je ale stavba, ne z každého prostředí je možno kvalitně extrahovat geometrická primitiva. Myšlenkou je reprezentovat velké množství sensorických dat malým množstvím geometrických útvarů při zachování obsažené informace. Geometrické mapy jsou vhodné pro plánování trajektorie.

**Topologické mapy** mají vyšší stupeň abstrakce než sensorické nebo geometrické mapy. Nepracuje se zde se souřadnicemi jednotlivých objektů, ale ukládají se jejich vztahy – například která lokace je z konkrétního místa přímo dostupná. Topologická mapa se používá pro plánování na vyšším stupni.

**Symbolické mapy** dosahují nejvyššího stupně abstrakce, informace v ní obsažené se většinou nedají zjistit sensorickými měřeními. Jsou to různé informace o vlastnostech jednoho nebo více předmětů.

Vzhledem k cílům a zadání bakalářské práce se dále zabývám pouze mapami sensorickými. Mapy topologické a symbolické jsou zde ze svého principu nevhodné. Práce je zaměřena na zpracování bodových dat, proto nebyla zvolena ani geometrická mapa.

### 2.3.1 Mřížka obsazenosti

Mřížka obsazenosti reprezentuje svět jako pole čtvercových buněk obsahující míry pravděpodobnosti, že daná buňka je obsazena (na jejím místě se nalézá překážka). Je odolná vůči šumu, aktualizace je jednoduchá, rychlá a její přesnost závisí na zvolené velikosti buňky. Při vyšších rozlišeních rychle rostou nároky na paměťový prostor a výpočetní výkon (nejvíce pokud je potřeba v mřížce vyhledávat). Z každého měření se určuje hodnota pravděpodobnosti pro jednotlivé buňky mřížky, do výsledné mřížky (mapy) jsou získané hodnoty přidávány za uplatnění různých pravidel. Podle způsobu přidávání nových měření dělíme mřížky do tří skupin:

- Bayesovy mřížky
- Dempster-Schaferovy mřížky
- Fuzzy mřížky

Implicitně je každá buňka nastavena na 0.5 (50% pravděpodobnost výskytu překážky), protože o ni zpočátku nic není známo. Základním nedostatkem mřížek obsazenosti je jejich paměťová náročnost, která se odvíjí od velikosti mapovaného prostoru a nikoli od složitosti a počtu překážek v prostředí. Právě struktura přechodu mezi volným prostorem a překážkou nese nejvíce informace. Problém nastává při mapování rozsáhlejších prostor (a malé velikosti buňky). Vlastnost ale může být i přínosem – mřížky obsazenosti dovolují dobře pracovat s volným prostorem, protože je v nich uložen také. Není tak známa pouze hranice prostoru, kterou je třeba dále interpretovat – je možné přímo odečíst kde je překážka a kde není. Mřížky jsou také vhodné pro slučování dat z různých sensorů, nebo je lze použít jako základ pro extrakci dat například pro geometrickou mapu.

### 2.3.2 Bodová mapa

Dalším důležitým reprezentantem sensorických map je bodová mapa – zejména její typ navržený v [8]. Je vhodnou alternativou k mřížkám obsazenosti pro účely lokalizace. Data se do bodové mapy nekládají přímo tak, jak přicházejí ze senzoru, ale jsou filtrovány s ohledem na stávající obsah mapy. Nový bod je do mapy vložen pouze, pokud se v jeho nastaveném okolí v mapě již nenachází jiný bod.

V této bodové mapě jsou data ukládána do vyhledávací struktury K-D stromu (viz. 3.2.2). Hledání blízkých bodů je jednou z nezákladnějších operací, které se nad bodovou mapou provádějí. Je klíčové nejen při samotném budování mapy, ale také během využívání mapy pro účely lokalizace. Ke každému bodu z nově příchozího nezlokalizovaného skenu se v mapě hledá nejbližší bod v rámci běhu *ICP* algoritmu. Bodová mapa je implementována v této práci, důkladněji je popsána v kapitole 3.2.1.

## 2.4 Lokalizace polohy

Pro lokalizaci existuje množství metod založených na různých principech. Provedeme rozdělení na pravděpodobnostní přístupy pracující s neurčitostí a na přístupy pracující bez ní. Pokud není brána v úvahu neurčitost, je výsledkem jen odhad stavu bez informace o míře důvěry a není tak nic známo o pravděpodobnosti a velikosti chyb.

### 2.4.1 Pravděpodobnostní přístupy

Pravděpodobnostní přístupy bývají robustnější, zejména při zpracování dat obsahujících silnější šum. Tyto přístupy tak dokáží mnohem lépe pracovat s nepřesnostmi v měřeních i použitých modelech a dokáží se samy zotavit z některých chyb. Nevýhodou naopak je zpravidla menší přesnost a větší výpočetní náročnost. V detekci a sledování pohyblivých objektů je často řešená úloha sledování osob statickými detektory. Hlavními problémy jsou zde zastínění sledovaného objektu a shlukování objektů. V úlohách tohoto charakteru se často používají částicové filtry.

#### 2.4.1.1 Částicové filtry

Částicové filtry (*particle filters*) jsou důležitým představitelem pravděpodobnostních metod. Založeny jsou na statistické metodě *Monte Carlo*, kde se pro simulaci využívá náhodných hodnot. Výhodou je, že nekladou omezení na tvar hustot pravděpodobnosti a proto pracují i s negaussovskými a nelineárními dynamickými procesy. Klíčovou myšlenkou částicových filtrů je reprezentace stavu váženou množinou vzorků (částic) tak, aby výsledná diskretní distribuce odpovídala výchozí spojitě distribuci. Metoda reprezentuje multimodální stavové hustoty, což má pozitivní vliv na robustnost při použití ve sledovacích algoritmech. Výpočetní náročnost může být vysoká, ale lze ji za běhu algoritmu měnit úpravou počtu částic.

Částicové filtry byly implementovány např. v [2], několik statických laserových skenerů bylo rozmístěno na rušné ulici. Detektor je založen na kombinaci Kalmanova filtru (jako srovnávací metody) a „RaoBlackwellized Monte Carlo data asociation“ filtru. Druhá jmenovaná metoda se s problémy zastínění, shlukování nebo korelovaného pohybu vyrovnávala



podle závěru autorů lépe a docházelo zde méně často k selháním. Detekce osob z pohyblivého robotu je zpracována v [10] a [1], v obou případech s využitím částicových filtrů.

### 2.4.1.2 Kalmanovy filtry

Kalmanův filtr (*KF*) patří mezi nejoblíbenější pravděpodobnostní metody pro určování stavu dynamických systémů (např. sledování polohy objektů v mobilní robotice). *KF* umožňuje pracovat s neurčitostí, stav systému reprezentuje střední hodnotou a rozptylem gaussovského rozložení. *KF* možná efektivně odhadovat stav procesu při minimalizaci střední hodnoty kvadratické odchylky. Myšlenka byla představena roku 1960 R. E. Kalmanem.

*KF* nejprve odhadne budoucí vývoj stavu procesu (přičemž se zvýší neurčitost gaussovského rozložení) a následně získá nová měření (zpětnou vazbu), která ale obsahují šum. Neurčitost se tak opět sníží. Formálně můžeme celý výpočetní cyklus rozdělit do dvou částí – časového a měřicího kroku. V časovém kroku probíhá projekce aposteriorních dat z minulého cyklu do apriorních dat pro aktuální cyklus. Měřicí krok zajišťuje zpětnou vazbu zapracováním nových měření do apriorního odhadu, čímž vytvoří aposteriorní odhad a cyklus může být zopakován.

$$\hat{X}_1^- = A\hat{X}_{k-1} \quad (2.1)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (2.2)$$

$$K_k = P_k^- C^T (CP_k^- C^T + R)^{-1} \quad (2.3)$$

$$\hat{X}_k = \hat{X}_k^- + K_k(Z_k - C\hat{X}_k^-) \quad (2.4)$$

$$P_k = (I - K_k C)P_k^- \quad (2.5)$$

V Kalmanově filtru se pracuje s neurčitostí (šumem) při určování stavu procesu a s šumem v naměřených datech. U šumů se předpokládá jejich vzájemná nezávislost, standardní pravděpodobnostní rozložení a také možnost aproximace bílým šumem. Šum měření je v rovnicích Kalmanova filtru reprezentován pomocí kovarianční matice  $R$ , šum při určení stavu procesu představuje kovarianční matice  $Q$ . Obě matice se mohou v každém výpočetním cyklu měnit, avšak je možno je uvažovat konstantní. Výsledná neurčitost odhadu stavu je zahrnuta v kovarianční matici  $P$ . Matice  $A$  je maticí dynamiky sledovaného procesu, která promítá stav v předchozím cyklu na odhad stavu v aktuálním čase, matice  $B$  určuje vliv případných řídicích signálů. Matice  $C$  (model senzoru) v sobě zahrnuje vztah mezi stavem procesu a měřeným výstupem. Matice  $A$  a  $C$  také mohou být proměnné, ale ve většině případů je možné uvažovat je konstantní. Matice  $K$  se nazývá „Kalmanovo zesílení“, určuje míru důvěry v nové měření oproti odhadům stavu a slouží k minimalizaci střední hodnoty kvadratické odchylky aposteriorního odhadu. Máme-li vysokou důvěru v kvalitu naměřených dat a tedy malé hodnoty v  $R$ , matice  $K$  obsahuje vyšší hodnoty. Dojde tak k posílení vlivu měření na úkor odhadu (viz. rovnice 2.4). Pokud předpokládáme konstantní matice  $R$  a  $Q$ , zjistíme,

že také matice  $P$  a  $K$  se v krátkém čase ustálí a budou moci být považovány za konstantní. Potom je možné spočítat obě matice dopředu a nikoli počítat průběžně za chodu systému. Předpoklady konstantních matic nemusí nutně představovat dobré řešení, protože podmínky se mohou měnit. Kalmanův filtr je optimální z hlediska střední hodnoty kvadratické odchylky odhadu, to ale platí pouze pro lineární systémy (je možné je popsat lineárními diferenciálními rovnicemi). Pro ostatní situace je třeba využít „rozšířeného Kalmanova filtru“ (*Extended Kalman Filter – EKF*), který celý problém linearizuje vůči aktuální poloze robotu.

Systémová matice  $A$ , zajišťující projekci minulého stavu systému do současnosti, je modelem systému. Zde se volí mezi modelem konstantní rychlosti nebo konstantního zrychlení. Model pro konstantní zrychlení je mnohem komplexnější a model konstantní rychlosti v sobě v podstatě zahrnuje (pro zrychlení rovno nule). Pro sledování konstantní rychlosti se pohybujících objektů je výhodnější použít jednodušší model, který dává i kvalitnější výsledky – podrobné porovnání viz. [11]. V publikaci výše zmíněného autora je představen *Interacting Multiple Model – IMM* algoritmus. *IMM* rozšiřuje množinu modelů o „stop model“, který je získán zjednodušením modelu konstantní rychlosti (model přesněji aproximuje stojící robot). Algoritmus následně přikládá výsledkům jednotlivých modelů váhy na základě úspěšnosti odhadů v nedávném čase.

## 2.4.2 Přístupy bez pravděpodobnosti

Tyto metody bývají přesnější a rychlejší než pravděpodobnostní, ale předpokladem pro jejich funkci je splnění přísnějších omezujících podmínek. Při jejich nesplnění nebo vážnější chybě metody úplně selhávají. Do kategorie patří tzv. *scan matching* metody (porovnávání skenů) nebo odometrie.

### 2.4.2.1 Odometrie

Odometrie je jedním ze základních systémů pro určování polohy u kolových robotů. Poloha robotu se vypočítává z rychlosti otáčení jednotlivých kol snímané inkrementálními senzory. Používají se různá optická nebo magnetická čidla, integrací jejich výstupů jsou získány délky drah jednotlivých kol, ze kterých je vypočítána poloha robotu. Přesnost odometrie záleží na přesnosti (jemnosti rozlišení) inkrementálních senzorů, na přesnosti identifikace fyzických parametrů robotu a na tření mezi koly a povrchem. Nejdůležitější fyzické parametry robotu jsou obvod kol a jejich rozteč – záleží na typu pohybového ustrojí robotu. Nejvhodnější je dvoukolový robot (s pomocnými kolečky) s osou otáčení ve středu nápravy. Odometrii lze použít i pro roboty s pásovými podvozky. Pásky ale zatáčejí smykem, kde i poměrně malá změna koeficientu tření mezi podlahou a podvozkem může vést k velkým chybám. Řešením může být dodatečné měřicí kolo. Při kvalitním nastavení všech konstant a příznivých okolních podmínkách může být odometrie velmi přesná. Vzhledem k integračnímu charakteru měření polohy však chyba neustále roste. Malá a ve většině případů stabilní chyba může za jistých podmínek nečekaně silně narůst (při prokluzu kol). Chyby tohoto druhu jsou velmi nepříjemné a ze samotného odometrického systému takové situace většinou není možné odhalit. Proto se pro další upřesnění lokalizačního algoritmu používá *ICP* pracující s daty z laserového skeneru. Upřesňující krok umožňuje selhání odometrie detekovat a opravit. Nevýhodou je, že algoritmus *ICP* je citlivý na rotační transformace, rotace jsou ale také největším zdrojem chyb i u odometrie.

### 2.4.2.2 Scan matching

Lokalizace *scan matching* hledá transformaci mezi nově naměřenými daty (nejčasněji body nebo úsečkami) a daty naměřenými v některém z předchozích kroků k určování pozice robotu. Ve většině aplikací lze i při pohyblivém robotu předpokládat, že sken byl naměřen z jednoho místa a vznikne tím pouze zanedbatelná chyba. V případě, že využíváme laserový skener při vyšších rychlostech – například za jízdy v automobilu, není tento předpoklad zcela splněn. V takovém případě je vhodné použít pro korekci údaj o rychlosti pohybu robotu a úhlové rychlosti rotace zrcátka skeneru. Úlohu *SLAM & DATMO* pro aplikaci v automobilu pohybujícím se běžnými rychlostmi řeší [12].

Metody *scan matching* lze podle struktury dat, se kterou pracují rozdělit do následujících skupin:

- Feature to Feature (F2F) – porovnávání geometrických útvarů
- Point to Point (P2P) – porovnávání bodů
- Kombinace obou metod

Metoda *F2F* má za běhu nejmenší nároky na výpočetní výkon, stovky bodů příchozích senzorických dat jsou redukovány na desítky geometrických primitiv (features) jako jsou přímky, kružnice nebo rohy. Postup je výhodný zejména pro využití ve vnitřních prostorech, kde je možno extrahovat geometrická primitiva robustně. Nevhodný je pro venkovní prostory s množstvím objektů jako keře, stromy a teréní nerovnosti. Jejich tvary jsou geometricky složité na popis. Myšlenkou *F2F* metod je minimalizace množství dat ze senzoru při co nejlepším zachování informace v nich obsažené. S menším množstvím dat klesá i výpočetní výkon potřebný pro zpracování a dochází tak k urychlení algoritmů. Pro kvalitu výsledků *F2F* metod je klíčové, jak dobře lze z prostředí extrahovat geometrická primitiva.

Oproti tomu metoda *P2P* využívá všechna senzorická data a stává se tak robustnější v případech nevhodných pro *F2F* přístup. Pro přímé zpracování dat z laserového skeneru je nejpobulárnější *ICP* algoritmus, kvalitní výsledky dává pouze pro malé odchylky a je zvláště citlivý na větší rotační transformace porovnávaných skenů. Pro správnou a funkci je důležitý dobrý počáteční odhad hledané transformace. V případech příliš velké odchylky (například vážně selhání prvotního odhadu pozice) je pravděpodobné, že výsledek nebude správný, což je hlavní slabinou *ICP*.

Při kombinaci obou metod (například *Point to Feature*) je snahou maximálně využít výhod jednotlivých postupů a minimalizovat jejich nedostatky.

### 2.4.2.3 ICP algoritmus

*ICP* je algoritmus pro hledání transformace mezi dvěma obrazy (zde pro série měření laserového skeneru). Akronym je zkratkou pro *Iterative Closest Point*, již z názvu vyplývá, že algoritmus pracuje iterativně. V každé iteraci se hledají nejbližší body z obou porovnávaných skenů. Z vlastností vektorů mezi dvojicemi nejbližších bodů je určena transformace minimalizující vzdálenost skenů. Vzdáleností skenů je myšlen součet čtverců vzdáleností

jednotlivých dvojic nejbližších bodů. *ICP* algoritmus je velmi citlivý na velikost počáteční rotace a translace mezi skeny – pokud jsou příliš velké, může se stát, že algoritmus nebude konvergovat a zcela selže. Protože *ICP* nepracuje s neurčitostí, nezískáme v takovém případě z navraceného řešení téměř žádnou informaci.

Algoritmem *ICP* lze pro účely lokalizace porovnávat dva sousledné skeny, nový sken s některým ze starších, nebo s mapou. Výsledkem ale v praxi nikdy není zcela přesná transformace – bude zatížena malou chybou. V případě porovnávání dvou sousledných skenů se bude tato chyba stále načítat a nebude zprincipu ničím omezena. Problém je podobný jako u odometrie, ale v menším rozsahu. Částečné nápravy dosáhneme porovnáváním nového skenu s vhodně zvoleným starším skenem, který je možné prohlásit za statický. Nejlepšího výsledku bude dosaženo při použití mapy místo staršího skenu. Mapa ale musí být ve vhodném formátu – tedy bodová (implementována v [8]), v jiném případě je postup komplikovanější. V [11] byla použita mřížka obsazenosti, problém byl z původního hledání korespondujících bodů transformován na problém hledání maximální pravděpodobnosti pro pozici nového skenu.

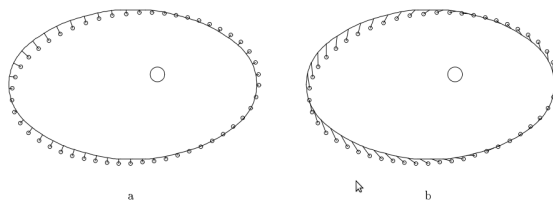
Porovnávání skenů je ztíženo, když se obou skenech vyskytují regiony, které se nepřekrývají a proto je není možno pro porovnávání *ICP* algoritmem použít. Pro body, které se v regionech nacházejí neexistuje vhodný bod do páru, algoritmus hledající nejbližšího souseda přesto ale bod nalezne. Situace je poměrně častá – příkladem může být robot jedoucí za roh, kdy nový sken odkrývá zatím neznámé prostory. Problém se řeší například filtrací jednotlivých párů nejbližších bodů na základě jejich vzdálenosti – pokud je vzdálenost vyšší než prahová vzdálenost  $d_r$ , pár se pro další výpočty nepoužije. Rozdělující vzdálenost  $d_r$  není konstantní, ale mění se s každou iterací, určována je samotnými daty. V prvních iteracích jsou vzdálenosti bodových párů přirozeně mnohem větší než v koncových, to musí být při stanovování prahu  $d_r$  zohledněno. Neodstraněné chybné páry narušují chod algoritmu, zpomalují jeho konvergenci a mohou vést až k selhání celé metody.

Jiné řešení problému potlačení vlivu měření, která k sobě nemají ekvivalent v druhém skenu (také platí pro chybná měření) je navrženo v [9]. Většina *P2P* lokalizačních systémů je založena na principu optimalizace kritéria nejmenších čtverců. Tato metoda však počítá s Gaussovským rozložením dat, to ale vychýlená měření nespĺňují a dokáží proto velmi významně zasáhnout do výsledku. Ve výše zmíněné práci je navrženo použití *Lorenzovského estimátoru*, který má nelineární funkci vlivu různých hodnot vzdáleností bodů na výslednou transformaci a odlehlá měření dokáže dobře potlačit.

Zvláštní pozornost je dobré věnovat také měřením obsahujícím maximální vzdálenost – podle navraceného chybového kódu senzoru se může jednat o volný prostor, měření mimo rozsah nebo nedetekovaný paprsek. Maximální měřitelná vzdálenost je známou konstantou, která se ale transformací jednoho ze skenů mezi souřadnými soustavami lehce změní, to působí problémy při další identifikaci. Je proto nutné zajistit, aby měření byla vyjmuta ze skenu nebo nebyla transformována.

Mezi nedostatky *ICP* patří také rychlost konvergence, hlavně v případě transformací obsahujících rotace. Vektory mezi korespondujícími body mají velmi nekonzistentní směry a navzájem se často vyruší. Přirozenou vlastností *ICP* je snaha o nahrazování rotace translací.

Problematika je podrobně řešena v [6], autoři zde navrhnou *Iterative Matching Range Point (IMRP)* algoritmus a kombinaci obou předchozích (*ICP* a *IMRP*) nazvanou *Iterative Dual Correspondence (IDC)* algoritmus.



Obrázek 2.2: Bodové páry vytvořené (a) ICP (b) IMRP (převzato z [6])

**IMRP** je v základech stejný jako *ICP*, pouze nahrazuje pravidlo hledání „vzdálenostně“ nejbližšího souseda jiným. Nové pravidlo ignoruje translace, respektive předpokládá, že je možno je zanedbat. K bodu  $P$  s naměřenou vzdáleností  $|P|$  a úhlem  $\theta$  v polárních souřadnicích bod do páru vyhledává ve stejné vzdálenosti  $|P|$  pod úhlem  $\omega$ , kde  $\theta - B_\omega \leq \omega \leq \theta + B_\omega$ .  $B_\omega$  zajišťuje hledání pouze v blízkém okolí, pokud není bod nalezen, druhotnou podmínkou se stává rozdíl naměřených vzdáleností bodů od hodnoty  $|P|$ . Práh  $B_\omega$  není konstantní, je to exponenciální funkce  $B_\omega(t) = B_\omega(0)e^{-\alpha t}$  protože rotace jsou v prvních iteracích mnohem větší než v koncových. *IMRP* se narodil od *ICP* inklinuje k reprezentaci translace pomocí rotace, translace se zde řeší až v závěrečných krocích, kdy je práh  $B_\omega$  dostatečně malý. Konvergence algoritmu je rychlejší, ale jeho stabilita není tak dobrá jako *ICP*. Pro porovnání vytvořených bodových párů viz. obr. 2.2.

V algoritmu **IDC** je účelem využít výhod obou algoritmů v odhadech rotační a translační složky transformace – zajistit rychlost konvergence *IMRP* s stabilitu *ICP*. Algoritmus pracuje v následujících krocích:

1. Pro každý vstupní bod  $P$  se nalezne bod do páru  $P_1$  pravidlem z *ICP* a bod  $P_2$  pravidlem z *IMRP*
2. Jsou spočtena řešení  $(\omega_1, T_1)$  pro množinu bodů  $P_1$  a  $(\omega_2, T_2)$  pro množinu bodů  $P_2$
3. Výsledek je složen z obou řešení ve tvaru  $(\omega_2, T_1)$

Základní myšlenkou *IDC* je tedy vzít pro výslednou transformaci rotační složku transformace od algoritmu *IMRP* a translační od *ICP*. Výsledný algoritmus dává kvalitnější výsledky, než kterýkoli z původních samostatně. *IDC* je stabilnější a také rychleji konverguje.

## Kapitola 3

# Implementace SLAM & DATMO

### 3.1 Přehled jednotlivých kroků

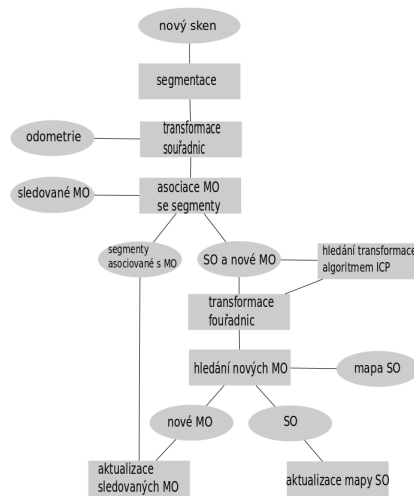
Postup algoritmu *SLAM* & *DATMO* je možné rozdělit do následujících kroků (také viz. obr. 3.1):

1. Segmentace nového skenu na samostatné (statické i pohyblivé) objekty
2. Úprava pozice robotu údaji z odometrie a odpovídající transformace nového skenu
3. Asociace již známých pohyblivých objektů s novými segmenty, ty jsou následně ze skenu odstraněny
4. Registrace obou skenů algoritmem *ICP* pro získání přesnějšího údaje o pozici robotu a odpovídající transformace nového skenu
5. Detekce nových pohyblivých objektů na základě hledání bodů skenu nekorespondujících s mapou
6. Doplnění bodů, které nebyly odfiltrovány jako součást pohyblivých objektů do mapy a aktualizace pozic pohyblivých objektů

#### 3.1.1 Získání a segmentace nových dat

Nedříve je získán nový sken, který se skládá ze 361 čísel odpovídajících vzdálenostem k překážkám (v metrech) v rozsahu  $0^\circ - 180^\circ$ . Druhým krokem je načtení aktuální pozice robotu z odometrie, odometrická data obsahují inkrement polohy robotu ve směru osy  $X$ ,  $Y$  a úhlu natočení  $\phi$ . Úhel  $\phi$  je počítán vzhledem k definovanému počátečnímu natočení, které je zvoleno jako  $90^\circ$ .

Sken je ve formě série vzdálenostních měření, kterou použitý algoritmus pro segmentaci předpokládá (je založen na porovnávání sousledných vzdáleností ve skenu – viz. 2.2.1). Počáteční a koncový index pro každý segment je uložen do seznamu indexů, který zůstává po



Obrázek 3.1: Postup při zpracování nového měření

dobu cyklu celého algoritmu neměnný. Indexy jsou proto platné i pro transformované body v kartézských souřadnicích, pro zachování pořadí nejsou při odstraňování body mazány, ale nahrazeny chybovou hodnotou.

Daty získanými z odometrie je upraven polohový vektor robotu, který obsahuje stejné tři složky – souřadnice  $X$ ,  $Y$  a úhel natočení  $\phi$ . Nový sken je transformován z polárních souřadnic do kartézských a následně transformací podle polohového vektoru robotu do souřadného systému mapy. V úloze lokalizace se předpokládá nepřesnost odometrického systému a nutnost korekce výsledné transformace algoritmem zpracovávajícím měření laserového skeneru, získaná transformace je proto považována za nepřesnou.

Dalším krokem je odstranění příliš vzdálených segmentů, které nemají velkou informační hodnotu – může také jít o protnutí roviny rozmítání laserového paprsku s rovinou země. Převážně jsou do kategorie zahrnuty body, u kterých nedošlo k detekci odraženého paprsku a navrácená hodnota je na maximu měřicího rozsahu (vlastnosti laserového dálkoměru podrobněji viz. 2.1.3). Odstraněny jsou také bodové segmenty, kde je pravděpodobné, že se jedná o chybné měření – tzv. *mixed pixel*. Jednobodový segment v sobě nese mnoho informace a je pravděpodobné, že v dalších skenech nebude nalezen. Pro samotný bod by jinak byl vytvořen nový model sledovaný objekt, který je jednak relativně náročný na vytvoření a likvidaci, ale také by mohlo dojít k mylnému spojení modelu s jiným segmentem po nenalezení původního. Pokud by se nejednalo o chybu, ale například o nohu židle, po případném přiblížení (při riziku kolize) by počet bodů narostl a objekt tak byl zaznamenán.

### 3.1.2 Asociace segmentů se stávajícími pohyblivými objekty

Každý z minulých cyklů rozpoznáný pohyblivý objekt má přiřazen svůj model v mapě pohyblivých objektů – *object tracker*. Pro každý model je hledán odpovídající segment z nového skenu (viz. detekce na základě porovnávání s mapou pohyblivých objektů v kap.

2.2.2.2). Sledovaný objekt je reprezentován obdélníkem „opsaným“ bodům segmentu (tzv. *bounding box*), s nímž byl model naposledy asociován. Pro každý nový segment je také vytvořen jeden obdélník. Obdélník příslušející modelu je nejdříve posunut do pozice, kde je objekt očekáván a ve všech směrech rozšířen o empiricky určenou konstantu  $B = 0.5m$  – jako pojistný mechanismus proti náhlé akceleraci (podrobněji viz. 2.2.2.2). Odhad pozice je realizován Kalmanovým filtrem (viz. 2.4.1.2), implementovaný dynamický model je modelem konstantní rychlosti. Tento model se je jednodušší než u konstantní akcelerace a pro tuto práci byl postačující. Absence akcelerace v modelu musí být kompenzována větší hodnotou konstanty  $B$ . Ve druhém kroku jsou ze všech segmentů vybrány takové, jejichž „opsaný“ obdélník je v blízkosti sledovaného objektu. Provedeno je porovnávání vzdáleností středů obdélníků s prahovou hodnotou  $d_m$ . Pro segmenty z vybrané skupiny je spočítáno, kolik procent bodů se nachází ve zvětšeném čtverci modelu a segment s nejvyšší hodnotou je vybrán. V případě, kdy by zvětšený obdélník modelu obsáhl dva malé segmenty celé, jsou dále porovnány vzdálenosti středů obdélníků. Segmenty, které jsou identifikovány s některým ze sledovaných pohyblivých objektů jsou ze skenu odstraněny. Obdélník segmentu, který byl smazán je přiřazen příslušnému modelu. Nedojde zde ale k aktualizaci pozice sledovaných objektů, to je provedeno až po budoucím upřesnění polohy robotu z *ICP*.

### 3.1.3 Upřesnění polohy nového skenu algoritmem *ICP* a detekce nových pohyblivých objektů

Nový sken po předchozím zpracování obsahuje (kromě zatím neodhalených pohyblivých objektů) pouze statické body. *ICP* (podrobněji viz. 2.4.2.3) tak může porovnat sken s mapou se zvýšenou přesností. Z výsledné transformace je upravena pozice skenu vzhledem k mapě do koncové podoby. Mapa byla zvolena bodová, důvodem byl poskytnutý *ICP* algoritmus s knihovnou pro správu *KD stromů*, nástroje jsou podrobněji popsány v následujících kapitolách 3.2.1 a 3.2.2.

Sledované pohyblivé objekty jsou odstraněny, proto všechny nesrovnalosti mezi skenem a mapou jsou potenciálními pohyblivými objekty (viz. jejich detekce v kapitole 2.2.2.1). Všem potenciálním pohyblivým objektům je přiřazen status *kandidáta* a model pro jejich sledování. Tento postup je zde proto, že objekty mohou být jen nově objevenou statickou překážkou. *Kandidáti* proto nejsou přidáni do mapy ale jsou jako pohyblivé objekty filtrováni (nejsou ale regulárními pohyblivými objekty), navíc jsou uloženy přímo naměřené body, které jim náleží. V příštím cyklu celého algoritmu jsou uschované body porovnávány s nově naměřenými body segmentu asociovaného s příslušným modelem. Algoritmus pro rozpoznání pohybu objektu je stejný jako při detekci nových pohyblivých objektů. Pokud se *kandidát*  $K$  krát nepohne, je prohlášen za statický objekt a body, které mu náleží jsou přidány do mapy. Konstantu  $K$  jsem experimentálně určil jako čtyři cykly algoritmu, hodnota je závislá na periodě skenování. V případě pohybu je pouze odstraněn status *kandidáta* a objekt je dále veden jako pohyblivý.

### 3.1.4 Aktualizace statické mapy a pozic sledovaných pohyblivých objektů

V závěru jsou pouze aktualizovány pozice sledovaných pohyblivých objektů – transformací získanou algoritmem *ICP* jsou upraveny uschované obdélníky v modelu pro každý sledovaný



objekt. Po upřesnění pozice sledovaného objektu jsou aktualizovány modely (zavolání metod Kalmanova filtru zpracovávajících nová měření).

## 3.2 Použité datové struktury

### 3.2.1 Bodová mapa

Data do mapy je možno ukládat přímo tak jak přicházejí ze senzoru – po konverzi měření laserového skeneru z polárního do kartézského souřadného systému. Podobné řešení by ale nebylo vhodné, protože v místech, kde se robot pohybuje delší dobu tak vzniká velká hustota bodů. Jedním z problémů je paměťová náročnost, která by rostla úměrně času a nikoli velikosti a struktuře prozkoumaného prostředí. Velké objemy dat jsou nevhodné i pro získávání informace z mapy. Algoritmy, které s daty pracují jsou v důsledku pomalé.

Filtrace nových dat před vložením do mapy je z výše uvedených důvodů nutností. Nejjednodušším přístupem je ukládat body pouze v případě, že se v jejich okolí o poloměru  $R$  v mapě nenachází jiné body. Hustota dat je přibližně rovnoměrná, množství závisí na velikosti zvolené konstanty  $R$ . Vyhledávání nejbližších sousedů je v mapě prováděno velmi často, struktura ukládání dat je proto důležitá pro jeho efektivitu. Vhodná je vyhledávací struktura *KD strom*.

Efektivní vyhledávání v *KD stromu* je mnohem výhodnější v případě, kdy je lokalizace založena na *ICP* algoritmu. Podstatou *ICP* je vyhledávání nejbližších sousedů za účelem tvorby bodových párů. Pro 361 bodů nového skenu a 40 iterací v jediném lokalizačním cyklu je provedeno 14440 operací vyhledávání nejbližšího souseda.

Volba bodové mapy je výhodná z hlediska chodu *ICP* a filtrace vstupních bodů. Další drobnou výhodou je vyšší přesnost porovnávání nového skenu s mapou než pro porovnávání dvou skenů. Chyba vznikající při každém hledání transformace algoritmem *ICP* se nesčítá tak rychle.

U mřížky obsazenosti lze regulovat hodnotu granularity prostředí a paměťovou náročnost uložení pomocí změny velikosti buňky v mřížce. U bodové mapy granularitu určuje filtrace příchozích dat prahováním na základě mezibodové vzdálenosti k nejbližšímu sousedu v mapě.

Jiný způsob filtrace nových dat vede současně k normalizaci mezibodových vzdáleností v novém skenu. Je zvolena minimální vzdálenost  $R_s$  dvou nejbližších bodů, podobně jako u předchozího algoritmu, dále se postup liší. Konstanta  $R_s$  se volí o několik řádů menší, než je rozlišení senzoru. Postup se skládá z následujících kroků:

1. Nad nově příchozími daty je vybudována vyhledávací struktura *KD strom* a pro každý z bodů je postupně hledán nejbližší soused
2. Pokud se soused nachází v menší vzdálenosti než je zvolená prahová hodnota  $R_s$ , je ověřeno zdali i pro souseda je bod, odkud bylo původně iniciováno hledání, nejbližším sousedem
3. Pokud je nalezena dvojice popsaná v předchozím bodě, je nalezený soused zrušen a původní bod posunut do váženého těžiště obou bodů

Celý cyklus je opakován dokud žádné dva body nemají mezi sebou vzdálenost menší než  $R_s$ . Stejný algoritmus je možno použít i pro normalizaci bodových vzdáleností při přidávání bodů do mapy. Jediným rozdílem je velikost konstanty minimální vzdálenosti bodů  $R_m$ . Volena je tak, aby data byla rovnoměrně rozprostřena v celém skenu a zároveň se zbytečně nezmenšovala kvalita výsledné mapy.

Zvláště může být provedena filtrace chybných měření, které se vyznačují menším počtem sousedů ve svém okolí než je průměr. Při jenorázovém průchodu nového skenu je ke každému bodu vyhledáno  $K$  nejbližších sousedů. Pokud je  $K$ -tý nejbližší soused dále, než jiný práh  $R_c$ , bod zrušíme jako chybné měření.

Na bodové mapě s normalizací vzdáleností v novém skenu i mapě a filtrací chybných měření je založena práce [8], která je zaměřena na mapování bez uvažování pohyblivých objektů. V této implementaci *SLAM & DATMO*, která má pracovat jako pomocný algoritmus pro detekci pohyblivých objektů, nejsou normalizace vzdáleností provedeny. Mapa zde není tak důležitá a je ušetřena výpočetní kapacita.

### 3.2.2 KD strom

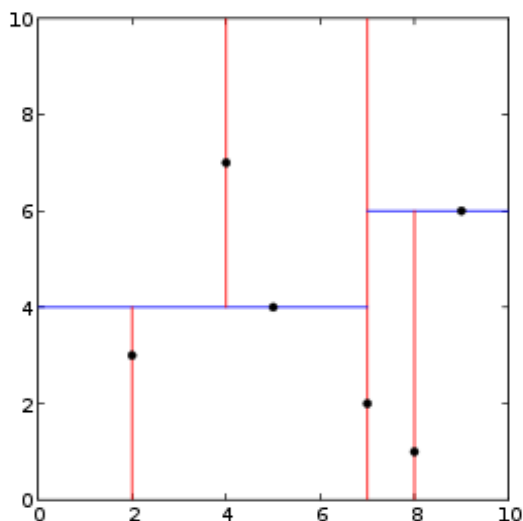
*KD strom* je binární vyhledávací strom, kde každý uzel je  $K$ -dimenzionální bod. Uzly stromu obsahují na každé úrovni levý a pravý podstrom a klíč – data skládající se z čísla dělicí dimenze a vlastních bodů. Body zde mohou vystupovat jako takzvaný „pivot“ – bod, kterým prochází další dělicí hyperrovina – nebo jako prostorový bod ležící v listu stromu. List stromu může být i prázdný. Obecně platí, že hodnoty klíčů v pravém podstromu jsou vždy větší než hodnota klíče mateřského uzlu. Naopak v levém podstromu jsou hodnoty klíče vždy menší než hodnota klíče mateřského uzlu. *KD strom* umožňuje vyhledávání v  $K$ -dimenzionálním prostoru, pravidelně se střídají jednotlivé dimenze klíčů. Stavbu *KD stromu* ve dvojdimenzionálním prostoru ilustruje obrázek 3.2.

Pokud je při stavbě stromu k dispozici celý soubor dat, postup je rekurzivní a spočívá v následujících krocích:

1. Výběr rozdělující dimenze a pivotu určujícího hyperrovinu dělicí zbylé body souboru na dvě části
2. Samotné rozdělení bodů do skupin (existují různá kritéria výběru)

Na každou z obou nových skupin bodů je aplikován postup znovu až jsou nakonec získány listy stromu (prázdné nebo obsahující jeden prostorový bod). Při postupném budování stromu je nejprve ve stromu provedeno hledání, kterým je zjištěno do jakého místa struktury nový bod zařadit. Do nalezeného listového uzlu je bod vložen v případě, že byl prázdný. Pokud se zde již bod nacházel, je rozhodnuto, který z nich se stane pivotem. Pivot určí další rozdělující dimenzi, zbylý bod je umístěn do odpovídajícího prázdného listu upravené větve stromu.

Hledání nejbližšího souseda v *KD stromu* probíhá rekurzivně. Postupným „odřezáváním“ nevyhovujících větví stromu je poměrně rychle eliminován velký prostor, až je nakonec dosaženo koncového listu. Strom je prohledáván do hloubky. Bod v koncovém listu (nebo poslední pivot, pokud je list prázdný) určuje maximální vzdálenost  $r_m$  – nejbližší soused není



Obrázek 3.2: Ukázka KD stromu dimenze dva

dále než tento bod. Vzdálenost  $r_m$  určuje hyperkružnici ve které se nejbližší soused vyskytuje. Pokud hyperkružnice protíná některé ze „sousedních sektorů“ (viz. obr. 3.2), musí být tyto části stromu také prohledány. Proběhne návrat ve struktuře stromu o příslušné úrovni výše, nalezení bodů v sousedních sektorech a spočtení vzdálenosti mezi nimi a bodem odkud bylo iniciováno hledání.

Efektivita prohledávání závisí na struktuře dat ve stromu, na poloze zadaného bodu a také na strategii výběru pivota. Vyjimečně proto mohou nastat situace, kdy je nakonec prohledán téměř celý strom.

*KD strom* může teoreticky pracovat s libovolným počtem dimenzí ukládaných dat. V praxi ale platí určitá omezení kvůli exponenciálně vzrůstajícímu počet uzlů, které je třeba prohledat. Uvádí se, že *KD strom* efektivně funguje pro dimenzi dat maximálně deset.

## Kapitola 4

# Experimenty

Kapitola blíže popisuje vlastnosti laserového dálkoměru a strukturu dat z odometrie. Jsou provedeny experimenty ověřující funkčnost algoritmu, zaměřením kapitoly budou ale převážně nedostatky použitých postupů a jejich analýza.

Vstupní data programu jsou načítána ze souboru obsahujícího měření laserového dálkoměru, výstupy inkrementálních senzorů odometrického systému a příslušné časové údaje.

### 4.1 Laserový dálkoměr

V této práci byl použit laserový dálkoměr *SICK LMS 200*. Senzor je velmi odolný a robustní, určen převážně pro průmysl – používán je například u automatických transportních vozidel nebo pro ostrahu areálů a prostor uvnitř budov. Senzor pracuje na principu měření doby letu laserového paprsku. Použitý laser vyzařuje v infračerveném spektru. Paprsek je rozmítán v jedné rovině pomocí zrcadla rotujícího rychlostí  $4500\text{ot}/\text{min}$ . Možné hodnoty úhlu rozmítání jsou  $180^\circ$  nebo  $100^\circ$ . Úhlové rozlišení senzoru je standardně 1 stupeň, je ale možno přepnout do režimu 2 nebo 4 prokládaných skenů a získat tak rozlišení 0.5 respektive 0.25 stupně. *SICK LMS 200* měří v módu do  $80\text{m}$  nebo do  $8\text{m}$  – s vyšším milimetrovým rozlišením. Komunikace se senzorem probíhá po seriové lince *RS232* nebo *RS485*. Chyba měření není v čase konstantní – její velikost je teplotně závislá. Při experimentech za pokojové teploty v [13] došlo k jejímu ustálení (ustálení tepelných poměrů uvnitř senzoru) po přibližně třech hodinách provozu. U senzoru se projevuje jev *mixed pixel* (viz. 2.1.3). V případě, že se nepodaří vyslaný paprsek detekovat, vrací dálkoměr chybový kód a vzdálenost maxima měřícího rozsahu. Obecně je maximální hodnota vzdálenosti vracena v případě, kdy není v měřitelné vzdálenosti překážka, nebo je odražený paprsek pro detekci příliš slabý.

Senzor je umístěn na robotu ve výšce  $50\text{cm}$ , rozsah měření je nastaven do  $80\text{m}$ , proutnutí poloroviny rozmítaného laserového paprsku se zemí při prudších pohybech robota bylo řešeno vzdálenostním prahováním měření rozdělených na segmenty (segment je odstraněn, pokud všechny jeho body leží mimo hranici  $d_{max} = 15\text{m}$ ). Nastavené úhlové rozlišení je  $0.5^\circ$  s celkovým rozsahem  $180^\circ$ . Algoritmus byl testován na chodbách budovy školy na Karlově náměstí – povrch, po kterém se robot pohyboval, bylo rovné linoleum. Struktura prostředí ve výšce, kde bylo prováděno měření vzdálenosti laserovým skenerem nebyla složitá (v chodbě

nebyl nábytek) a proto zde například k jevu *mixed pixel* docházelo zřídka, v takových případech byl ale potlačen vstupní filtrací bodů.

## 4.2 Odometrie

Data z odometrického systému jsou načtena ve formě časové značky a dvou čísel vyjadřujících součty pulsů z inkrementálního senzoru levého a pravého kola robotu. Měření z odometrického systému je méně než z laserového dálkoměru a mají rozdílné časy, proto jsou načtená data předzpracována – je provedena interpolace odometrických měření pro získání přesnějších hodnot v časech laserových měření. Inkrementální senzor generuje 171019 pulsů na dráze délky 1 m.

## 4.3 Popis jednotlivých experimentů

Většina pohyblivých objektů v testovacích datech jsou lidé. Pohyb lidských nohou (měření, které člověku odpovídají) je poměrně komplikovaný, zejména pokud pro ně není algoritmus uzpůsoben. Podle zjednodušeného modelu v práci [14], která se zabývá sledováním lidí ze statických detektorů lze pohyb nohy aproximovat modelem konstantního kladného a záporného zrychlení. V této práci je sledována je každá noha odděleně a nepředpokládá se žádný model člověka jako soustavy dvou nohou. Problém nastává v místě, kde jsou si obě nohy nejbližší nebo jsou vzhledem k detektoru v zákrytu a jsou označeny jako jediný objekt. Pokud člověk současně rychleji zatáčí, model, který v důsledku splnutí nohou v jeden segment nenalezl cíl bude pravděpodobně v dalších cyklech algoritmu hledat na nesprávném místě. V okamžiku, kdy se nohy opět rozloží na dva segmenty, tento model „svou“ nohu nenalezne a v krátkém čase zanikne. Noha je současně rozpoznána jako nový objekt a je jí přiřazen jiný model. Důvodem problémů je neplatnost předpokladu konstantní rychlosti pohybu nohy.

Bylo provedeno několik experimentů detekce pohyblivých objektů ze statického a pohyblivého robotu:

1. Detekce pohybu krabice (krychle o straně 1m) ze statického robotu
2. Detekce chodce z robotu jedoucího konstantní rychlostí
3. Test lokalizace a detekce chodců z robotu pohybujícího se po obecné trajektorii

Poloha robotu je na obrázcích znázorněna velkým červeným bodem. Zdi a překážky, které jsou považovány za statické jsou znázorněny žlutě, pohyblivé objekty modře. Oranžovou barvu mají objekty, které jsou *kandidáty* na pohyblivý objekt. Každý pohyblivý objekt je nejdříve označen jako *kandidát*, následně se čtyři cykly algoritmu čeká na „potvrzení“ pohybu (podrobněji viz 2.2.2.1). Objekt je poté zařazen mezi pohyblivé/statické a je příslušně barevně znázorněn. Některé důležité objekty jsou zvýrazněny baravným obdélníkem. Pro testování byla využita i data z práce [7] ze statického detektoru.

## Experiment 1

V experimentu je sledována poloha krabice ze statického detektoru. Obr. 4.1 ukazuje počáteční stav, kdy se krabice nachází uprostřed chodby. Na obr. 4.2 je znázorněn pohyb objektu, který byl považován za statický (zvýrazněný je také člověk, který krabici tlačil). Obrázek je složen ze sedmi sousledných skenů při pohybu krabice k levému dolnímu okraji obrázku, na obr. 4.3 jsou pouze detekované pohyblivé objekty a statické body v mapě na původním místě krabice.

Body, které zbyly v místě, kde se původně krabice nalézala vytváří dojem, že je stále přítomna na původním místě. Účinný způsob odstraňování neexistujících statických bodů z mapy nebyl v této práci nalezen. Případy, kde se vyskytují jsou ve většině případů složitější – body bývají spojeny se statickou překážkou.

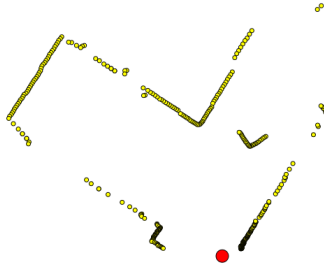
U objektů, které jsou jednou označeny jako statické a začínají se pohybovat s pomalým zrychlením dochází k dalšímu nepříznivému jevu, kdy je neexistující překážka (viz. předchozí odstavec) v mapě „rozmazána“. Případ je dobře patrný na obr. 4.2, kde žluté body v mapě zůstanou. Důvodem „rozmazání“ je princip algoritmu detekce pohybu a poměrně nepříznivý tvar objektu. Pro rozpoznání pohybu je nutné, aby minimálně polovina bodů patřících objektu neměla v mapě v určeném okolí souseda. Zde tato podmínka není pro první kroky splněna a body jsou proto přidány do mapy jako statické. Možným řešením je snížení prahu pro počet pohyblivých bodů (ze současné poloviny), tím se ale zvýší riziko nesprávného označení statických překážek za pohyblivé.

Na obrázcích je možné rozpoznat i člověka, který krabici tlačí. Jedna jeho noha se pohybovala příliš blízko rohu zdi a při rozdělení skenu na segmenty byla chybně určena jako její součást. Body jsou proto přidány do mapy – dobře patrné v rámečku na obr. 4.4. Další případ pohybu objektů blízko zdi je na obr. 4.5 – označené body jsou důsledek situace, kdy se pohyblivý objekt zastavil velmi blízko u statické překážky a následně se znovu dal do pohybu.

## Experiment 2

Robot se při experimentu pohyboval konstantní pomalou rychlostí, v okolním prostředí se pohyboval člověk. Rohy červených rámečků označují plochu, ve které je sledovaný objekt očekáván (50% bodů objektu z nového skenu musí ležet uvnitř, aby došlo ke shodě s jeho modelem). Body jsou umístěny přesně programem, doplnění stran vzniklého obdélníku je provedeno ručně. Prohledávaný prostor je obdélníkový z důvodu jednoduché implementace, rozdíl oproti kruhovému tvaru by nebyl významný. Na následujících třech obrázcích je demonstrováno očekávání dalšího pohybu modelem sledovaného objektu při jeho skrytí ve výklenku. Jedna noha má podle posledních naměřených dat značnou rychlost a je očekáváno, že i nadále zůstane přibližně konstantní. Prohledávaná oblast se proto v dalších krocích rychle posunuje směrem určeným modelem nohy. Na obr. 4.6 obě nohy splynuly v jeden objekt, který je identifikován s modelem druhé nohy. Druhá noha byla v posledních měřeních ve fázi kroku, kdy se nepohybuje a proto i po ztrátě objektu na obr. 4.8 je očekávána přibližně na stejném místě.

Na sérii tří obrázků počínaje obr. 4.9 je znázorněna situace, kdy sledovaný objekt náhle změnil směr. Původní směr pohybu člověka byl na obrázku zprava doleva, v dalším okamžiku obě nohy splynou (viz. obr. 4.10) a jedna se začne pohybovat druhým směrem (obr.



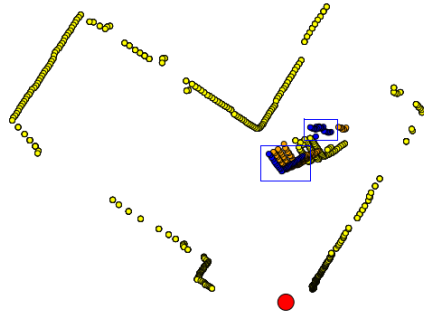
Obrázek 4.1: Počáteční stav

4.11). Noha pohybující se druhým směrem je přesně na pomezí oblasti, kde je očekávána modelem, ale není splněna podmínka 50% obsažených bodů a proto je pro stávající model ztracena. Je vytvořen nový model – oranžová barva značí nově objevený objekt označený jako *kandidát* na pohyblivý objekt. Problém neočekávaných zrychlení je možné účinně řešit zvětšením prohledávaného prostoru, současně ale vznikne problém pokud by ke ztrátě i tak došlo. Model s velkým prohledávacím prostorem může být identifikován s velkými statickými objekty (segmenty zdí). Velikost prohledávaného prostoru byla určena jako velikost obdélníku „opsaného“ bodům objektu zvětšeného o experimentálně určenou konstantu  $0.5m$  do všech čtyř směrů.

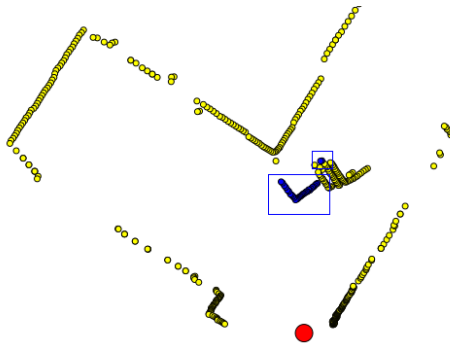
### Experiment 3

Zde byl testován rozdíl v lokalizaci při použití samotné odometrie a při zpřesňování porovnáním skenů algoritmem *ICP*. Z obrázku 4.12 je vidět, že i při relativně přesné odometrii, rovném a i z hlediska tření vhodném povrchu se po několika prudších pohybech na krátké trajektorii robotu objeví zratelná chyba. Chyba je převážně rotačního charakteru, což je pro odometrii obvyklé. Použitý robot se při rychlých rotacích pohybuje trhavě, kdy může dojít k prokluzu kol – to je nejpravděpodobnější důvod poměrně velké chyby vzhledem k ujeté vzdálenosti.

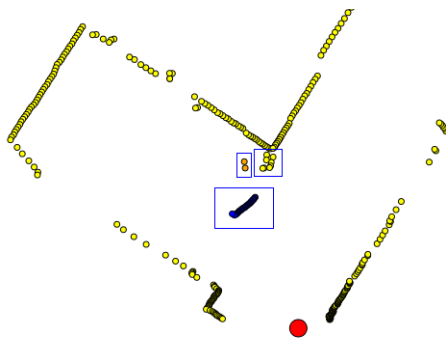
Na obr. 4.13 je zapnuta korekce polohy využívající porovnávání skenů algoritmem *ICP*. Lokalizace je v tomto případě dostatečně přesná a žádná zratelná chyba se neprojevila. V porovnání se zde dosaženým výsledkem vznikla bez použití *ICP* na dráze dlouhé  $3m$  chyba v natočení  $1.17\%$  a chyba v určení souřadnic  $(X, Y)$   $3.61\%$  resp.  $0.85\%$ . Na obrázku 4.14 je stejná situace o několik vteřin dříve, kdy byl detekován pohyblivý objekt, ale nebyl odstraněn před zpracováním *ICP*. „Rozhození“ v určení rotační složky transformace je znečné, což potvrzuje důležitost filtrace pohyblivých objektů.



Obrázek 4.2: Pomalé uvedení statického objektu do pohybu

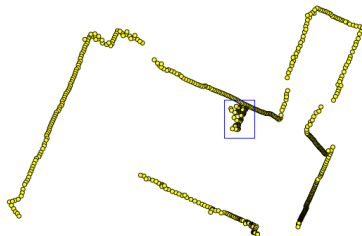


Obrázek 4.3: Detekovaná krabice tlačena člověkem

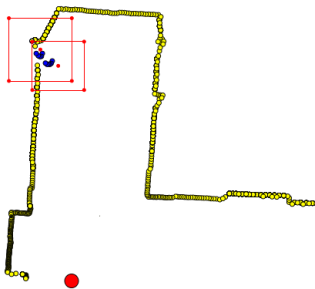


Obrázek 4.4: Chybné přidání bodů do mapy při pohybu člověka blízko zdi

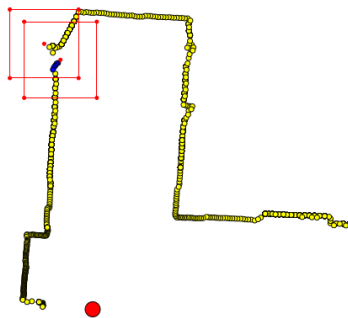




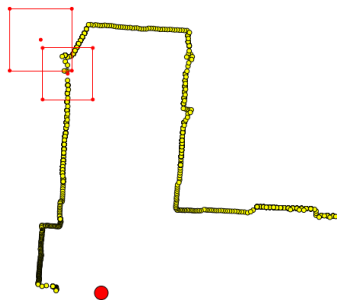
Obrázek 4.5: Důsledek zastavení člověka těsně u zdi



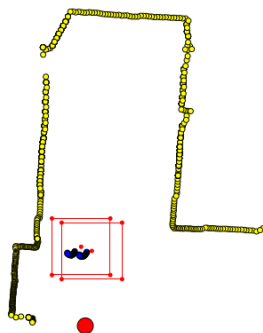
Obrázek 4.6: Člověk směřující do výklenku (1)



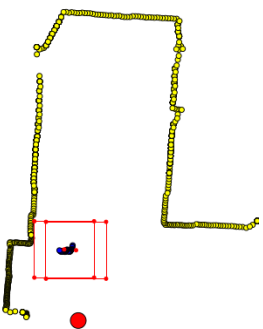
Obrázek 4.7: Člověk směřující do výklenku (2)



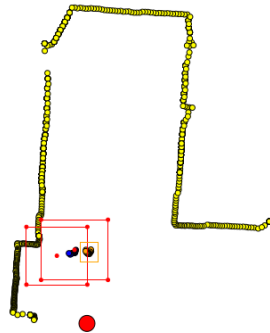
Obrázek 4.8: Člověk směřující do výklenku (3)



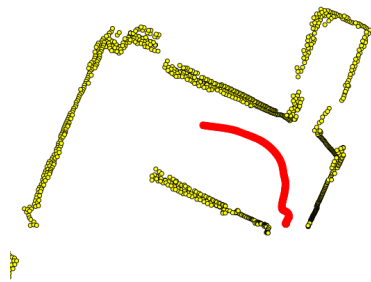
Obrázek 4.9: Náhlá změna směru pohybu (1)



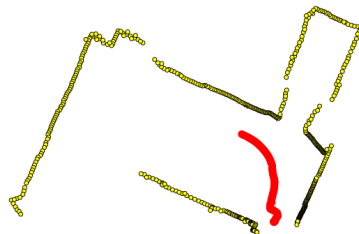
Obrázek 4.10: Náhlá změna směru pohybu (2)



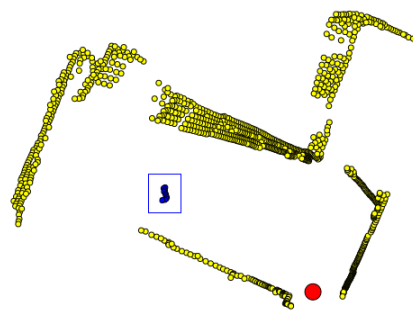
Obrázek 4.11: Náhlá změna směru pohybu (3)



Obrázek 4.12: Lokalizace pouze z odometrie



Obrázek 4.13: Lokalizace z odometrie s upřesněním ICP algoritmem



Obrázek 4.14: chyba ICP při neodstraněných pohyblivých objektech

# Kapitola 5

## Závěr

V bakalářské práci byl implementován algoritmus simultánní lokalizace a mapování s detekcí a sledováním pohyblivých objektů. K lokalizaci byl zvolen přístup *scan matching*, kde je změna polohy určována výpočtem transformace mezi vytvářenou mapou a novým měřením laserového skeneru. Cílem práce je kromě úspěšné lokalizace také detekce pohyblivých objektů, pro zamezení přidání objektům příslušných měření do budované mapy.

Algoritmus úspěšně detekuje pohyblivé objekty, je schopen je sledovat a ve většině situací zabránit jejich zanesení do mapy. Odometrický systém robotu použitého pro testování je relativně přesný a prostředí chodby s rovným linoleem se pro něj blíží ideálním podmínkám. Korekce algoritmem *ICP* zde přesto byla nutná, z důvodů větších odometrických chyb, které byly způsobeny pravděpodobně trhanými pohyby při rychlejším otáčení. Algoritmus pro korekci odometrických chyb byl proto užitečný i pro pohyb robotu v takto vhodném prostředí – zabraňuje chybám v určení polohy a tím i zhoršení kvality celé mapy.

Největší vliv mají nepřesnosti určení polohy na detekci pohyblivých objektů. Body nového skenu ke kterým není v mapě v nastaveném okolí nalezen soused jsou označeny jako pohyblivé. Sken je také rozdělen na segmenty příslušející různým objektům – pokud je více než polovina bodů segmentu označena jako pohyblivá, bude jako pohyblivý označen i segment. Při nepřesném určení polohy a tedy i transformace nového skenu mohou být některé ze segmentů reprezentujících statické objekty chybně označeny jako pohyblivé. Algoritmus detekce pohyblivých objektů požaduje, aby segment byl jako pohyblivý označen alespoň dvakrát a tím riziko nesprávné klasifikace vlivem náhodné chyby minimalizuje. Opatření filtruje nově objevené statické objekty, které také splňují výše uvedenou definici pohybu.

Podobné postupy pro detekci pohyblivých objektů mají jistá omezení – slabší stránkou je zejména nutnost rozdělit nový sken do segmentů odpovídajících samostatným objektům. Detekce pohyblivých objektů pak často selhává při pohybu sledovaného objektu těsně u překážky nebo při shlukování více objektů k sobě. Rozdělení probíhá na základě rozdílu vzdálenosti u sousledných měření ve skenu. Při pohybu sledovaného objektu blízko statické překážky nebo jiného objektu splývají oba objekty v jeden. Pokud se výsledný objekt nalézá v oblasti, kde je jeden z objektů očekáván a má podobnou velikost, je jeden z modelů sledovaného objektu upraven pro sledování nového spojeného objektu a druhý model po několika cyklech bez nalezení „svého“ objektu smazán. Segmenty reprezentující statické překážky jsou zpravidla velkých rozměrů (vzdáleností okrajových bodů i jejich počtem) a po splnutí se segmentem sledovaného objektu nesplní výsledný segment podmínku podobné

velikosti a polohy. Výsledný segment je tedy označen jako statický a všechny body jsou přidány do mapy. V oblasti pohyblivého objektu proto v mapě chybně vznikne překážka. Pokud je sledovaný objekt zakryt jiným objektem nebo statickou překážkou, očekáváme u něj konstantní rychlost pohybu pro určení lokace, kde má být hledán. Předpoklad ale není vždy splněn a dochází tak ke ztrátě sledovaného objektu.

## Možná vylepšení algoritmu

Případné budoucí vylepšení této práce by se mohlo týkat zdokonalení systému segmentace, který je nejslabším místem algoritmu. Schopnost rozlišit i objekty pohybující se v blízkosti statických překážek by pro některé druhy prostředí významně zlepšila kvalitu mapy. Jedno možné řešení je navrženo v kapitole [2.2.1](#).

Jiným vylepšením by mohla být implementace modelu lidské chůze pro efektivnější sledování lidí jako soustavy dvou nohou – model by měl brát v úvahu periodické zrychlování a zpomalování nohou a také jejich častý zákryt.

# Literatura

- [1] J. Almedia, A. Almedia, and R. Araujo. Tracking multiple moving objects for mobile robot navigation. Technical report, University of Coimbra, Portugal.
- [2] J. Cua, H. Zha, H. Zhao, and R. Shibasaki. Evaluation of a laser-based multi-people detection and tracking system. Technical report, Peking University, University of Tokyo, 2006.
- [3] A. Fod, M. Mataric, and G. Sukhatme. Laser tracking and classification of multiple objects. Technical report.
- [4] B. Jung and G. S. Sukhatme. Tracking multiple moving targets using a camera and laser rangefinder. Technical report, University of Southern California, Department of Computer Science, USA, 2001.
- [5] B. Kluge, C. Kohler, and E. Prassler. Fast and robust tracking of multiple moving objects with a laser range finder. Technical report, Research Institute for Applied Knowledge Processing, Ulm, 2001.
- [6] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. Technical report, York University, Canada.
- [7] T. Mrázek. Detekce pohyblivých objektů v datech laserového dálkoměru. Technical report, CVUT, Fakulta elektrotechnická, 2008.
- [8] R. Mázl. *Lokalizace pro autonomní systémy*. PhD thesis, ČVUT Praha, Fakulta elektrotechnická, Katedra Kybernetiky, 2007.
- [9] L. Romero, M. Jesus, and A. Pimentel. Robust local localization of a mobile robot using a 180° 2-d laser range finder. Technical report, Universidad Michocana, Mexico, 2005.
- [10] D. Schulz, W. Burgard, D. Fox, and A. Cremers. Tracking multiple moving objects with a mobile robot. Technical report, University of Bonn, University of Freiburg, University of Washington, 2001.
- [11] C.-C. Wang. *Simultaneous Localization, Mapping And Moving Object Tracking*. PhD thesis, Carnegie Mellon University, USA, 2004.
- [12] C.-C. Wang and C. Thorpe. Simultaneous localization and mapping with detection and tracking of moving objects. Technical report, Carnegie Mellon University, USA, 2003.

- [13] C. Ye and J. Borenstein. Characterization of a 2-d laser scanner for mobile robot obstacle negotiation. Technical report, University of Michigan, USA, 2002.
- [14] H. Zhao and R. Shibasaki. Pedestrian tracking using multiple laser range scanners. Technical report, University of Tokyo, Japan.



## Dodatek A

# Seznam použitých zkratek

**SLAM** Simultaneous Localization And Mapping

**DATMO** Detection And Tracking of Moving Objects

**KF** Kalman Filter

**ICP** Iterative Closest Point

**IMRP** Iterative Matching Range Point

**IDC** Iterative Dual Correspondence

**IR** Infra Red

## Dodatek B

# Obsah přiloženého CD

CD obsahuje kromě elektronické verze tohoto dokumentu také složku s programem a složku s naměřenými daty. Program je dále rozčleněn na část s knihovnou *ANN* pro ukládání dat do struktur *KD stromů*, grafické prostředí pro vizualizaci výsledků a složku se třídami a pomocnými třídami samotného algoritmu.