



## Bakalářská práce

Lukáš Kuhajda

Akademický rok 2018/2019

# Obsah

<b>1</b>	<b>SLAM - Simultaneous Localization and Mapping</b>	<b>3</b>
1.1	Úvod . . . . .	3
1.2	Historie . . . . .	3
1.3	Formulace a struktura SLAM . . . . .	3
1.3.1	Legenda . . . . .	3
1.3.2	Pravděpodobnostní SLAM . . . . .	4
1.3.3	Struktura pravděpodobnostního SLAMu . . . . .	4
1.4	Řešení problému SLAM . . . . .	5
1.4.1	EKF-SLAM . . . . .	5
1.5	Rao-Blackwellizedův filtr . . . . .	6
<b>2</b>	<b>Simultaneous localization and mapping: part II.</b>	<b>7</b>
2.1	Výpočetní složitost . . . . .	7
2.1.1	Rozšířený stav . . . . .	7
2.1.2	Rozdělené aktualizace . . . . .	8
2.1.3	Rozčlenění . . . . .	8
2.1.4	Globální submapy . . . . .	9
2.1.5	Submapy vztažné . . . . .	9
2.2	Sdružování dat . . . . .	9
2.2.1	Validace várky dat . . . . .	9
2.2.2	Popis vzhledu . . . . .	10
2.2.3	Multihypoziční sdružování dat . . . . .	10
2.3	Reprezentace prostředí . . . . .	10
2.3.1	Částečná pozorovatelnost a zpožděné mapování . . . . .	10
2.3.2	3D SLAM . . . . .	10
2.3.3	Trajektorií orientovaný SLAM . . . . .	11
2.3.4	Vložené pomocné informace . . . . .	11
2.3.5	Dynamická prostředí . . . . .	11
<b>3</b>	<b>Improved techniques for grid mapping with Rao-Blackwellized particle filters</b>	<b>11</b>
3.1	Introduction . . . . .	11
3.2	Mapping with RBPFs . . . . .	12
3.3	RBPF with improved proposals and adaptive resampling . . . . .	12
3.4	Implementation issues . . . . .	14
3.5	Complexity . . . . .	14
3.6	Experiments . . . . .	15
<b>4</b>	<b>Real-Time Loop Closure in 2D LIDAR SLAM</b>	
	<b>CARTOGRAPHER</b>	<b>16</b>
4.1	Úvod . . . . .	16
4.2	System overview . . . . .	16
4.3	Local 2D SLAM . . . . .	16
4.4	Closing loops . . . . .	17

<b>5</b>	<b>A Flexible and Scalable SLAM System with Full 3D Motion Estimation</b>	
	<b>Hector SLAM</b>	<b>18</b>
5.1	Úvod . . . . .	18
5.2	System overview . . . . .	18
5.3	2D SLAM . . . . .	19
5.4	3D State Estimation . . . . .	20

# 1 SLAM - Simultaneous Localization and Mapping

## 1.1 Úvod

Simultaneous localization and mapping, v překladu simultánní lokalizace a mapování (SLAM) je problém, při kterém řešíme otázku, zda-li dokáže robot, umístěný do neznámého prostředí a s neznámou lokací, postupně utvářet mapu svého okolí a zároveň určovat místo, ve kterém se nachází. Řešení tohoto problému je nezbytně nutné pro tvorbu zcela autonomně se pohybujících robotů. Jedná se o aktuální téma vzhledem k trendům moderní doby, kdy dochází k pozvolnému přechodu na samořiditelná auta.

## 1.2 Historie

Jako počátek, kdy byl daný problém poprvé zíněn, se bere konference Robotics and Automation Conference konaná v roce 1986. Pravděpodobnostní metody byly tehdy ještě velmi nerozvinuté, jak v robotice, tak i v umělé inteligenci, a došlo tak pouze k debatě na dané téma.

K většímu posunu kupředu se dostalo o pár let později, kdy vyšla práce pojednávající o vztahu mezi orientačními body (landmarky) a zmenšení geometrické nepřesnosti. Důležitým prvkem zde bylo zjištění, že mezi odhady landmarků na mapě musí být velký stupeň korelace, přičemž korelace roste s dalšími pozorováními.

Ve stejném období vznikaly základy vizuální navigace a navigace pracující se sonarem s použitím Kalmanova filtru. Práce byly v základu dosti podobné, ukazovaly, že odhady landmarků získané pohybem robota prostředím, jsou v korelaci s ostatními kvůli chybě v odhadu pozice robota a je třeba mít stav složený z pozice robota a landmarků. Vznikal tak však velký stavový vektor s náročností rostoucí v kvadrátu a tendence korelaci landmarků snižovat.

Později došlo ke sjednocení problémů lokalizace a mapování a závěru, že snaha minimalizovat korelaci mezi landmarky byla chybná, naopak bylo v zájmu korelaci co nejvíce zvýšit. Struktura SLAMu, a celkově první použití tohoto akronymu, byla prezentována v roce 1995 na International Symposium of Robotics Research (ISRR). Poté v roce 199 se na ISRR odehrálo první zasedání pojednávající přímo o SLAM a došlo k představení práce dosahující stupeň konvergence mezi SLAMem na Kalmanově filtru a pravděpodobnostními metodami pro lokalizaci a mapování.

## 1.3 Formulace a struktura SLAM

Jedná se o proces, při kterém robot vytváří mapu prostředí, v němž se pohybuje a na základě mapy určuje svoji pozici v prostoru. Pro určování trajektorie robota a rozložení landmarků není třeba znalosti jeho lokace, neboť odhad těchto parametrů probíhá současně.

### 1.3.1 Legenda

$k$  ... čas

$x_k$  ... stavový vektor popisující lokaci a orientaci robota

$u_k$  ... vektor aplikovaný v čase  $k - 1$  pro dostání robota v čase  $k$  na pozici  $x_k$

$m_i$  ... vektor popisující pozici  $i$ -tého landmarku, která je brána jako neměnná se v čase

$z_{ik}$  ... pozorování lokace  $i$ -tého landmarku z robota v čase  $k$

$x_{0:k}$  ... historie umístění robota  
 $u_{0:k}$  ... historie vstupních dat pro pohyb robota  
 $z_{0:k}$  ... množina pozorování landmarků  
 $m$  ... množina landmarků

### 1.3.2 Pravděpodobnostní SLAM

V této formě je zapotřebí výpočet rozložení pravděpodobnosti

$$P(x_k, m | Z_{0:k}, U_{0:k}, x_0)$$

pro každý časový okamžik  $k$ . Jedná se o sdruženou posteriorní hustotu landmarků, stavu vozidla s ohledem na zaznamenané pozorování, řídicí vstupy a zahrnuje čas  $k$  s počátečním stavem vozidla. Pro výpočet je využit Bayesův teorém, je tedy potřeba, aby přechodový stavový model a model pozorování popisovaly vliv vstupního řízení a pozorování.

Model pozorování popisuje pravděpodobnost zisku pozorování  $z_k$ , pokud známe polohu vozidla a landmarků.

$$P(z_k | x_k, m)$$

Model pohybu vozidla může být popsán jako stavový přechodový model. Předpokládáme přechodový stav jako Markovův proces, při kterém následující stav  $x_k$  je závislý pouze na předchozím stavu  $x_{k-1}$  a aplikovaném řízení  $u_k$  a není tak závislý ani na mapě, ani na pozorování.

$$P(x_k | x_{k-1}, u_k)$$

Máme tak implementaci ve dvoustupňové rekurzivní formě korekce předpovědi.

Aktualizace času:

$$P(x_k, m | Z_{0:k-1}, U_{0:k}, x_0) = \int P(x_k | x_{k-1}, u_k) \times P(x_{k-1}, m | Z_{0:k-1}, U_{0:k-1}, x_0) dx_{k-1}$$

Aktualizace měření:

$$P(x_k, m | Z_{0:k}, U_{0:k}, x_0) = \frac{P(z_k | x_k, m) P(x_k, m | Z_{0:k-1}, U_{0:k}, x_0)}{P(z_k | Z_{0:k-1}, U_{0:k})}$$

### 1.3.3 Struktura pravděpodobnostního SLAMu

Model pozorování udává závislost polohy vozidla a pozorování landmarků, sdružená posteriorní pravděpodobnost nemůže být klasicky rozdělena stylem

$$P(x_k, m | z_k) \neq P(x_k | z_k) P(m | z_k),$$

neboť by to vedlo k chybným odhadům. Dalším zdrojem chyb je špatný odhad pozice robota, landmarky jsou ale silně korelované, takže chybný odhad landmarku vůči mapě nevede k chybné poloze dvou landmarků navzájem.

Velmi důležitým poznatkem bylo zjištění, že korelace mezi landmarky monotónně vzrůstá s počtem jejich pozorování (potvrzeno pouze pro lineární Gaussovský případ), tudíž odhad pozice landmarků bude s narůstajícím počtem pozorování monotónně přesnější. Tento jev nastává díky, v podstatě, skoro nezávislému měření relativních pozic mezi landmarky, které jsou zcela nezávislé na natočení vozidla a úspěšné pozorování z tohoto bodu

může nést další nezávislá měření relativních rozložení landmarků.

Pohybem robota v prostoru získává pozorováním novou pozici známých landmarků vůči sobě a dle této informace aktualizuje svoji odhadovanou pozici a také pozici landmarků. Pokud ji nějaký landmark není pozorován, tak je jeho pozice aktualizována dle změny landmarků pozorovaných a pokud jsou zpozorovány landmarky nové, koreluje se s zostatnými, čímž se vytváří síť. Čím častěji jsou dva landmarky pozorovány při jednom měření, tím je síla korelace větší. Opětovným projížděním mapy tak získáváme přesnější a robustnější mapu.

## 1.4 Řešení problému SLAM

Při řešení je potřeba adekvátně obsáhnout jak složku modelace prostředí, tak i tvorbu pohybového modelu. Máme řadu možností jak tento problém řešit, například princip Monte Carlo, kdy rozdělujeme hustoty pravděpodobnosti odhadu pozice robota. Další možností je Markovova lokalizace, s tímto přístupem jsme se setkali v pravděpodobnostní formě SLAM. Nejčastěji se setkáváme s reprezentací problému ve formě stavového modelu zatíženého šumem, což vede k použití rozšířeného Kalmanova filtru (v originále *extended Kalman filter* → *EKF*). Jinou možností je ještě rozčlenit pohybový model vozidla na vzorky s obecnějším negausovským rozdělením pravděpodobnosti, v tomto případě mluvíme o použití Rao-Blackwellizedova partikulárního filtru.

### 1.4.1 EKF-SLAM

Popis pohybu vozidla:

$$P(x_k|x_{k-1}, u_k) \leftrightarrow x_k = f(x_{k-1}, u_k) + w_k$$

$f(\cdot)$  ... funkce modelující pohyb vozidla

$w_k$  ... chyby měření

$Q_k$  ... kovariance

Model pozorování:

$$P(z_k|x_k, m) \leftrightarrow z_k = h(x_k, m) + v_k$$

$h(\cdot)$  ... geometrické vlastnosti

$v_k$  ... chyby měření

$R_k$  ... kovariance

Tyto dvě definice využijeme v EKF metodě k výpočtu průměru a kovariance sdruženého posteriorního rozložení

průměr:

$$\begin{bmatrix} \hat{x}_{k|k} \\ \hat{m}_k \end{bmatrix} = E \begin{bmatrix} x_k \\ m \end{bmatrix} | Z_{0:k}$$

kovariance:

$$P_{k|k} = \begin{bmatrix} P_{xx} & P_{xm} \\ P_{xm}^T & P_{mm} \end{bmatrix}_{k|k} = E \left[ \begin{pmatrix} x_k - \hat{x}_k \\ m - \hat{m}_k \end{pmatrix} \begin{pmatrix} x_k - \hat{x}_k \\ m - \hat{m}_k \end{pmatrix}^T | Z_{0:k} \right]$$

Mezi hlavní problémy této metody se řadí konvergence, výpočetní složitost, sdružování

dat a nelinerita. Prvním z nich, konvergence, se projevuje postupným přechodem determinantu kovarianční matice mapy a všech podkategorií dvojic landmarků k nule. Jednotlivé odchylky landmarků pak konvergují dle původních nepřesností z odhadu pozice robota a jeho pozorování.

Výpočetní složitost je zde brána jako kvadraticky rostoucí s počtem zaznamenaných landmarků, neboť při každém zaznamenaném pozorování se aktualizují již uložené landmarky, tento problém již ale prošel vývojem a existují metody pracující v reálném čase s tisíci landmarky.

Metoda EKF-SLAM je velmi náchylná na chybné spojení pozorování se známými landmarky. Jedná se zejména o problém s uzavřením smyčky, kdy dochází k opětovnému návratu na místo, ze kterého robot začínal nebo ve kterém se již nacházel.

Nelinerita je posledním z významných problémů, kvůli níž můžeme dojít k větším nepřesnostem ve výsledku, neboť EKF-SLAM využívá lineárních modelů pro vyjádření nelineárního pohybu a modelu pozorování, konvergence a konzistence modelu je tedy jistá pouze v lineárním případě.

## 1.5 Rao-Blackweillizedův filtr

Forma SLAM založená na Rao-Blackellizedově filtru, jinak nazývaná také jako FastSLAM, je na bázi rekurzivního Monte Carlo modelu a dokáže reprezentovat nelineární stavový model. Je výpočetně nemožné aplikovat na stavový prostor s vysokým počtem dimenzí partikulární filtry, je však možné redukovat velikost vzorků.

Sdružený stav může být jako faktor komponentů vozidla a podmíněných komponentů mapy:

$$P(X_{0:k}, m | Z_{0:k}, U_{0:k}, x_0) = P(m | X_{0:k}, Z_{0:k}) P(X_{0:k} | Z_{0:k}, U_{0:k}, x_0)$$

Rozdělení pravděpodobnosti zde není na jednotlivých pozicích  $x_k$ , ale na celou trajektorii  $X_{0:k}$  a tím se stávají jednotlivé landmarky na sobě nezávislými, mapa je tedy reprezentována jako soubor nezávislých gaussianů, což znamená lineární složitost oproti kvadratické u formy EKF. Hlavními ukazateli FastSLAMu je mapa, jež je počítána analyticky a vážené vzorky, jimiž je reprezentována trajektorie pohybu. Rekurzivní odhad je proveden partikulárním filtrem pro stav pozice a EKF pro stav mapy.

Zpracování každého landmarku probíhá zvlášť, pozice se aktualizuje stejným způsobem jako v EKF a landmarky, které nebyly zpozorovány, zůstávají na původní pozici a neaktualizují se. Vzájemnou neprovázaností landmarků však vzniká chyba v odhadu, která s časem roste.

V čase  $k - 1$ , je sdružený stav reprezentovaný jako

$$\{w_{k-1}^{(i)}, X_{0:k-1}^{(i)}, P(m | X_{0:k-1}^{(i)}, Z_{0:k-1})\}_i^V$$

V prvním kroce, pro každou částici vypočítáme návrh distribuce, jež je podmíněná svojí specifickou historií a z ní odebereme vzorek  $x_k$ , který je poté sdružen k historii částice  $X_{0:k}^{(i)}$ . Krokem dva, dle funkce důležitosti, stanovíme váhy vzorků. Třetím krokem je případné převzorkování, které se provádí různě často dle implementace. Krokem posledním je provést EKF update, na každou, již zpozorovanou částici, která je při aktuálním pozorováním zaznamenána.

## 2 Simultaneous localization and mapping: part II.

### 2.1 Výpočetní složitost

Daný problém SLAM má neobvyklou strukturu, jelikož je odhad sdruženého stavu složený z pozice robota a landmarků, procesní model má tak vliv pouze na stav pozice robota a model pozorování na pár robot-landmark, a to vedlo k vytvoření spousty metod využívajících zmíněnou strukturu, pro redukci výpočetní složitosti. Základním rozdělením metod redukujících výpočetní složitost, je rozlišování optimálních, konzervativních a nekonzistentních metod. První typ, optimální metody, jsou založené na redukci daného výpočtu, výsledkem jsou pak odhady a kovariance, stejně tak, jako je tomu v případě plnohodnotné formy SLAM, rozebírané v předchozích kapitolách. U metod konzervativních dochází k odhadům s vyšší neurčitostí nebo kovariancím, většinou ale, i přes větší nepřesnost, jsou implementovány v reálném použití. Poslední možností jsou nekonzistentní metody. Jsou to algoritmy, které mají nižší neurčitost nebo kovarianci, než algoritmy optimální. Pro řešení SLAM se však nepoužívají.

Prvním přístupem pro redukci výpočetní složitosti, je využití struktury SLAM, pro omezení požadovaného výpočtu rovnicí aktualizace pozorování. Výpočet aktualizace času může být omezen metodami využívající rozšířený stav, výpočet stavu pozorování pak metodami oddělujícími rovnice dané aktualizace a obě tyto omezení vedou k redukci výpočtů, typicky jsou to optimální algoritmy. Další možností, jak snížit složitost procesu, je reformulace stavového prostoru do informační podoby, která umožňuje rozdělení výsledné matice s informacemi pro snížení výpočtů, což obvykle bývají algoritmy konzervativní. Obvykle je v díky nim znatelně redukována výpočetní složitost, stále však má dostatečně dobré odhadovací schopnosti. Dalším přístupem je submapping, který rozděluje mapu na regiony, kdy následné aktualizace mohou nastávat pouze v dané oblasti a s určitou periodou pak v i rámci ostatních oblastí.

#### 2.1.1 Rozšířený stav

Sdružený stavový vektor  $x_k$ , v čase  $k$ , se skládá z pozice robota a jím zaznamenaných landmarků, kdy model robota ovlivňuje pouze stav pozice, a to vlivem vstupního řízení, stav mapy se tím tedy nemění.

$$x_k = \begin{bmatrix} f_v(x_{vk-1}, u_k) \\ m \end{bmatrix}$$

Pokud při volbě typu SLAM jako EKF, má výpočet předpovědi kovariance kubicky rostoucí složitost s počtem landmarků

$$P_{k|k-1} = \nabla f_x P_{k-1|k-1} \nabla f_x^T + \nabla f_u U_k \nabla f_u^T,$$

to se však dá předělat na formu s pouze lineární složitostí

$$P_{k|k-1} = \begin{bmatrix} \nabla f_{vx} P_{vv} \nabla f_{vx}^T + \nabla f_{vu} U_k \nabla f_{vu}^T & \nabla f_{vx} P_{vm} \\ P_{vm}^T \nabla f_{vx}^T & P_{mm} \end{bmatrix}$$

Přidání nového landmarku má podobný tvar, kdy je nový landmark inicializován jako funkce pozice robota a pozorování a rozšířený stav pak získáme z malého množství existujících stavů

$$x_k^+ = \begin{bmatrix} x_{vk} \\ m \\ g(x_{vk}, z_k) \end{bmatrix},$$



kde  $g(x_{vk}, z_k) = m_{new} \rightarrow$  přidání nového landmarku

Rozšíření stavu můžeme aplikovat vždy, když je nový stav funkcí podmnožiny již existujících stavů

$$\begin{bmatrix} x_1 \\ x_2 \\ f(x_2, q) \end{bmatrix},$$

$$\begin{bmatrix} P_{11} & P_{13} & P_{13} \nabla f_{x_2}^T \\ P_{12}^T & P_{23} & P_{23} \nabla f_{x_2}^T \\ \nabla f_{x_2} P_{11}^T & \nabla f_{x_2} P_{32} & \nabla f_{x_2} P_{32} \nabla f_{x_2}^T + \nabla f_q Q \nabla f_q^T \end{bmatrix}$$

### 2.1.2 Rozdělené aktualizace

Jedná se o metody vytvářející optimální odhady. Při implementaci základní podoby aktualizace pozorování, se při každém novém měření aktualizuje jak stav vozidla, tak i mapy, což vede ke kvadratickému nárůstu složitosti s množstvím landmarků. Při této metodě si však rozdělíme mapu na menší oblasti, které se aktualizují při průjezdu robota danou částí, zatímco aktualizace celé mapy probíhá s výrazně nižší frekvencí.

Rozlišujeme dva způsoby možné implementace. První z nich pracuje na zmenšené oblasti, ale stále si drží globální referenční souřadnice, jedná se například o algoritmus CEKF (compressed EKF). Druhou možností je tvorba menších map s vlastním souřadnicovým rámcem neopouštějícím danou submapu, jedná se o algoritmy CLSF (constrained local submap filter  $\rightarrow$  omezený lokální submapový filtr). Pokračovat budeme v rozboru druhé možnosti, neboť je jednodušší a při provádění operací s velkou frekvencí opakování je méně ovlivněna linearizačními chybami, je stabilnější a zabráňuje příliš velkému nárůstu globální kovariance.

Logaritmus submapy se skládá z dvou nezávislých odhadů, které si stále udržuje. Jde o vektory  $x_G$  a  $x_R$ , kdy  $x_G$  je mapa složená z globálně referencovaných landmarků a globálně referencované pozice dané submapy a  $x_R$  je lokální submapa s lokálně referencovanou pozicí robota lokálně referencovanými landmarky. Při získání pozorování se aktualizují pouze landmarky náležící aktuální submapě, ve které se robot nachází. Celkový globální odhad pak získáváme periodicky, zaevidováním submapy do mapy celé a použitím aktualizace omezení na společné vlastnosti obou map.

### 2.1.3 Rozčlenění

Do této doby jsme brali stavový odhad  $\hat{x}_k$  a matici kovariance  $P_k$ , jako produkty klasicky implementovaného EKF-SLAM, ty pak spolu popisovaly první dva centrální momenty hustoty Gaussovy pravděpodobnosti skutečného stavu  $x_k$ . Jinou možností je vyjádřit stavový odhad a matici kovariance v informační formě pomocí matice informací  $Y_k = P_k^{-1}$  a vektoru informací  $\hat{y}_k = Y_k \hat{x}_k$ . Tato změna je výhodná pro mapy s větším měřítkem, kdy spousta nediagonálních prvků bude velmi blízkých nule, což vede k nastavení těchto prvků na hodnotu nula, přičemž však může vznikat malá ztráta optimality při vzniku map.

Důležitým poznatkem je fakt, že rozšíření stavu je rozčleňovací operace vedoucí k přesnému rozčlenění informační formy SLAMu a má tak ekvivalentní informační formu, kde pro zjednodušení považujeme šum za nulový přírůstek  $f(x_2, q) = f(x_2) + q$ . Podmnožinu stavů  $x_1$  je pak předpokládána jako obsahující většinu stavů mapy a po rozčlenění, dosahuje pouze konstantní složitosti v čase. Můžeme tedy získávat přesné řešení rozšiřováním

stavu novým odhadem pozice robota v každém kroce a zachovat všechny předchozí pozice. Díky tomu jsou nenulové nediagonální prvky pouze ty, které jsou spojené napřímo s měřenými daty.

Dále sem musíme zahrnout marginalizaci, jež je nezbytná pro odstranění předchozích stavů pozice. Máme možnost marginalizovat všechny předchozí stavy, což vede na zhuštěnou matici informací, to je stav, kterého dosáhnout nechceme. Správnou volbou ukotvení pozice můžeme marginalizovat velkou část pozic, aniž bychom vyvolaly nadměrnou hustotu matice informací. Pro reálné využití v praxi se zde stále vyskytuje problém s obnovováním průměru a kovariance v každém kroce, což může být velmi výpočetně náročné, dá se však poměrně efektivně získat konjugacími gradientními metodami.

#### 2.1.4 Globální submapy

U globálních metod rozlišujeme základní dva typy, globálně a lokálně referencované, přičemž oba mají společné, že submapa stanovuje místní souřadnicový rámec a landmarky z jejího okolí jsou odhadovány s ohledem na daný místní rámec.

Lokální metoda získává odhady pomocí optimálního algoritmu SLAM, používajícího pouze lokální landmarky. Tato metoda, i přes výpočetní efektivitu, je při tvorbě struktury submap neoptimální. Globální metoda je mnohem zajímavější. Dokáže z kvadraticky rostoucí složitosti, udělat lineární, či dokonce v čase konstantní složitost. Je to možné díky údržbě a konzervativním odhadům celkové mapy. Metoda stojí na odhadování globální pozice submapy v rámci společného základního rámu, nevede však ke zmírnění problémů s linearizací, způsobenou velkými nepřesnostmi v pozicích.

#### 2.1.5 Submapy vztažné

Základním rozdílem od metody globálních submap, je absence společného základního rámu. V této metodě se submapy zaznamenávají dle sousedství s ostatními a celkovou mapu pak můžeme získat souhrnem vektoru cestou po vytvořené síti submap. Submapy jsou, díky vyhýbání se globálním spojením, velmi zajímavé z hlediska výpočetní složitosti a problémům týkajících se nelinearity. Velkým kladem je například tvorba lokálně optimální mapy s výpočetní složitostí nezávislou na celkové velikosti mapy a dále, díky úpravám pouze na lokální úrovni, je velmi stabilní.

### 2.2 Sdružování dat

Jedná se velmi důležitý problém, neboť i když během procesu tvorby mapy, dojde k jedné chybné asociaci dat, může to vést k destabilizaci odhadu mapy, často dokonce k pádu celého algoritmu

#### 2.2.1 Validace várky dat

Zprvu se k problému přistupovalo způsobem, kdy každé jednotlivé zachycení landmarku, se porovnávalo se všemi odhady nacházejícími se v blízkém okolí, tento individuální přístup je neproveditelný, pokud je pozice robota velmi nejistá, což znamená selhání ve všech, obzvláště ve strukturovaných a málo osídlených prostředích. Později se již podařilo násobné asociace zvažovat simultánně.

### 2.2.2 Popis vzhledu

Jedním ze způsobů snímání okolí je vidění, kdy zaznamenáváme tvar, barvu, strukturu, a tím dokážeme rozlišovat různé balíčky data, což v SLAMu využíváme pro přepověď dané asociace. Nejčastěji tak pro problém s uzavřením smyčky, ke kterému se dostalo z původního využití, a to rozpoznání míst v topologických mapách a indexování databáze obrázků. Pokrok této metody přišel s prací Newmana a spol., kdy se začala počítat metrika podobnosti přes sekvenci obrazů, místo původního jednoho a pro použití v běžném režimu je použita metoda valstních čísel.

### 2.2.3 Multihypoziční sdružování dat

Jedná se o metodu nezbytně potřebnou pro robustní sběr cílů v zaplněném prostředí, kdy tvorbou oddělených odhadů trasy jízdy pro každou asociaci hypotézu, řeší problém ne-jednaznačnou hypotéz. Tato funkce je však silně limitována dostupným výpočetním výkonem. Dale je metoda využívána, zvláště ve velkých prostředích, v implementaci robustního SLAMu, kdy je při uzavírání smyčky vytvořena hypotéza pro smyčku uzavřenou, tak i pro stále neuzavřenou, a tím se bere v potaz, že je prostředí pouze podobné.

## 2.3 Reprezentace prostředí

Původně se modeloval svět pomocí SLAM pouze jako soubor landmarků, majících svůj určitý tvar, později však, zejména ve venkovním, podvodním a podzemním použití, se ukázala tato metoda jako nevyhovující.

### 2.3.1 Částečná pozorovatelnost a zpožděné mapování

Základními dvěma typy pro pozorování je vidění, pomocí kamery, nebo nějaký typ dálkového senzoru. Kamera, pokud je na robotovi samostatně, zaznamenává informace neobsahující přesnou vzdálenost objektu. Tento problém měření pomocí senzoru nemá, neboť měření vzdálenosti bývá velmi přesné, za to se pro tento typ vyskytuje problém se šířkou vysílaného paprsku a postranními žlábkami. S tímto senzorem nedokážeme jedním pozorováním ani vytvořit přesně položený landmark, jedním měřením totiž získáváme negaussovské rozložení pro pozici landmarku a potřebujeme tak větší počet jeho zpozorování pro vytvoření odhadu.

Generalizované rozložení umožňují nezpožděné sledování landmarků, zpožděním inicializace však můžeme získat hned gaussovský odhad pozice landmarku, je při tom ale potřeba zaznamenávat pozici robota v každém okamžiku měření, což provedeme rozšířením stavu o vektor pozic. Nejedná se pouze o částečnou pozorovatelnost, ale shromažďováním informací a zpožděným rozhodováním, zvyšuje robustnost procesu.

### 2.3.2 3D SLAM

Jedná se v podstatě pouze o rozšíření 2D SLAMu, které má však výrazně větší výpočetní složitost a komplikovanější modelování, způsobené složitějším snímáním. Rozlišujeme základní tři možnosti. První je klasický 2D SLAM s přidanou schopností vytvářet třetí dimenzi, což se užitečné, pokud se robot pohybuje po rovině. Druhý typ vytváří 3D obraz extrakcí diskrétních landmarků a sdruženého odhadu pozice vozidla a mapy. Využití je vhodné v robotech s jedním senzorem, který umožňuje pohyb se šesti stupni volnosti. Poslední možnost se od těch předchozích dost odlišuje. Sdružený stav se totiž skládá z předchozích

pozic robota, na každé pozici se udělá 3D sken prostředí a odhad pozice se vyrovná korelaí skenů.

### 2.3.3 Trajektorií orientovaný SLAM

Základní formulací problému, je odhadovaný stav jako pozice robota a zaznamenané landmarky, jinou, novější, možností je odhadovat místo toho trajektorii vozidla. Mapa tak není součástí stavu, každá pozice robota má ale přidružený sken svého okolí, které se pak srovnají a vytvoří globální mapu. Z toho je možno vidět jeden z problémů, což je neustálý růst stavového prostoru, který se nijak nepromazává. Využití se nachází například při tvorbě topologickým map.

### 2.3.4 Vložené pomocné informace

Jak je vidět u trajetoriemi orientovaného SLAMu, můžeme ke stavu připojit data, v tomto případě to dále rozšířit například o teplotu, charakteristiku povrchu a mnoho dalších věcí, které pak napomáhají při tvorbě mapy. Tvorba takovéto struktury je ale poměrně komplikovaná. Pohybem robota prostředím, ukládají se pomocná data do datové struktury tak, že každá buňka v této struktuře je přiřazena daným landmarkům v mapě, při aktualizaci jejich pozice se tak s nimi přemísťují i jim přidružené informace.

### 2.3.5 Dynamická prostředí

Ve světě se setkáváme převážně z dynamicky se vyvíjejícím prostředím, kdy nám do pozorování můžou zasahovat lidé, zvířata, nebo třeba nábytek jako židle, nebo zaparkovaná auta. Musí se tedy nějak určit, co s takovými objekty bude a jak je za pohyblivé určit. Máme možnost tyto objekty do mapy vůbec nepřidávat, nebo je mít označené a pohyblivé, nesmí se ale stát, že zaznamenáme pohyblivý objekt a uložíme ho jako statický.

Klasická implementace SLAM umí odstranit landmark, i poměrně velké množství landmarků, bez většího vlivu na hodnotu konvergence. To je také využíváno pro úpravu mapy, kdy se odstraňují přebytečné landmarky, které se již na svém místě nenacházejí.

## 3 Improved techniques for grid mapping with Rao-Blackwellized particle filters

### 3.1 Introduction

- tvorba mapy - základní funkce mobilního robota (problém mapování = SLAM)
- pro odhad pozice dobrá mapa, pro získání mapy dobrý odhad pozice
- → RBPF - effective solution
- redukce dat, převzorkování může eliminovat správné částice → problém vyčerpání částice
- 2 přístupy zlepšení RBPF
- - návrhové rozložení zahrnuje přesnost senzoru → přesné vykreslení částic
- - vzorkovací technika udžující rozumné množství částic, umožňuje algoritmu určovat přesnou mapu a snižuje riziko vyčerpání částic
- návrhové rozložení - vyhodnocováním pravěpodobné pozice (kombinace informací z laseru a odometrie), poslední pozorování použito pro tvorbu nových částic → odhad stavu na základě více informací než pouze na odometrii
- - přesnější mapa - zvážení vlivu pozorování na pozici, poté až změny v mapě → redukce

chyby odhadu

- - adaptivní vzorkování - udělání vzorku pouze když ho potřebujeme, udržení rozumné různosti částic → snížení rizika vyčerpání částice

### 3.2 Mapping with RBPFs

- základní úkol - odhad sdružené posteriorní pravděpodobnosti
- - cíl - mapa, trajektorie
- - odhad na základě - pozorování, odometrie
- odhad trajektorie, poté až mapy
- mapy tvořeny pozorováním, trajektorie odpovídajícími částicemi (každá částice reprezentuje část trajektorie)
- částicové filtry - SIR (sampling importance resampling) filter, pro mapování postupně zpracovává data ze senzoru, pak odometrii, updatuje sadu vzorků reprezentujících posterior o mapě a trajektorii
- - vzorkování - nové částice jsou získávány z předchozí generace odběrem vzorků z návrhového rozložení (často pravděpodobnostní odometrie) - - vážená důležitost - individuální váhu důležitosti má každá částice podle zásady důležitostního vzorkování. Váhu zavádíme kvůli tomu, že cílové rozložení není rovno tomu navrhovanému.
- - převzorkování - částice jsou přepisovány úměrně jejich vážené důležitosti, nezbytný krok pro potřebnou konečnost částic k aproximaci kontinuálního rozložení, dále převzorkování umožňuje použít částicový filtr v situaci neshodujícího se cílového rozložení od toho navrhovaného (po převzorkování všechny částice mají stejnou váhu)
- - odhad mapy - pro každou částici je vypočítána mapa na základě trajektorie vzorku a historie pozorování
- pro implementaci potřeba vyhodnocovat váhy trajektorií při každém pozorování, po delší době neefektivní
- omezením navrhovaného rozložení získáme rekursivní formulaci pro výpočet vážené důležitosti

### 3.3 RBPF with improved proposals and adaptive resampling

- zlepšení návrhového rozložení - pro získání nové generace částic je potřeba vykreslení vzorků z návrhového rozložení (čím lepší návrh tím lepší výsledek, kdyby přímo jako cílové rozložení → stejné vážené důležitosti částic, není potřeba převzorkování)
- - typický částicový filtr - návrhové rozložení = odometrický pohybový model (+ jednoduchý výpočet pro většinu robotů, vážená důležitost je počítána podle modelu pozorování)
- toto rozložení není optimální, zvláště pokud je senzor výrazně přesnější než odhad pozice (laser)
- - vyhlazování pravděpodobné funkce - zabránění, částicím v blízkosti významné oblasti, přílišného poklesu vážených důležitostí (- vyřazuje užitečnou informaci ze senzoru → nepřesnější mapa)
- - překonání problému - zvažovat poslední pozorování při generování nových vzorků, integrováním posledního pozorování do návrhu se dá zaměřit na vzorkování ve významné oblasti pravděpodobnosti pozorování

- efektivní výpočet zlepšeného návrhu - nepředvídatelná funkce pravděpodobnosti pozorování → nedostupná uzavřená aproximace informačního návrhu
- - teorie - získání aproximace upraveným částicovým filtrem - návrh pro každou částici je vypočten odhadem vzorků z optimálního návrhu
- - SLAM - navzorkování potenciální pozice z pohybového modelu → zvážení vzorků podle pravděpodobnosti pozorování (zisk aproximace optimálního návrhu - pokud již je, stále je potřeba vysoký počet vzorků pozice z pohybového modelu)
- - většinou cílové rozložení omezený počet maxim (většinou jedno) → vzorkování pozic kolem maxima (snížení složitosti) - zvažujeme pravděpodobnost pozorování i pohybový model v době použití senzoru - lokální aproximace posteriorní pravděpodobnosti kolem maxima pravděpodobnostní funkce
- - - na těchto datech počítáme Gaussovskou aproximaci - efektivní vykreslování nových vzorků
- - - porovnání scanů pro zisk významé oblasti z funkce pravděpodobnosti pozorování → navzorkujeme tuto oblast, vyhodnotíme vzorky podle cílového rozložení
- - - pro každou částici  $i$  jsou Gaussiánské parametry  $(\mu_t^i, \Sigma_t^i)$  určené individuálně pro daný počet vzorků na intervalu kdy je aktivní senzor + bereme v potaz odometrii při výpočtu střední hodnoty  $\mu^i$  a výchylky  $\Sigma^i$  → zisk uzavřené aproximace optimálního návrhu → efektivní zisk nové generace částic - pro výpočet váhové funkce používáme stejný normalizační faktor jako u Gaussiánské aproximace
- zlepšený návrh - získání parametrů Gaussiánského návrhu zvlášť pro každou částici (uvažuje odometrii a pozorování, mezitím efektivně vzorkuje), snížená neurčitost výsledných hustot
- - porovnávač pozorování určuje režim významné oblasti funkce pravděpodobnosti pozorování - vzorkování z důležitých oblastí - většina porovnávačů maximalizuje pravděpodobnost pozorování, dává mapu a nástřel pozice robota
- - vícerežimová pravděpodobnostní funkce (uzavření smyčky) - porovnávač vrátí pro každou částici maximum nejbližší původnímu odhadu - může způsobit ztrátudalších maxim v pravděpodobnostní funkci, protože je hlášen jen 1 režim
- - přílišná důvěřivost filtru (teoreticky) - přeplněný prostor + odometrie silně zatížena šumem
- - - řešení - sledovat více režimů porovnávače měření + opakovat vzorkování v každém uzlu
- adaptivní převzorkování
- - převzorkování - důležitý aspekt výkonu částicového filtru, nahrazování vzorků s nízkou váhou důležitosti těmi s vysokou
- - - nezbytné - konečný počet částic je použit k aproximaci cílového rozložení (- může odstanit dobré vzorky z filtru → ochuzení částic) → důležité převzorkování ve správný čas, mít dobré rozhodovací kritérium
- - Liu - efektivní velikost vzorkování pro odhad, jak aktuální sada částic reprezentuje cílový posterior

$$N_{eff} = \frac{1}{\sum_{i=1}^N (w^{(i)})^2}$$

- - -  $w^{(i)}$  ... normalizovaná váha částice  $i$
- - -  $N_{eff}$  - vzorky z cílového rozložení → stejné váhové důležitosti (zhoršující se aproximace cílového rozložení → větší odchylka váhových důležitostí)

- - - Doucet - převzorkování, když  $N_{eff} \leq N/2$  ( $N$  ... počet částic) - výrazná redukce možnosti nahrazení dobrých částic (redukován počet převzorkování, převzorkování když je potřeba)
- - algoritmus
- - - nástřel pozice reprezentovaný danou částicí, získáno z předchozí pozice této částice a odometrického měření od poslední aktualizace filtru
- - - na základě mapy provedeno porovnání pozorování z místa úvodního nástřelu pozice, vyhledává se pouze v okolí tohoto bodu, v případě selhání - pozice a váhy počítány podle pohybového modelu (další 2 kroky přeskočeny)
- - - vybrána sada vzorků kolem dané pozice, vypočítání průměru a kovarianční matice návrhu bodovým hodnocením cílového rozložení v pozici vzorku, také počítán váhový faktor
- - - nová pozice částice je zakreslena podle z Gaussovské aproximace podle zlepšeného návrhového rozložení
- - - aktualizace váhových důležitostí
- - - mapa částice je aktualizována podle její zakreslené pozice a pozorování

### 3.4 Implementation issues

- porovnávač "vasco" - hledání pozice porovnáním scanu vůči mapě se zanesením i úvodního odhadu (sestupné gradientní hledání v pravděpodobnostní funkci aktuálního pozorování vůči grid mapě - hledání maxima funkce)
- - může být použit každý porovnávač umožňující najít nejlepší zarovnání mezi mapou a aktuálním pozorováním s ohledem na úvodní odhad
- Bayesovo pravidlo - hledání pozice
- model koncového bodu - výpočet pravděpodobnosti pozorování
- - paprsky nezávislé - pravděpodobnost paprsku počítána na základě vzdálenosti konce paprsku od nejbližší překážky od tohoto bodu - pro rychlý výpočet využití konvolované lokální grid mapy
- návrh - výpočet 2 komponent - jejich vyhodnocování pro každý vzorek
- - 1. - výpočet podle modelu koncového bodu
- - 2. - Gaussovská aproximace odometrie pohybového modelu - aproximace získána přes rozšíření Taylorovy řady
- 0.5 m mezi 2 použitými filtry - aproximace funguje dobře, žádný rozdíl mezi modelem na EKF a přesnějším modelem rychlosti pohybu na bázi vzorků

### 3.5 Complexity

Výpočet návrhového rozložení ...  $O(N)$

Aktualizace grid mapy ...  $O(N)$

Výpočet vah ...  $O(N)$

Testovací potřeby převzorkování ...  $O(N)$

Převzorkování ...  $O(N \cdot M)$

- naučení se grid map s RBPF

- - sada vzorků - posteriorní pravděpodobnost o mapách a pozicích

- - počet vzorků - centrální množství

- - návrhové rozložení - vzorkování u nejpravděpodobnější pozice danou porovnávačem

pozorování

- - vzorkování pro každou částici, žádná závislost mezi nimi při výpočtu návrhu
- - poslední pozorování - výpočet Gausiánských parametrů
- - - kryje část mapy → komplexita závisí jen na počtu vzorků
- - - - to samé platí pro aktualizaci jednotlivých map spojených s každou částicí
- - kopírování informace náležící částici při převzorkování - nejhůře  $N-1$  částic nahrazeno jednou
- - každá částice si nese svou grid mapu → duplikování částice - kopírování její mapy → převzorkování má složitost  $O(N \cdot M)$  -  $M$  ... velikost grid mapy
- - - používání adaptivního převzorkování

### 3.6 Experiments

- ActivMedia Pioneer2 AT, Pioneer 2 DX-8, iRobot B21r
- venkovní i vnitřní prostory
- až rozlišení do 1 cm bez problémů
- 250mx250m - max 80 částic
- výsledky mapování
- - Intel Research Lab - vnitřní prostor, 28mx28m, 15 částic, Pioneer II, SICK senzor, velmi přesné
- - Freiburg Campus - venku, 250mx250m, 30 částic, keře, stromy, pohyblivé předměty (auta, lidi) i tak přesná mapa
- - MIT Killian Court - vnořené smyčky (potenciální vyčerpání částic), dlouhé chodby, důležitá vlastnost adaptivního převzorkování, 60 částic (občas dvojité stěny), 80 částic (vysoká kvalita)
- vyčíslitelné výsledky - porovnání počtu částic s přístupem dirka Hahnela, počet částic které potřebuje RBPF pro vytvoření topologicky přesné mapy alespoň v 60% použití - Intel (8 vs. 40), Freiburg (20 vs. 400), MIT (60 vs. 400) - mnohem méně částic + lepší mapa (lepší vzorkování)
- účinky lepších návrhů a adaptivního vzorkování
- - zlepšené návrhové rozložení - generování vzorků s vysokou pravděpodobností,
- - adaptivní vzorkování -  $N_{eff}$
- - návrhy bez celé vstupní historie snižují v čase  $N_{eff}$  (čím horší návrh, tím rychleji) - obnovení na max hodnotu po převzorkování
- - v této implementaci
- - - zlepšení poklesu  $N_{eff}$  - klesá pomaleji i v neznámém prostředí (návrhové rozložení není tak vypíchnuté)
- - - částice při průchodu známým prostředím zůstávají ve vlastní mapě (díky zlepšenému návrhovému rozložení a kvůli váhám)
- - - uzavírání smyčky - některé částice jsou správně seřazeny (velká váha) a některé chybně (nízká váha) → zvětšení výchylek vah → snižování  $N_{eff}$  - většinou vynucení převzorkování
- - - výrazné snížení pravděpodobnosti vyčerpání částice - uchovávání vzorků v sadě částic
- vliv odometrie na návrh - většinou je návrh jen z laseru dobrý pro odhad pohybu částic, někdy potřeba odometrie pro návrhové rozložení - např. slabé vlastnosti dat z laseru (pustá místa, koridory)
- situace, kdy selhá porovnávač pozorování - nemusí na základě dat dojít k odhadu pozice
- - vzorkujeme rovnou z odometrie pro získ nových částic



- - vevnitř se tento problém nestává, venku v otevřených prostorech, když laser nikde ve svém rozsahu nic nezaznamenává
- analýza běhu - PC 2.8 GHz procesor, scan při pohybu o 0.5m nebo otočení o  $25^\circ$
- - výpočet návrhového rozložení, vah, aktualizace mapy (1910 ms), testování potřeby převzorkování (41 ms), převzorkování (244 ms)

## 4 Real-Time Loop Closure in 2D LIDAR SLAM CARTOGRAPHER

### 4.1 Úvod

- velké mapy, optimalizované výsledky v reálném čase
- porovnávání scan-to-scan - rychlé hromadění chyb
- porovnávání scan-to-map - omezení chyb, pokud správný odhad pozice a kvalitní data z LIDAR → efektivnost, robustnost
- porovnávání přesnosti pixelů - redukce hromadění lokálních chyb oproti scan-to-scan, náročnější, použití při uzavírání smyčky
- částicové filtry - tvoří stav systému pro každou částici, pro SLAM s grid-mapami s velkým prostorem rychle roste náročnost - řešení - použití systému děající grid-mapu po čase a ne v každé částici
- SLAM na grafu - kolekce nodů reprezentující pozice a funkce, hrany - omezení získaná pozorováním

### 4.2 System overview

- real-time 2D mřížková mapa s rozlišením na 5 cm
- submapa do odhadované pozice
- porovnávání vůči poslední submapě → hromadění globálních chyb
- neobsahuje částicový filtr - snížení HW požadavků
- pořízení submapy, dále již nepřepisování - všechny vzniklé připraveny k porovnávání na uzavření smyčky - blízká odhadovaná pozice + dostatečná shoda → přidání omezení uzavření smyčky do optimalizačního problému (odhad pozice) - optimalizace po pár vteřinách - uzavření smyčky je vidět hned

### 4.3 Local 2D SLAM

- lokální a globální přístup - obě optimalizace pozice (x, y, natočení)
- IMU - inertial measurement unit - odhad směru gravitace, při nerovné ploše
- submapa - kousek světa - lokální chyba porovnává je odstaněna v globálním přístupu
- Sken - iterativní, zarovnávání skenu a souřadnicových snímků submapy
- -  $H$  ... informace o bodech skenu
- -  $M$  ... pravděpodobnostní mřížka
- -  $\xi$  ... pozice snímání skenu
- -  $T_\xi$  ... pozice skenu vůči submapě - transformace - body skenu do submapy
- Submapy - pár skenu do submapy - pravděpodobnostní mřížky mapované z diskretních

bodů mřížky v určitém rozlišení (5 cm)

- - všechny body mřížky - odpovídající pixel ze všech pixelů nejbližše danému bodu
- - přidání skenu do pravděpodobnostní mřížky - výpočet množiny zasažených bodů mřížky a minutých bodů
- - zásah - vložíme nejbližší bod mřížky do množiny zásahů
- - minutí - vložíme bod mřížky sdružený se všemi pixely, které jsou protínány jedním paprskem mezi počátkem skenování a každým snímacím bodem - nepřidáváme body již přidané do množiny zásahů
- - dosud nepozorovaný bod mřížky má přiřazenu pravděpodobnost minutí či zásahu, podle toho jestli je v jedné z těchto množin
- - již zpozorovaným bodům aktualizujeme pravděpodobnosti minutí a zásahu
- Ceres scan matching - před vložením do submapy - optimalizace pozice skenu vůči submapě - maximalizace pravděpodobnosti

$$\operatorname{argmin}_{\xi} \sum_{k=1}^K (1 - M(T_{\xi} h_k))$$

#### 4.4 Closing loops

- hromadění lokálních chyb, jen pár snímků za sebou má vždy minimální chybu vůči sobě
- Spare Pose Adjustment - optimalizace pozice snímků
- - relativní pozice skenů jsou ukládány, všechny další páry ze skenů a submap jsou brány v potaz při uzavírání smyčky, pokud se submapa nezmění
- běh v pozadí, shoda - relativní pozice uložena mezi optimalizační problémy
- optimalizační problém - nelineární nejmenší čtverce, jednoduché přidávání zbytků - zohlednění dalších dat
- - 1x za sekundu Ceres - optimalizace pozice skenu v souladu vůči daným omezením - relativní pozice, kovarianční matice

$$\operatorname{argmin}_{\Xi^m, \Xi^s} 0.5 \sum_{ij} \rho(E^2(\xi_i^m, \xi_j^s, \Sigma_{ij}, \xi_{ij}))$$

- -  $\rho$  ... ztrátová funkce
- - Huberova ztráta - snížení vlivu odlehlých hodnot - přidávají nesprávná omezení do optimalizačního problému - lokálně symetrická prostředí
- branch-and-bound scan matching - větev a mez
- - přesnost pixelů

$$\xi^* = \operatorname{argmax}_{(\xi \in W)} \sum_{k=1}^K M_{nearest}(T_{\xi} h_k)$$

- -  $W$  ... vyhledávací okno
- -  $M_{nearest}$  ... rozšíření  $M$  na všechny  $R^2$  zaokrouhlením argumentů do nejbližšího bodu mřížky - rozšířená hodnota bodu mřížky ukazuje na odpovídající pixel
- - vylepšení účinnosti - úhlový krok takový, aby body skenu v maximálním rozsahu nebyly větší než šířka žádného pixelu - pomalé
- - použití branch-and-bound - podmnožiny možností jako uzly ve stromu, kořenový uzel jsou všechny možné řešení ( $W$ )
- - uzly - potomci dohromady tvoří stejný soubor možností jako rodičovský uzel
- - listy - singlety - jediné proveditelné řešení
- - stejné řešení jako předchozí přístup pokud je  $score(c)$  vnitřních uzlů horní mezí skóre jeho prvků - neexistuje lepší řešení než to doposud známé

- výběr uzlů - prohledávání do hloubky - DFS
- - efektivnost závisí na tom jaký je strom - dobrá horní mez a dobré aktuální řešení
- - prahová hodnota skóre - pod ní nemáme zájem o dané řešení - nepřidáváme špatné shody jako omezení pro uzavírání smyčky
- - - často nepřekonané - snížení výnamu výběru uzlů nebo nalezení počátečního heuristického nastavení
- - pořadí průchodu stromu - výpočet horní hranice skóre pro každého potomka, navštívení nejlibnějšího uzlu - největší mezní počet
- pravidlo větvení - každý uzel popsán pomocí tuple integerů ( $c = (c_x, c_y, c_\Theta, c_h) \in \mathbb{Z}^4$ )
- -  $c_h$  ... výška uzlu -  $c_h = 0 \rightarrow$  uzel je list
- výpočet horních mezí - výpočet na vnitřních uzlech - zájem o výpočetní úsilí a kvalitu spojení

$$score(c) = \sum_{k=1}^K \max_{(j \in \bar{W})} M_{nearest}(T_{\xi_j} h_k)$$

- - pro výpočet použití předvypočítaného  $M_{precomp}^{ch}$  - můžeme vypočítat skóre s lineární náročností s body skenu
- - - udržení nízké náročnosti - čekání než pravděpodobnostní mřížka není již aktualizována, poté výpočet sbírky předvypočítaných mřížek, znovu a začínáme porovnávat

## 5 A Flexible and Scalable SLAM System with Full 3D Motion Estimation

### Hector SLAM

#### 5.1 Úvod

- nízké výpočetní nároky - malé autonomní systémy
- ne pro uzavírání velkých smyček
- pro rychlý pohyb terénem
- 2D SLAM (soft real time), 3D navigace (hard real time), IMU (inertial measurement unit)
- nerovný povrch - narušil od gmappingu
- frontend SLAM - Hector - odhad pohybu robota v reálném čase
- backend SLAM - optimalizace pozičního grafu vzhledem k omezením mezi pozicemi

#### 5.2 System overview

- pro 6DOF - z 3DOF odhadem naklonění a rotace - navigační filtr spojí měření z inerciální jednotky a dalších senzorů (3D řešení), informace o poloze (2D SLAM) - volně spojeno, synchronizace v čase
- $x = (\Omega^T p^T v^T)^T$  ... reorientace 3D stavu
- $\Omega = (\phi, \theta, \psi)^T$  ... otáčení, stoupání, natočení
- $p, v$  ... pozice, rychlost platformy v navigačním rámci
- $u = (\omega^T a^T)^T$  ... vstupní vektor pro inerciální měření
- $\omega = (\omega_x, \omega_y, \omega_z)^T$  ... úhlová rychlost
- $a = (a_x, a_y, a_z)^T$  ... zrychlení

pohyb tuhého tělesa

$$\dot{\Omega} = E_{\Omega} \cdot \omega$$

$$\dot{p} = v$$

$$\dot{v} = R_{\Omega} \cdot a + g$$

- $R_{\Omega}$  ... matice směrových cosinů
- $E_{\Omega}$  ... mapování natočení těla na deriváty Eulerova úhlu
- $g$  ... vektor gravitace
- integrovaná rychlost a pozice - signifikantní šum  $\rightarrow$  další informace o senzoru - porovnávání snímků (interiéry), snímač magnetického pole (směr), senzor barometrického tlaku (výška), měření rychlosti kol

### 5.3 2D SLAM

- mřížková mapa obsazenosti - reprezentace prostředí, osvědčená lokalizace robota
- transformace skenu na lokální stabilizovaný souřadnicový rámec odhadovanou polohou
- odhad orientace + sdružené hodnoty  $\rightarrow$  bodový oblak bodů - možnost předzpracování - odstranění odlehlých bodů
- tady filtrace na základě koncového bodu  $z$  - při shodě skenu použity pouze koncové bodyprahové hodnoty skenovací roviny
- přístup k mapě - mřížková mapa obsazenosti - omezení přesnosti, neumožnění přímého výpočtu interpolovaných hodnot a derivátů - interpolační schéma - submřížková přesnost buňky bilineárním filtrem
- - aproximace souřadnice  $P_m$  čtyřmi nejbližšími integerovými souřadnicemi ( $P_{00} - P_{11}$ )
- - vzorkovací body (buňky mřížky) od sebe na vzdálenost 1
- porovnávání snímků - zarovnávání snímků vůči sobě nebo celé mapě
- - laserové skenery - malý šum, vysoká rychlost dělán snímků  $\rightarrow$  možnost velmi přesného měření
- - - mnohem větší přesnost než odometrie, pokud vůbec je - v tomto případě nemáme vůbec
- - zarovnání koncových bodů s již známou mapou - není zapotřebí hledání spojení mezi koncovými body - porovnání se všemi předchozími skeny
- - hledání transformace s nejlepším sladěním skenu s mapou

$$\xi^* = \operatorname{argmin}_{\xi} \sum_{i=1}^n [1 - M(S_i(\xi))]^2$$

- -  $M(P_m)$  ... hodnota obsazenosti
- -  $S_i(\xi)$  ... souřadnice koncového bodu  $s_i = (s_{i,x}, s_{i,y})^T$
- -  $\xi$  ... souřadnice robota
- -  $M(S_i(\xi))$  ... hodnota mapy v souřadnicích  $S_i(\xi)$  - odhad  $\Delta\xi$  - optimalizace chyby měření podle

$$\sum_{i=1}^n [1 - M(S_i(\xi + \Delta\xi))]^2 \rightarrow 0$$

- reprezentace map s více rozlišeními - riziko dosahu pouze lokálního minima - tímto se riziko zmírňuje - více mřížkových map vždy s polovičním rozlišením oproti předchozí
- uložených více map, součastné aktualizace odhady pozice generovaných zarovnávacím procesem
- - konzistence map napříč stupnicemi, vyhnutí se převzorkování

- - zarovnávání skenů na nejvyšší mapě, odhad pozice poté pro všechny ostatní - okamžitá dostupnost hrubých zrnitých map - dobré pro odhad trasy

## 5.4 3D State Estimation

- odhad úplného 3D stavového vektoru
- navigační filtr - 100 Hz - asynchronní aktualizace s pozicí z porovnávače a dalších informací ze senzorů jakmile přijdou
- navigační filtr - EKF - odhad 6D pozice - rozšíření ještě o náklon gyroskopů a akcelerometry
- - nelineární filtr, inerciální měření se berou jako známé vstupy systému
- - aktualizace rychlosti a pozice integrací zrychlení
- - aktualizace pseudo-nulové rychlosti - zabránění nezávislému růstu odhadu stavu pokud nejsou k dispozici měření - při dosažení odchylky určité prahové hodnoty - zajištění stability
- - pro odhad potřeba barometr, snímač rozsahu
- integrace SLAM - 2D SLAM + 3D EKF - komunikace v obou směrech - nesynchronizovány, EKF větší frekvence obnovování
- - odhadovaná pozice EKF počáteční odhad pro optimalizaci porovnávání - nebo integrace odhadované rychlosti a úhlové rychlosti
- - opačně - kovarianční křižovatka (CI - covariance intersection) - spojení pozice ze SLAM s celkovým odhadovaným stavem