Bakalářská práce

Lukáš Kuhajda

Akademický rok 2018/2019

Obsah

1 Úvod

Ve své bakalářské práci se budu věnovat systémům, které pomocí měření z LIDARu (Light Detection And Ranging) utváří mapu prostředí, v němž se pohybují. V první části práce jsem se zaměřil obecně na problém simultánní lokalizace a mapování (SLAM). Mobilní robot je umístěn do neznámého prostředí a jeho úkolem je určovat svoji pozici a utvářet mapu. Zabývám se zde vnitřními principy a různými přístupy k řešení problému. Daná tématika momentálně vstupuje do podvědomí i širší veřejnosti, neboť pomalu dochází k přechodu na autonomní vozidla, která fungují na podobných principech. Tyto automobily však využívají i mnoho dalších senzorů, jako jsou například kamery. Já se zaměřuji na systémy využívající čistě jen 2D LIDAR, tedy o sezor měřící vzdálenosti pouze v jedné výškové úrovni.

V další části své práce jsem se podrobněji věnoval třem nejrozšířenějším 2D SLAM systémům, přesněji se jedná o GMapping, HectorSLAM a Google Cartographer. Popsal jsem zde jejich základní rysy, kterými se liší od ostatních a jimiž je dostatečně popsána jejich funkčnost.

měření, výsledky, porovnání, změna v gmappingu, proč jsem si vybral ...

2 Simultánní lokalizace a mapování

2.1 Historie

Jako počátek řešení problému se považuje konference Robotics and Automation Conference konaná v roce 1986. Pravděpodobnostní metody byly tehdy ještě velmi nerozvinuté, jak v robotice, tak i v umělé inteligenci. Došlo tedy 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 je velká korelace, které je rostoucí 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. Je tak třeba mít stav složený z pozice robota a landmarků, tím však vznikal velký stavový vektor s náročností rostoucí v kvadrátu. V daném období byla tendence korelaci landmarků snižovat.

Později došlo k 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 se v roce 1999 na ISRR odehrálo první zasedání pojednávající přímo o SLAM a došlo k představení práce dosahující dostatečné konvergence mezi SLAMem na bázi Kalmanova filtru a pravděpodobnostními metodami pro lokalizaci a mapování.

2.2 Formulace a struktura

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.

2.2.1 Pravděpodobnostní SLAM

V této formě je cílem získat odhad hustoty pravěpodobnosti

$$P(\mathbf{x}_k, \mathbf{m} | \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0) \tag{1}$$

pro každý časový okamžik k, kde $\mathbf{x}_k$ je stavový vektor popisující pozici a orientaci robota, $\mathbf{m}$ je množina landmarů, tedy mapa, $\mathbf{Z}_{0:k}$ jsou všechna pozorování landmarků, $\mathbf{U}_{0:k}$ je historie vstupů a $\mathbf{x}_0$ je počáteční stav. Jedná se tedy o srduženou posteriorní hustotu mapy, stavu vozidla s ohledem na zaznamenané pozorování, řídící vstupy a zahrnuje čas k s počátečním stavem robota. Pro výpočet je využit Bayesův teorém, je tedy potřeba, aby pohybový model a model pozorování popisovaly vliv vstupního řízení a pozorování.

Model pozorování popisuje pravděpodobnost zisku pozorování $\mathbf{z}_k$, pokud známe polohu vozidla a landmarků.

$$P(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m}) \tag{2}$$

Model pohybu vozidla může být popsán jako stavový přechodový model.
Předpokládáme přechodový stav jako Markovův proces, při kterém následující stav $\mathbf{x}_k$ je

závislý pouze na předchozím stavu $\mathbf{x}_{k-1}$ a aplikovaném řízení $\mathbf{u}_k$ a není tak závislý ani na mapě, ani na pozorování.

$$P(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_k) \tag{3}$$

Máme tak implementaci ve formě dvoustupňového rekurzivního algoritmu.

Aktualizace času (predikce):

$$P(\mathbf{x}_k, \mathbf{m}|\mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k}, \mathbf{x}_0) = \int P(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_k) \times P(\mathbf{x}_{k-1}, \mathbf{m}|\mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k-1}, \mathbf{x}_0) d\mathbf{x}_{k-1} \tag{4}$$

Aktualizace měření (korekce):

$$P(\mathbf{x}_k, \mathbf{m}|\mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0) = \frac{P(\mathbf{z}_k|\mathbf{x}_k, \mathbf{m}) P(\mathbf{x}_k, \mathbf{m}|\mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k}, \mathbf{x}_0)}{P(\mathbf{z}_k|\mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k})} \tag{5}$$

2.2.2 Struktura

Model pozorování udává závislost polohy vozidla a pozice landmarků, sdružená posteriorní pravděpodobnost nemůže být rozdělena na

$$P(\mathbf{x}_k, \mathbf{m}|\mathbf{z}_k) \neq P(\mathbf{x}_k|\mathbf{z}_k) P(\mathbf{m}|\mathbf{z}_k), \tag{6}$$

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). Odhad pozice landmarku je tedy s narůstajícím počtem pozorování monotónně přesnější. Tento jev nastává díky, v podstatě, skoro nezávislému měření relativních pozic mezi landmarky. Ty jsou zcela nezávislé na natočení vozidla a úspěšné pozorovnání 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 jejich pozici a též svoji odhadovanou polohu. Pokud již nějaký landmark není pozorován, tak je jeho pozice aktualizována dle změny landmarků pozorovaných. Při pozorování nových landmarků dochází ke korelaci s již známými, čímž se vytváří síť. Čím častěji jsou dva landmarky pozorovány při jednom měření, tím je síla korelace větší. Opětovným projížděním prostředím tak získáváme přesnější a robustnější mapu.

2.3 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, jedná se o pravděpodobnostní formu 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* $\rightarrow$ *EKF*). Jinou možností je ještě rozčlenit pohybový model vozidla na vzorky s obecnějším negausovským rozdělením pravděpodobnosti, v tomto případě mluvíme o použití Rao-Blackwellizedova částicového filtru.

2.3.1 EKF-SLAM

Pohybu robota je zde popsán rovnicí

$$P(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_k) \leftrightarrow \mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k, \tag{7}$$

kde $\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k)$ je funkce modelující pohyb robota a $\mathbf{w}_k$ jsou chyby měření.

Model pozorování je vyjádřen vztahem

$$P(\mathbf{z}_k|\mathbf{x}_k, \mathbf{m}) \leftrightarrow \mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{m}) + \mathbf{v}_k \tag{8}$$

kde funkce $\mathbf{g}(.)$ popisuje geometrické vlastnosti pozorování a $\mathbf{v}_k$ jsou chyby měření.

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

průměr:

$$\begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{m}}_k \end{bmatrix} = E \begin{bmatrix} \mathbf{x}_k|\mathbf{Z}_{0:k} \\ \mathbf{m} \end{bmatrix} \tag{9}$$

kovariance:

$$\mathbf{P}_{k|k} = \begin{bmatrix} \mathbf{P}_{xx}\mathbf{P}_{xm} \\ \mathbf{P}_{xm}^T\mathbf{P}_{mm} \end{bmatrix}_{k|k} = E\left[\begin{pmatrix} \mathbf{x}_k - \hat{\mathbf{x}}_k \\ \mathbf{m} - \hat{\mathbf{m}}_k \end{pmatrix} \begin{pmatrix} \mathbf{x}_k - \hat{\mathbf{x}}_k \\ \mathbf{m} - \hat{\mathbf{m}}_k \end{pmatrix}^T |\mathbf{T}_{0:k} \right] \tag{10}$$

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 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 ní může 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.

2.4 Rao-Blackweillizedův částicový filtr (RBPF)

Forma SLAM založená na Rao-Blackellizedově filtru, jinak nazývaná také jako FastSLAM, je na bázi rekurzivního Monte Carlo modelu a dokáže reprezentovat nelineání 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 jednotivých 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 gaussiánů, 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 provézt EKF update, na každou, již zpozorovanou částici, která je při aktuálním pozorováním zaznamenána.

2.5 Výpočetní složitost

Neobvyklá struktura problému, sdružený stav složený z pozice robota a landmarků, je využita v řadě metod pro redukci výpočetní složitosti, zde má model procesu vliv pouze na stav pozice robota a model pozorování na dvojici robot-landmark. 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 kovariancí, většinou ale, i přes větší nepřesnost, jsou implementovány v reálném použití. Poslední možností jsou nekonzistentní metody a jedná se o algorithmy, které mají nižší neurčitost nebo kovarianci, než algoritmy optimální, ale pro řešení SLAM se v praxi nepoužívají.

Prvním přístupem pro redukci výpočetní složitosti, je omezení požadovaného výpočtu rovnicí aktualizace pozorování. Výpočet časové aktualizace 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, je reformulace stavového prostoru do informační podoby, která umožnuje rozdělení výsledné matice s informacemi pro snížení výpočtů, což bývají algoritmy konzervativní. Obvykle je díky nim znatelně redukována výpočetní složitost a stále je zachována dostatečně dobrá odhadovací schopnost. Dalším přístupem je submapping, který rozděluje mapu na regiony, kdy následné aktualizace se konají pouze v dané oblasti a s určitou periodou poté i v rámci celé mapy.

2.5.1 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, výpočet předpovědi kovariance má kubicky rostoucí složitost s počtem landmarků

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

to se však dá předělat na formu s pouze lineární složitostí

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

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

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

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

2.5.2 Oddělné 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. V této metodě si však rozdělíme mapu na menší oblasti, které se aktualizují pouze při jejím průjezdu, 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, což jsou algorithmy 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 zabraňuje 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 a 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 vlatnosti obou map.

2.5.3 Rozčlenění stavového prostoru

V této metodě vyjadřujeme stavový odhad ($\hat{x}_k$) a matici kovariance (P_k) v informační formě pomocí matice informací $Y_k = P_k^{-1}$ a vektoru informací $\hat{y}_k = Y_k \hat{x}_k$. Je výhodné pro mapy s větším měřítkem, kdy spousta nediagonálních prvků bude velmi blízkých nule, což vede k možnosti nastavení těchto prvků na hodnotu nula. Může tím však vznikat malá ztráta optimality při vzniku map.

Rozšíření stavu je rozčleňovací operace vedoucí k přesnému rozčlenění informační formy a má tak ekvivalentní informační formu. Předpokládáme, že podmnožina stavů x_i obsahuje většinu stavů mapy a po rozčlenění dosahuje pouze konstantní složitosti v čase. Můžeme tedy získávat přesné řešení díky rozšiřování stavu novým odhadem pozice robota v každém kroce a zachovat všechny předchozí pozice. Nenulové nediagonální prvky jsou tak 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, a tedy stav, kterého dosáhnout nechceme. Správnou volbou ukotvení pozice můžeme marginalizovat velkou část pozic, aniž bychom vyvolaly nadměrnou hustotu matice informací.

2.5.4 Globální submapy

Rozlišujeme dva základní typy submap a to globálně a lokálně referencované, přičemž oba mají společné, že submapa stanovuje místní souřadnicový rámec a landmarky z jejího okolí jsou odhadovány s ohledem na tento rámec.

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

2.5.5 Submapy vztažné

Základním rozdílem od metody globálních submap, je absence společného základního rámce. V této metodě se submapy zaznamenávají dle sousedství s ostatními a celou mapu pak můžeme získat souhrnem vektoru cesty. 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.

2.6 Asociace dat

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

pádu celého algoritmu

2.6.1 Validace várky dat

Z prvu se k problému přistupovalo způsobem, kdy se každé jednotlivé zachycení landmarku porovnávalo se všemi odhady nacházejícími se v blízkém okolí. Tento individuální přístup je neproveditelný, pokud je nejistá pozice robota, cožje tedy neproveditelné obzvlášť v málo zaplněných prostředích.

2.6.2 Popis vzhledu

Jedním ze způsobů snímání okolí je vidění, kdy zaznamenáváme tvar, barvu, strukturu, a tím dokážeme rozlišovat různé balíčky dat, což využíváme pro přepověd dané asociace, nejčastěji pro problém s uzavřením smyčky. Pokrok této metody přišel, když se začala počítat metrika podobnosti přes sekvenci obrazů, místo původního jednoho.

2.6.3 Multihypoziční asociace dat

Jedná se o metodu nepostradatelnou pro robustní sběr dat v přeplněním prostředí, kdy vytváříme oddělené odhady trasy jízdy pro každou asociační hypotézu. Tato funkce je však silně limitována dotupným výpočetním výkonem. Dále je metoda vyžívána při implementaci robustního SLAMu ve velkých prostředích, kdy je při uzavírání smyčky vytvořena hypotéza pro smyčku uzavřenou i pro stále neuzavřenou, a tím se bere v potaz, že je prostředí pouze podobné.

2.7 Reprezentace prostředí

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

2.7.1 Částečná pozorovatelnost a zpožděné mapování

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

Obecná 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 shromažďování informací a zpoždění rozhodování zvyšuje robustnost procesu.

2.7.2 3D SLAM

Jedná se v podstatě pouze o rozšíření 2D SLAMu, které má však výrazně větší výpočetní složitost a komplikovanější modelování. Rozlišujeme základní tři možnosti. První je kla-

sický 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 umožnuje 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 korelací skenů.

2.7.3 SLAM orientovaný pomocí trajektorie

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 hned patrný problém, neustálý růst stavového prostoru, který se nijak nepromazává. Využití metpdy se nachází například při tvorbě topologických map.

2.7.4 Vložené pomocné informace

Ke stavu můžeme připojit rozšiřující data, v tomto případě například teplotu, charakteristiku povrchu a mnoho dalších věcí, které pak napomáhají při tvorbě mapy. Tvorba takovéto struktury je ale poměrně komplikovaná. Pohybem robota prostředím se ukládají pomocná data do datové struktury tak, že každá buňka v této struktuře je přiřazena danému landmarku v mapě, při aktualizaci jaho pozice se tak přemisťují i přidružené informace.

2.7.5 Dynamická prostředí

Ve světě se setkáváme převážně s 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 za 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 GMAPPING

3.1 Úvod

Jako velice podstatný úkol mobilního robota, bereme schopnost tvorby mapy. Tu můžeme kvalitně vytvářet, pokud máme dobrý odhad pozice a pozici zase správně získávat, pokud máme dostatečně kvalitní mapu. Efektivním řešením je tedy RBPF, který můžeme ještě zlepšit. Jednou možností, je zahrnout přesnost měření do návrhového rozložení, tím získáme přesné vykreslení částic. Druhou možností, je volba vzorkovací techniky udržující rozumný počet částic. Udržujeme tak přesnou mapu a snižujeme riziko vyčerpání částice, což je problém vznikající při převzorkování.

Návrhové rozložení získáváme vyhodnocováním pravděpodobnosti pozice robota z kombinace informací ze senzoru a odometrie. Poslední pozorování využíváme pro tvorbu nových částic, odhad stavu tedy provádíme na základě více informací, než jen z odometrie.

3.2 Mapování pomocí RBPF

Základním úkolem je odhad posteriorní pravděpodobnosti a získat tím mapu a trajektorii pohybu, kdy odhad provádíme díky pozorování a informaci z odometrie. Nejprve dochází k odhadu mapy, kterou utváříme z pozorování, poté až trajektorii. Tu získáváme z částic, které měly v rozhodovací době největší pravděpodobnost a každá částice tedy reprezentuje část trajektorie.

Pro výběr správné částice používáme částicový filtr, v tomto případě SIR (Sampling Importance Resampling), který postupně při mapování zpracovává data ze senzoru, poté odometrii a aktualizuje sadu vzorků reprezentující posteriorní pravděpodobnost o mapě a trajektorii.

Celý proces začíná ziskem nových částic odběrem vzorků z předchozí generace návrhového rozložení. Dále nastavujeme částicím váženou důležitost, abychom neměli cílové rozložení rovno tomu navrhovanému. Poté dochází k převzorkování, kdy jsou částice přepisovány úměrně jejich váženým důležitostem a nakonec odhad mapy, kdy je pro každou částici, na základě trajektorie vzorku a historie pozorování, vypočítána mapa.

3.3 RBPF s vylepšenými návrhy a adaptivním převzorkováním

Pro zisk nové generace částic je třeba vykreslení vzorků z návrhového rozložení, kde platí úměra, čím lepší návrh, tím lepší výsledek. Kdybychom měli návrh rovný cílovému rozložení, měli bychom částice se stejnou váženou důležitostí a nebylo by třeba převzorkování.

Typicky je návrhové rozložení odpovídající odometrickému pohybovému modelu, který však není optimální a to zejména pokud je senzor výrazně přesnější než odhad pozice. Využívá se také vyhlazování pravděpodobnostní funkce, což zabraňuje částicím v okolí významné oblasti, přílišnému poklesu vážených důležitostí. Následkem je ale zkreslení mapy. To se však dá vyřešit zahrnutím posledního pozorování do generování nových vzorků. Díky tomu se můžeme aměřit na vzorkování ve významné oblasti pro pravděpodobnost pozorování.

Díky zlepšenému návrhu můžeme získávat pro každou částici zvlášť její parametry Gaussiánského návrhu a snižuje se také neurčitost výsledných hustot pravěpodobnosti. Porovnávač pozorování určuje režim výnamné oblasti pravděpodobnostní funkce pozorování.

Také má většinou funkci maximalizace pravděpodobnosti pozorování, tvorby mapy a počáteční odhad pozice robota. Pokud je pravděpodobnostní funkce vícerežimová, například při uzavírání smyčky, porovnávač vrací pro každou částici nejbližší maximum, což může způsobit vynechání některých maxim v pavděpodobnostní funkci.

Převzorkování je velice důležitým aspektem určujícím výkon částicového filtru, dochází k nahrazování vzorků s nízkou váhou, těmi s váhou vysokou. Je to nezbytný proces, neboť je potřeba konečného počtu částic pro aproximaci cílového rozložení. Může však odtranit dobré vzorky a ochudit tak částice, je proto důležité mít pro převzorkování vhodné rozhodovací kritérium a provádět ho ve správný čas.

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

$w^{(i)}$ ... normalizovaná váha částice i
N_{eff} - vzorky z cílového rozložení $\rightarrow$ stejné váhové důležitosti (zhoršující se aproximace cílového rozložení $\rightarrow$ větší odchylka vážených důležitostí)
Základní nastavení převzorkování je následující:

$$N_{eff} < N/2$$

N ... počet částic
Dochází tak k výrazné redukci možnosti nahrazení dobrých částic a počtu převzorkování, které se vykonává pouze pokud je potřeba.

Algoritmus probíhá následovně. Zisk odhadu pozice, který je reprezentovaný danou částicí. Dostaneme ho z předchozí pozice částice a odometrického měření od poslední aktualizace. Na základě mapy je provedeno porovnání pozorování z místa úvodního odhadu pozice, kdy se vyhledává pouze v okolí tohoto bodu. V případě selhání se pozice a váhy počítají dle pohybového modelu a následující dva kroky jsou přeskočeny. Prvním z nich vybrání sady vzorků v okolí dané pozice, vypočítání průměrů a kovarianční matice návrhu bodovým hodnocením cílového rozložení v pozici vzorku. Druhým, potenciálně přeskočeným krokem, je zakreslení nové pozice částice z Gaussovské aproximace podle zlepšeného návrhového rozložení. Dále, a to již vždy, dojde k aktualizaci vážených důležitostí a podle zakreslené pozice a pozorování je aktualizována i mapa částice.

3.4 Složitost

Výpočet návrhového rozložení ... $O(N)$
Aktualizace grid mapy ... $O(N)$
Výpočet vah ... $O(N)$
Testováí potřeby převzorkování ... $O(N)$
Převzorkování ... $O(N * M)$
M ... velikost grid mapy

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

4.1 Úvod

Systém Cartographer je vyvíjen společností Google. Je vhodný pro tvorbu rozsáhlých map a zisk optimalizovaných výsledků v reálném čase. Při porovnávání snímků typem scan-to-scan dochází rychlému hromadění globálních chyb, v případě porovnávání stylem scan-to-map dochází, při správném odhadu pozice a kvalitnímu snímku z LIDARu, k velké redukci chyb, což vede k větší efektivnosti a robustnosti algoritmu. Dále se využívá také porovnávání přesnosti pixelů. Jde o metodu redukující hromadění lokálních chyb a je vhodná při řešení problému uzavírání smyčky.

4.2 Přehled

Cartographer vytváří v reálném čase 2D mřížkovou mapu s rozlišením na 5 cm. Submapy se vkládají na odhadované místo a daná submapa se porovnává vůči poslední zaznamenané, dochází tedy k hromadění globální chyby z odhadu pozice. Systém neobsahuje žádný částicový filtr. Je to z důvodu snížení hardwarových nároků na běh algoritmu.

Submap se již po pořízení dále nijak nepřepisuje a je zařazena mezi ostatní k porovnávání na uzavření smyčky. Pokud dojde k blízkému odhadu pozice a zároveň dostatečné shodě snímků, přidá se omezení uzavření smyčky do optimalizačního problému, tím je odhad pozice. Odhad se aktualizuje po několika vteřinách a uzavření smyčka je tedy okamžitě viditelné.

4.3 Lokální 2D SLAM

Rozlišujeme dva možné přístupy, lokální a globální, v obou případech se jedná o optimalizaci pozice, přičemž pozice se zkládá z hodnoty na ose x, z hodnoty na ose y a z natočení robota. V systému je také obsažená jednotka IMU (inertial measurment unit), která slouží k odhadu směru gravitace, což má využití při pohybu na nerovné ploše.

Skeny prostředí se iterativně zarovnávají se snímky již obsaženými v submapě. Submapa je tedy v postatě zachycení kousku světa, který je složený z pár skenů. Lokální chyba, vznikající při její tvorbě, je poté odstraněna v globálním přístupu. Pro každý bod mřížky zadefinujeme odpovídající pixel skládající se ze všech pixelů nejblíže danému bodu.

Při přidání skenu do pravděpodobnostní mřížky, počítáme množinu zasažených bodů mřížky a bodů minutých. Při zásahu vložíme nejbližší bod mřížky do množiny zásahů. Při minutí vložíme bod mřížky sdružený se všemi pixely, které jsou protínány jedním paprskem mezi počátkem skenování a každým snímacím bodem. Nepřidáváme sem body již přidané do množiny zásahů. Doposud nepozorované body mřížky mají přiřazenou pravděpodobnost minutí či zásahu, podle toho, jestli se v jedné z těchto množin vyskytují. Již pozorovaným bodům pak aktualizujeme pravděpodobnosti minutí a zásahu.

Před vložením skenu do submapy se ještě využívá Ceres scan matching, který optimalizuje pozici skenu vůči submapě. Jedná se o maximalizaci pravděpodobnosti výskytu v dané oblasti.

$$argmin_{\xi} \Sigma_{k=1}^{K} (1 - M(T_{\xi} h_k))$$

H ... informace o bodech skenu
M ... pravděpodobnostní mřížka
ξ ... pozice snímání skenu
T_ξ ... pozice skenu vůči submapě - transormace (body skenu do submapy)

4.4 Uzavírání smyčky

Daný systém pracuje v oblasti submap s porovnáváním scan-to-scan, hromadí se v něm tedy lokální chyby, ale pár snímků za sebou má vůči sobě chybu minimální. Relativní pozice skenů se ukládají a v případě, že se submapa nezmění, všechny další páry ze skenů a submapy se předkládají k porovnávání pro uzavření smyčky. To vše běží na pozadí a pokud je nalezena shoda, dojde k uložení relativní pozice mezi optimalizační problémy.

Optimalizační problém je problém nelineárních nejmenších čtverců. Díky tomu můžeme jednoduše přidávat zbytky pro zohledňování dalších dat. Jednou za pár sekund je, pro optimalizaci pozice skenu vůči daným omezením, spuštěn Ceres scan matcher. Omezeními jsou myšlena relativní pozice ξ_{ij} a kovarianční matice Σ_{ij}.

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

ρ ... ztrátová funkce (např. Hubertova ztráta - snížení vlivu odlehlých hodnot, ty přiávají nesprávná omezení do optimalizačního problému)

Při uzavírání smyčky se vyžívá také branch-and-bound scan matching, což se dá přeložit jako porovnávání větví a mezí. Zde se zaměřujeme na přesnou shodu pixelů. Podmnožiny možností jsou zde popsány jako uzly stromu kdy kořenový uzel obsahuje všechna možná (W). Potomci uzlu dohromady utváří stejný soubor možností, jako samotný rodičovský uzel. Listy jsou brány jako singlety, odpovídají jedinému proveditelnému řešení. Tento přístup nám dává stejné řešení jako ten předchozí, dokud je hodnota $score(c)$ vnitřních uzlů horní mezní skóre jeho prvků, neexistuje tedy žádné řešení, než to doposud známé.

$$\xi^* = argmax_{(\xi \epsilon W)} \Sigma_{k=1}^{K} M_{nearest}(T_\xi h_k)$$

W ... vyhledávací okno
$M_{nearest}$ ... rozšíření pravděpodobnostní mřížky M na všechny R^2 zaokrouhlením argumentů do nejbližšího bodu mřížky - rozšířená hodnota bodu mřížky ukazuje na odpovídající pixel

Výběr uzlů probíhá prohledáváním do hloubky, Efektivnost algoritmu hodně závisí na podobě stromu, zda-li má pro výpočet dobrou horní mez a dobré aktuální řešení. Důležitým termínem je zde prahová hodnota skóre. Jedná se o hodnotu, pod níž nemáme zájem jít a řešení je pro nás tedy nevyhovující. Díky tomu nepřidáváme špatné shody jako omezení pro uzavírání smyčky. Pro rychlost algoritmu je důležité rozhodování o průchodu stromem. Pro každého potomka se vypočítá horní hranice skóre a vybíráme toho nejslibnějšího, což je uzel s největším mezním počtem.

Každý z uzlů je popsán pomocí tuple integerů:

$$c = (c_x, c_y, c_\Theta, c_h) \epsilon Z^4$$

c_h ... výška uzlu - $c_h = 0 \rightarrow$ uzel je list

Horní meze jsou počítány na vnitřních uzlech, máme zde zájem o výpočetní úsilí a kvalitu spojení.

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

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

5.1 Úvod

Hlavním znakem systému Hector SLAM, je jeho rychlost a nízké výpočetní nároky oproti předchozím dvěma typům, které jsou v této práci rozebírány. Je tedy vhodný pro implementaci v menších autonomních systémech, pro rychlý pohyb terénem a nehodí se pro uzavírání velkých smyček.

Vnitřní složení je z hlavních tří částí, a to 2D SLAM, běžící jako soft real time, 3D navigace, která je hard real time a jednotka IMU, která se nachází i v systému Cartographer. Hector SLAM je braný jako frontend SLAM, což znamená, že dochází k odhadu robota v reálném čase. Backend SLAM optimalizuje poziční graf vzhledem k omezením mezi pozicemi.

5.2 Přehled

Daný systém se musí převézt z 3DOF pomocí naklonění a rotace na 6DOF, kdy navigační filtr spojí měření z inerciální jednotky (IMU) a dalších senzorů, tím získáváme 3D řešení a díky 2D SLAMu máme informaci o poloze v prostoru. Oba odhady se aktualizují nezávisle na sobě a jsou propojeny jen velmi málo.

Kvůli znatelnému posunu odhadů integrované pozice a rychlosti, který je způsoben šumem v senzorech, zahrnujeme do systému další informace, jako je například porovnávání snímků, snímání magnetického pole, senzor barometrického tlaku a nebo měření rychlosti kol.

pohyb tuhého tělesa:

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

$$\dot{p} = v$$

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

$x = (\Omega^T p^T v^T)^T$ ... reprezentace 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
R_{Ω} ... matice směrových cosinů
E_{Ω} ... mapování natočení těla na deriváty Eulerova úhlu
g ... vektor gravitace

5.3 2D SLAM

Jako reprezentace prostředí je zde vybrána osvědčená metoda mřížkových map. Odhadovaná pozice robota slouží pro transformaci skenu na lokální stabilizovaný souřadnicový rámec. Odhadovanou orientací a přidruženými hodnotami je ze skenu vytvořen oblak bodů, který si pro odstranění odlehlých bodů předzpracováváme. Filtrace probíhá pouze

na základě souřadnic koncového bodu, kdy jsou pro porovnávání skenu použity pouze koncové body v rámci skenovací roviny.

Jak již bylo poznamenáno, je zde použita struktura mřížkových map, to vede k omezení přesnosti a neumožnění přímého výpočtu interpolovaných hodnot a derivátů. Pro oba odhady se tedy využívá interpolační schéma, díky kterému tu možnost máme. Jakoukoliv souřadnici, kterou potřebujeme nahradit, aproximujeme pomocí čtyř nejbližších integerových souřadnic.

Laserové skenery jsou již velice přesná zařízení, jejich měření zatěžuje minimální šum a snímky se dají vytvářet s vysokou frekvencí. Měření pomocí laseru je tedy mnohem přesnější než to odometrické, což je jeden z důvodu, proč je odometrie v tomto systému zcela vynechána. Při zarovnávání snímků s již známou mapou, nepotřebujeme hledat žádná spojení mezi koncovými body, ale porovnáváme s předchozími skeny.

Hledání transformace s nejlepším sladěním skenu s mapou:

$$\xi^* = argmin_\xi \Sigma_{i=1}^n [1 - M(S_i(\xi))]^2$$

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

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

Při tvorbě mapy dochází často k nalezení pouze lokálního minima, reprezentací mapy pomocí více rozlišení se tak toto riziko znatelně redukuje. Utváří se mřížkové mapy, kdy každá další má poloviční rozlišení, než ta předchozí. Máme tak uloženo více map, které se součastně aktualizují podle odhadu aktuální pozice, který je generovaný zarovnávacím procesem. Díky tomuto přístupu jsou mapy konzistentní a nepotřebují převzorkovávání. Zarovnávání skenů probíhá pouze na mapě s nejvyšším rozlišením a tento odhad je pak, jak je uvedeno, použit pro ostatní mapy. Mapy s nízkým rozlišením máme tedy v podstatě dostupné okamžitě, a můžeme je hned použít pro odhad trasy.

5.4 Odhad 3D stavu

Pro odhad úplného 6D stavu (3D stav + informace o naklonění z gyroskopů a informace z akcelerometrů) slouží navigační filtr běžící v reálném čase, který je asynchronně aktualizován při příchodu odhadované pozice z porovnávače nebo jiných informací ze senzorů. Filtr je implementován ve formě EKF, jedná se tedy o nelineární filtr a jako jeho známé vstupy se berou inerciální měření. Rychlost a pozici aktualizujeme integrací zrychlení. Abychom se ale zabránilo nezávislému růstu odhadu stavu při nepřítomnosti měření, dochází k aktualizace pseudo-nulové rychlosti, které se spouští při dosažení odchylky určité prahové hodnoty a zajišťuje stabilitu systému.

Při integraci systému máme tedy základní dva celky, 2D SLAM a 3D stavový odhad, které spolu musejí komunikovat v obou směrech. Jejich fungování neni nijak synchronizováno, odhad stavu běží typicky s vyšší frekvencí. Odhadovaná pozice z EKF je brána jako počáteční odhad pro optimalizaci porovnávání. Opačným směrem je použit pro spojení pozice ze SLAM s celkovým odhadovaným stavem kovarianční průsečík (CI - covariance interseption).