

2D SLAM FOR INDOOR ENVIRONMENT

Jan Klečka

Master Degree Programme (2), FEEC BUT

E-mail: xkleck01@stud.feec.vutbr.cz

Supervised by: Karel Horák

E-mail: horak@feec.vutbr.cz

Abstract: This paper describes one of the possible ways to solve task of 2D SLAM (Simultaneous localization and mapping) in complex indoor environment. Localization part of the algorithm is realized by data registration using the ICP algorithm and mapping part uses discrete 2D data field for representing mapped environment.

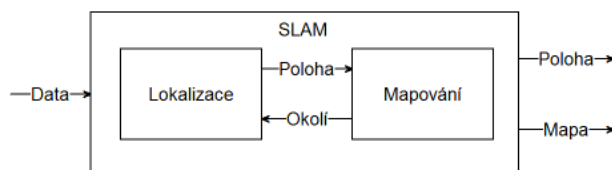
Keywords: EEICT, SLAM, mapping, SICK LMS 111, ICP

1. ÚVOD

Algoritmy označované jako SLAM jsou v oblasti robotiky již delší dobu zkoumané téma prozatím bez univerzálního řešení. V této práci se budu zabývat 2D variantou SLAMu pro mapování prostředí uvnitř budov, která pracuje s daty dostatečně členitými a ležícími pouze v jedné rovině, dále také předpokládá pohyb snímače pouze v této rovině.

2. OBECNÝ PRINCIP

Význam zkratky SLAM je možné do češtiny překládat jako „souběžná lokalizace a mapování“. Je tedy nasnadě na algoritmus nahlížet jako na dva samostatné, nicméně silně provázané problémy: část lokalizační a část mapovací. Základní myšlenka těchto algoritmů je potom taková, že přijdou-li nová data, je lokalizačním algoritmem nalezena optimální korelace s již zmapovaným prostředím, díky čemuž je nalezena aktuální poloha snímače. Dále jsou na základě zjištěné polohy zpracovávaná data transformována ze souřadnicového prostoru snímače do souřadnicového prostoru mapy a předána mapovacímu algoritmu. Mapovací algoritmus potom na základě takto připravených dat aktualizuje mapu. Graficky toto můžeme vidět na Obrázku 1.



Obrázek 1: Schematické znázornění principu algoritmů SLAM

Je zřejmé, že při startu vyžaduje algoritmus apriorní informaci buď o poloze, nebo o okolí. Nejčastěji je tento předpoklad řešen určením polohy v okamžiku startu měření např. při startu je poloha snímače předpokládána $[0,0]$ s nulovou rotací další měření probíhá relativně k této poloze a rotaci.

3. LOKALIZACE

Lokalizace slouží pro zjištění aktuální polohy snímače, což je jeden z výstupů algoritmu typu SLAM, a zároveň i vstup pro mapovací část algoritmu. Problém lokalizace jsem v tomto případě řešil pomocí registrace aktuálního snímku ze snímače vůči datům poskytnutým mapovacím algoritmem. Pro registraci jsem použil algoritmus ICP (Iterative Closest Point) [3], který byl původně

vyvinut pro registraci trojrozměrných dat, nicméně bez rozsáhlejších úprav je možné ho použít i pro registraci dvourozměrných množin bodů.

ICP je algoritmus iterativní. V každé iteraci najde nejbližší bod pro každý bod z registrované množiny v množině, vůči níž provádíme registraci, a následně minimalizuje sumu kvadrátů odchylek mezi těmito páry bodů vůči rotaci a translaci. Vypočtená transformace je následně aplikována na registrovaná data, čímž se algoritmus přiblíží k výslednému řešení. Pro rychlejší vyhledávání nejbližších bodů jsem implementoval k-d stromy. Korektní použití tohoto algoritmu ale vyžaduje, aby celá registrovaná množina byla obsažena v množině, vůči níž ji registrujeme – nemusí jít o naprosto stejné body (ICP je poměrně odolný vůči šumu), ale registrovaná data by neměla obsahovat informace, které nemají ekvivalent v druhé množině. Tento problém řeším tak, že vždy po nalezení dvojic nejbližších bodů, pro zbytek iterace odstráním 10-20% párů s největší vzdáleností od sebe.

4. MAPOVÁNÍ

Cíle mapovacího algoritmu jsou vytvořit mapu, která bude věrně reprezentovat okolní prostředí (výstup algoritmu SLAM), a především zpracování této mapy do podoby, kdy může efektivně sloužit jako vstup pro algoritmus lokalizace. Hlavním prostředkem pro splnění těchto cílů je zajištění přijatelného poměru mezi množstvím dat a množstvím informací, jež nese mapa. Tedy je třeba zajistit, aby mapa nekumulovala data nesoucí stejnou informaci. Dalším důležitým prvkem, který je součástí mapování, je mechanismus aktualizace (resp. zapomínání) starých dat. Změní-li se během funkce algoritmu okolní prostředí, mapovací algoritmus tuto změnu detekuje a přizpůsobí se.

Pro realizaci mapovacího algoritmu jsem zvolil přístup, u kterého je mapa reprezentována dvourozměrným polem. Pole je datového typu integer z důvodu pravděpodobnostního přístupu k zpracování dat. Hodnota uložená v buňce sice kvantitativně neodpovídá pravděpodobnosti výskytu překážky, ale buňky jsou díky tomu mezi sebou kvalitativně porovnatelné. Tato vlastnost rozšiřuje možnosti práce s uloženými daty.

4.1. ZÁPIS NOVÝCH DAT DO MAPY

Přijdou-li z lokalizačního algoritmu v čase t nová data, jsou ve formě množiny 2D bodů. Prostorové souřadnice každého bodu jsou přepočítány na indexy mapy pomocí měřítka a středu mapy. Následně jsou buňky odpovídající vypočteným indexům postupně spojeny pomocí úseček a ke zpracovávaným indexům jsou přidány indexy buněk, které tyto úsečky protínají. Následně jsou hodnoty p uložené v buňkách s těmito indexy (x, y) navýšeny pomocí diferenční rovnice (1).

$$p(x, y, t + 1) = 0,75 \cdot p(x, y, t) + 2500 \cdot e^{-ud} \quad (1)$$

Kde u je koeficient, který nastavuji v rozsahu $\langle 0,5;1 \rangle$ a d je manhattanovská vzdálenost zpracovávaného indexu od nejbližší buňky, do které připadá bod ze vstupních dat.

4.2. AKTUALIZACE STARÝCH INFORMACÍ

Tato část algoritmu je založena na myšlence, že data z laserového snímače neobsahují jen informaci o tom, v jaké vzdálenosti od snímače je překážka, ale i informaci o segmentu prostoru, ve kterém žádná překážka není. Je-li hodnota v buňkách odpovídajícím těmto prostorovým segmentům různá od nuly, je daná buňka penalizována pomocí diferenční rovnice (2).

$$p(x, y, t + 1) = 0,75 \cdot p(x, y, t) \quad (2)$$

4.3. GENEROVÁNÍ DAT PRO LOKALIZAČNÍ ALGORITMUS

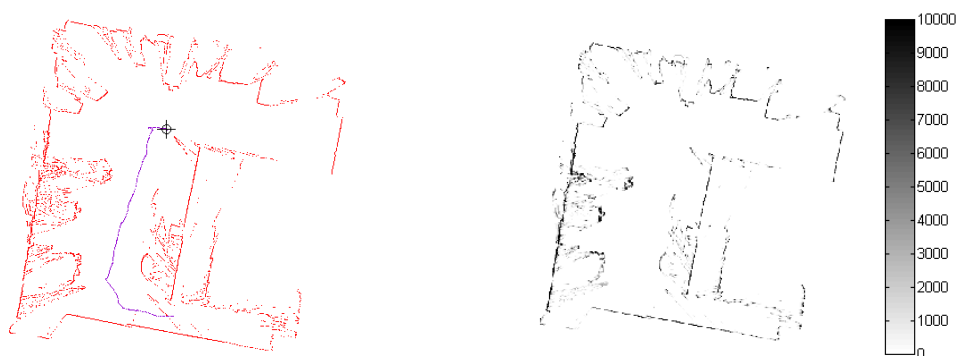
Tato část algoritmu je vykonána jako úplně první věc po příchodu nových dat ze senzoru, tedy ještě před lokalizací. V prvním kroku je na mapu aplikován filtr non-maximum suppression [4]. A dále jsou vysegmentovány buňky vyfiltrované mapy, jejichž odpovídající prostorové souřadnice mají od

polohy snímače v minulém snímku menší vzdálenost, než je maximální naměřená vzdálenost překážky od snímače v současném snímku. Výstup této části algoritmu potom tvoří prostorové souřadnice buněk uvnitř segmentované oblasti, které mají nenulovou hodnotu pravděpodobnosti.

5. VÝSLEDKY

Pro testování navrženého algoritmu jsem použil laserový snímač vzdálenosti SICK LMS 111 [1], který jsem pro účely testování připevnil k pojízdné konstrukci, která udržovala konstantní výšku snímače a zároveň rovnoběžnost roviny měření s rovinou podlahy. Mapování jsem uskutečnil v místnosti laboratoře. Obrázek 2 zobrazuje výslednou mapu jak formou pravděpodobnostní, tak i binární, která vznikla filtrací pravděpodobnostní mapy pomocí non-maximum suppression. V binární mapě je navíc zobrazena trajektorie pohybu snímače.

Test považuji za úspěšný. Algoritmus se po celou dobu trvání testu správně lokalizoval a výstupní mapa odpovídá kvalitě dat přijímaných ze snímače.



Obrázek 2: Filtrovaná mapa s trajektorií (vlevo) a pravděpodobnostní mapa (vpravo)

6. ZÁVĚR

V této práci popisují algoritmus schopný v reálném čase (na běžném kancelářském PC 10-20 FPS) zmapovat komplexní dynamické prostředí. Z výsledků je vidět, že výsledné mapy obsahují jisté množství šumu. Možného zlepšení tohoto problému by bylo možné dosáhnout buď filtrací dat ze snímače, nebo pokročilejší filtrací dat obsažených v mapě.

PODĚKOVÁNÍ

Tato publikace vznikla za podpory grantu číslo FEKT-S-14-2429 - "Výzkum nových řídicích metod, měřicích postupů a inteligentních prostředků v automatizaci" financovaného z Interní grantové agentury Vysokého učení technického v Brně a také za podpory Centra kompetence realizované TACR (reg. číslo TE01020197).

REFERENCE

- [1] SICK. Laser Measurement Sensors of the LMS1xx Product Family [online]. 2008 [cit. 2014-01-27]. Dostupné z: <https://mysick.com/saqqara/pdf.aspx?id=im0031331>
- [2] RIISGAARD, Søren a Morten Rufus BLAS. SLAM for Dummies [online]. 2005 [cit. 2013-11-04]. Dostupné z: http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-412j-cognitive-robotics-spring-2005/projects/1aslambblas_repo.pdf
- [3] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," IEEE Trans. Pattern Anal. Mach. Intell., 14, 1992, 239-256.
- [4] Canny Edge Detection: 09gr820. [online]. s. 7 [cit. 2014-01-27]. Dostupné z: http://www.codeproject.com/KB/cs/canny_edge_detection/canny_edge_detection_c_.zip