

Semestrální práce - BI-ZUM

Barbora Karlíková

May 20, 2024

1 Definice problému

Jako téma semestrální práce jsem zvolila téma: "Ztracený robot". Cílem úlohy je lokalizovat robota ve známém bludišti na základě bloků, které vidí. Tuto úlohu jsem specifikovala na případ robota, jenž neuvažuje světové strany, neotáčí se a pohybuje se pouze 4 směry - nahoru, dolů, doleva, doprava. Ve své úloze jsem si dala za cíl porovnat dva typy simulovaných senzorů. Tzv. LiDAR je senzor který měří vzdálenost na základě rychlosti šíření laserového paprsku v prostoru. Jeho simulaci provádím tak, že robot s tímto senzorem vidí n bloků v každém směru, nebo méně, pokud se zde nachází překážka.

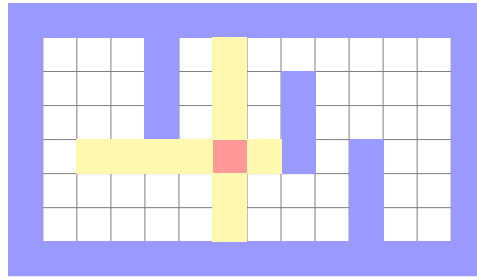


Figure 1: Ukázka robota s LiDAR senzorem o rozsahu $n = 4$

Jako druhý typ senzoru jsem zvolila klasický kernel, tedy čtverec o hraně n , kde n je striktně liché číslo. V tomto případě robot vidí oprotu LiDARu i za překážku.

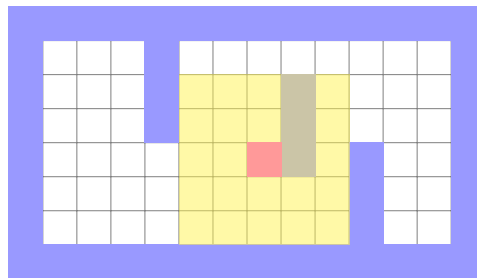


Figure 2: Ukázka robota s kernelovým senzorem o rozsahu $n = 5$

2 Algoritmus pro nalezení polohy robota

Úlohu jsem se rozhodla vyřešit za pomoci hashování a DFS. V prvním kroku po načtení mapy si spočítám hash pro každé pole, na kterém se potenciálně může nacházet robot. Hash je počítán z možného pokledu, který by měl robot na daném místě. Konkrétně:

- pro **LiDar** se hash skládá ze 4 16-bitových neznamímkových integerů, které vyjadřují n volných bloků v každém směru. Ty pak tvoří 64-bitové neznamímkový integer, který je hashem.

- pro **kernel** je hashem maximálně 64-bitový neznamínkový integer, jehož každý bit označuje v pořadí od levého horního rohu, zda je pole zeď, či nikoliv.

Všechny pozice jsou pak uloženy v hash tabulce.

Po předzpracování mapy, robot spočítá hash na základě toho co "vidí" a uvažuje pozice z hash tabulky. Dokud uvažovaná pozice není pouze jedna jediná, pohybuje se po mapě pomocí prohledávání do hloubky (DFS). Pro zaznamenávání prozkoumaných pozic si robot drží zároveň množinu objevených vrcholů v relativních souřadnicích. V každém kroku nejprve robot zvolí pohyb vpřed. Pokud existuje možný pohyb vpřed, posune se a spočítá hash okolí, které vidí. Vytáhne si možné pozice z hash tabulky a dále uvažuje jen ty startovní pozice, jejichž součet s absolutním posunutím v mapě se shoduje s některou z právě nalezených pozic. Pokud neexistuje možný pohyb vpřed, robot se vrátí o jeden pohyb zpět a pokusí se najít pohyb zde. Algoritmus končí v momentě, kdy uvažovanou pozicí je pouze jedna pozice, nebo se robot dostal do některé z uvažovaných začátečních pozic a nemůže udělat pohyb vpřed. Pak nejspíš existuje více řešení.

3 Statistiky

Statistiky provedu na jedné relativně velké mapě. Jako první je zde běh robota s LiDARem, pro $n = 3$

```
Size of map: 417 * 288 = 120096
-----
Duration of map preparation: 46425 microseconds
Duration of search: 64131000 microseconds
Number of steps: 22
Found position 394, 169
-----
Original position: 394, 169
```

Dále jsem provedla dalších 6 běhů s různými hyperparametry a porovnála jejich časy. Všechny běhy byly přesné v určení počáteční polohy. V případě robota s LiDARem, jenž má dostatečný rozsah, jsou výsledky, co se týče kroků, velmi příznivé. Snižuje se tak čas hledání.

Figure 3: Porovnání různých senzorů a jejich konfigurací na mapě o velikosti 417x288

Senzor type	n	Duration of map prep.	Search duration	Number of steps	Found	Expected
LiDAR	2	57 901 μ s	149 178 000 μ s	22	394, 169	394, 169
LiDAR	3	50 042 μ s	105 130 000 μ s	22	394, 169	394, 169
LiDAR	7	59 468 μ s	59 468 μ s	14	394, 169	394, 169
LiDAR	10	50 346 μ s	35 000 μ s	5	394, 169	394, 169
LiDAR	INT16_MAX	65 797 μ s	13000 μ s	3	394, 169	394, 169
kernel	3	46 644 μ s	120 555 000 μ s	21	394, 169	394, 169
kernel	5	39 485 μ s	62 665 000 μ s	20	394, 169	394, 169
kernel	7	65 049 μ s	37 179 000 μ s	17	394, 169	394, 169

Z tabulky můžeme vidět, že ani jeden přístup se neliší v časové složitosti předpřípravě mapy. Nicméně u samotné lokalizace můžeme vidět rapidní zlepšení už pro LiDAR o rozsahu 7 bloků. Velmi se snížil čas hledání a přirozeně tedy počet kroků robota. I když například robot s kernelem s $n = 7$ mohl vidět za stěny a mít tak daleko komplexnější informaci o celkovém rozsahu 49 bloků, takový robot s LiDAR senzorem také s $n = 7$ měl informaci maximálně o 47 blocích.

Na závěr si netroufám říci, který přístup je lepší, ale pokud bychom se měli zaměřit na počet zkoumaných bloků v jednom pohledu robota, jistě by se vyplatil přístup simulující LiDAR. I když vidí jen 4 pruhy v mapě, má většinou větší rozsah, nebo cennou informaci o překážce, které ve výsledku přináší podstatnější informaci a tak i rychlejší lokalizaci.