

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Evidenčné číslo: FEI-100863-87902

**Metódy určovania najlepšieho nasledujúceho pohľadu priemyselného
roboťa v procese mapovania neznámeho prostredia**

Študijný program:	Robotika a kybernetika
Študijný odbor:	kybernetika
Školiace pracovisko:	Ústav robotiky a kybernetiky
Vedúci záverečnej práce/školiťel':	Ing. Michal Dobiš

Bratislava 2021

Artur Shults

Zadanie:

Aplikácie autonómnej navigácie priemyselného robota využívajú existujúcu mapu prostredia (lokálnu alebo globálnu). Mapa prostredia môže byť vytvorená aj počas samotnej navigácie. V tomto prípade môžu existovať požiadavky o zmapovanie neznámeho prostredia. Aby takáto činnosť bola čo najefektívnejšia a najrýchlejšia, je dôležité minimalizovať počet presunov robota. Touto problematikou sa zaoberajú metódy o určení najlepšieho nasledujúceho pohľadu robota, zvyčajne označovaných aj ako View Planning. Cieľom tejto práce je naštudovať problematiku o určení nasledujúceho najlepšieho pohľadu na scénu s robotom a zanalyzovať dostupné algoritmy so zameraním na optimalizáciu počtu polôh robota pri mapovaní prostredia. Výsledky je potrebné overiť simulačne aspoň na jednom vybranom algoritme. Ako simulačné prostredie je odporúčené použiť ROS Gazebo.

Úlohy:

1. Oboznámte sa s problematikou o určovaní najlepšieho nasledujúceho pohľadu (View Planning) robota v 3D priestore.
2. Analyzujte dostupné algoritmy View Planning so zameraním na optimalizáciu počtu polôh robota.
3. Oboznámte sa so simulačným prostredím.
4. Aspoň jednu vybranú metódu overte simulačne.
5. Spracujte a vyhodnoťte výsledky.
6. Vypracujte dokumentáciu k dosiahnutým výsledkom.

Abstrakt

Slovenská technická univerzita v Bratislave

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Študijný odbor: Kybernetika

Študijný program: Robotika a kybernetika

Autor: Artur Shults

Metódy určovania najlepšieho nasledujúceho pohľadu priemyselného robota v procese mapovania neznámeho prostredia

Vedúci diplomovej práce: Ing. Michal Dobiš

Kľúčové slová:

Pre navigáciu robota v prostredí je dôležitá mať mapu miestnosti, v ktorej sa robot nachádza. V prípade kedy mapu vytvárame v procese navigácie jedným z najväčších problémov je to, ako umiestniť kameru, aby najrýchlejšie a najpresnejšie vytvoril mapu prostredia. Aby vyriešiť tento problém existuje množstvo algoritmov view palnning. Cieľom tejto práce je pomocou simulácie porovnať existujúce algoritmy view planing. Výsledkom práce by malo byť overenie efektívnosti metód view palnnig.

Obsah

Zadanie:.....	2
Abstrakt.....	3
Úvod.....	6
1 Problematika určovania najlepšieho nasledujúceho pohľadu (View Planning).....	7
1.1 Reprezentácia prostredia.....	7
1.1.1 Metrická reprezentácia.....	7
1.1.2 Topologická reprezentácia.....	8
1.1.3 Hybridná reprezentácia.....	8
1.2 Mapovanie prostredia.....	9
1.3 Mračno bodov.....	9
1.4 Mriežka obsadenosti.....	10
1.5 Metódy určovania najlepšieho nasledujúceho pohľadu založení na modeli.....	11
1.5.1 Metódy založené na teórii množín.....	11
1.5.2 Metódy založené na teórii grafov.....	12
1.5.3 Metódy založené na výpočtovej geometrii.....	12
1.6 Metódy určovania najlepšieho nasledujúceho pohľadu s neznámou geometriou modeli.....	12
1.6.1 Povrchovo založené metódy.....	13
1.6.2 Volumetrické metódy.....	14
1.6.3 Globálne metódy.....	17
2 Úvod do ROS a opis simulačného prostredia.....	18
2.1 ROS.....	18
2.2 Simulačné prostredie.....	18
2.3 MoveIt.....	19
2.4 Opis robotického manipulátora.....	20
2.5 Hlbkova kamera.....	22
3 Návrh riešenia a implementácia.....	23
3.1 Príprava ROS komponentov.....	23
3.2 Metódy hodnotenia pohľadov.....	23
3.3 Implementácia volumetrickej metódy.....	24
3.4 Parametre metódy.....	27
4 Simulácia metódy volumetrickeho najlepšieho nasledujúceho pohľadu.....	29
4.1 Simulácia pre rôzne polomery.....	29
4.2 Simulácia pre rôzne uhly medzi pohľadmi.....	31

4.3 Simulácia pre rôzne váhy vzdialenosti k ďalšiemu pohľadu.....	33
4.4 Vyhodnotenie výsledkov.....	35
Záver.....	36

Úvod

Jednou z najväčších výziev pri práci s dnešnými laserovými skenermi s vysokým rozlíšením je obrovské množstvo dát, ktoré vydávajú. Aj keď sa výkon počítačov za posledných pár rokov výrazne zvýšil, zostáva problém so spracovaním veľkého množstva dát prijatých z moderných skenerov s vysokým rozlíšením. Budovanie, prieskum a mapovanie trojrozmerných autonómnych modelov sú oblasti, kde sa často vyskytujú problémy s veľkým objemom údajov. Jednou z dôležitých častí algoritmov používaných vo všetkých týchto aplikáciách je riešenie problému určovania najlepšieho nasledujúceho pohľadu alebo "Next-Best-View". Pojem Next Best View označuje problém, keď pri vytváraní mapy oblasti alebo modelu objektu hľadáme pohľad, z ktorého je možné získať čo najviac informácií. V tejto práci je predstavená metóda volumetrického plánovania nasledujúceho pohľadu. Praca sa obmedzuje na volumetrické metódy, pretože sú všestranné a väčšina z nich nevyžaduje znalosti scény, ktorá sa má modelovať. Výsledkom práce je evaluácia výhodnosti metód s ohľadom na rôzne hodnotiace kritériá.

Prvá kapitola zaoberá problematikou určovania najlepšieho nasledujúceho pohľadu, tomu aké spôsoby reprezentácii prostredia existujú, metódami určovania najlepšieho nasledujúceho pohľadu a analýze dostupných algoritmov. V druhej kapitole opísaný použitý softvér, hardvér. Tretia kapitola obsahuje návrh implementácii a opisuje metódy hodnotenia pohľadov. V štvrtej kapitole opísané experimenty a zhodnotení ich výsledky.

1 Problematika určovania najlepšieho nasledujúceho pohľadu (View Planning)

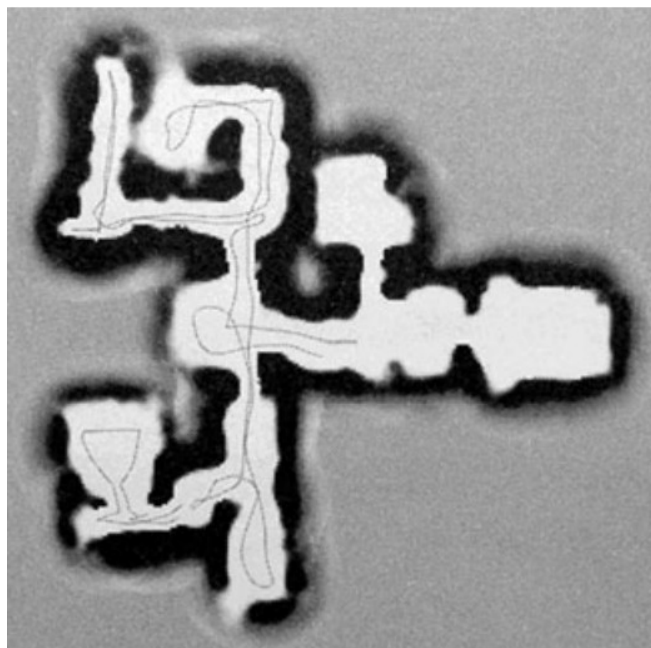
Pre porovnanie metód najlepšieho nasledujúceho pohľadu je potrebné určiť aké metódy a prostriedky použiť. Minimálne pre úlohu je potrebné nástroje na simuláciu a kontrolu robota, robotický manipulátor a hĺbková kamera, pomocou ktorých je možné otestovať algoritmy.

1.1 Reprezentácia prostredia

Prostredie je možné reprezentovať pomocou mapy prostredia, ktorá sa vytvára v reálnom čase. Mapa by mala reprezentovať geometriu priestoru okolo robota, aby robot prispôbil pohyb v závislosti od prekážok okolo neho. Rviz použitý v tejto práci má funkciu na vytvorenie oktmapy pomocou dát z hĺbkovej kamery. Rovnako sa pri plánovaní pohybu robota rviz zohľadňuje údaje z oktmapy. Ako takú reprezentáciu môže použiť metrickú, topologickú alebo hybridnú mapu prostredia.[1]

1.1.1 Metrická reprezentácia

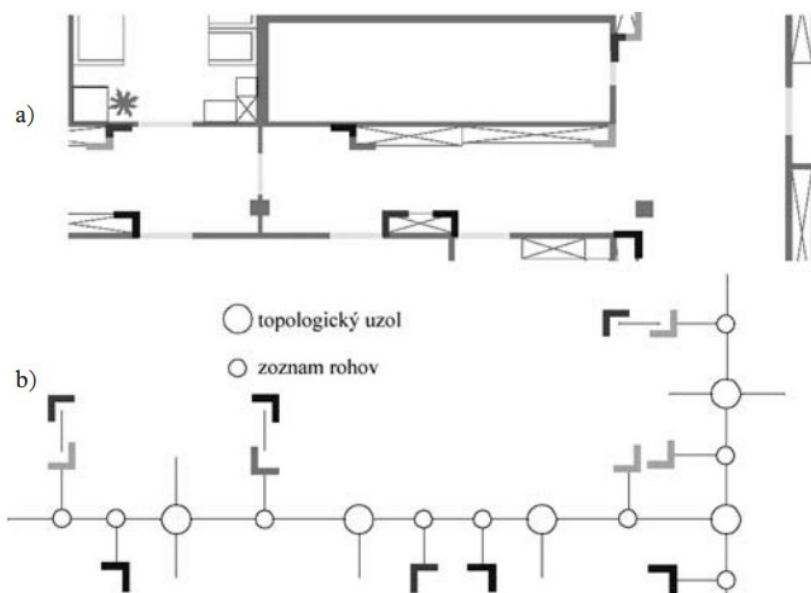
Metrická mapa - Je reprezentovaná mriežkovým rozložením priestoru a každej bunke prislacha hodnota - voľný, obsadený, neznámy. Vychodou metrickej mapy je možnosť vytvoriť podrobnú reprezentáciu prostredia. Nevýhodou je veľký objem dát na zaznamenanie [2]



Obr. 1: Metrická mapa

1.1.2 Topologická reprezentácia

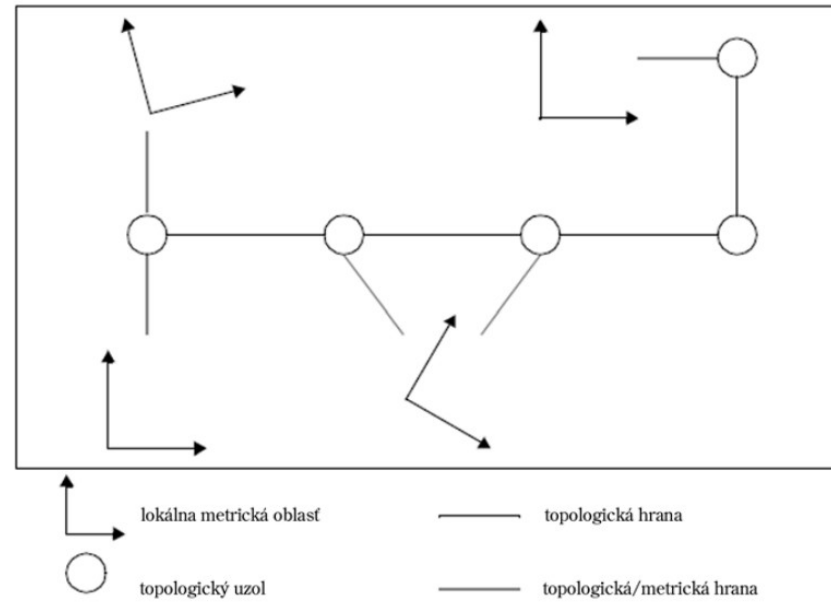
Topologická mapa - je formou grafu, kde uzly grafu reprezentujú vlastnosti prostredia a hrany grafu vzťahy medzi vlastnosťami prostredia. Výhodou je nízka kapacita dát na zaznamenávanie. Nevýhodou je problém s rozpoznávaním a identifikáciou vlastností prostredia na základe ktorých tvorí sa mapa[3].



Obr. 2: a) časť chodby s extrahovanými vlastnosťami prostredia – rohmi a vchodmi, b) topologická mapa reprezentovaná grafom

1.1.3 Hybridná reprezentácia

Hybridná mapa - je kombináciou metrickej a topologickej mapy. Reprezentujúci význačné miesta v prostredí uzly môžu byť opísané metrickou a topologickou formou. Má výhody metrickeho a topologickeho metóda, ale má problém určenia miest, kde potrebné prepnúť formu mapovania[2].



Obr. 3: Hybridná mapa

1.2 Mapovanie prostredia

Pre mapovanie prostredia je možné použiť hĺbkovú kameru. Výstupom kamery je množstvo bodov s označením vzdialenosti od kamery. Porovnaním výstupu z kamery a polohy robota je možné vytvoriť mračno bodov reprezentujúce reálnu polohu objektov v okolí robota. Postupným pridávaním nových údajov z mračna bodov do mapy je možné vytvoriť celkovú mapu prostredia. View planning algoritmy umožňujú znížiť čas na zostavenie mapy prostredia.

1.3 Mračno bodov

Mračno bodov je množina dátových bodov v priestore. Body môžu reprezentovať 3D povrch alebo objekt. Každý bod má svoje koordinaty v karteziánskej súradnici (X , Y , Z). Mračivo bodov je možné vytvoriť pomocou hĺbkovej kamery alebo fotogrametrického softvéru, ktorý meria veľký počet bodov na vonkajších povrchoch objektov. Výstupom procesu 3D skenovania je mračivo bodov používané na rôzne účely, vrátane vytvárania 3D modelov CAD pre vyrobené diely, na metrológiu a kontrolu kvality a na množstvo aplikácií na vizualizáciu, animáciu a vykresľovanie.[4]



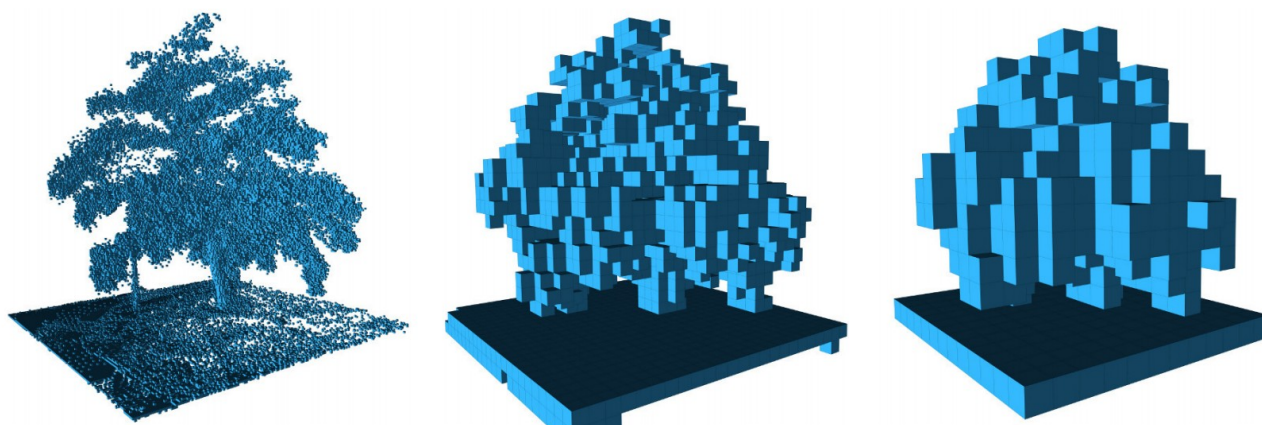
Obr. 4: Mračno bodov

1.4 Mriežka obsadenosti

Základnou myšlienkou mriežky obsadenosti je predstaviť mapu prostredia ako rovnomerne rozložené pole binárnych náhodných premenných, z ktorých každá predstavuje prítomnosť prekážky na danom mieste v prostredí. Algoritmy mriežky obsadenosti počítajú približné zadné odhady pre tieto náhodné premenné. Dôležitým je rozlíšenie bunky v mape. Čím menšia bunka tým presnejšia je mapa, ale taká mapa využíva viac pamäte.[5]

Mriežka obsadenosti by mala spĺňať tieto požiadavky:

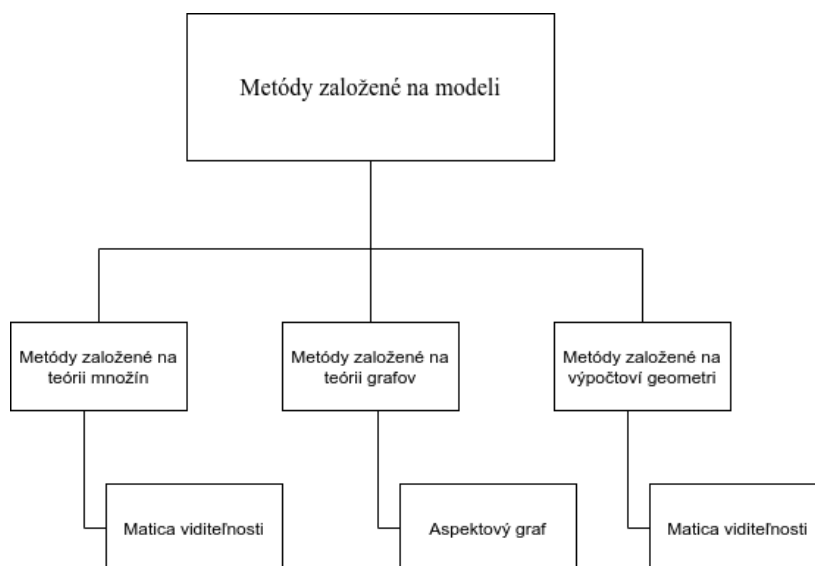
- Mapa by mala byť schopná modelovať ľubovoľné prostredie bez predchádzajúcich znalostí o ňom.
- Novú informáciu alebo údaje zo snímačov by malo byť možné kedykoľvek pridať do mapy.
- Ak nie je vopred známy rozsah mapy, mapa by mala podľa potreby dynamicky rozširovať.
- Mapa by mala byť efektívne uložená v pamäti aj na disku. Malo by byť možné vygenerovať komprimované súbory pre neskoršie použitie alebo pohodlnú výmenu medzi robotmi aj pri obmedzeniach šírky pásma.



Obr. 5: Obsadené bunky sa zobrazené v rozlíšení 0,08 m, 0,64 a 1,28 m

1.5 Metódy určovania najlepšieho nasledujúceho pohľadu založené na modeli

Metódy určovania najlepšieho nasledujúceho založené na modeli je možné kategorizovať podľa začiatkových znalosti modelu o objekte.



Obr. 6: Metódy založení na modeli

1.5.1 Metódy založené na teórii množín

Jadrom metódy množín je matica viditeľnosti, ktorej prvky kódujú v jednej dátovej štruktúre viditeľnosť diskretných povrchových bodov jednotlivých objektov z každej pózy v priestore pohľadu. Sofistikovanejšie algoritmy nahrádzajú viditeľnosť merateľnosťou, v ktorej dátové prvky

kódujú odhadovanú kvalitu merania. Predpokladané že objekt leží úplne v zhone senzora. Pomocou mapy obsadenosti bunkami predstavoval snímaný objem. Tento prístup plánovania pohľadu bol prispôsobený pre konkrétny senzor s aktívnou trianguláciou, takže problém plánovania pohľadu sa stáva problémom evaluácii množiny všetkých usporiadaných párov bodov na sfére pohľadu [6]

1.5.2 Metódy založené na teórii grafov

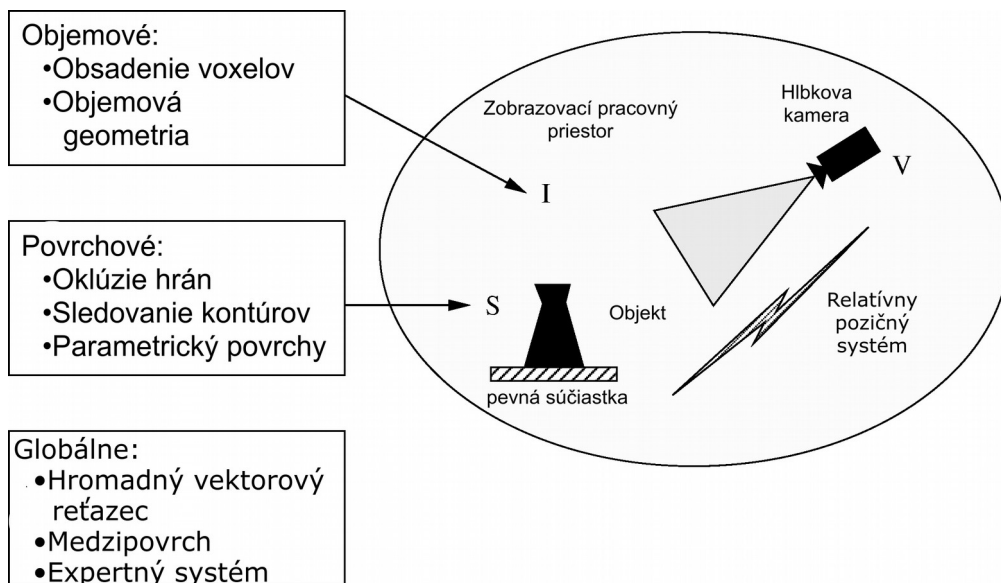
Aspektový graf objektu má uzol predstavujúci každý aspekt tohto objektu a oblúky spájajúce všetky susedné aspekty objektu. Aspekt je voľne definovaný ako množina pohľadov na objektu, takže časť objektu videná z všetkých pohľadov je kvalitatívne rovnaká. Inými slovami, priestor pohľadov je rozdelený do oblastí poskytujúcich rovnocenné pohľady. Každý uzol predstavuje jednu takúto oblasť, kde oblúky predstavujú ich susednosť.

1.5.3 Metódy založené na výpočtovej geometrii

Môže byť prevedený ako klasický „problém s umeleckou galériou“, kde je potrebné nájsť minimálny počet a umiestnenie strážcov, aby videli všetky steny v umeleckej galérii. Pri hľadaní najlepšieho nasledujúceho pohľadu je potrebné nájsť minimálny počet a polohu pozícií potrebných na to, aby bol viditeľný celý povrch objektu. Dodatočná zložitosť prichádza z práce v 3D priestore.

1.6 Metódy určovania najlepšieho nasledujúceho pohľadu s neznámou geometriou modeli

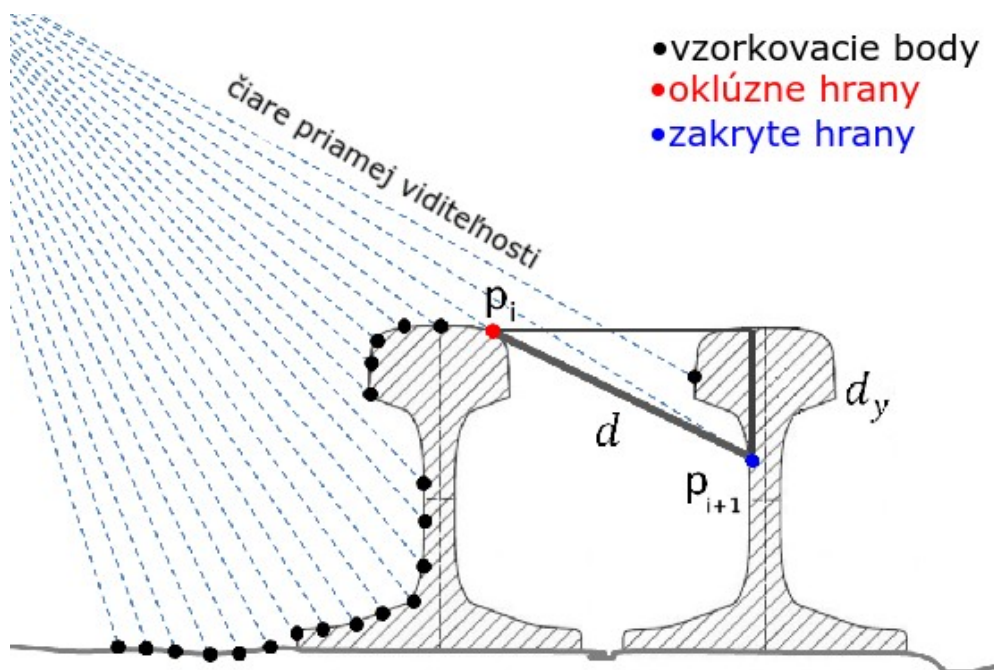
Je vhodné klasifikovať existujúce metódy určovania najlepšieho nasledujúceho pohľadu podľa toho, ako je možné urobiť evaluáciu pohľadu - to znamená, že sú dostupné volumetricka, povrchová alebo globálna metóda. Niektoré metódy kombinujú niekoľko techník. Väčšina spadá do dvoch pod kategórií - obsadenie buniek alebo okraje oklúzie.



Obr. 7: Metódy s neznámou geometriou modeli

1.6.1 Povrchovo založené metódy

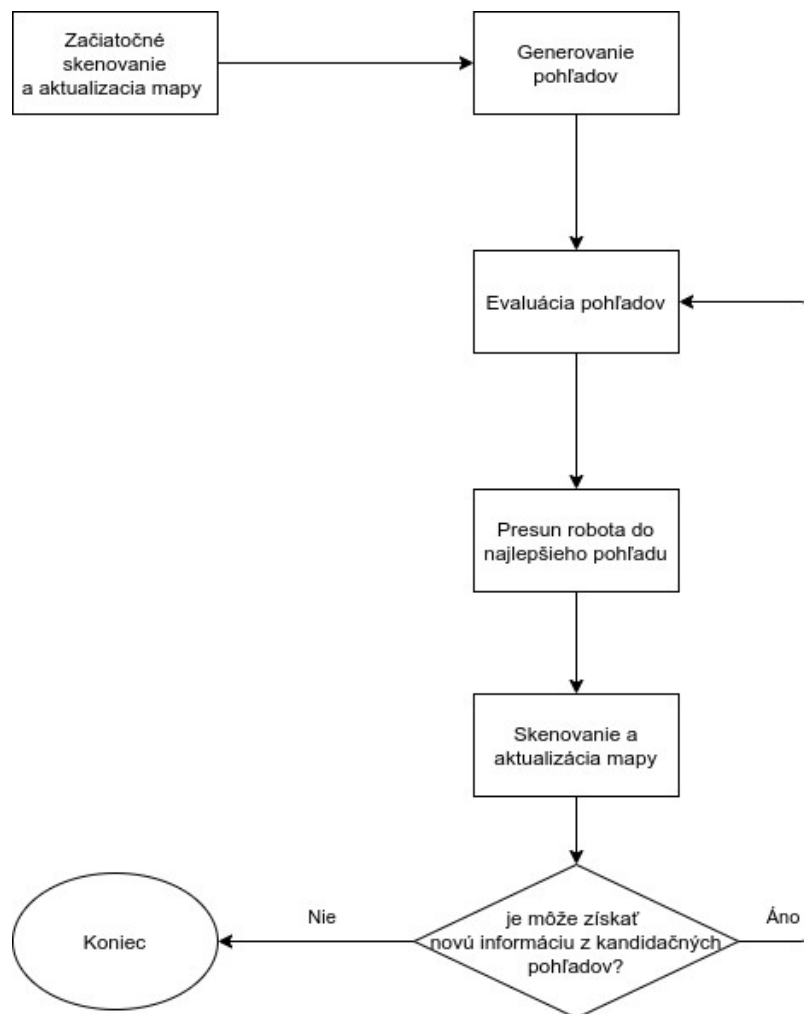
Algoritmy založené na znalosti priestora povrchu objektu. Metod oklúzných hrán sú najčastešie používanou tradičnou metódou využívajúcou geometrické ostre hrany. Tento prístup je založený na predpoklade, že oklúzne hrany vnútorné k obrázku naznačujú povrchové oblasti, ktoré ešte neboli vzorkované a ostre hrany predstavujú hranicu nepozorovaného objemu. V oboch prípadoch poskytujú oklúzne hrany podnety pre ďalší najlepší smer sledovania [7].



Obr. 8: Povrchovo založené metódy

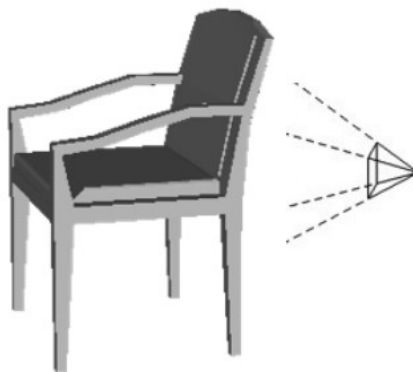
1.6.2 Volumetrické metódy

Metod je založený na tom, že robot vie polohu a rozmery objektu. V prvom kroku algoritmu, robot vykoná skenovanie hĺbkovou kamerou. Vytvorí mračno bodov zhodne s referenčnou polohou. Druhý krok je vytvoriť množstvo kandidačných pohľadov. Každý kandidačný pohľad ocení podľa toho koľko novej informácie vie zistiť. Tretí krok skenovanie hĺbkovou kamerou z najlepšieho kandidačného pohľadu a pridanie nových bodov do mapy. Opakuje druhy a tretí krok pokiaľ môžeme zistiť novú informáciu z kandidačných pohľadov[8].



Obr. 9: Volumetrické metódy

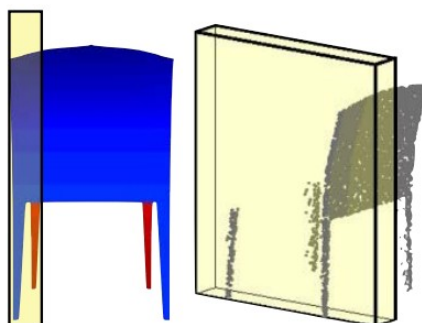
Na to, aby zistiť koľko novej informácie má každý pohľad je možné použiť bunkovú mapu(voxel map). Každá bunka má tri možné stavy: voľné miesto, neznámy alebo viditeľný povrch. Najlepším je pohľad, z ktorého bude možné uvidieť najviac neznámych kociek. Takže bunkovú mapu je možné použiť pri plánovaní pohybu, aby zabránil kolízii s predmetmi v okolí robota.



(1) objekt

Obr. 10: Ilustruje, že hĺbková mapa prevzatá z fyzického objektu v 3D svete

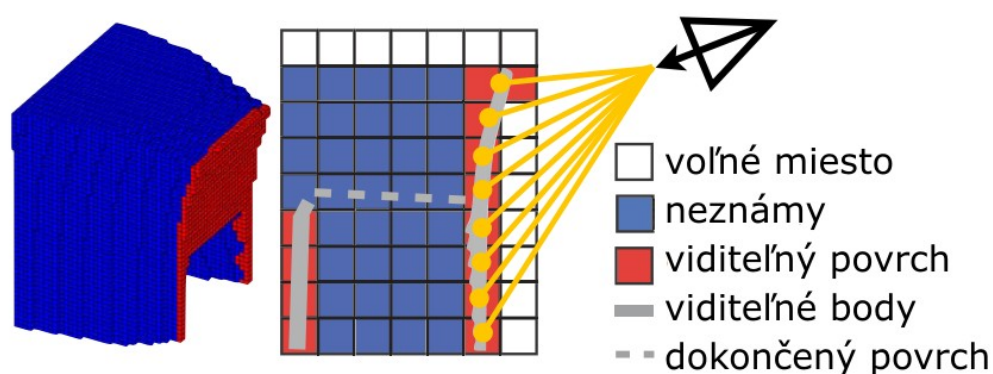
Najprv v priestore, ktorý chceme skenovať je potrebné rozmiestniť objekt približnú polohu a rozmery, ktorého zadáme do algoritmu a hĺbkovú kameru, ktorá vie svoju polohu v priestore.



(2) hĺbka a mračno bodov

Obr. 11: Zobrazuje hĺbkový obraz zachytený zo zadnej strany stoličky

Pri skenovaní výstupom z kamery bude obrázok, na ktorom ukazana vzdialenosť každého bodu od kamery. Ak vieme polohu kamery a vzdialenosť bodov od kamery tak je možné dostať polohy známych bodov v karteziánskej sústave.



(3) volumetricka reprezentácia

Obr. 12: Zobrazuje profil rezu a rôzne typy kociek.

Známe polohy z predchádzajúceho kroku pridáme do mapy. Kocky, na ktorých nachádzajú body z senzora označíme ako obsadené. Kocky, ktorí sa nachádzajú medzi kamerou a obsadenou kockou označíme ako voľné. Všetky ostatne kocky necháme, ako neznáme.



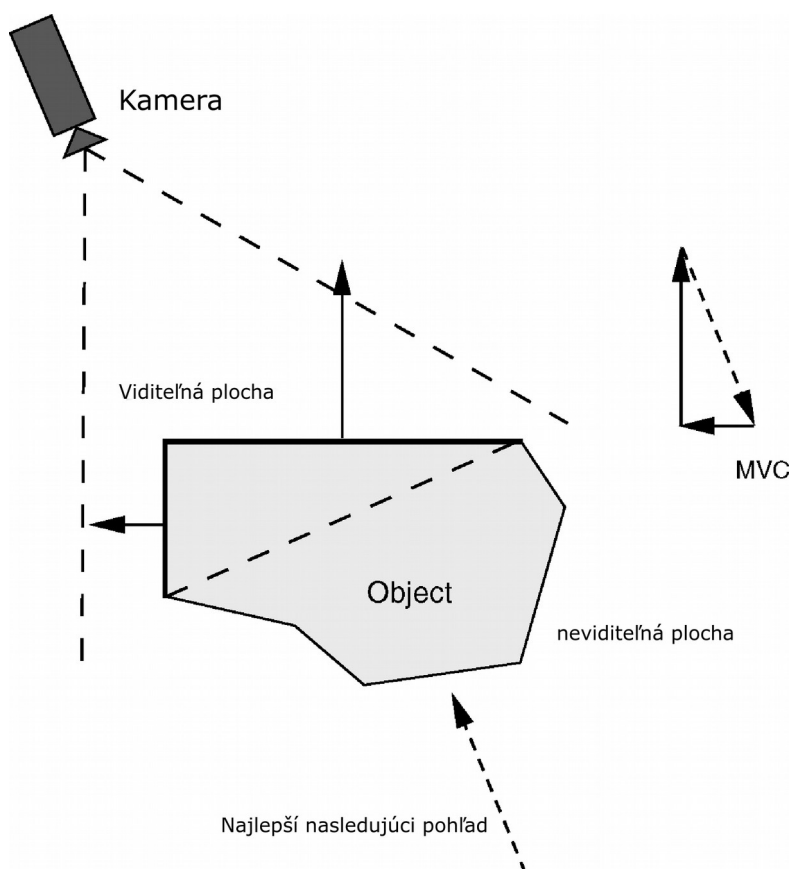
(4) uznanie a dokončenie

Obr. 13: Zobrazuje výsledok rozpoznávania a dokončenia tvaru.

Pokračujeme skenovať a aktualizovať mapu pokiaľ ostávajú neznáme kocky alebo pokiaľ budeme mať možnosť dostať novú informáciu.

1.6.3 Globálne metódy

Metódy globálneho plánovania používajú globálnu charakteristiku objektu, nie lokálnu ako iné metódy. Ilustrovaný na obrázku 11 prístup segmentuje povrch do množiny malých škvŕn a popisuje objekt pomocou reťazca hmotnostných vektorov (mass vector chain alebo MVC). Podľa definície MVC je rad vektorov váhy. Hraničné povrchy objektu tvoria uzavretú plochu povrchu iba vtedy, keď ich hromadné vektory tvoria uzavretý reťazec. Nasledujúci najlepší pohľad, teda môže byť byť nastavený na zápor kumulatívneho MVC, ktorý by mal definovať smer pohľadu, ktorého hmotnostný vektor uzavrie reťaz alebo aspoň skráti medzeru [9].



Obr. 14: Global View Planning

2 Úvod do ROS a opis simulačného prostredia

Na testovanie algoritmov je potrebné mať možnosť simulovať prostredie a riadiť robota v simulovanom prostredí.

2.1 ROS

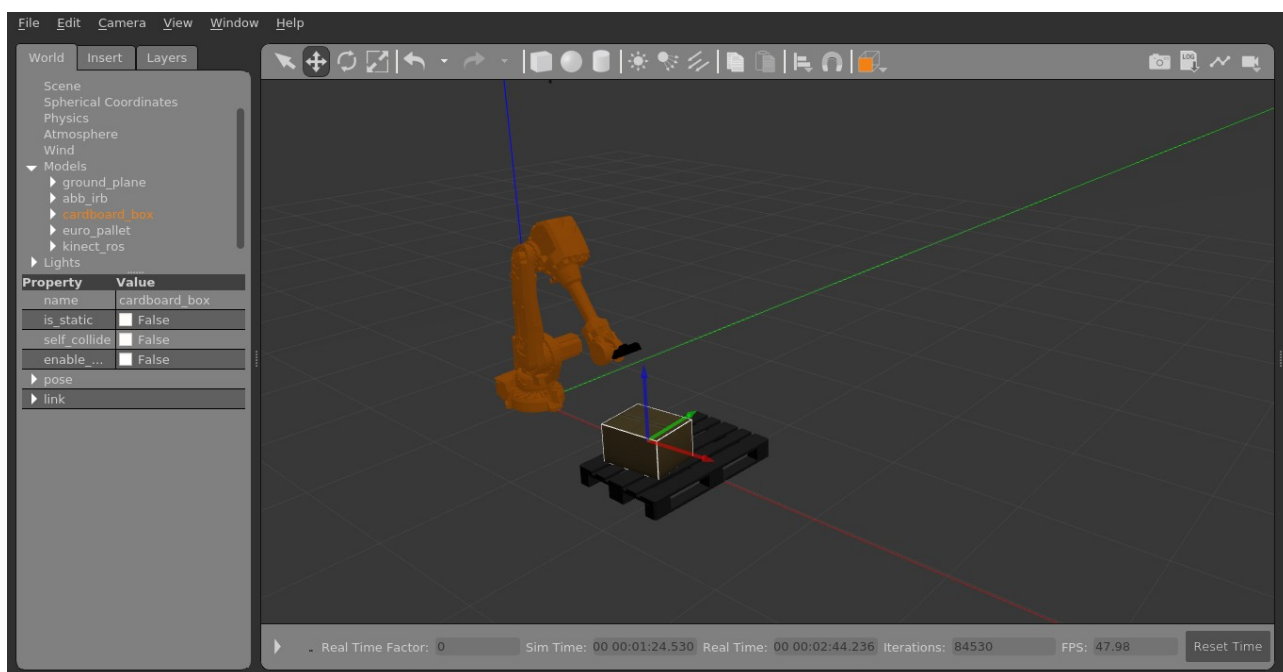
Robotický operačný systém (Robotic operating system - ROS) predstavuje voľne dostupný softvér, ktorý výrazne uľahčuje prácu pri riešených rôznych zložitých úloh robotiky. ROS je kolekcia frameworkov a knižníc pre vývoj robotického softvéru, abstrakciu hardvéru a nízkoúrovňové riadenie zariadení a výmenu správ medzi procesmi. Pomocou softvéru v systéme ROS je možné realizovať simuláciu, riadenie robota a vizualizáciu procesu vytvárania priestorovej mapy [10].

Pre potreby tejto práce sa budú využívať nasledovne ROS balíky:

- rviz - na vizualizáciu robota
- moveit - na plánovanie pohybov robota
- gazebo - na simuláciu robota a priestoru okolo neho

2.2 Simulačné prostredie

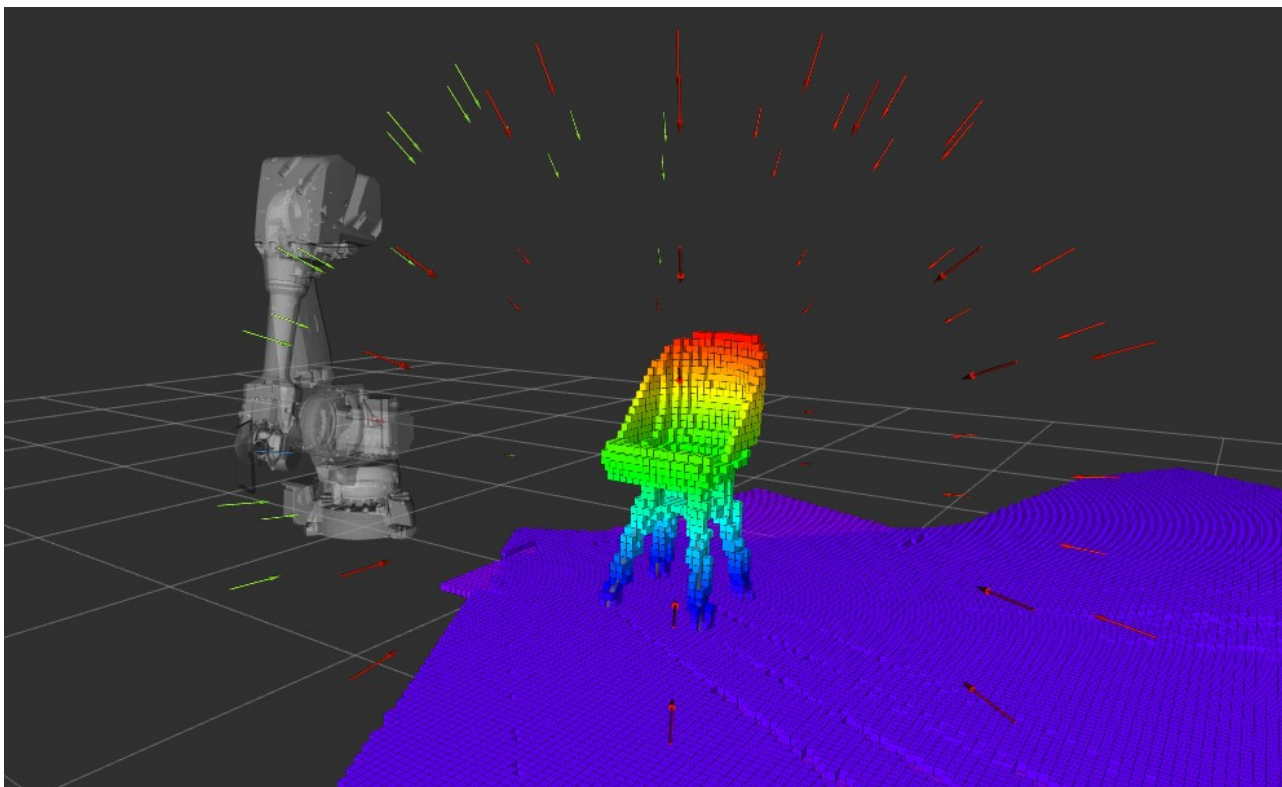
Gazebo je open source platforma na vytváranie a testovanie riadiacich algoritmov robotov [11]. Na testovanie view planning algoritmov je použitá schopnosť gazebo simulovať robota, riadiaci systém robota a čítať informáciu zo senzorov a kamery. Tiež Gazebo umožňuje vytvárať a meniť prostredie robota. Na testovanie v Gazebo sa použila model manipulátora ABB IRB 4600-40 / 2.55 s nainštalovanou kamerou kinect, z ktorej je možné načítať dáta a aktualizovať mapu prostredia.



Obr. 15: Gazebo

2.3 MoveIt

MoveIt voľne dostupná platforma pre vývoj robotických aplikácií, vytváranie prototypov a testovanie algoritmov. Súčasťou MoveIt je Setup Assistant. Hlavnou funkciou Setup Assistant je generovanie Semantic Robot Description Format (SRDF) súboru pre vybraný model robota. Dodatočne generuje aj iné potrebné konfiguračné súbory pre použitie v MoveIt pipeline. Pomocou vytvoreného SRDF je možné vizualizovať robota, data z kamery a mapu v rviz. Ovládanie pohybu robota s prihliadnutím na mapu sa vykonáva ovládaním pohybu, ktoré je tiež súčasťou MoveIt [12].



Obr. 16: rviz

2.4 Opis robotického manipulátora

Robotický manipulátor - zariadenie na manipuláciu s materiálmi bez priameho fyzického kontaktu s operátorom.

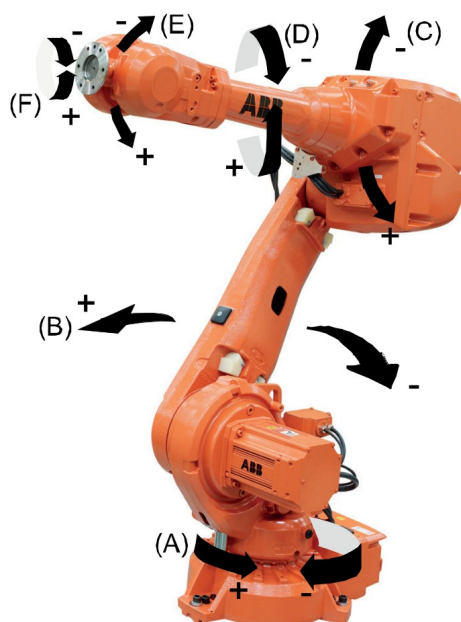
Cieľom priamej a inverznej kinematiky je určiť polohu koncového efektora a hodnoty stavového vektora segmentovej štruktúry [13].

- Priama kinematika - postupným výpočtom jednotlivých stavov stavového vektora pre všetky segment štruktúry vieme nájsť konečnú polohu.
- Inverzná kinematika(cieľom riadený pohyb) - definovaná iba konečná poloha efektora. Parametre a orientáciu všetkých kĺbov nachádzajú kinematické rovnice.

Ako robotický manipulátor používame ABB IRB 4600-40/2.55 Obr. 17. Model robota pri použití v simulácii je možné stiahnuť priamo od výrobcov a implementovať pomocou MoveIt.

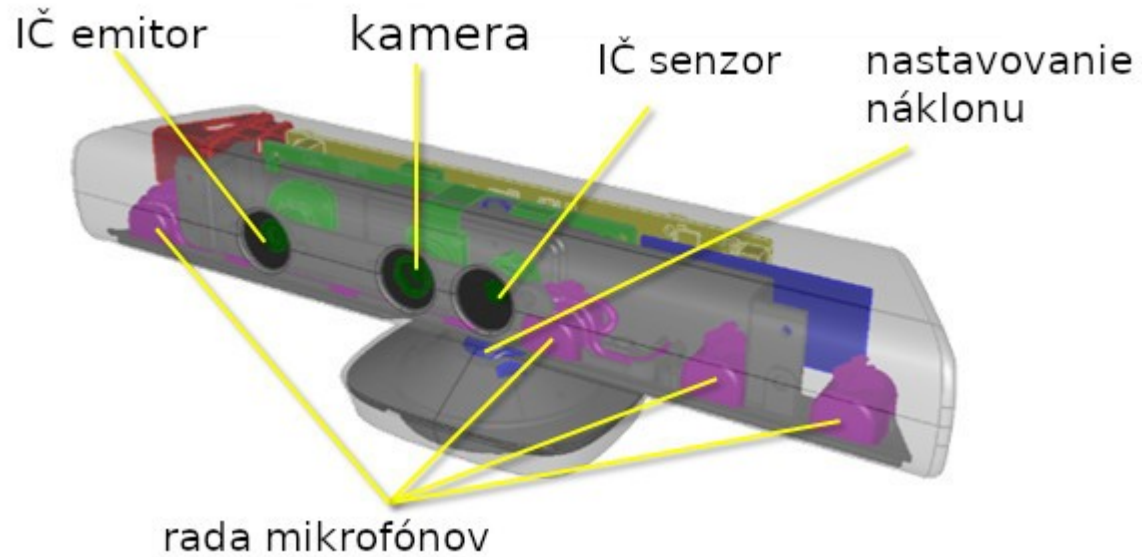
Parametre robota:[<https://new.abb.com/products/robotics/sk/roboty/irb-4600>]

Nosnosť	40 kg
Dosiachnuteľnosť	2.25 m
Hmotnosť	435 kg



Obr. 17: ABB IRB 4600-40/2.25

2.5 Hĺbková kamera



Obr. 18: Vnútorná štruktúra Microsoft Kinect

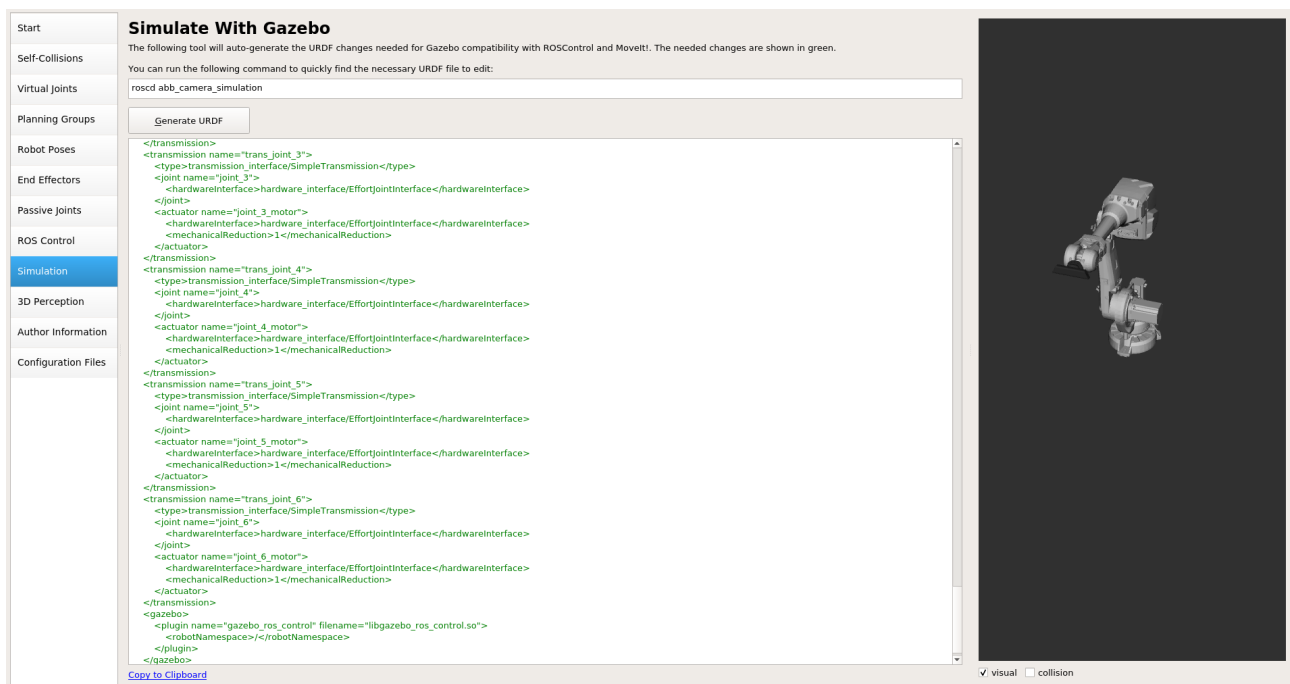
\

Pre vytvorenie mapy prostredia potrebujeme hĺbkovú kameru s možnosťou montáže na robota a obyčajnou kamerou pre kontrolu. Ako hĺbkovú kameru používame *Microsoft Kinect* (640 x 480 px) Obr. 18. Kamera ma IR senzor pomocú, ktorého je možné zistiť vzdialenosť k bodu na ploche, na ktorú pozerá kamera a vytvoriť mapu prostredia okolo kamery [14].

3 Návrh riešenia a implementácia

3.1 Príprava ROS komponentov

Implementácia metódy začína inštaláciou ROS. Na vytvorenie nového pracovného priestoru a súboru so skriptom metódy je použitý catkin. Ďalším krokom je nastavenie modelu robotického manipulátora s riadiacim systémom simulovaným v Gazebo. Ešte v tomto kroku je pridaná hĺbková kamera a je nastavený ROS node, na ktorý bude simulovaná kamera posielat' dáta. Potom pomocou MoveIt Setup Assistant vytvorit' súboru SRDF na vizualizáciu a riadenie robota v rviz a napísanie vlastných skriptov. Skripty je možné kompilovať pomocou catkin_make. Rviz sa používa na vizualizáciu robota a údajov.



Obr. 19: MoveIt Setup Assistant

3.2 Metódy hodnotenia pohľadov

Pri simulácií boli použité dva spôsoby evaluácii. Prvá metóda generuje guľu okolo pohľadu a robí evaluáciu počtu neznámych buniek v mape prostredia nachádzajú vo vnútri gule (na obrázkoch dole označená ako guľa). Druhá metóda generuje lúč medzi pohľadom a stredom objektu a robí evaluáciu počtu neznámych buniek, ktoré sa nachádzajú na lúče (na obrázkoch dole označená ako lúč). Takže

hodnotenie ovplyvňuje vzdialenosť pohľadu to terajšej polohy kamery. Nastaviteľný parameter váhy vzdialenosti ovplyvňuje to aká časť konečnej evaluácii pohľadu bude závisieť od vzdialenosti.

$$numberOfVoxels_{normalized} = \frac{(numberOfVoxels)}{(maxVoxelPerView)} \quad (3.1)$$

Normalizovaná známka pre počet voxelov. *NumberOfVoxels* - počet kociek viditeľných z pohľadu, *MaxVoxelPerView* - teoreticky maximálny počet viditeľných kociek.

$$dist_{normalized} = \frac{DistanceToNextTarget}{maxDistance} \quad (3.2)$$

Normalizovaná známka vzdialenosti. *DistanceToNextTarget* - vzdialenosť do ďalšieho pohľadu, *MaxDistance* - teoretická maximálna vzdialenosť

$$Mark = voxelWeight \times NumberOfVoxels_{norm} - distanceWeight \times distance_{norm} \quad (3.3)$$

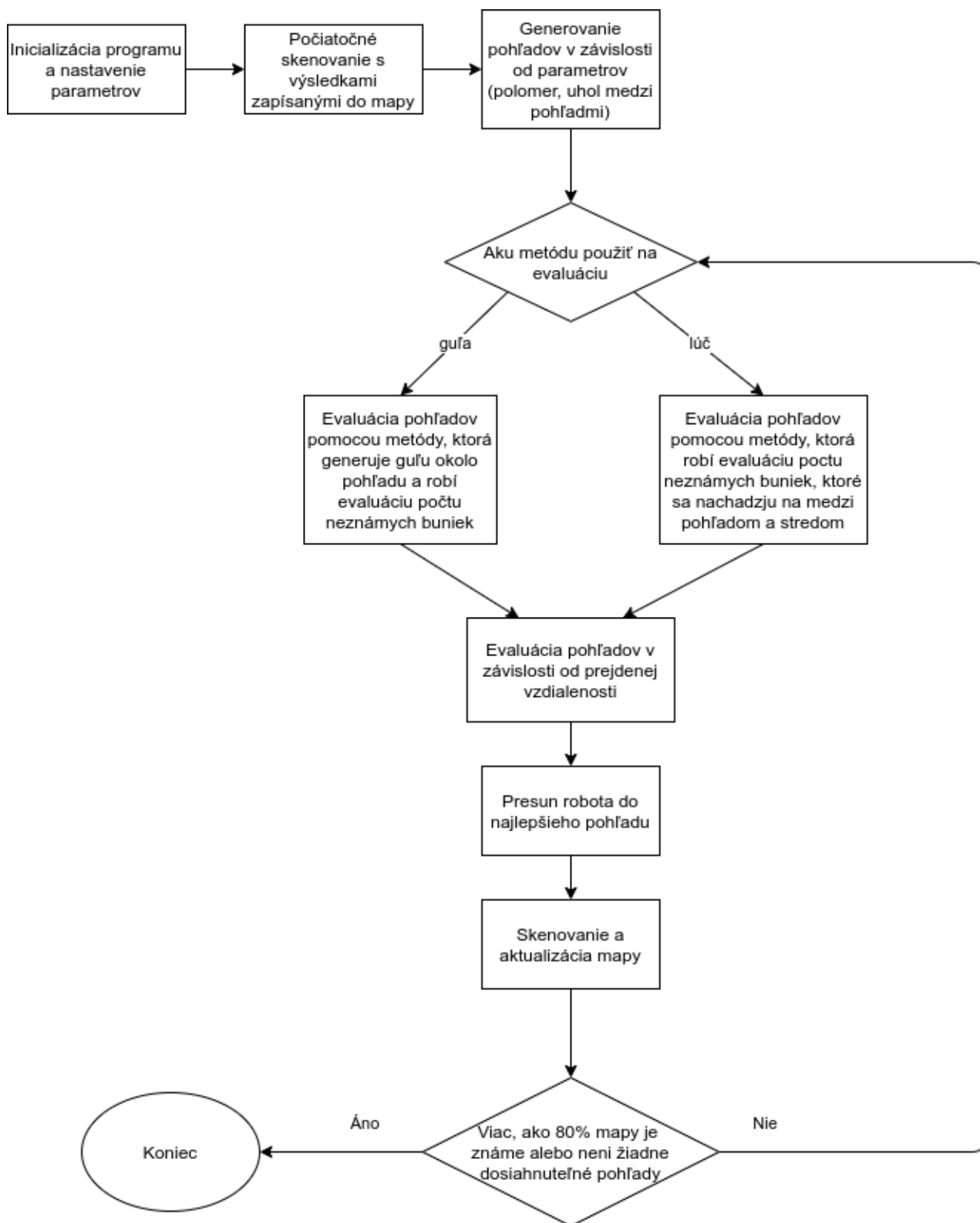
Finálna známka - *Mark*. *voxelWeight* a *distanceWeight* váha známky vzdialenosti a známka počtu viditeľných kociek.

3.3 Implementácia volumetrickej metódy

Skript implementácii volumetrickej metódy využíva algoritmus :

1. Zoberie parametre zo súboru s parametrami (polomer, uhol medzi pohľadmi, spôsob evaluácii, váha vzdialenosti).
2. Urobí začiatkové skenovanie výsledky, ktorého zápise do mapy prostredia.
3. Podľa zadaných parametrov (polomer a uhol) vytvorí pohľady okolo objektu
4. Pomocou inverznej kinematiky zisti, ktoré pohľady sú dosiahnuteľné (na Obr. 13 dosiahnuteľné pohľady označenou zelenou farbou, nedosiahnuteľne pohľady označení červenou).
5. Urobí evaluáciu pohľadov a nájde najlepší. Ako sa robí evaluácia popísané v kapitole 3.2.

6. Pomocou MoveIt move planning knižnici presune kameru robota do novej polohy.
Plánovanie pohybu berie do úvahy mapu prostredia a vyhýba sa kolízií.
7. Zapiše údaje z kamery do mapy a vymaže pohli zo zoznamu dostupných pohľadov.
8. Ak po pridaných údajov do mapy je viac, ako 80% mapy známe alebo neni žiadne dostupné pohľady program skončí ináč opakuje body 6, 7, 8 a 9 pokiaľ nebudú splnené podmienky ukončenia.

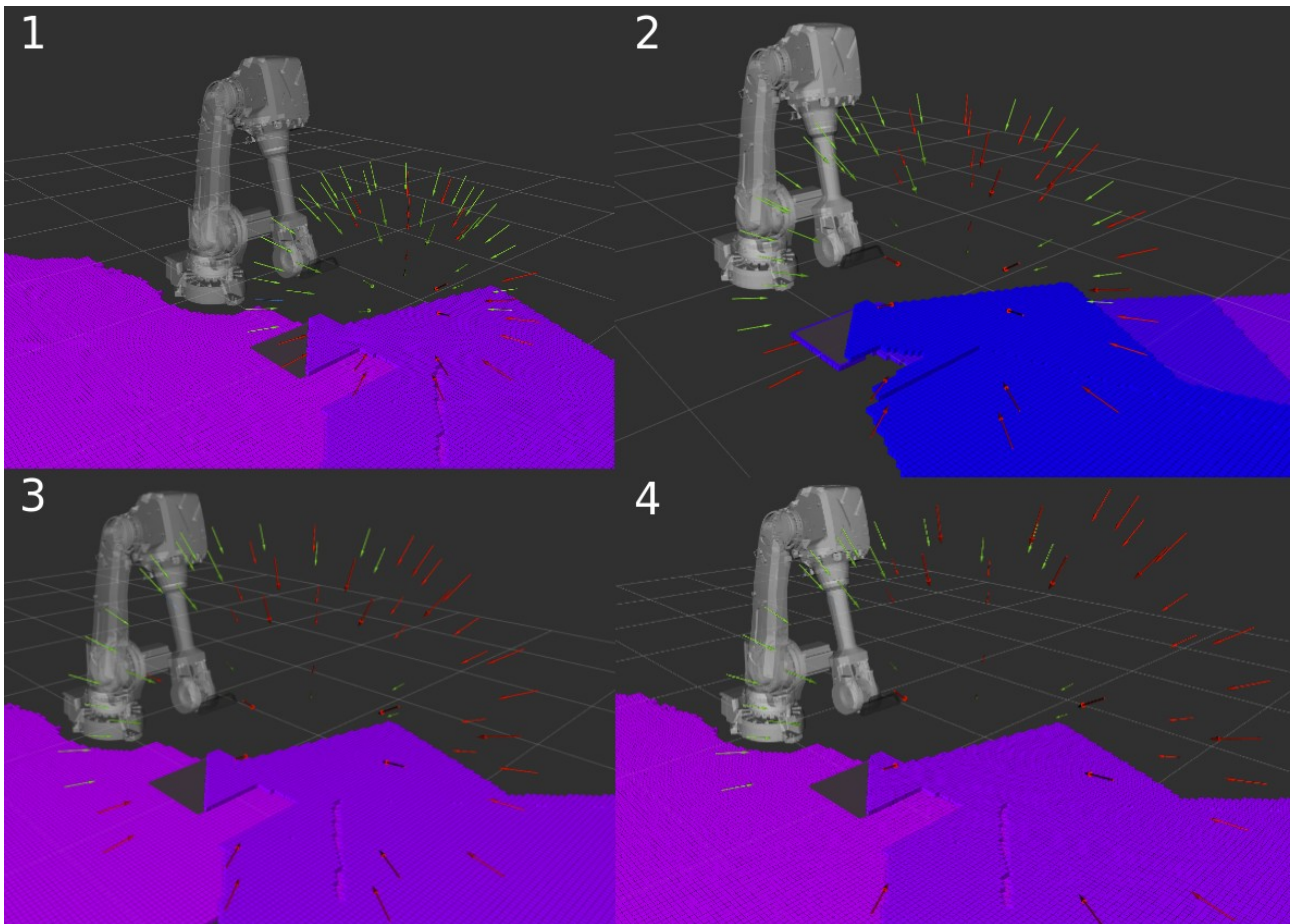


Obr. 20: Vývojový diagram

3.4 Parametre metódy

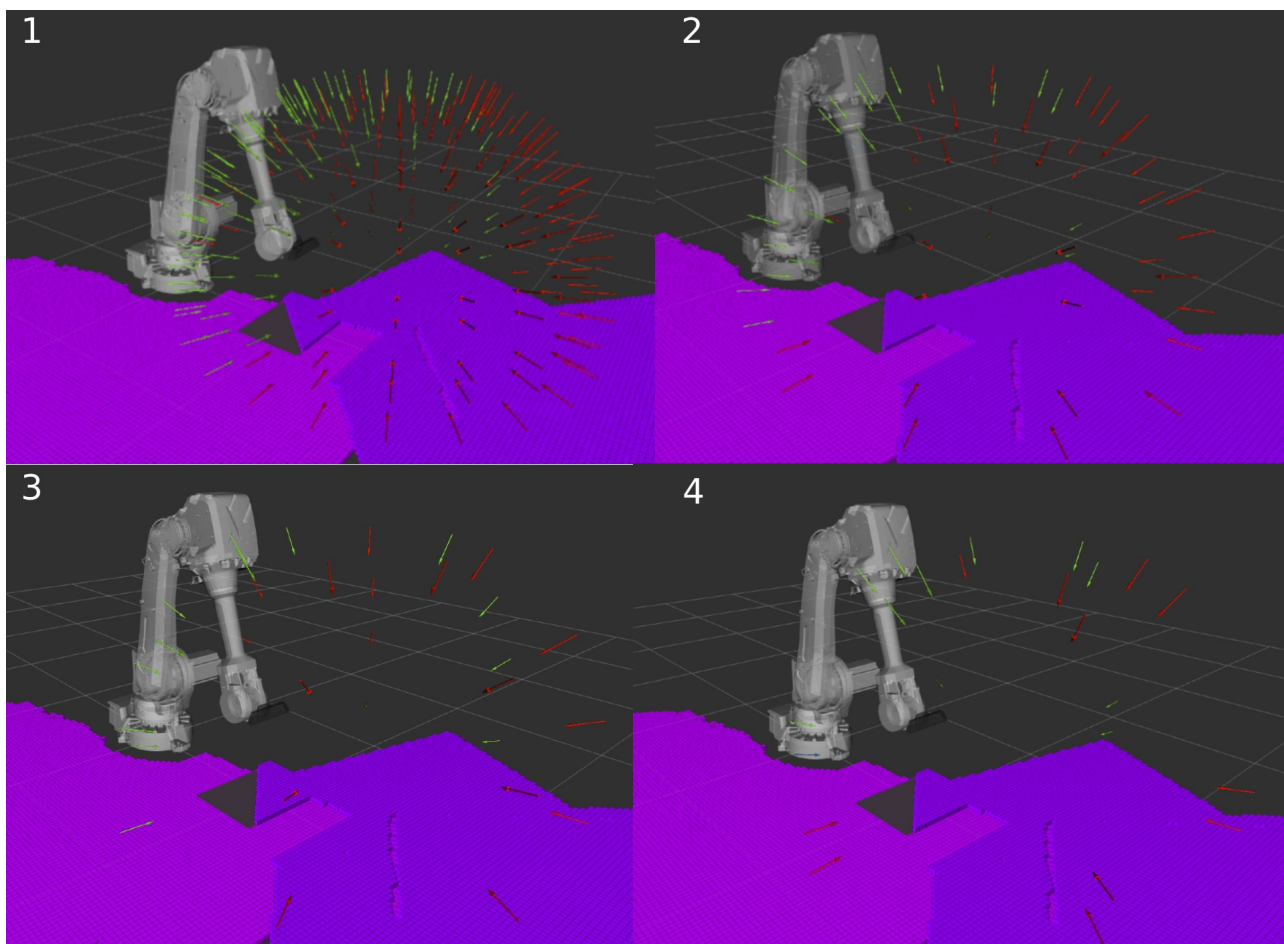
Program umožňuje pri spustení zadať niektoré parametre skenovania. Do týchto parametrov patria:

- Polomer - vzdialenosť vygenerovaných pohľadov od stredu objektu. Pre tento parameter je potrebné vedieť približne rozmery objektu, lebo pri príliš malom polomere môže nastať kolízia s objektom a príliš veľký polomer zmenší presnosť.



Obr. 21: **Generovanie pohľadov pre rôzne polomery.** (1) $R = 0,7$. (2) $R = 0,9$. (3) $R = 1,1$.
(4) $R = 1,3$

- Uhol - uhol medzi pohľadmi. Uhol ovplyvňuje počet pohľadov. Pri menšom počte pohľadov rýchlejšie prechádza evaluácia, ale môže vzniknúť situácia, keď nebude dost' pohľadov na vytvorenie mapy.



Obr. 22: **Generovanie pohľadov pre rôzne uhly medzi pohľadmi.** (1) $\angle = 15^\circ$. (2) $\angle = 30^\circ$.
(3) $\angle = 45^\circ$. (4) $\angle = 60^\circ$.

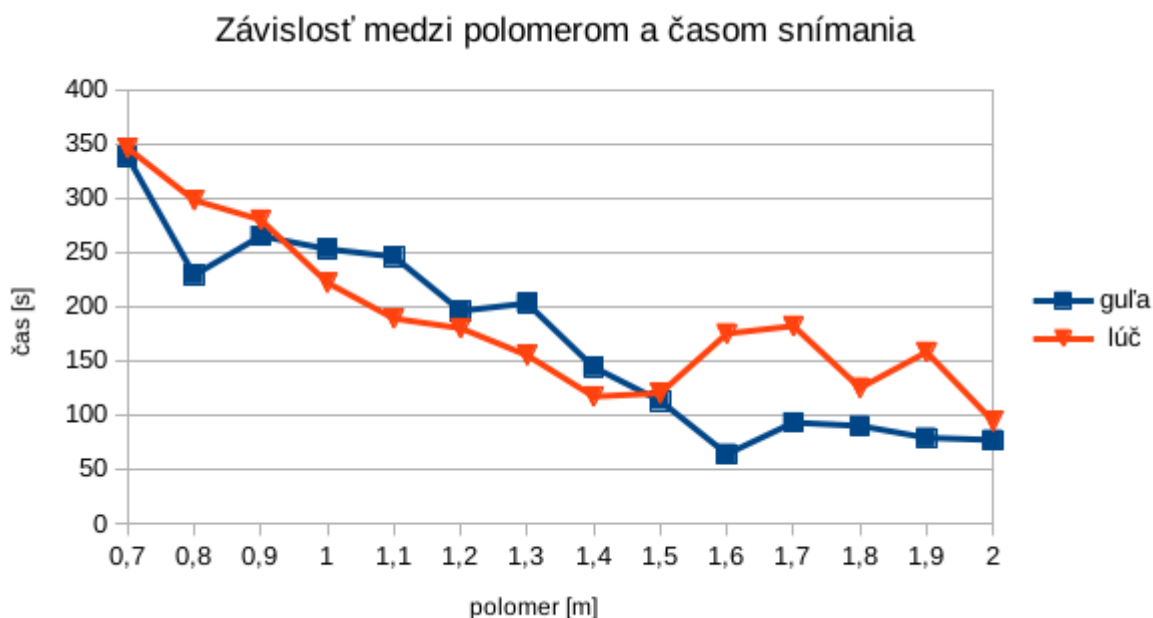
- Metóda evaluácii pohľadu
- Váha vzdialenosti v finálnom hodnotení
- Váha množstva novej informácie v finálnom hodnotení

4 Simulácia metódy volumetrickeho najlepšieho nasledujúceho pohľadu

Pomocou simulácie je možné zistiť, ako efektívna metóda a ako môžu vstupné parametre ovplyvniť konečné výsledky skenovania. Za týmto účelom sa v sérii simulácií s rôznymi parametrami meria ich vplyv. Ako môže polomer od stredu skenovaného objektu, uhol medzi pohľadmi a vaha vzdialenosti k ďalšiemu pohľadu pri hodnotení ovplyvniť čas skenovania, cestu prejdenu kamerou a potrebný počet pohľadov, aby bolo možné naskenovať 80% priestoru.

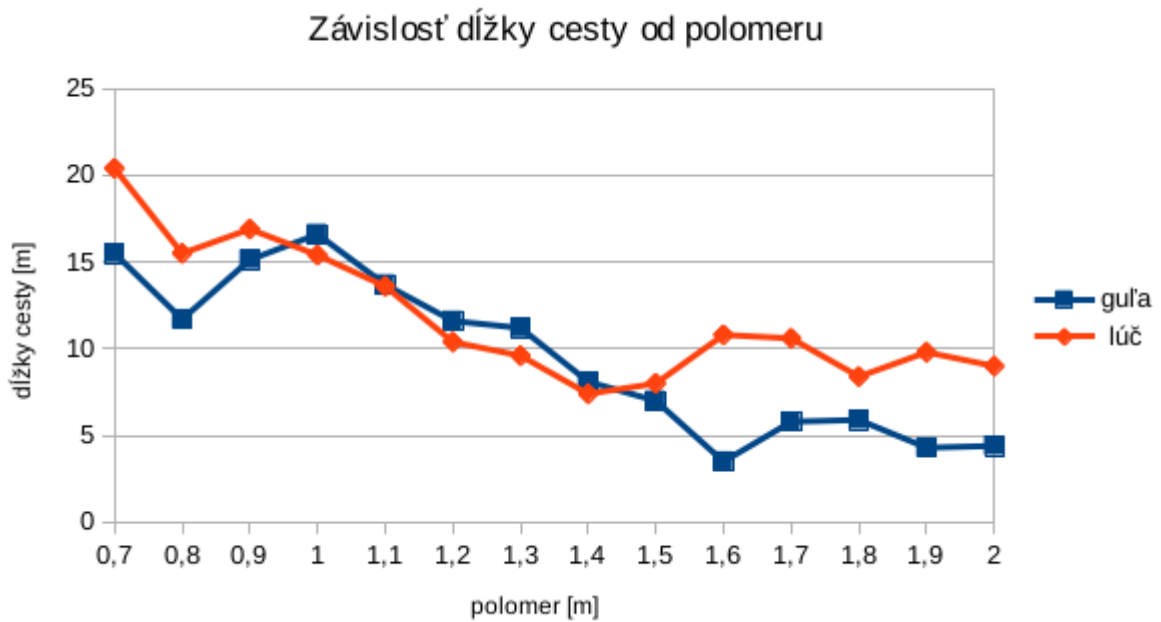
4.1 Simulácia pre rôzne polomery

Pohľady generované na gule s centrom v približnom strede objektu. V prvom experimente úlohou je zistiť, ako polomer gule ovplyvní čas, dĺžku cesty a počet potrebných pohľadov.



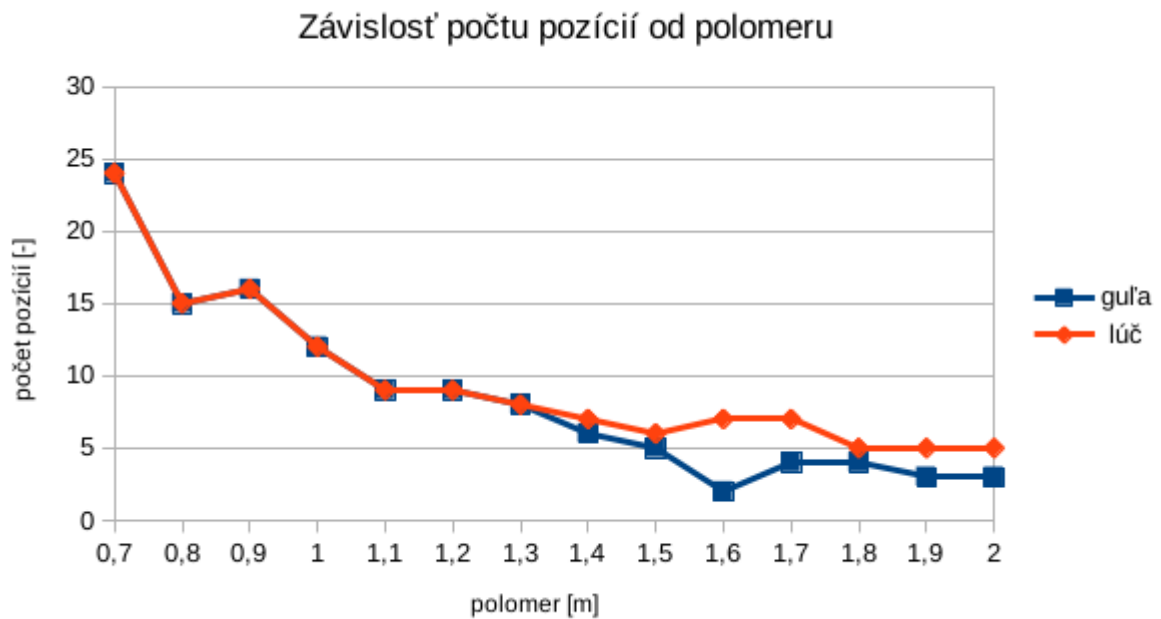
Obr. 23: Porovnanie závislosti času od polomeru pri rôznych metódach evaluácií

Väčší polomer umožňuje zmenšiť potrebný čas, lebo z každého pohľadu je viditeľná väčšia plocha



Obr. 24: Porovnanie závislosti dĺžky cesty od polomeru pri rôznych metódach evaluácií

Väčší polomer umožňuje zmenšiť dĺžku cesty, ktorou prechádza sensor, lebo pri väčšom polomere nemusí ohýbať objekt

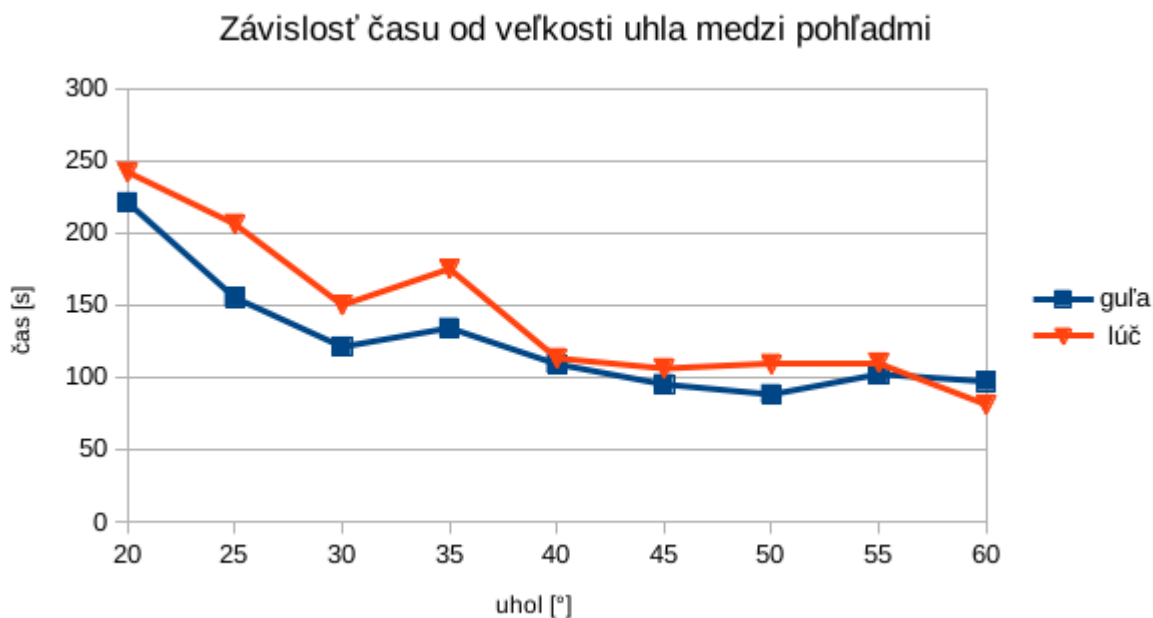


Obr. 25: Porovnanie závislosti počtu pozícií od polomeru pri rôznych metódach evaluácií

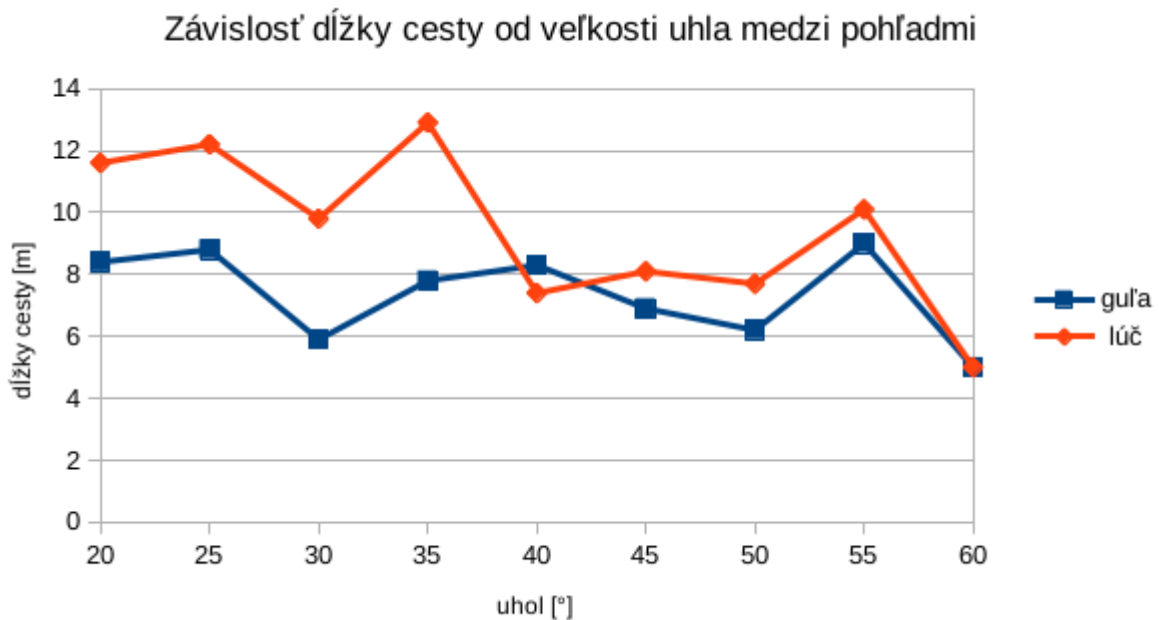
Väčší polomer znižuje počet pohľadov, lebo z jedného skenovania mapa dostane viac informácií

4.2 Simulácia pre rôzne uhly medzi pohľadmi

V druhom experimente úlohou je zistiť, ako ovplyvní simuláciu nastavenie veľkosti uhlov medzi pohľadmi aj tým zmeniť a počet pohľadov.

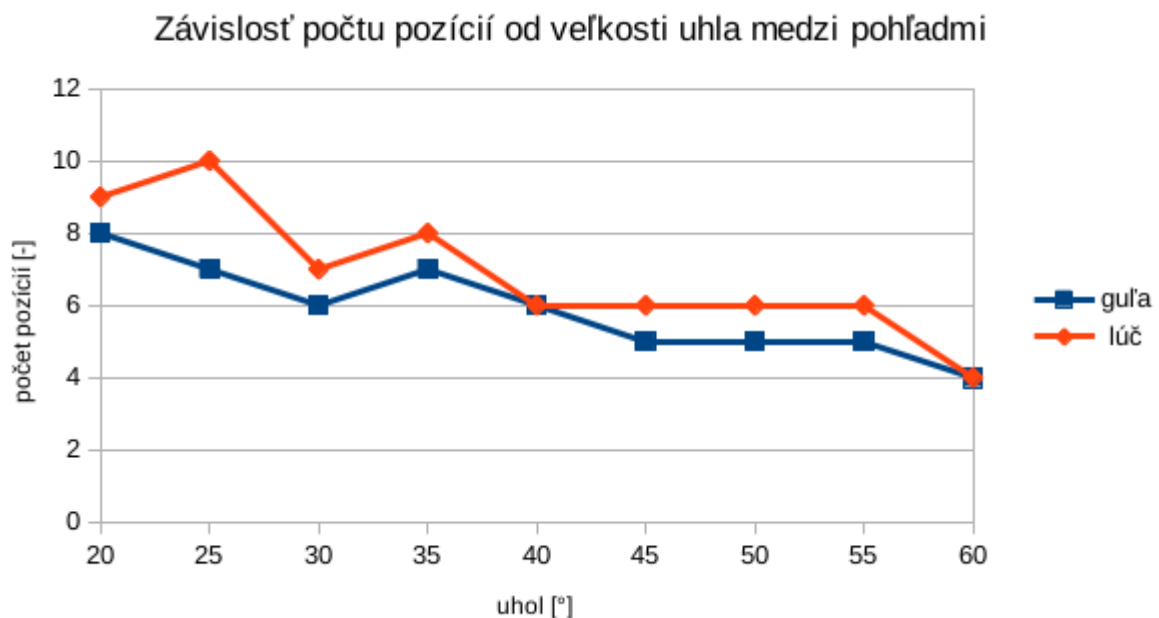


Obr. 26: Porovnanie závislosti času od uhla medzi pohľadmi pri rôznych metódach evaluácií
S menším počtom pohľadov je menší aj čas potrebný na evaluáciu.



Obr. 27: Porovnanie závislosti dĺžky cesty od uhla medzi pohľadmi pri rôznych metódach evaluácií

Relatívne slabo ovplyvňuje dĺžku cesty, lebo s menším výberom polôh a väčšou vzdialnosťou medzi nimi niekedy musí prejsť väčšom vzdialenosť, ako pri väčšom počte pohľadov

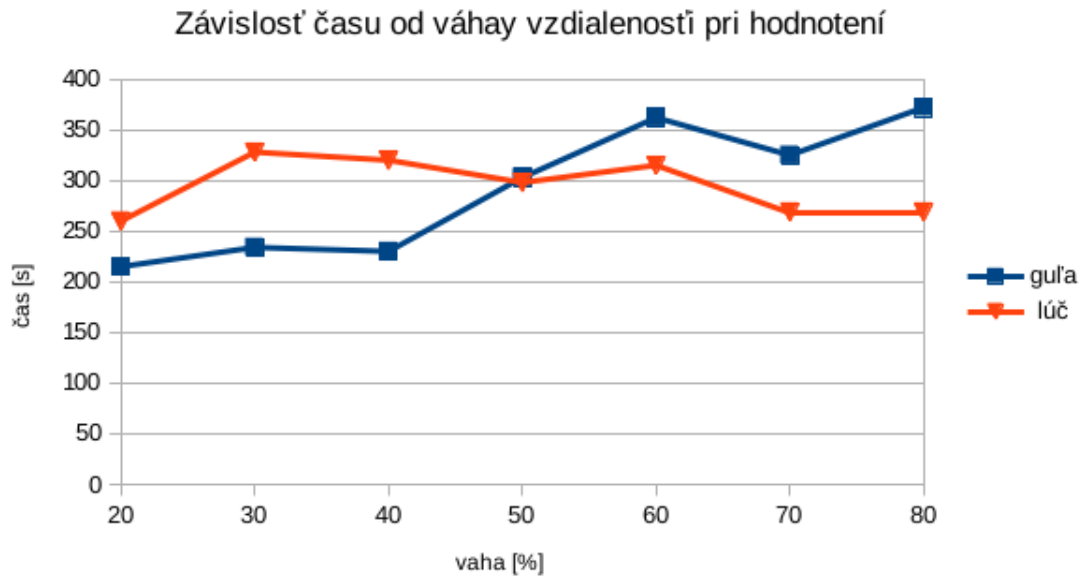


Obr. 28: Porovnanie závislosti počtu pozícií od uhla medzi pohľadmi pri rôznych metódach evaluácií

Pri väčšom uhle medzi pohľadmi používa menej polôh, lebo má menší výber, ale môže vzniknúť situácia kedy robot prejde všetky možné polohy, ale bude mať neúplnú mapu objektu

4.3 Simulácia pre rôzne váhy vzdialenosti k ďalšiemu pohľadu

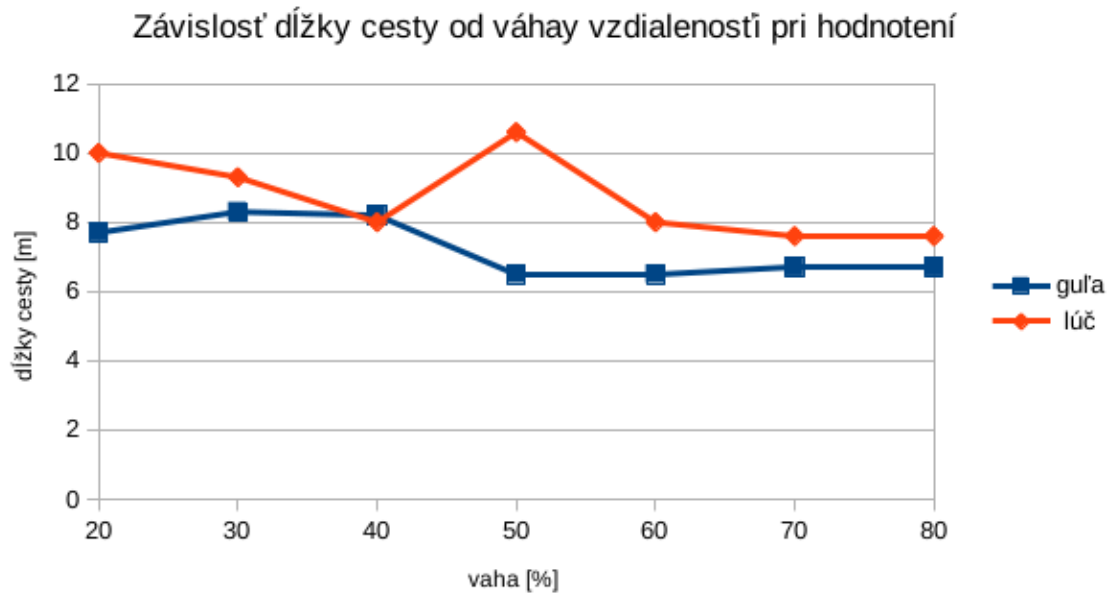
V treťom experimente úlohou je zistiť vplyv toho, čo bude preferované pri výbere nasledujúceho pohľadu množstvo novej informácie alebo najkratšia cesta. V grafoch dole na osi X označená váha



Obr. 29: Porovnanie závislosti času od váhy vzdialenosti k ďalšiemu pohľadu pri rôznych metódach evaluácií

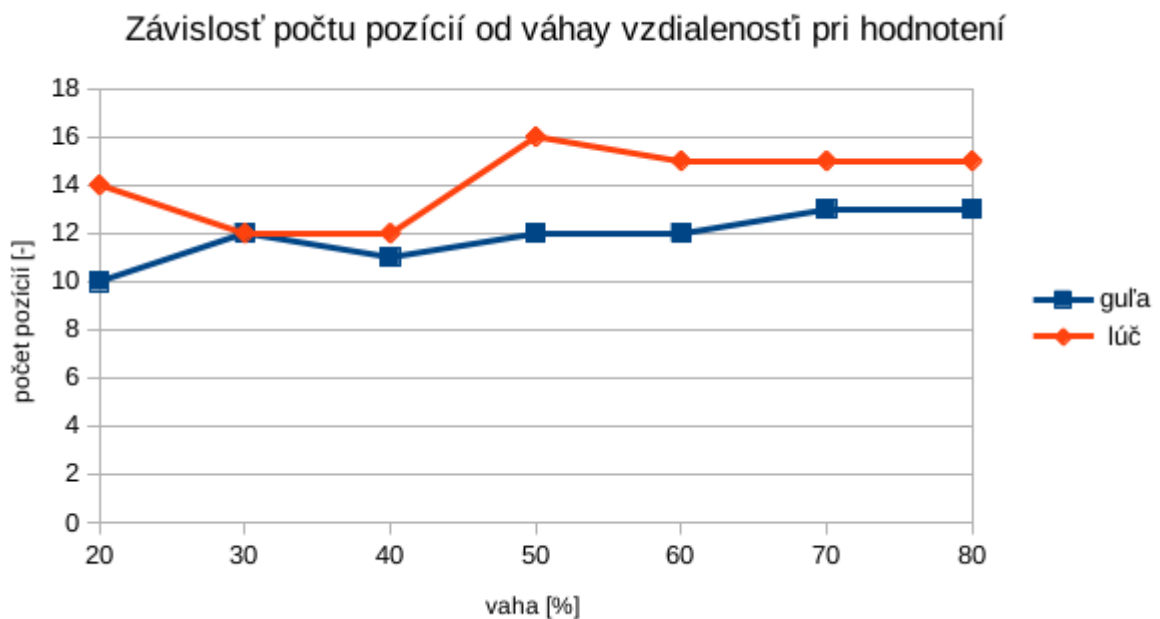
vzdialenosti pri evaluácii

Väčšia váha vzdialenosti zväčší čas, lebo ak možné vidia na Obr. 19 počet potrebných pohľadov tiež zväčší



Obr. 30: Porovnanie závislosti dĺžky cesty od váhy vzdialenosti k ďalšiemu pohľadu pri rôznych metódach evaluácií

Pokles dĺžky cesty je minimálny (v niektorých prípadoch dĺžka cesty môže aj zväčšiť). Toto spôsobené tým, že preferuje najbližší pohľad, ktorá nie vždy najoptimálnejší, preto musí prejsť viac pohľadov.



Obr. 31: Porovnanie závislosti počtu pozícií od váhy vzdialenosti k ďalšiemu pohľadu pri rôznych metódach evaluácií

4.4 Vyhodnotenie výsledkov

Experimenty ukázali, že metódy na nájdenie najlepšieho nasledujúceho pohľadu môžu významne ovplyvniť proces mapovania. Najmä volumetrická metóda, ktorá bola testovaná v kapitole 4, ukázala, že správnym výberom parametrov je možné znížiť čas a dĺžku cesty skenovania viac ako dvakrát. Napríklad so zväčšením polomeru v guli, na ktorej sa vytvárajú pohľady, je možné znížiť čas a dĺžku cesty, pretože manipulátor sa nemusí ohýbať okolo prekážok. Ale pri príliš veľkom polomere, môže stratiť presnosť. V experimente 4.3, kde sa oceňovalo nielen množstvo nových informácií, ale aj vzdialenosť od pohľadu, je možné vidieť závislosť váhy vzdialenosti pri ocenovaní a čase. Napriek tomu, že zmenšenie dĺžky cesty bolo malé, čas sa zvýšil o 1, 5-krát. Pri experimentovaní s rôznym počtom vygenerovaných pohľadov je možné skrátiť čas pri menšom počte pohľadov, ale pri nedostatočnom počte môže nastať situácia, keď už budú všetky pohľady prekonané, ale mapa ešte nie je plná. Na základe výsledkov experimentov je možné povedať, že algoritmus zvládol úlohu hľadania najlepšieho nasledujúceho pohľadu

Záver

Cieľom tejto práce je naštudovať problematiku o určení nasledujúceho najlepšieho pohľadu na scénu s robotom a zanalyzovať dostupné algoritmy so zameraním na optimalizáciu počtu polôh robota pri mapovaní prostredia.

V prvej kapitole bola popísaná problematika určovania najlepšieho nasledujúceho pohľadu. Kapitola pozostávala z opisu typov máp prostredia, spôsoby mapovanie prostredia a toho aké algoritmy určovania najlepšieho nasledujúceho pohľadu existujú. V druhej Kapitole boli opísané nástroje použité na testovanie algoritmov view planning: manipulátor, kamera softvér a ich implementácia. Tretia kapitola obsahuje implementáciu volumetrickej metódy a experimenty účelom, ktorých je overenie účinnosti metódy pri použití rôznych parametrov. V štvrtej kapitole sú hodnotenie výsledkov experimentov.

V tejto práci bolo ukázané, ako použiť ROS na zostavenie mapy prostredia. Bol popísaný postup, ako vytvoriť simulačné prostredie a nastaviť riadenie robota v ňom. Bolo ukázané, že rýchlosť mapovania prostredia závisí od nastavenia parametrov algoritmu.

Táto práca ukázala, že algoritmy určovania najlepšieho nasledujúceho pohľadu (View Planning) zvládajú úlohu minimalizovať počet presunov robota a čas potrebný na zmapovanie neznámeho prostredia.

Použitá literatúra

- 1: Ing. Roman Murár prof. Ing. Ladislav Jurišica, Reprezentácia prostrediamobilného robotického systému, 2005
- 2: Ing. Roman Murár prof. Ing. Ladislav Jurišica, Prieskum neznámeho prostredia mobilným robotickým systémom a tvorba topologickej mapy prostredia, 2004
- 3: Ing. Roman Murár prof. Ing. Ladislav Jurišica, Reprezentácia prostrediamobilného robotického systému(2), 2005
- 4: R. B. Rusu and S. Cousins, 3D is here: Point Cloud Library (PCL), 2011
- 5: Kai M. Wurm, Armin Hornung, Maren Bennewitz, Cyrill Stachniss, Wolfram Burgard, OctoMap: A Probabilistic, Flexible, and Compact 3D Map Representation for Robotic Systems, 2010
- 6: Scott, William & Roth, Gerhard & Rivest, Jean-François, View planning for automated three-dimensional object reconstruction and inspection, 2003
- 7: Jasna Maver and Ruzena Bajcsy, Occlusions as a Guide for Planning the Next View, April 1991
- 8: J. Irving Vasquez-Gomez, L. Enrique Sucar, Rafael Murrieta-Cid, Efrain Lopez-DamianEfrain Lopez-Damian, Volumetric Next-best-view Planning for 3D Object Reconstruction with Positioning Error, 2014
- 9: Tsardoulis, Emmanouil & Iliakopoulou, Katerina & Kargakos, Andreas & Petrou, Loukas., A Review of Global Path Planning Methods for Occupancy Grid Maps Regardless of Obstacle Density, 2016
- 10: Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, ROS: an open-source Robot Operating System, 2009
- 11: Nathan Koenig and Andrew Howard, Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator, 2004
- 12: David Coleman, Ioan A. Şucan, Sachin Chitta, Nikolaus Correll, Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt!, 2014
- 13: Serdar Kucuk and Zafer Bingul , Robot Kinematics: Forward and Inverse Kinematics, 2006
- 14: J. Han, L. Shao, D. Xu and J. Shotton, Enhanced Computer Vision With Microsoft Kinect Sensor: A Review, 2013