

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

MODUL PRO LOKALIZACI DVEŘÍ A KLIKY PRO PR2

BAKALÁŘSKÁ PRÁCE

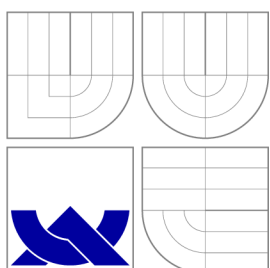
BACHELOR'S THESIS

AUTOR PRÁCE

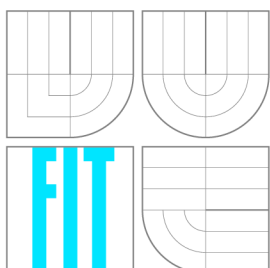
AUTHOR

ROLAND BOTKA

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

MODUL PRO LOKALIZACI DVEŘÍ A KLIKY PRO PR2

DOORS AND DOOR HANDLE LOCALIZATION FOR PR2

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ROLAND BOTKA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MICHAL KAPINUS

BRNO 2015

Abstrakt

Cieľom tejto práce je navrhnúť a vytvoriť modul pre lokalizovanie dverí a kľučky pre robotickú platformu PR2. Hľadanie dverí by malo prebiehať samostatne, čo znamená, že robot nájde dvere v miestnosti bez pomoci. Pozícia kľučky je uložená v súbore, z ktorého sa načítajú informácie na základe AR kódu nalepeného na dverách. Výsledkom tohoto modulu by mal byť program pre Robotický operačný systém, ktorý lokalizuje dvere v miestnosti a získa o nich informácie. Tieto informácie sú základom pre prácu ďalšieho modulu, ktorý otvára dané dvere. Výsledné testovanie prebehlo vo fakultnom robotickom laboratóriu.

Abstract

The aim of this work is to design and create a module for locating doors and handles for the robotic platform PR2. Finding the doors should be done separately, which means that the robot finds a door in the room without assistance. Handle position is stored in a file which the data are read from based on the AR code affixed to the door. The result of this module should be a program for the Robotic operating system that will locate the door in the room, and give information about them. These informations will be the basis for the work of another module, which open the door. The final test took place in a faculty robotic laboratory.

Klíčová slova

ROS, PR2, Gazebo, robot, Kinect, AR kód, OpenCV, PCL, Lokalizácia dverí, robotický operačný systém.

Keywords

ROS, PR2, Gazebo, robot, Kinect, AR code, OpenCV, PCL, Door localization, robotic operating system.

Citace

Roland Botka: Modul pro lokalizaci dverí a kliky pro PR2, bakalářská práce, Brno, FIT VUT v Brně, 2015

Modul pro lokalizaci dveří a kliky pro PR2

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Michala Kapinusa.

.....

Roland Botka
20. května 2015

Poděkování

Ďakujem vedúcemu práce, Ing. Michalovi Kapinusovi, za pomoc, cenné rady a venovaný čas, čo mi pomohlo pri riešení problémov spojených s touto prácou.

Ďalej by som chcel poďakovať Danielovi Senčuchovi za spoluprácu, ktorá dopomohla k prepojeniu dvoch odlišných modulov pre jeden spoločný cieľ.

© Roland Botka, 2015.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	2
2 Teória	3
2.1 Robotický operačný systém(ROS)	3
2.2 Personal Robot 2	4
2.3 Kódovanie informácií	8
2.4 Knížnice	9
3 Návrh riešenia	10
3.1 Popis problému	10
3.2 Koncept riešenia	12
3.3 Existujúce riešenie	13
4 Implementácia	14
4.1 Balík find_door	14
4.2 Balík detect_door	16
4.3 Formáty správ	18
5 Testovanie	20
5.1 Experimenty	20
5.2 Vyhodnotenie experimentu	25
5.3 Námety k rozšíreniu práce	25
6 Záver	27
A Obsah CD	29

Kapitola 1

Úvod

Od pradávna sa ľudia snažia uľahčiť si život a to v mnohých smeroch. Servisné roboty sú jedným z veľa spôsobov ako to docieľiť. Už dnes existuje veľa činností, kde roboty pomáhajú ľuďom, avšak v domácnosti je to stále ojedinelé. Cieľom tejto práce je vytvorenie modulu pre lokalizáciu dverí a kľučky pre robota PR2. V práci sa zameriam hlavne na čo najpresnejšiu lokalizáciu kľučky, pri prípadnom ďalšom použití údajov pre modul otvárania dverí. Aby bola práca robota samostatná, veľmi dôležité je nájdenie dverí v priestore miestnosti. Na tieto 2 hlavné problémy je zameraná táto práca.

Takéto problémy už boli riešené v minulosti, avšak výsledný produkt pre platformu PR2 a Robotický operačný systém ROS nebol vyriešený dostatočne. Na túto platformu existoval modul pre detekciu a lokalizáciu, avšak ten bol dostupný iba na staršie verzie robotického operačného systému. Mnou písaný modul bol vytvorený pre najaktuálnejšiu distribúciu pre Ubuntu 12.04 Hydro Medusa, ktorá je dostupná na Fakulte informačných technológií v Brne na platforme PR2. Všetky podrobné informácie o Robotickom operačnom systéme ale aj platforme PR2 a jej možnostiach sa nachádzajú v sekcii 2.1 a sekcii 2.2.

Výsledkom môjho modulu sú získané informácie o dverách na základe ktorých prebieha otváranie dverí. Mnou získané informácie o lokalizácii kľučky, sú veľmi dôležité pre následné otvorenie dverí modulom, ktorým sa zaoberá práca Daniela Senčucha [1]. Môj modul spolu s modulom pre otváranie dverí by mohol pomôcť starším alebo menej mobilným ľuďom pri uľahčení života. Robot sa tak pohodlne a bez akýchkoľvek problémov dostane do druhej miestnosti.

Nasledujúca kapitola 2 obsahuje teoretické informácie o Robotickom operačnom systéme, robotovi PR2 a taktiež popíšem všetky potrebné balíky, využité v tejto práci. Ďalšia kapitola 3 popisuje návrh môjho riešenia, graficky ukáže všetky potrebné informácie o dverách a taktiež pozíciu, kde má byť nalepený AR kód. V kapitole 4 sa nachádzajú informácie o implementácii mnou vytvorených balíkov. Samotné testovanie sa nachádza v kapitole 5. Tiež sa tu nachádza popis jednotlivých testov a návrh možnosti rozšírenia tejto práce.

Kapitola 2

Teória

V tejto časti sa nachádzajú teoretické informácie, ktoré bolo potrebné naštudovať. Sú tu informácie o operačnom systéme pre prácu s robotmi(ROS). Ďalej o senzore Kinect, ktorý slúži na zachytávanie a získavanie potrebných informácií z okolia, taktiež o knižnici OpenCV pre spracovanie obrazu. Na záver sa tu nachádzajú informácie o AR kódach.

2.1 Robotický operačný systém(ROS)

The Robot Operating System(ďalej ROS) je flexibilný framework pre návrh a implementáciu softwaru pre robotov.[3] Je to súbor nástrojov, knižníc a konvencií, ktoré majú zjednodušiť vytváranie komplexných platforiem pre robotov. Keďže vytváranie funkčných programov je zložité, využívanie ROS frameworku šetrí čas a taktiež zjednodušuje implementáciu. ROS je veľký projekt, na ktorom pracovalo značné množstvo ľudí. Na Stanfordskej Univerzite v roku 2000 začali s vývojom predchodcu ROS a v roku 2007 bolo vytvorené novšie a lepšie otestované prostredie. Tomuto pomohlo veľa výskumníkov, ktorí prispeli svojimi odbornými znalosťami do ROS frameworku. Po celú dobu bol software vyvíjaný pod BSD open-source licenciou. Vďaka tomuto sa nachádza veľa návodov a tutoriálov na wiki stránkach ROSu [4]. ROS bol vyvíjaný v niekoľkých inštitúciách pre niekoľko robotov. V súčasnosti môže každá inštitúcia zdieľať svoje ROS kódy na svojom serveri, kde môžu byť voľne dostupné. ROS systém dnes pozostáva z desiatok tisíc užívateľov po celom svete, ktorí pracujú v oblastiach od hobby projektov až po veľké priemyslené automatizačné systémy.

Pri programovaní je možné jednotlivé úlohy rozdeliť do uzlov (nodes), ktoré medzi sebou komunikujú pomocou správ(messages) a služieb(services).

ROS je vydávaný vo forme distribúcií. Keďže na platforme PR2 je nainštalovaný operačný systém Ubuntu 12.04, na ktorom je podporovaná distribúcia Hydro, rozhodol som sa ju využiť v mojej práci. Distribúcia Hydro je najnovšia dostupná distribúcia. Pre túto distribúciu som sa rozhodol aj preto, že podporuje veľa knižníc a balíkov.

Uzly

Uzol (anglicky Nodes)[2] je v ROS nazvaný proces, ktorý vykonáva výpočet. Uzly sú spojené v grafe a komunikujú medzi sebou pomocou správ. Aplikácie sa môžu skladať z niekoľkých uzlov.

Výhoda tejto architektúry je v znovupoužitelnosti jednotlivých uzlov. Každý uzol riadi jednu úlohu (plánovanie cesty, spracovanie obrazu z kamery a iné). Je možné nahradiť existujúci, napríklad zastaralý, uzol novším, ktorý využíva rovnaké správy.

Ďalšou výhodou je spracovanie chýb. Pri vzniknutí chyby v aplikácii sa táto chyba nachádza typicky v jednom uzle. Ten môže byť reštartovaný bez nutnosti reštartovania celého systému a tak môže aplikácia pokračovať vo vykonávaní procesov.

Správy

Správy (anglicky Messages)[7] slúžia na komunikáciu jednotlivých uzlov. Správa je jednoduchá dátová štruktúra, štandardne jednoduchých typov (integer, boolean, floating point a iné). Správy môžu mať neobmedzené množstvo odoberateľov a prispievateľov. Využívajú sa napríklad pre predanie obrazu z kamery do uzlu, ktorý spracúvajú.

Služby

Služba[9] je spôsob komunikácie medzi uzlami na princípe RPC¹. Uzol, ktorý implementuje službu, čaká kým ho zavolá iný uzol, ktorý požaduje vykonanie služby. Dáta sa neprenášajú neustále, ale len keď je to vyžiadané klientom. Klient volá službu odoslaním správy s požiadavkou a čaká na odpoveď.

2.2 Personal Robot 2

Výsledok tejto práce bude zameraný na lokalizovanie dverí na platforme PR2(Personal Robot 2) od Willow Garage. Tento robot je veľkosťou podobný človeku.

Platforma PR2 a jej aplikácie majú za úlohu pomôcť ľuďom zvýšiť ich produktivitu v práci, ale aj v domácnosti. Vďaka robotickému operačnému systému je tento robot použiteľný na veľa spôsobov. Platforma PR2 podporuje viac ako 1000 knižníc, ktoré sú voľne prístupné. PR2 je otvorená platforma, ktorú je možné upravovať akokoľvek pre svoje účely. Manipuláciu predmetov robotovi umožňujú dve ramená. Ramená majú niekoľko kĺbov, čo robotovi umožňuje maipulovať s rôznodruhými objektmi. Na konci ramena je zverák (gripper). Ten môže mať v sebe senzor tlaku, vďaka čomu robot môže uchopovať aj krehké objekty. Pohyblivosť robota zabezpečuje základňa s kolesami a teleskopické torzo. Zo všetkých senzorov sa dajú ľahko získať potrebné informácie pomocou už existujúcich knižníc.

Na PR2[8] sa nachádzajú 2 počítače s 8-jadrovými i7 procesormi, 24GB ram pamäť a 2TB HDD pre ukladanie dát. Robot má 2 ramená so zaťažením 1.8 kg. Na každom ramene sa nachádza kamera pre pohľad na prácu s jednotlivými ramenami. Hlavu robota je možné otáčať v rozsahu do 350° a umožňuje ju nakláňať do 115°. K nej je pripojený senzor Kinect. Pohyb robota zabezpečujú kolesá, pričom maximálna rýchlosť pohybu je 1 m/s. Na PR2 sa nachádzajú senzory:

- 5-megapixelová kamera
- Kinect
- Dva laserové snímače
- Dva páry stereo kamier
- Kamery v ramenách robota
- Senzory tlaku

¹Remote procedure call - vzdialené volanie procedúr

Na platforme sa nachádza gigabitový ethernet, cez ktorý sa pracuje so všetkými kamerami. Ďalej je tu dual-radio WiFi a prístupový bod Bluetooth. Napájanie platformy je zabezpečené pomocou 1.3 kWh batérie, ktorá vydrží v pohotovosti dve hodiny, avšak robot môže pracovať aj pri napájaní z elektrickej siete.



Obrázek 2.1: Platforma PR2 na našej fakulte

Kinect

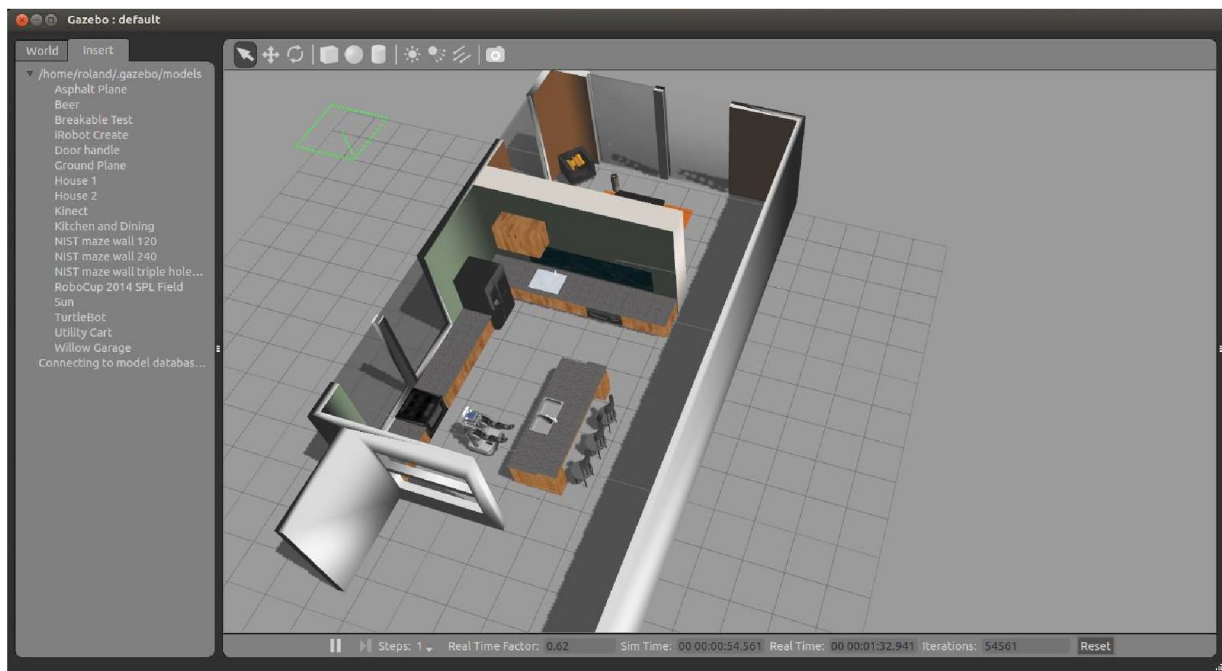
Kinect [6] je vstupné zariadenie, so senzorom pohybu, Microsoft-u, určený na ovládanie hernej konzoly XBOX 360 a hranie hier. Umožňuje užívateľom ovládať a pracovať s konzolou cez gestá. Zariadenie sa skladá z RGB kamery, hĺbkového senzoru a mikrofónu. RGB video používa 8-bitové rozlíšenie VGA(640x480 pixelov) s farebným filtrom Bayer, avšak toto rozlíšenie je možné hardwarovo vylepšiť na 1280x1024 pixelov.



Obrázek 2.2: Senzor kinect²

Gazebo

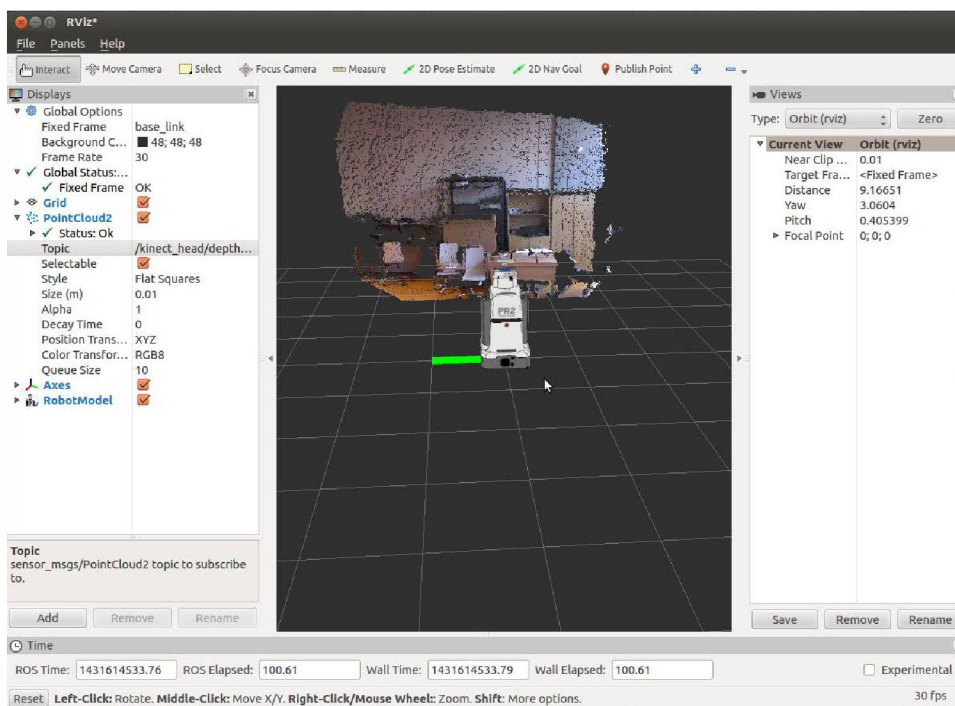
Pri vytváraní aplikácií pre robotov je dôležité pracovať so simulátorom. Aplikácie sa nemusia testovať priamo na fyzickom stroji, ktorý by sa mohol poškodiť, ale testujú sa za pomoci simulátora. Gazebo [5] je simulačný program, ktorý umožňuje pracovať s niekoľkými robotmi. Umožňuje vytvoriť vnútorné, ale i vonkajšie prostredie a testovať v ňom správanie robotov.



²Prevzaté z: <http://wiki.ipisoft.com/File:Kinect-small.jpg>

Rviz

Rviz je balíček ROS pre vizualizáciu dát zo senzorov robota. Služi na zobrazenie rôznych dát napríklad body z kamier kinectu, obraz z rôznych kamier, hĺbkový obraz, vytvorená mapa z odometrického senzora a iné. Zobrazí sa dajú buď hodnoty nahrané alebo reálne z kamier a senzorov. V Rviz-e sa dá pracovať aj so zadávaním cieľových pozícií robota (knížnica navigation).



Obrázek 2.3: Prostredie Rviz

2.3 Kódovanie informácií

S kódovaním informácií sa môžeme stretnúť skoro všade, avšak kód s ktorým budem pracovať vychádza práve z čiarového kódu. Tento kód existuje už viac ako 60 rokov. Čiarový kód pozostáva z čiar, ktoré majú rôznu šírku a veľkosť. Tieto čiary potom reprezentujú dáta, ktoré sú načítané a rozpoznané skenerom[11]. Dáta v tomto kóde môžu reprezentovať číslo výrobku, číslo výrobcu, miesto uloženia v sklade, ale aj rôzne iné potrebné informácie. Čiarové kódy sú používané dodnes a to najmä pre ich presnosť, rýchlosť a flexibilitu.

Existujú rôzne typy čiarových kódov, z ktorých má každý svoju charakteristiku(EAN, Kód 128, PDF 417 a iné)[14]. Široké uplatnenie nachádzajú aj takzvané 2D kódy(dvojdimenzionálne), ktoré rozširujú údajovú kapacitu štandardných kódov.



Obrázek 2.4: EAN kód³

QR kód

QR kód je špecifický kód, ktorý patrí k skupine 2D kódov. V súčasnosti je používaný veľmi často a okrem výhody vysokej kapacity je totiž snímateľný aj bežným telefónom. Táto jeho schopnosť sa už využíva pre rýchly prístup mobilom na internetové stránky, ale aj v reklame[14].



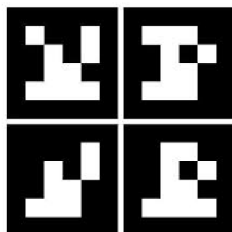
Obrázek 2.5: QR kód⁴

³Prevzaté z: <http://www.attentionmedia.se/grafisk-layout/grafiskt-layout-ean-kod.html>

⁴Prevzaté z: <http://www.kodys.sk/stranka/trochu-teorie-o-ciarovom-kode>

AR kód

AR kód je iná variant 2D kódovania využívaná pre rozšírenú realitu. AR (augmented reality-rozšírená realita) je označenie používané pre reálny obraz sveta doplnený počítačom vytvorenými objektami. Je to v podstate zobrazenie reality. AR kód slúži na interakciu počítačových zariadení s reálnymi objektami.



Obrázek 2.6: Ukážka niekoľkých AR kódov

2.4 Knižnice

V nasledujúcej časti sa nachádza popis jednotlivých knižníc, ktoré budem vo svojej práci využívať.

PCL

PCL (Point Cloud Library) je samostatný otvorený projekt, pre 2D alebo 3D obraz a bodové spracovanie obrazu[13]. PCL je vydaná pod BSD licenciou. Obsahuje rôzne algoritmy, ako napríklad algoritmy pre filtrovanie, odhad funkcie, rekonštrukcia povrchu alebo segmentácia. Všetky tieto funkcie môžu byť použité napríklad pre filtrovanie ľubovolnej hodnoty od zašumených dát, spájanie 3D mračien bodov dohromady, segmentovanie príslušnej scény alebo vypočítanie deskriptorov pre rozpoznanie na základe ich geometrických tvarov a iné. PCL vyvíja veľké množstvo inžinierov a vedcov z veľa organizácií po celom svete. Point cloud je dátová štruktúra, reprezentujúca kolekciu multi-dimenzionálnych bodov, ktorá sa bežne používa na reprezentáciu trojrozmerných dát. V 3D bodoch sa body reprezentujú tromi geometrickými súradnicami x , y a z . Tieto mračná bodov možno získať zo senzorov, ako sú napríklad stereo kamery, 3D skenery alebo môžu byť vytvorené počítačom. Najznámejším takýmto zariadením je Kinect od Microsoft-u.

Navigation stack

Táto knižnica slúži na navigáciu PR2 po miestnosti. Je založená na 2D a používa informácie z odometrických senzorov, na základe ktorých vytvorí bezpečnú trasu do cieľovej pozícií.[12]

Ar track alvar

Táto knižnica bola vytvorená na generovanie a rozpoznávanie AR kódov. Pre rozpoznávanie sa využíva mračno bodov, z ktorého je možné získať aj vzdialenosť ku kódu ako aj jeho hodnota[10].

Kapitola 3

Návrh riešenia

Návrh riešenia tohoto modulu je skutočne dôležitý. Pri nesprávnom návrhu a následnej nesprávnej implementácii by mohla robotická platforma PR2 poškodiť okolité predmety a taktiež samu seba. Ďalším problémom môže byť zvolenie nevhodných metód pre vyhľadávanie dverí. Dvere môžu mať rovnakú farbu ale taktiež aj môžu byť rôznofarebné takže hľadať dvere na základe konštantnej farby je veľmi ťažké. Samotná detekcia kódu bude taktiež dôležitá, pretože súbor s týmto kódom obsahuje dôležité informácie pre prípadné bezpečné otvorenie dverí. Pre predídanie akýchkoľvek problémov je veľmi dôležité vytvoriť správny návrh modulu.

V tejto kapitole sa nachádza popis problému a spôsob, ktorým bude daná problematika vyriešená.

3.1 Popis problému

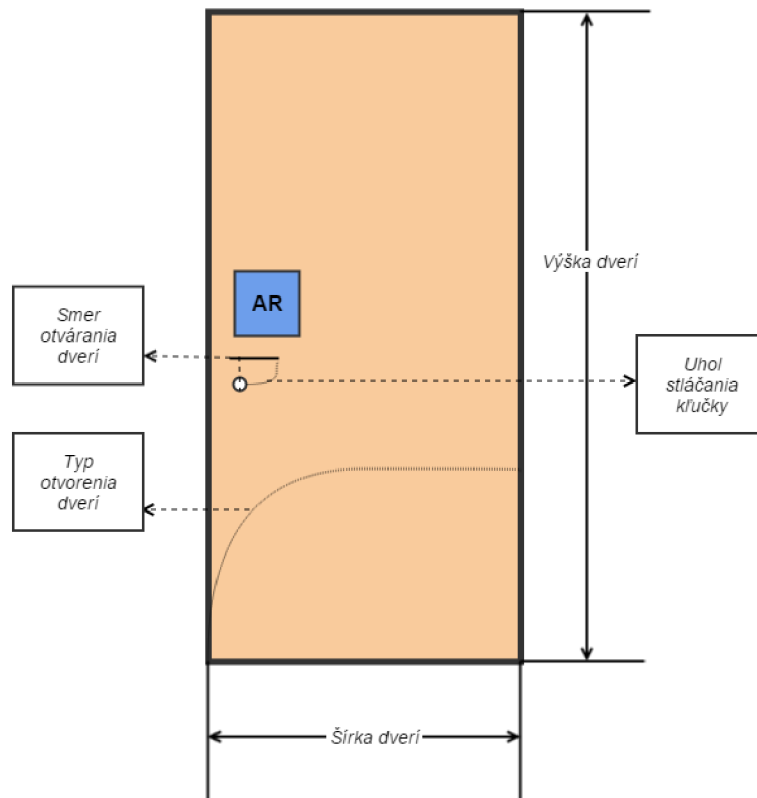
Cieľom tejto bakalárskej práce je vytvoriť modul pre detekciu a lokalizáciu kľučky. Tento modul bude schopný nájsť v miestnosti dvere, prísť k nim a rozpoznať typ dverí za pomoci AR kódu.

Modul by mal teda nájsť potencionálne dvere v miestnosti a mal by k nim prísť. Na to, aby robot rozpoznať skutočné dvere bude treba použiť vhodnú implementáciu. Pre detekciu dverí bude využívať svoje senzory a kamery, ktoré robot už používa. Cesta robota PR2 ku dverám by mala byť samostatná, ktorú by si mal naplánovať sám.

Keď sa robot PR2 už bude nachádzať pri dverách, vtedy nastupuje detekcia AR kódu. AR kód bude nalepený na dverách v oblasti kľučky. Senzor kinect rozpoznať identifikačný kód AR kódu pomocou ktorého pristúpy k určitým dátam priamo k požadovaným dverám. Tieto dáta budú obsahovať informácie o výške a šírke dverí, uhlom pod akým sa otvára kľučka, v ktorom smere sa dvere otvárajú (dvere tlačíť alebo dvere ťahať) a v akom smere sa dvere budú otáčať (zavesené vpravo alebo vľavo). Taktiež tam bude informácia o tom či sú dvere zamknuté alebo odomknuté. Tieto informácie sa potom posunú modulu pre otváranie dverí, ktorý je zodpovedný za otvorenie dverí a prejdenie nimi.

Informácie o dverách

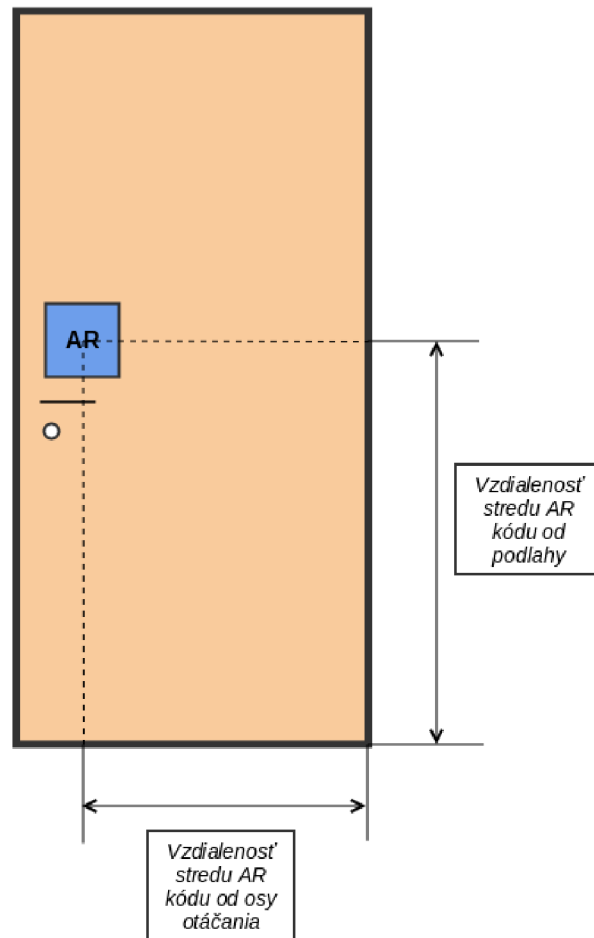
Na obrázku 3.2 sú názorne zobrazené všetky informácie, ktoré bude potrebné zistiť a pri rozpoznaní dverí následne poslať do modulu pre otvorenie dverí. Tieto informácie zostávajú z nasledujúcich údajov:



Obrázek 3.1: Vizualizácia potrebných údajov o dverách

- Šírka dverí (v metroch)
- Výška dverí (v metroch)
- Stav dverí:
 - Zamknuté
 - Otvorené
 - Zatvorené
- Uhol stláčania kľučky (v radiánoch)
- Smer, v ktorom sa dvere otvárajú vzhľadom na polohu ich zavesenia:
 - V smere hodinových ručičiek
 - Proti smeru hodinových ručičiek
- Smer, v ktorom sa dvere otvárajú vzhľadom na typ dverí:
 - Tlačiť
 - Ťahať

Poloha AR kódu



Obrázek 3.2: Správna poloha AR kodu

3.2 Koncept riešenia

Aby sa tento problém dal vyriešiť, musí sa navrhnuť dôkladná detekcia dverí. Robot musí zvládnuť bezpečne detekovať dvere a prísť k nim bez rôznych kolízií. Ďalej musí detekovať AR kód a vysporiadať sa s viacerými AR kódmi v priestore dverí. Pre tieto 2 rôzne problémy som sa rozhodol implementovať dva rozličné balíky. Jeden bude spravovať vyhľadávanie dverí a druhý bude obsahovať rozpoznávanie AR kódov. Tieto dva balíky budú medzi sebou komunikovať prostredníctvom správ a získané informácie o dverách budú taktiež prostredníctvom správ posielané ďalej modulu pre otvorenie dverí a bezpečné prejdienie nimi.

Robot by mal otáčať hlavou (keďže robot vie otočiť hlavu takmer o 360°), a s využitím PointCloudu detekovať potencionálne dvere. Dvere by sa mali detekovať podľa určitej výšky

a šírky a farby, ktorá bude iná ako je stena. Táto koncepcia problému by však nemala byť tá správna keďže dvere nemusia mať konštantú farbu. Ďalším spôsob, ktorý by mal byť lepší, sa skladá z viacerých podčastí. Najprv sa získajú súradnice bodov pri stene, ku ktorej príde, následne bude obchádzať miestnosť popri stene, s hlavou otočenou ku stene. Pri nájdení AR kódu sa robot otočí ku dverám. Po detekovaní identifikačného kódu, sa vyhľadá záznam s týmto kódom, v ktorom budú uložené všetky dôležité informácie o dverách, ktoré sú znázornené na obrázku 3.2. Tieto informácie budú poskytnuté modulu pre otvorenie dverí, ktorý následne tieto dvere otvorí a bezpečne nimi prejde.

3.3 Existujúce riešenie

Na internete je informácia o existencii balíku pre detekciu dverí a kľučky z 3D point cloud dát[15]. Táto detekcia bola vytvorená vývojármi z Willow Garage. V tomto riešení tohoto problému však vývojári používali zložitejší plánovač. Willow Garage však tento modul implementovali pre staršiu verziu ROSu. Modul pracoval len pre pár typov dverí a taktiež zdrojové materiály a potrebné údaje k tomuto balíku už na internetových stránkach nie sú aktuálne. Práve tento problém bol jednou z motivácií k vytvoreniu vlastnej detekcie dverí a kľučky, použiteľnej na rôzne typy dverí.

Kapitola 4

Implementácia

Pri realizácii tohoto modulu som sa snažil použiť to, čo ponúka robotická platforma PR2 a robotický operačný systém. Modul je vyvíjaný na platforme PR2, ktorá sa nachádza na Fakulte informačných technológií ale taktiež som vo veľkej miere využil pre vývoj simulačný nástroj Gazebo. Pri používaní simulačného nástroja som tak mohol vyvíjať modul bez priameho prístupu k tejto platforme. Pri záverečnom testovaní som však používal reálnu platformu v robotickom laboratóriu. Tu však pri neodladenom module mohlo nastať poškodenie robota PR2.

V nasledujúcej kapitole sa nachádza popis implementácie a problémy, ktoré som riešil pri vytváraní modulu.

Robot bude spolupracovať s počítačom, ktorý bude ovladaný užívateľom alebo bude spustený priamo na PR2. Pri spustení z počítača, tento počítač musí byť pripojený k robotovi cez internetový kábel, pretože WiFi pripojenie nie je dostatočne rýchle pre plynulý chod modulu. Po spustení modulov (detekcia dverí, otváranie dverí) už robot využíva vlastné poznatky zo senzorov a hľadá dvere, ktoré by mal následne otvoriť. Keďže moduly budú spúšťané na viacerých počítačoch, bude za potreby využiť synchronizáciu a správne nastavenie všetkých častí. V rámci vývoja modulu som využil vopred napísané balíky, ktoré mi pomohli a uľahčili výslednú implementáciu. Z viacerých distribúcií ROSu som využil distribúciu Hydro, ktorá je najnovšia distribúcia podporovaná na Ubuntu 12.04 a dobre popísaná v dokumentáciách na stránkach ROS. S touto distribúciou veľmi dobre pracoval aj simulátor Gazebo, kde boli testované viaceré implementácie. Pri preklade bol využitý systém catkin.

Ako už bolo spomínané v kapitole Návrh riešenia³, implementácia sa nachádza v dvoch balíkoch. Sú to balík `find_door`, ktorý slúži pre nájdenie dverí v miestnosti a balík `detect_door`, čo je balík pre detekciu dverí a získanie informácií o nich. V tejto časti budú popísané tieto dva balíky, správy, ktoré využívajú a príklad spustenia a otvárania dverí.

4.1 Balík `find_door`

Implementácia tohoto balíku prebiehala za pomoci simulátora Gazebo. No veľká časť implementácie bola vytváraná na reálnej platforme PR2 na fakulte informačných technológií v Brne, keďže práca so skutočným kinectom je oveľa presnejšia a rýchlejšia v reálnej podobe.

V tomto balíku sa nachádza implementácia algoritmu pre hľadanie dverí v miestnosti. Implementácia vyhľadávania dverí bola jednou z najťažších vecí na celom module. Pri implementácii som postupne testoval, ktorá možnosť bude najlepšia pre hľadanie dverí. Sa-

mozrejme som chcel použiť knižnicu PCL, ktorá ponúka veľa algoritmov a hĺbkový obraz. Z hĺbkového obrazu som chcel využiť hlavne body pre smer a vzdialenosť ku dverám.

Ako prvé som sa snažil využiť jednoduchú rovinnú segmentáciu bodov. Táto segmentácia ma za úlohu nájsť všetky body v rámci mračien bodov, ktoré sa nachádzajú v jednej rovine. V tejto segmentácii sa využíva aj výber bodov z mračien na základe indexov. Tieto body ďalej využíva už spomínaná jednoduchá segmentácia. Z tohoto algoritmu vznikne nakoniec obraz farebne rozlíšených plôch. Z týchto plôch by sa mali nájsť skutočné dvere, a to tak, že veľkosť skutočných dverí sa porovná s referenčnou veľkosťou typických dverí. Po následnej implementácii sa však tento algoritmus ukázal ako nevyhovujúci, pretože výsledné plochy boli často deformované a to z dôvodu zlých svetelných podmienok alebo chybného výpočtu tohoto algoritmu. Pre tento dôvod som musel hľadať iné riešenie tohoto problému. Ďalším implementovaným algoritmom bol algoritmus region growing. Úloha tohoto algoritmu je zlúčiť body, ktoré sú dostatočne blízko a sú farebne podobné. Výstupom je sada klastrov, kde každý jeden klaster je považovaný za jeden rovnaký a hladký povrch. Tento algoritmus pracuje na porovnávaní uhlov medzi bodmi normály. Taktiež tento algoritmus ako samotný nestačil na detekovanie samotných dverí a to aj napriek tomu, sa pri ňom dalo nastaviť niekoľko parametrov, ako napríklad minimálna a maximálna veľkosť klastrov alebo počet susedných bodov a iné. Region growing nevyhovoval pre detekciu samotných dverí pre zle rozoznávanie zo vzdialených pozícií, čo je dosť potrebné pre správne hľadanie dverí.

Nakoniec implementácia hľadania dverí v balíku `find_doors` prebehla na základe obchádzania robota okolo miestnosti a hľadania dverí podľa AR kódu. Po predchádzajúcich neúspechoch som hľadal vhodné riešenia, kde som sa rozhodol pre chodenie robota po miestnosti a hľadania dverí. Na toto sa využije knižnica `navigation stack`, ktorá zabezpečuje bezpečný pohyb robota po miestnosti. Robot nájde potrebné body do ktorých sa bude pohybovať za pomoci vyššie spomínaného algoritmu `region growing`. Pri presúvaní do tohoto bodu sa robot pohybuje za pomoci `navigation stack-u`, pri čom využíva odometrický senzor pre vytváranie mapy v miestnosti. Pri chôdzi má hlavu otočenú na stenu(dvere) a snaží sa detekovať na nej AR kód za pomoci môjho balíku `detect_code`. Pri nájdení AR kódu sa robotická platforma PR2 zastaví a otočí k dverám, kde je následne pripravená k otvoreniu daných dverí.

Špecifickejší popis

V tejto sekcii sú napísané konkrétne informácie o implementovanom uzle v balíku `find_doors`. Základný zdrojový súbor je `find_doors_node.cpp`, kde sa nachádza implementácia vyhľadávania dverí. Ďalej však v tomto balíku využívam triedy `driver.cpp` a `head.cpp`. Triedu `driver.cpp` som využil pre prácu s robotom a to konkrétne pre otáčanie robota o určitý uhol pri stenách. Ďalšia trieda (`head.cpp`) je využitá pre otáčanie hlavy robota. Táto trieda sa využíva pri hľadaní steny a AR kódu, kedy je zapotreby otáčať hlavou.

Samotný súbor `find_door_node.cpp` sa skladá z hlavnej funkcie `int main(int argc, char** argv)`, dvoch callbackov `void find_wall(const sensor_msgs::PointCloud2ConstPtr& msg)`, `void code(const find_door::Find_code::ConstPtr& msg)` a pomocných funkcií `void turnRobot(int tmp, ros::NodeHandle &nh)`, `void setHead(int tmp)`.

Popis činnosti

Po spustení uzlu `find_door_node.cpp` sa nastaví za pomoci pomocnej funkcie `setHead()` využívajúcej triedu `head.cpp` nasmeruje hlava rovno pred platformu PR2 a vytvorí sa sub-

scriber, ktorého vstupm je point cloud. Tento subscriber spustí callback `find_wall()`, kde prebieha samotné detekovanie steny nasledovne:

- Ako prvé sa musí pretransformovať správa z typu ROS správy na typ pcl, s ktorým umožňuje pracovať point cloud knižnica.
- Nasleduje samotný algoritmus Region growing detekovania bodov, ktoré sa nachádzajú blízko seba a majú rovnakú farbu.
- Algoritmus Region growing požaduje normály pre výpočet týchto plôch. Trieda `pcl::NormalEstimation` slúži na výpočet potrebných normál.
- Ďalším bodom je nastavenie ľubovoľných parametrov. V tomto prípade som nastavil minimálnu a maximálnu veľkosť detekovaných klastrov, ale aj napríklad počet požadovaných susedných bodov.
- Po vyhľadání požadovaných klastrov, vznikne štruktúra všetkých nájdených klastrov. Z nich sa vyberie klaster, ktorý nie je nulový.
- Z tohoto klastru je vybraný jeden bod, z ktorého sa získajú súradnice x, y, z.
- Keďže tento bod je pôvodom z kinect, a hlava robota so senzorom kinect sa pri pohybe otáča doprava, bolo potrebné tieto súradnice pretransformovať, aby sa robot pohyboval v požadovanom smere.
- Po transformácii sa nastaví cieľová pozícia so získaných súradníc a robot sa dostáva do pohybu.
- Keď robot narazí na stenu zastane otočí sa a znova hľadá body za pomoci senzoru kinect.
- Pri nájdení AR kódu uzol `detect_doors_node.cpp` pošle uzlu `find_doors_node.cpp` správu o detekovaní AR kódu a robot sa zastaví.

Po nastavení súradníc v callbacku `find_wall` sa opäť použije funkcia `setHead()` pre otočenie hlavy o 90° doprava a platforma PR2 prevezme cieľové súradnice, do ktorých sa snaží presunúť za pomoci knižnice `navigation stack`. Presun robota je spustený príkazom `ac.sendGoal(goal)`, kde `ac` je klient pre presúvanie platformy PR2. Keďže mapovanie miestnosti prebieha odometrickým senzorom, hlava robota môže byť stále otočená vpravo a tak detekovať kód pri presúvaní sa po miestnosti. Pre prijatie správy z uzlu `detect_code` je potrebné ďalšie vlákno. V jednom vlákne prebieha presúvanie robota na cieľovú pozíciu a v druhom subscriber s callbackom `code()`. V tomto callbacku sa čaká na prijatie správy `Find_code.msg` popísanej v tabuľke 4.2. Po následnej detekcii AR kódu robot PR2 zruší plánovanie cesty (`ac.cancelGoal`) a následne sa otočí smerom k dverám. Pri neúspešnom hľadaní a dorazení na cieľovú pozíciu sa robot otočí o 90° doľava a následne opäť začína s detekciou protihľanej steny a získaní súradníc.

4.2 Balík `detect_door`

Pri detekovaní AR kódu je využitá knižnica `ar_track_alvar`, ktorá detekuje a dekoduje AR kódy. Pri implementácii tohoto balíku bolo využité najmä simulačné prostredie, kde bolo

za potreby ako prvé nahráť AR kódy kinectom do podoby bag súborov¹. Avšak pri vytváraní týchto súborov bol nahraný najprv obraz z kamery. Tento obraz však pri využívaní knižnice `ar_track_alvar` nebol vyhovujúci a preto bolo potrebné vytvoriť nové súbory, kde sa využívajú aj `point cloud-y`. Po nahraní týchto súborov a spustení knižnice `ar_track_alvar` sa táto knižnica vracia požadovanú hodnotu AR kódu. Na základe tohoto kódu program vyhledá požadovaný súbor s informáciami zo špeciálneho príčinku. Z tohoto súboru získa informácie potrebné k otvoreniu dverí, z ktorých vytvorí správu. Táto správa sa posiela pre modul, ktorý otvára požadované dvere.

Špecifickejší popis

V tejto časti sa nachádza popis implementovaného uzlu `detect_doors_node.cpp` balíka `detect_doors`.

Uzol `detect_doors_node.cpp` sa skladá z hlavnej funkcie `int main(int argc, char** argv)`, callbacku `void arPoseCallback(const ar_track_alvar::AlvarMarkers::ConstPtr& msg)` a pomocnej funkcie `void setDoorInfo(int code)`.

Popis činnosti

Ako prvé sa vytvorí rozhranie pre odosielanie správ a to pre druhý uzol `find_door pub1 = n.advertise<detect_doors::Find_code>("find_door/find_code",10)`; pri detekovaní AR kódu a modul pre otváranie dverí `pub2 = n.advertise<detect_doors::Doors>("open_doors/doors",10)`; . Následne sa vytvorí subscriber s callbackom `arPoseCallback()` pre detekovanie AR kódu. V tomto callbacku sa pri detekovaní vybere jeden kód, ten ktorý bol detekovaný ako prvý, a uloží sa jeho hodnota do premennej `marker_id`. Pri úspešnom nájdení AR kódu sa publikuje správa pre uzol `find_door`, ktorá slúži pre zastavenie hľadania dverí robotom PR2. Z informácií, ktoré poskytuje AR kód, je potrebné zistiť polohu dverí vzhľadom k robotovi. Zo získaných súradníc dverí (rámec `ar_code` sa súradnice transformujú pre rámec `odom_combined`. Po úspešnom transformovaní sa riadenie programu presúva späť do hlavnej funkcie, kde sa začínajú vysielat' TF rámce pre modul otvárania dverí. Po poslaní niekoľkých rámcov sa pošle správa `Doors.msg` (popísaná v tabuľke 4.1. Táto správa je nastavovaná v pomocnej funkcii `setDoorInfo()`, ktorá ako parameter prijíma kód. Tento kód slúži pre prístup do súboru s informáciami o daných dverách. V tejto funkcii sa po otvorení daného súboru načítajú všetky potrebné informácie, ktoré sú následne poslane správou `Doors.msg` do modulu pre otváranie dverí robotom PR2.

Príklad spustenia

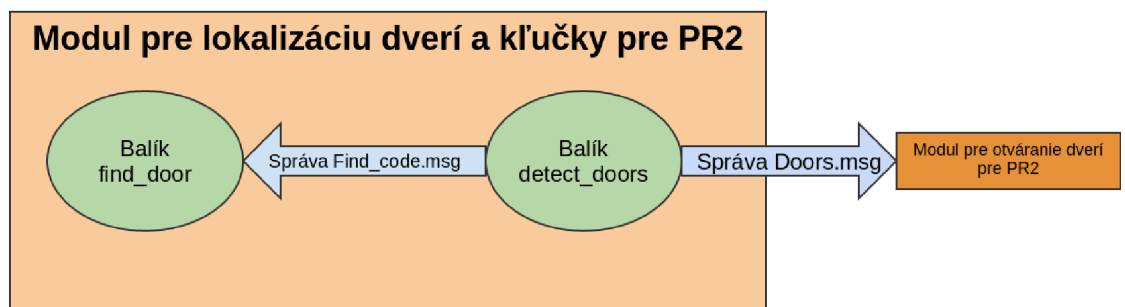
1. Ako prvé po spustení modulu robot nasmeruje hlavu pred seba a použije algoritmus `region growing` k vyhledaniu plochy (steny).
2. Po nájdení tejto plochy, sa získajú súradnice a hlava robota sa otočí doprava.
3. získané súradnice sa použijú ako cieľové súradnice pre `navigation stack` a robot sa začne presúvať na tieto súradnice.
4. Pri tejto ceste robot svojou otočenou hlavou hľadá AR kód. Pri nájdení AR kódu sa robot zastaví a otočí sa smerom ku dverám. Prechádza sa na bod 6.

¹špeciálny typ súboru využívaný v robotickom operačnom systéme

5. Ak sa robot už nachádza v cieľových súradniciach a AR kód nebol nájdený, robot sa otočí o 90° doľava, opäť pokračuje bodom jedna.
6. Robot detekuje AR kód načíta potrebné informácie zo súboru a pošle správu s potrebnými informáciami ďalšiemu modulu.

4.3 Formáty správ

Pre komunikáciu medzi uzlami som využil správy. V nasledujúcich podkapitolách sa nachádzajú popísané formáty správ, ktoré som v práci využil. Na schéme v obrázku 4.1 je graficky naznačené posielanie správ v mojom module medzi uzlami `find_door_node.cpp` a `detect_doors_node.cpp` a uzlom `detect_doors_node.cpp` a modulom pre otváranie dverí.



Obrázek 4.1: Schéma posielania správ

`detect_doors/Doors.msg`

Táto správa sa používa pre komunikáciu 2 modulov. Môj modul pre detekciu dverí a kľučky pre PR2 a modul pre otvorenie dverí pre PR2. V tejto správe sa posielajú získané informácie mojím modulom pre bezpečné otvorenie dverí. `width` a `height` premenné určujú šírku a výšku dverí, `latch_state` určuje stav dverí z konštánt(`LOCKED`, `LATCHED`, `UNLATCHED`), premenná `handle_rot_angle` určuje uhol v radiánoch akým sa stláča kľučka, ďalšia premenná `rot_dir` určuje smer v ktorom sa dvere otvárajú a nakoniec premenná `push_pull` určuje či sú dvere tlačené alebo ťahané pri otváraní. Celý popis sa nachádza v tabuľke 4.1.

Typ	Názov
float32	width
float32	height
int32	LOCKED=1
int32	LATCHED=2
int32	UNLATCHED=3
int32	latch_state
float32	handle_rot_angle
int32	CLOCKWISE=1
int32	COUNTERCLOCKWISE=2
int32	rot_dir
int32	PUSH=1
int32	PULL=2
int32	push_pull

Tabulka 4.1: Správa detect_doors/Doors

detect_doors/Find_code.msg

Táto správa komunikuje medzi mojimi uzlami find_door a detect_doors. Platforma PR2 hľadá dvere obchádzaním miestnosti po obvode a pri detekovaní AR kódu je poslaná uzlu find_door správa z tabuľky 4.2. Parameter door označuje nájdenie dverí.

Typ	Názov
int32	FIND=1
int32	NOTFIND=2
int32	door

Tabulka 4.2: Správa detect_doors/Find_code

Kapitola 5

Testovanie

V nasledujúcej kapitole sa nachádza popis jednotlivých testov, ktoré boli vytvorené pre účel experimentovania. Jednotlivé experimenty mali určiť výslednú kvalitu vytvoreného modulu. Ďalej sa tu nachádza popis jednotlivých testov a v závere sú tieto výsledky vyhodnotené.

Cieľom tejto práce bolo vytvoriť samostatnú detekciu a lokalizáciu kľučky a dverí pre robota PR2. To znamená robot samostatne po zapnutí modulu nájde dvere a získa informácie o týchto dverách.

5.1 Experimenty

V tejto časti sa nachádza popis experimentu a následne ukážka niekoľkých testovaných experimentov.

V tomto experimenta sa platforma PR2 nachádza v miestnosti s jednými dverami a niekoľkými prekážkami(stenami). V jednotlivých testoch sa rozmiestnenie prekážok v miestnosti mení a tým vzniká rôznodruhosť tohoto experimentu. Keďže tento modul spolupracuje s modulom pre otváranie dverí robotom PR2, ďalším testom bolo testovanie spolupráce jednotlivých modulov. Táto úloha pozostáva z týchto častí:

- Spustenie modulu a potrebných knižníc.
- Lokalizácia dverí
- Lokalizácia kľučky na základe AR kódu.
- Odoslanie potrebných informácií modulu pre otváranie dverí platformou PR2.

Očakávaný priebeh experimentu sa potom skladá z týchto častí:

1. Po spustení modulu platforma PR2 nájde protiľahlú stenu a prechádza k nej.
2. Pri presúvaní sa ku stene sa platforma PR2 snaží detekovať dvere tým, že hľadá AR kód.
3. Pri neúspešnom lokalizovaní dverí a dorazní na cieľovú pozíciu sa platforma PR2 otočí o 90° doľava a opäť hľadá protiľahlú stenu.
4. Pri úspešnej lokalizácii dverí sa platforma PR2 otočí ku dverám a načíta informácie na základe AR kódu.

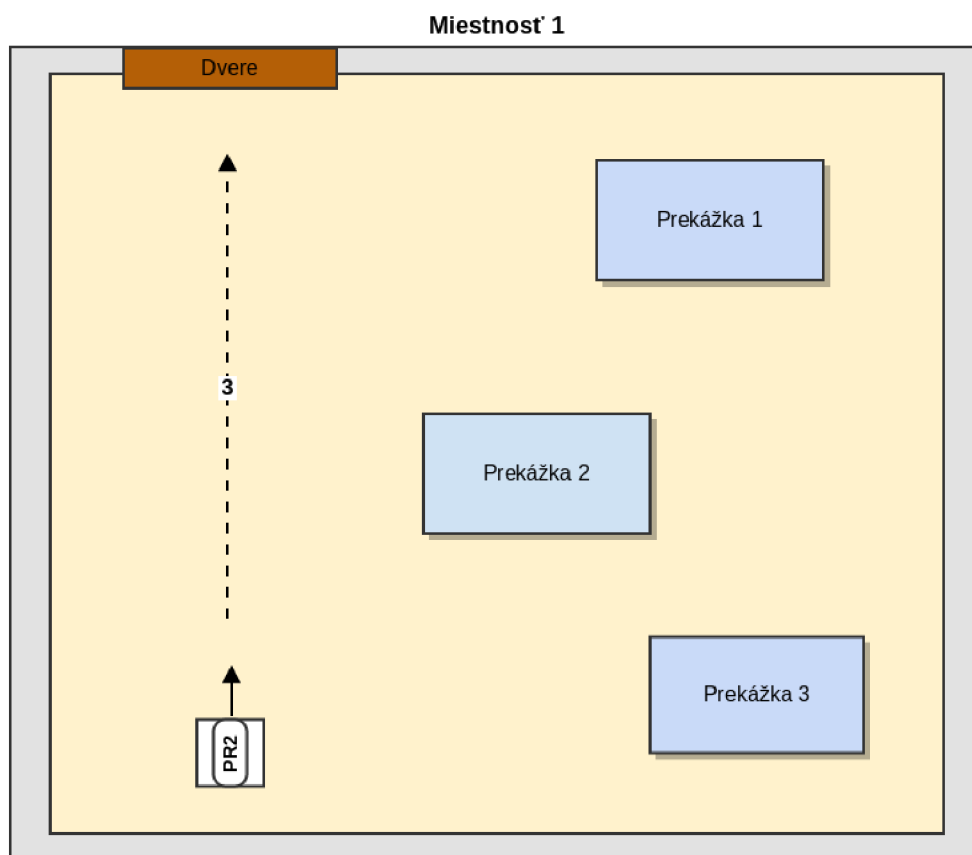
5. Tieto informácie sú poslané modulu pre otváranie dverí, ktorý následne na základe týchto informácií dvere otvorí.

Tento priebeh bol testovaný z rôznych pozícií, z ktorých najčastejšie vznikali nižšie popísané 3 prípady.

Lokalizácia dverí

V tejto podkapitole sú uvedené 3 typy testov, ktoré boli riešené v rámci testovania.

Test 1

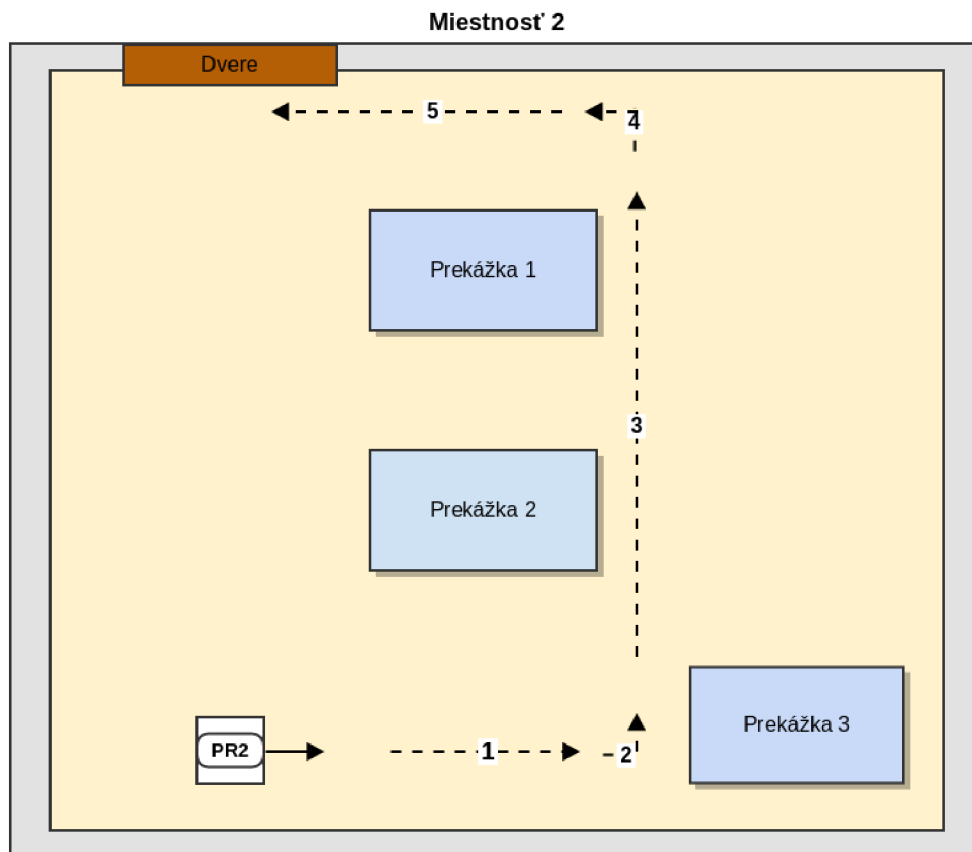


Obrázek 5.1: Prvý test

Na obrázku 5.1 je znázornená miestnosť a najjednoduchšie detekovanie dverí, ktoré sa nachádzajú oproti robotu. Robot už pri dojazde do cieľovej pozícií lokalizuje dvere pomocou AR kódu a tak spustí modul pre otvorenie dverí. Táto detekcia sa skladá len z jedného bodu:

1. Nájdenie protíľahlého bodu a následné presúvanie sa platformy do tohoto bodu v ktorom sa dvere rozpoznajú.

Test 2



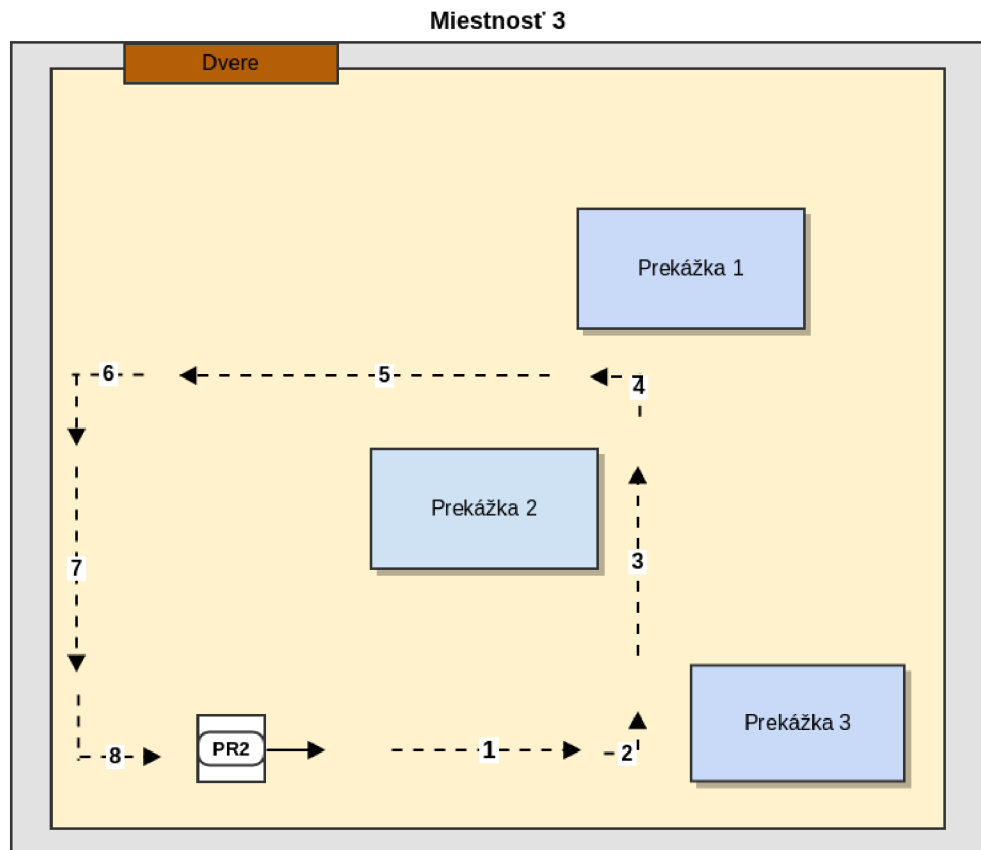
Obrázek 5.2: Druhý test

Ďalší prípad testu je znázornený na obrázku 5.2, kde robot po neúspešnom hľadaní narazí na prekážku, následne sa otočí a hľadá iný bod, do ktorého sa bude presúvať.

1. Ako prvé sa nájde protiľahlý bod a nastáva presun na tento bod.
2. Následne narazí robot na prekážku a zastane otočí sa o 90° doľava a opäť pokračuje hľadaním protiľahlého bodu.
3. Následne sa tento bod použije ako cieľový bod a robot sa začína presúvať s otočenou hlavou a hľadaním AR kódu.
4. Robot po dorazení na cieľový bod a neúspešnom detekovaní dverí zastane opäť sa otočí o 90° doľava. Pokračuje hľadaním protiľahlého bodu.
5. Tento bod sa znovu použije ako cieľový bod a pri presune robota sa už úspešne detekujú dvere a robot zastane.

Tento test však prebiehal len od bodu 3, pretože v učebni, kde prebiehali testy nebolo dostatok miesta. Po narazení na stenu sa robot otočil, a hľadá priľahlý bod. Po nájdení bodu sa presúval na tento bod až kým nenarazil na dvere, pri ktorých zastavil a detekoval AR kód. Tým sa dvere rozpoznali.

Test 3



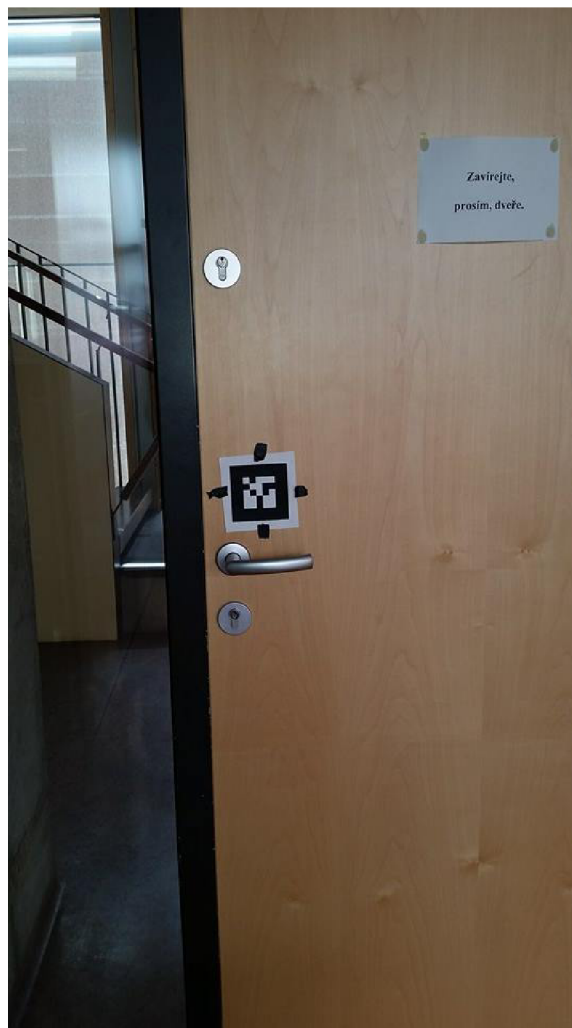
Obrázek 5.3: Tretí test

V tomto prípade, ktorý je zobrazený na obrázku 5.3, sa však robotovi vôbec nepodari lokalizovať dvere. Robot narazí na viaceré prekážky a tým sa vôbec nedostane ku dveram, takže AR kód ostane nedetekovateľný.

1. Po spustení sa nájde protilhlá stena, robot sa začne presúvať avšak narazí na prekážku tak zastane.
2. Robot sa otočí o 90° doľava a znovu hľadá protilhlú stenu.
3. Po jej nájdení sa začne presúvať do tohoto bodu a opäť narazí na prekážku.
4. Znovu sa robot otočí o 90° doľava a hľadá protilhlú stenu.

5. Po doterajšej neúspešnej lokalizácii dverí robot pokračuje na ďalšie súradnice. Robot síce prejde s hlavou otočenou ku dverám avšak dvere sa nachádzajú ďaleko od robota tak ostajú nedetekované.
6. Po dosiahnutí cieľových súradníc sa znovu robot otočí o 90° .
7. Robot prechádza popri stene s hlavou otočenou na stenu kde sa hľadá AR kód nalepený na dverách.
8. Nakoniec sa robot otočí opäť doprava a dostáva sa do miesta z ktorého vychádzal.

Lokalizácia kľučky



Obrázek 5.4: AR kód nalepený na dverách

Detekovateľnosť AR kódu bola testovaná v rámci simulácie. Testy prebiehali vo vzdialenejších pozíciách, za zhoršených svetelných podmienok ale aj z rôznych uhlov.

5.2 Vyhodnotenie experimentu

Tento experiment bol testovaný niekoľkokrát priamo na platforme PR2 na Fakulte informačných technológií. Z hore uvedených testov je zrejmé, že lokalizácia dverí nemusí byť vždy úspešná. Keďže robot sa vždy pri narazení na prekážku alebo stenu otočí o 90° doľava, môže sa stať to, že dvere ostanú nelokalizované. Tento prípad je znázornený v predchádzajúcej podkapitole na obrázku 5.3. Naopak v ďalších dvoch prípadoch (obrázok 5.1 a 5.2) sa dvere podarilo detekovať, avšak táto detekcia tiež nefungovala na 100%. Detekcia znázornená na obrázku 5.1 je najjednoduchšia, pretože robot sa len presunie rovno k protiľahlým dverám. Táto verzia fungovala vždy. Naopak druhá detekcia (obrázok 5.2) je zložitejšia. Ako už bolo spomínané pri popise testu číslo 2, tento test prebehol v menšom rozsahu. Tu niekedy vznikali komplikácie pri tvorení mapy, ktorá slúži pre knižnicu `navigation stack`. Niekedy vznikne chyba pri získavaní cieľového bodu, čo má za dôsledok zlý pohyb robota, respektíve nesprávny presun do tohoto bodu.

Pri lokalizácii kľučky rozhoduje veľkosť AR kódu, vzdialenosť robota od dverí a svetelné podmienky v miestnosti. Pri dobrých podmienkach je detekovateľnosť tohoto kódu veľmi vysoká a to aj vďaka použitej knižnici `ar_track_alvar`, ktorá to zvláda bez väčších problémov. Tu však rozhodujú aj uložené údaje o dverách, ktoré sa načítávajú zo súboru. Pri zlých údajoch je otvorenie dverí nemožné. Ďalší problém však nastáva pri detekovaní viacerých AR kódov. To je vyriešené tým, že sa použije len prvý detekovaný kód, avšak tento kód nemusí byť vždy ten správny. Pri detekovaní nesprávneho kódu uzol nepošle správnu verziu správy.

5.3 Námety k rozšíreniu práce

V tejto podkapitole je popis možných vylepšení tohoto modulu.

Vylepšenie hľadania dverí

Hľadanie dverí by malo fungovať len v miestnostiach tvaru štvorec a obdĺžnik, preto ako prvé vylepšenie by mohlo byť hľadanie dverí po rôznotvarých miestnostiach. Táto funkcia by pomohla detekciu viac zlepšiť a prispôbiť do rôznych miestností, nie len miestností jedného typu.

Vylepšiť by sa dala aj rýchlosť otvárania dverí. Bolo by potrebné však rýchlejšie vyhľadávať protiľahlý bod, to znamená cieľový bod, do ktorého sa robot presúva.

Ďalším bodom optimalizácie by mohlo byť lepšie využívanie mapy, ktorú poskytuje knižnica `navigation stack`. Získanie rohov miestnosti a určenie ich ako cieľových pozícií by mohlo fungovať správne, avšak problém by nastal pri miestnostiach iného typu.

Vylepšenie detekcie dverí a kľučky

Detekcia dverí funguje pomerne dobre, avšak veľmi dôležitú rolu zohráva postavenie robota pri prechode riadenia z môjho modulu pre lokalizáciu dverí a kľučky do modulu pre otvorenie dverí.

Editácia dverí

Pre uľahčenie používania tohoto modulu by prospela aplikácia, ktorá by slúžila na editáciu rôznych dverí. Tá by vytvárala súbory so všetkými potrebnými informáciami pre úspešné otvorenie dverí do ktorých by mal prístup uzol `detect_doors_node.cpp`.

Kapitola 6

Záver

Cieľom tejto práce bolo vytvoriť modul pre lokalizáciu dverí a kľučky pre robota PR2. K dosiahnutiu tohoto cieľa bolo potrebné navrhnuť, implementovať a otestovať vytváraný modul. Tento modul spolu s modulom vytváraným v inej bakalárskej práci [1] vytvoril automatizovaný systém otvárania dverí od robotického laboratória.

Tento cieľ bol splnený a všetky potrebné informácie sú popísané v tomto dokumente. Zo štúdia potrebných informácií, ktoré boli potrebné pre vytvorenie modulu, som tieto informácie spracoval do teoretickej kapitoly. Návrh modulu je popísaný v kapitole Návrh riešenia, ktorá obsahuje aj koncept modulu. Programové riešenie tohto modulu sa nachádza v kapitole Implementácia, popisom dvoch balíkov. Testy a experimenty sa nachádzajú v záverečnej kapitole. V tejto práci však vzniklo mnoho neočakávaných problémov, ktoré museli byť riešené a preto sa nepodarilo splniť niektoré vyššie ciele. Lokalizácia dverí funguje len vo obdĺžnikovej miestnosti s malým počtom prekážok. Naopak predávanie informácií AR kódom funguje veľmi dobre.

Z výsledku experimentov bolo zistené, že najjednoduchšia lokalizácia je lokalizácia dverí napriamo. To znamená, že robot sa nachádza kolmo od dverí, prípadne v minimálne odchyľke od tejto kolmice. S presvedčivou funkčnosťou pracuje aj hľadanie dverí v miestnosti kde sa nachádza málo prekážok alebo kde robot nemá veľmi zložitú cestu k dverám. Avšak pri veľkom množstve prekážok sa robot k týmto dverám dostane ojedinele. Získanie informácií prebieha za pomoci AR kódu, ktorý sa nachádza na dverách. Po detekovaní tohoto kódu sa získajú všetky potrebné informácie z uloženého súboru.

Vytvorená aplikácia je samostatná a je možné ju rozšíriť o ďalšie balíky a moduly. Hlavným rozšírením by bolo zrejme vytvorenie efektívnejšieho hľadania dverí v miestnosti rôznych typov a aplikácie pre vytváranie a editáciu dverí, do ktorých by sa ukladali všetky potrebné informácie pre otváranie daných dverí. Táto aplikácia by slúžila hlavne pre uľahčenie užívateľovi pri používaní nových typov dverí. Pri zefektívnení daných uzlov by som chcel tento modul zverejniť na stránkach Robotického informačného systému.

Literatura

- [1] Sencuch, D.: *Modul pro otevreni dveri pro PR2*. Bakalarska prace, Vysoke uceni technicke v Brne, 2015.
- [2] WWW stranka: Nodes [online]. <http://wiki.ros.org/Nodes>, [cit. 2015-01-17].
- [3] WWW stranka: The Robot Operating System [online]. <http://www.ros.org/about-ros/>, [cit. 2015-01-17].
- [4] WWW stranka: ROS Documentation [online]. <http://wiki.ros.org/>, [cit. 2015-01-17].
- [5] WWW stranka: Gazebo [online]. <http://gazebosim.org/>, [cit. 2015-01-18].
- [6] WWW stranka: Kinect [online]. http://wiki.ipisoft.com/Depth_Sensors_Comparison, [cit. 2015-01-18].
- [7] WWW stranka: Messages [online]. <http://wiki.ros.org/Messages>, [cit. 2015-01-18].
- [8] WWW stranka: Personal robot 2 [online]. <http://www.smartroboticsys.eu/>, [cit. 2015-01-18].
- [9] WWW stranka: Services [online]. <http://wiki.ros.org/Services>, [cit. 2015-01-18].
- [10] WWW stranka: Ar track alvar [online]. http://wiki.ros.org/ar_track_alvar, [cit. 2015-01-19].
- [11] WWW stranka: Barcode [online]. <http://www.computerhope.com/jargon/b/barcode.htm>, [cit. 2015-01-19].
- [12] WWW stranka: Navigation stack [online]. <http://wiki.ros.org/navigation>, [cit. 2015-01-19].
- [13] WWW stranka: Point cloud library [online]. <http://pointclouds.org/>, [cit. 2015-01-19].
- [14] WWW stranka: Teoria o ciarovych kodoch [online]. <http://www.kodys.sk/stranka/trochu-teorie-o-ciarovom-kode>, [cit. 2015-01-19].
- [15] WWW stranka: Door handle detector [online]. http://wiki.ros.org/door_handle_detector, [cit. 2015-04-20].

Příloha A

Obsah CD

Zložka find_doors

Obsahuje prvý balík, ktorý bol predmetom tejto bakalárskej práce. Tento balík slúži pre lokalizáciu dverí a je priamo pripravený pre prácu na reálnom robotovi.

Zložka detect_doors

Obsahuje druhý balík, ktorý bol predmetom tejto bakalárskej práce. Tento balík slúži pre lokalizáciu kľučky na dverách, je pripravený pre spustenie na reálnom robotovi ako aj v simulátore.

Zložka ar_track_alvar

Obsahuje balík, ktorý slúži pre detekciu AR kódu.

Zložka door_open

Táto zložka obsahuje uzol Daniela Senčucha, ktorý vytvoril vo svojej bakalárskej práci ???. Zložka slúži pre demonštráciu spolupráce môjho a Danielovho uzlu.

tex

Obsahuje zdrojové kódy a grafický materiál pre vytvorenie dokumentu Modul_pro_lokalizaci_dveri_a_kliky_...

Súbor Detekcia a otvorenie dveri pre PR2.avi

Video demonštrujúce výsledok prác mňa a Daniela Senčucha.

Súbor Modul pro lokalizaci dveri a kliky pro PR2.pdf

Tento dokument.

Súbor manual.txt

Obsahuje návody pre nainštalovanie a spustenie všetkých balíkov pre spustenie uzlov `find_door`, `detect_doors` a `pr2_open_doors` a taktiež ako ich správne spustiť.