VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

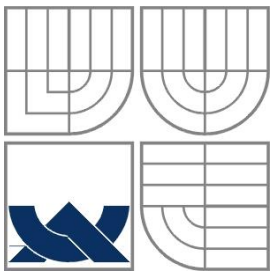
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ROBOTICKÝ NÁKUPNÍ KOŠÍK

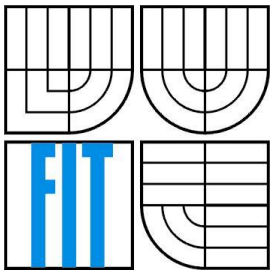
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE JIŘÍ KUŘÁTKO
AUTHOR

BRNO 2013



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

ROBOTICKÝ NÁKUPNÍ KOŠÍK

ROBOTIC SHOPPING CART

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — JIŘÍ KUŘÁTKO
AUTHOR

VEDOUCÍ PRÁCE — Ing. ZDENĚK MATERNA
SUPERVISOR

BRNO 2013

Abstrakt

Bakalářská práce se zabývá návrhem aplikace pro plánování pohybu robota tak, aby následoval zadaného člověka. Práce se zpočátku věnuje senzoru Microsoft Kinect, který je použit pro snímání okolních objektů. Dále jsou rozebírány možnosti pro rozpoznávání lidských postav z hloubkových dat získaných senzorem Microsoft Kinect. Čtenář je také seznámen s metodami pro zjišťování směru příchodu akustických signálů. Poslední kapitoly jsou věnovány řízení pohybu robota. Na závěr je popsána celá implementace softwaru robota pro platformu Robotic Operating System (ROS).

Abstract

This bachelor's thesis deals with the project of application about planning the robot's movement in order to follow the specified person. At the beginning of the work, the Microsoft Kinect sensor which is used for sensing nearby objects is described. Hereinafter the possible methods how to identify human figures on the basis of depth data from the Microsoft Kinect sensor are discussed as well. The reader is also familiar with the methods determining the direction of arrival of acoustic signals. Last chapters of the work are dedicated to the motion control of the robot. Finally the complete implementation of the robot software for the platform of Robotic Operating System (ROS) is described.

Klíčová slova

Kinect, ROS, PCL, OpenNI, libfreenect, NITE, PID, PSD, TDOA, TurtleBot, odometrie

Keywords

Kinect, ROS, PCL, OpenNI, libfreenect, NITE, PID, PSD, TDOA, TurtleBot, odometry

Citace

Jiří Kuřátko: Robotický nákupní košík, bakalářská práce, Brno, FIT VUT v Brně, 2013

ROBOTICKÝ NÁKUPNÍ KOŠÍK

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Zdeňka Materny. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Jiří Kuřátko
15. května 2013

Poděkování

Poděkování patří především vedoucímu mé bakalářské práce panu Ing. Zdeňkovi Maternovi za odborné vedení, rady i připomínky při vypracovávání této práce.

Obsah

1 Úvod

Cílem této bakalářské práce je vytvořit program pro robota TB2. Tento program musí umět detekovat a rozpoznávat osoby v prostoru. TB2 si vybere první detekovanou osobu a její pohyb posléze celou dobu následuje. V praxi by toto chování mohlo být využito jako např. robotický nákupní košík. K rozpoznávání a sledování pohybu člověka je použito zařízení Microsoft Kinect. Microsoft Kinect poskytuje data ve formě tzv. hloubkových dat (point clouds). Hloubková data jsou množina bodů ve 3D souřadném systému reprezentující vnější povrch objektů. Takto je získán trojrozměrný obraz prostoru nacházejícího se před Kinectem. Za pomocí knihovny OpenNI je z těchto dat prováděna detekce přítomnosti osob ve snímaném prostoru. Práce obsahuje také porovnání OpenNI s knihovnou libfreenect, která je rovněž v mé práci použita. Dále popisuji i možnost detekování lidí za pomocí knihovny PCL pro zpracování hloubkových dat. Kinect kromě hloubkového snímače obsahuje mikrofonové pole, které je využíváno v případech, kdy je pozice osoby mimo snímaný prostor. Vygenerováním zvukového signálu, písknutím či jiným způsobem dojde na tomto mikrofonovém poli k vyhodnocení směru (úhlu) příchodu akustického zvuku. Tento změřený úhel je vyžit k natočení robota tak, aby se příslušná osoba dostala do snímaného prostoru. Princip vyhodnocení směru je založen na měření časového rozdílu v příchodu zvuku na jednotlivé mikrofony (TDOA). Celý software (SW) je implementován ve formě balíčku pro Robot Operating Systém (ROS)[1]. Tento systém obsahuje knihovny a nástroje, které usnadňují vývoj robotických aplikací. Dále poskytuje hardwarovou (HW) abstrakci, obsahuje správce balíčků, umožňuje vytváření závislostí na jednotlivých komponentech, obsahuje nástroje pro předávání zpráv, vizualizéry a další.

2 Popis konstrukce robota

K testování vytvářeného SW jsem měl možnost použít školního robota, ale zkonstruoval jsem si vlastního. Tento robot, stejně jako robot školní, jsou inspirovány továrním výrobkem TurtleBot, který je de facto nízkonákladovou robotickou platformou. Podvozkem mého robota je upravený robotický vysavač iRobot Roomba 530. Bližší popis je v kapitole 2.2.

K podvozku je kolmo připevněná tyč, na níž je instalován senzor Microsoft Kinect, kterým jsou získávána RGB-D a zvuková data. Toto zařízení je podrobněji popsáno v kapitole 2.1. Řídící SW robota je instalován v notebooku, který je spolu s nosnou tyčí Kinectu připevněn k podvozku robota.

[1] http://www.ros.org/wiki/

Tento SW je schopen ovládat i verzi školního robota. Školní robot obsahuje mimo již popsaných zařízení ještě další vybavení jako přídavné baterie, laserový skener a robotické rameno, které pro tuto práci nejsou potřebné. Bližší informace o rozdílech mezi roboty a důsledcích jsou uvedeny v kapitole 8.

Obrázek 1: Bakalářská verze robota

Obrázek 2: Školní verze robota

2.1 Microsoft Kinect

Microsoft Kinect je zařízení, které je v této bakalářské práci použito jako senzor pro snímání okolních objektů.

Zařízení disponuje klasickou barevnou kamerou v rozlišení 1280 x 1024 bodů, ale kvůli menšímu objemu přenášených dat po USB se rozlišení snižuje na 640 x 480 bodů [1]. Dále obsahuje infračervený vysílač a snímač měřící vzdálenosti okolních objektů od Kinectu pracující se stejným rozlišením. Tyto vzdálenosti jsou prezentovány ve formě hloubkových dat (point clouds). Tím je získán 3D model snímaného prostoru.

Ke snímání příchodu směru akustického signálu slouží mikrofonové pole tvořené čtyřmi mikrofony, integrovanými přímo v zařízení za čelní stěnou (Obrázek 13: Umístění mikrofonů v zařízení Kinect). Analogový signál z mikrofonů je přiveden na A/D převodníky. Převod na digitální signál probíhá u všech mikrofonů současně se vzorkovací rychlostí 16 KHz a s rozlišením 32 bitů. Pomocí integrovaného motorku v kloubu Kinectu je možné zařízení vertikálně naklánět. Naklápěním je kompenzována různá výška sledovaných osob. Je snaha, aby osa snímače vždy mířila na střed sledované osoby. U školního robota TB2 není tento pohyblivý kloub využíván, protože nosník příslušného Kinectu má vlastní kloub se servomotorem.

Výpočetní jednotku Kinectu zajišťuje čip od společnosti PrimeSense, který se stará o zpracování všech informací od senzorů a zajišťuje komunikaci s PC.

2.1.1 Komunikace Microsoft Kinectu s PC

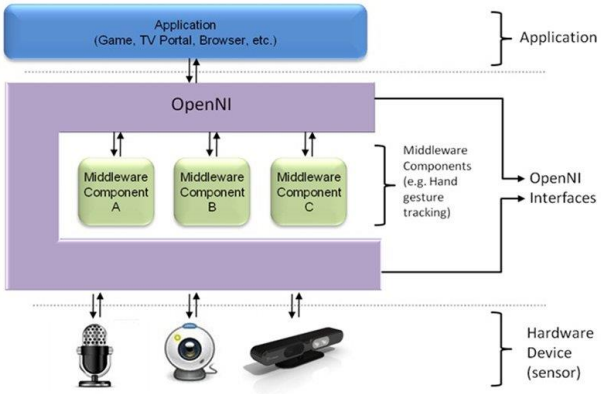
Ve své bakalářské práci jsem uvažoval komunikaci Kinectu pouze s PC s nainstalovaným systémem Linux.

Kinect se k PC připojuje přes USB. Uvnitř Kinectu je USB rozbočovač, ke kterému jsou připojeny následující části: *Xbox NUI Motor*, *Xbox NUI Camera* a *Xbox NUI audio*. Část *Xbox NUI Motor* ovládá kloubový motor a zároveň i LED diodu na čelní straně zařízení. *Xbox NUI audio* poskytuje data z integrovaných mikrofonů a poslední část *Xbox NUI Camera* zpřístupňuje data z hloubkového senzoru. K načtení těchto dat lze použít knihoven libfreenect a OpenNI.

Knihovna libfreenect je vyvíjena skupinou lidí pod projektem OpenKinect. Tato knihovna je open-source software a je možné ji používat v operačních systémech Windows, Linux a Mac.

Jak bylo zmíněno v kapitole 2.1, hlavní výpočetní jednotkou Kinectu je čip od společnosti PrimeSense. Tato společnost se po vzniku knihovny libfreenect rozhodla vydat vlastní knihovnu OpenNI vytvářenou stejnojmennou organizací OpenNI, jejímž cílem je rozvíjet možnosti přirozené

komunikace mezi člověkem a počítačem např. pomocí gest nebo různých pohybů. Knihovna OpenNI rovněž definuje rozhraní ke snímacím zařízením a zároveň definuje rozhraní k možným binárním blobům, kterým OpenNI říká middleware, což jsou části, ke kterým nejsou volně k dispozici zdrojové kódy.



Obrázek 3: Architektura OpenNI, převzato z [2]

Jedním z možných, a zároveň mnou používaným middlewarem, je NITE obsahující algoritmy pro rozpoznávání osob v hloubkových datech.

Při porovnání obou knihoven jsou rozdíly následující. Obě knihovny poskytují hloubková data ze senzoru k dalšímu zpracování. Knihovna libfreenect, na rozdíl od OpenNI v důsledku middlewarů (např. NITE), již nedokáže tato data dále zpracovávat. K tomu by bylo zapotřebí použít další knihovnu pro zpracování hloubkových dat, jako je např. PCL (této knihovně se více věnuje kapitola 3.2). OpenNI oficiálně nepodporuje zařízení Microsoft Kinect. Do oficiálních zdrojových kódů je proto nutné přidat modul pro jeho podporu (avin2)[2]. Další rozdíly mezi těmito knihovnami jsou shrnuty v následující tabulce.

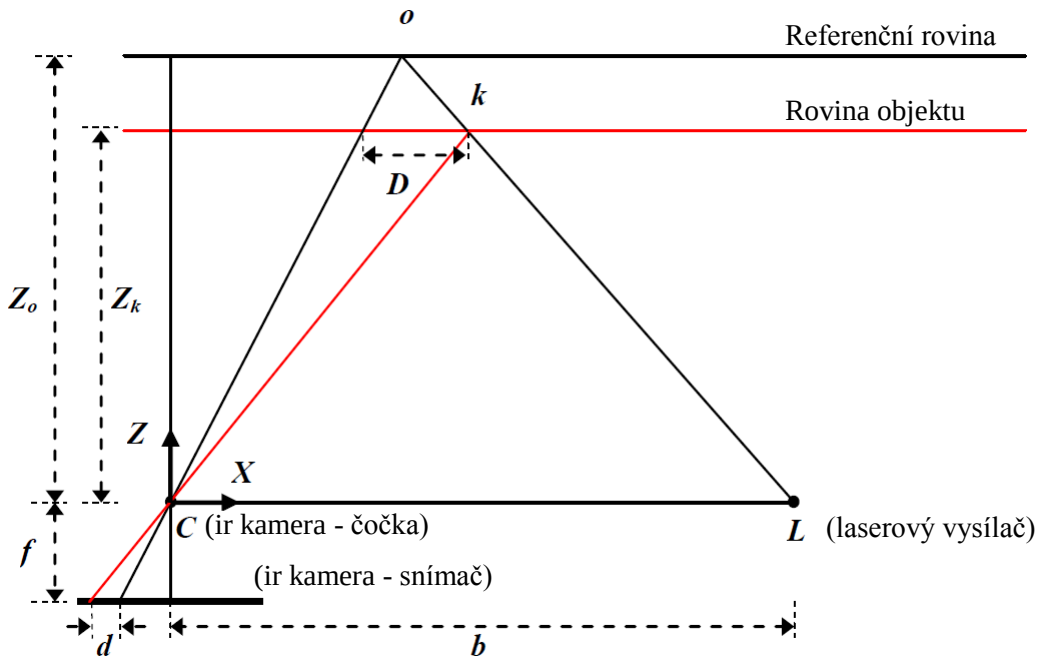
[2] Modul je dostupný na https://github.com/avin2/SensorKinect

	libfreenect	OpenNI
Podpora point cloud.	ANO	ANO
Podpora RGB kamery.	ANO	ANO
Podpora snímaní zvuku.	ANO	NE
Podpora mechanického kloubu.	ANO	NE
Podpora čelní LED.	ANO	NE
Algoritmy pro rozpoznávání objektů (middleware).	NE	ANO

Tabulka 1: Rozdíly mezi knihovnami libfreenect a OpenNI

2.1.2 Princip snímání hloubkových dat

Hloubková data jsou získáváná pomocí zařízení Kinect, obsahující infračervený laserový vysílač a infračervenou kameru. Kinect má také zabudovanou RGB kameru, která však pro tento účel není použita. Laserový vysílač emituje paprsek, který dále prochází difrakční mřížkou, čímž vznikne konstantní vzorek bodů projektovaných do scény. Ze scény odražený vzorek bodů je zachycen infračervenou kamerou a je porovnáván s referenčním vzorkem uloženým v paměti zařízení. Referenční vzorek představuje odraz od referenční roviny ve známé, výrobcem určené vzdálenosti od zařízení. Porovnání těchto vzorků (zachyceného a referenčního) je provedeno matematicky triangulační metodou, čímž jsou zjištěny vzdálenosti, v nichž se nachází každý bod odraženého vzorku. Podrobnější informace jsou v dokumentu [1].



Obrázek 4: Schéma triangulační metody, převzato z [1]

Veličiny D a d jsou na obrázku (Obrázek 4: Schéma triangulační metody) posuvy bodu k v rovině objektu a v rovině snímače vzniklé porovnáním s odrazem od referenční roviny a roviny objektu. Písmeno b označuje vzdálenost mezi osou kamery a osou laserového vysílače. Ohnisková vzdálenost ir kamery je označena písmenem f a Z_o je hloubka referenční roviny od kamery, která je uložena v paměti zařízení.

Výpočet hloubky Z_k bodu k je následující.

Z podobnosti trojúhelníků vyplývá:

$$\frac{D}{b} = \frac{Z_o - Z_k}{Z_o} \tag{1}$$

$$\frac{d}{f} = \frac{D}{Z_k} \tag{2}$$

Z obou rovnic lze vypočítat:

$$Z_k = \frac{Z_o}{1 + \frac{Z_o}{fb} d} \tag{3}$$

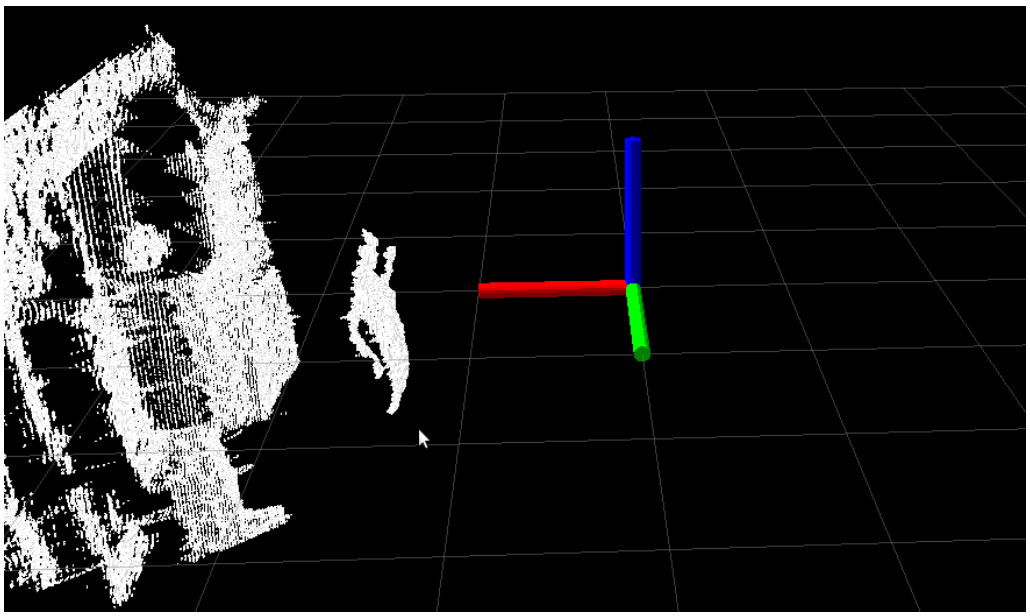
2.2 Podvozek – iRobot Roomba 530

Jedná se o upravený robotický vysavač iRobot Roomba 530. Jeho nádobka na nečistoty spolu s čistícími kartáči byly odebrány. Všechny robotické vysavače Roomba lze ovládat pomocí portu, který je skryt pod jeho horním krytem. Toto rozhraní je obyčejný sériový port pracující v TTL logice. Pomocí sériových příkazů přes tento port lze ovládat pohyb robota a načítat údaje ze všech jeho senzorů.

3 Rozpoznávání osob ve snímaném prostoru

Rozpoznávání osob ve snímaném prostoru probíhá na základě informací získaných Kinectem ve formě hloubkových dat (Point cloud). Hloubková data tvoří množinu multidimenzionálních bodů. Obvykle se jedná o třídimenzionální body, které tvoří navzorkovaný povrch objektů ve snímaném prostoru. Grafické vyjádření získaných dat ze senzoru je na obrázku (Obrázek 5: 3D vizualizace hloubkových dat ze senzoru). Přidáním barevné informace ke každému hloubkovému bodu, rovněž získané Kinectem, vznikne 4D point cloud.

V dalším procesu probíhá zjišťování osob v naměřených hloubkových datech. V této kapitole se budu věnovat knihovně PCL a middlewaru NITE pro detekci osob z hloubkových dat.



Obrázek 5: 3D vizualizace hloubkových dat ze senzoru

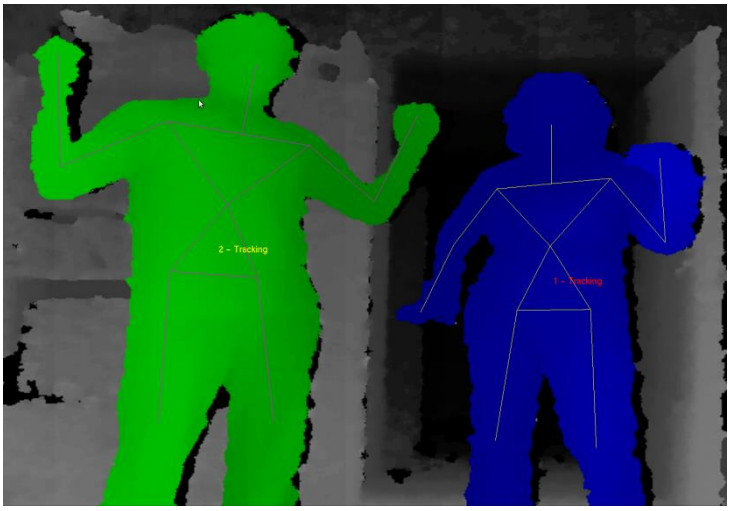
3.1 NITE

NITE je binární blob (middleware) pro knihovnu OpenNI zaměřený na rozpoznávání osob z hloubkových dat. Ačkoliv jsou zdrojové kódy knihovny OpenNI otevřené, tak tento binární blob (middleware) je dostupný pouze v binární podobě. Jeho implementace algoritmů pro rozpoznávání a detekci osob je skryta. Ani výrobcem není nikde zveřejněn přibližný popis principu tohoto middlewaru.

Při instalaci tohoto blobu je nutné zadat licenční klíč, který je veřejně známý. Pokud je blob NITE v pořádku nainstalován, lze přes rozhraní OpenNI používat třídu `UserGenerator`. V opačném případě je v době inicializace objektem `UserGenerator` vyvolána výjimka. Po úspěšném vytvoření instance třídy `UserGenerator` a následné inicializaci lze volat metodu `GetUsers()`. Tato metoda vrací přes své argumenty dvě proměnné. V první proměnné je uložen celkový počet detekovaných osob ve snímaném prostoru a ve druhé proměnné je vektor jejich pozic. Každé detekované osobě se navíc přiřadí jednoznačný identifikátor, který si osoba udržuje po celou dobu, kdy se pohybuje ve snímaném prostoru. Poté svůj identifikátor ztrácí.

NITE také dokáže na detekované osobě rozeznat jednotlivé části těla jako hlava, krk, trup, ruce a nohy (Obrázek 6: Kostra osob po provedení kalibrace). Nevýhodou ale je, že NITE v tomto případě vyžaduje počáteční kalibraci, kdy daná osoba musí zaujmout postoj Ψ (upaženo, pokrčené lokty, předloktí vzhůru). Tato kalibrace trvá cca 10 s v závislosti na vzdálenosti osoby od senzoru a okolních objektů. Pokud dojde ke krátkodobé ztrátě osoby ze snímaného prostoru, proces kalibrace se musí znovu opakovat. Pokud toto neprovedeme, NITE nám pouze poskytne množinu hloubkových bodů, které představují detekovanou osobu. Víme tedy, kde se jednotlivé osoby v prostoru nacházejí, ale již nejsme schopni určit jejich gesta. Na obrázku (Obrázek 6: Kostra osob po provedení kalibrace) jsou detekované dvě osoby a množina jim příslušejících hloubkových bodů (znázorněno zelenou a modrou barvou). Jejich kalibrací je vyhodnocena vnitřní kostra.

Pro potřeby mé bakalářské práce není proces kalibrace osoby nutný. Pro řízení robota je podstatné znát pouze polohu osoby v prostoru. Pokud bychom chtěli program rozšířit o možnost ovládání robota gesty tj. pohybem rukou, proces kalibrace by byl nezbytný.



Obrázek 6: Kostra osob po provedení kalibrace

3.2 Point Cloud Library

Tuto knihovnu jsem zkoušel jako další variantu k rozpoznávání osob místo middlewaru NITE.

Point cloud library (PCL) je knihovna pod BSD licencí, která obsahuje různé algoritmy pro pokročilé zpracování hloubkových dat, např. algoritmy pro filtrování, rekonstrukci povrchu, registraci a segmentaci. Knihovna PCL je plně integrována do ROS systému. Pro zjednodušení vývoje je kód PCL rozdělen do několika menších knihoven, které mohou být kompilovány samostatně, jako jsou např.:

- libpcl_filtrs – implementuje datové filtry jako podvzorkování, odstranění odlehlých hodnot, projekce atd.,
- libpcl_regitration – implementuje registrační algoritmy (ICP),
- libpcl_surface – implementuje techniky pro rekonstrukci povrchu, mřížkování atd.

Novým modulem, kterým byla rozšířena verze 1.7.0 je `libpcl_people`. Tento modul rozpoznává jednotlivé osoby v prostoru v reálném čase na základě naměřených RGB-D dat. RGB-D data jsou hloubková data rozšířená o barevnou informaci. Navržený způsob implementace modulu pracuje pouze tehdy, pokud je senzorem snímaná i podlaha, se kterou je v kontaktu detekovaná osoba.

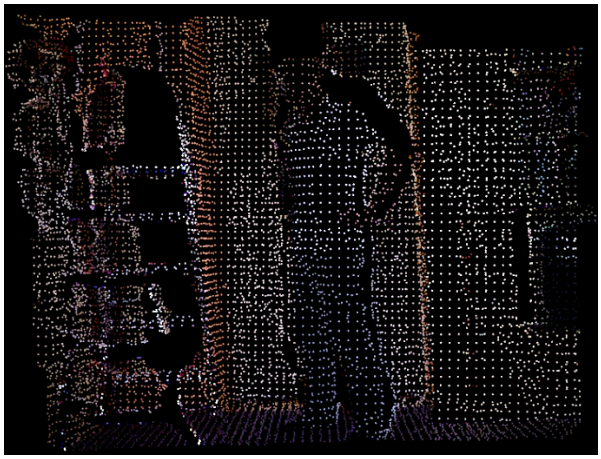
Algoritmus je rozdělen na dva moduly: detekční a sledovací. Detekční modul rozpoznává jednotlivé osoby ve vstupních datech a jejich polohu v prostoru předává sledovacímu modulu. Ten na základě předaných informací o pozici osoby predikuje jejich další pohyb. Predikce může být prováděna např. Kálmánovým filtrem (viz kapitola 6). Tento odhad pozice se v každém kroku koriguje na základě nově získané informace o poloze osoby z detekční části. V případě, kdy tuto

informaci z detekční části nedostaneme (krátkodobý výpadek), se následující pozice objektu pouze odhaduje bez následné korekce. Tím samozřejmě roste chyba (nejistota) o odhadnuté poloze sledovaného objektu, protože nedostáváme žádnou informaci o jeho chování. Pokud míra nejistoty o poloze objektu přesáhne uživatelem nastavenou hranici, objekt se prohlásí za ztracený a tím sledování osoby končí. První část detekčního modulu potřebuje ke své práci pouze naměřená hloubková data ze snímaného prostoru.

Dále budu diskutovat naměřená hloubková data znázorněná na obrázku (Obrázek 7: Znázorněná vstupní data).

Obrázek 7: Znázorněná vstupní data

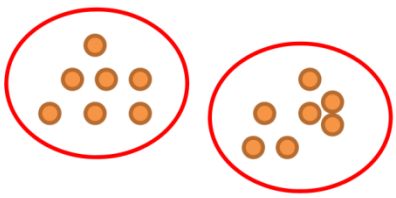
Nejdříve se vstupní hloubková data podvzorkují (down-sampling). Každý snímek prostoru je rozdělen na množinu voxelů (objemový pixel). Všechny body uvnitř každého voxelu jsou aproximovány do společného těžiště (Obrázek 8: Podvzorkovaní vstupního obrazu). Výchozí velikost voxelu jsem při testování zvolil o hraně 0,04 m. Tím se celkový počet hloubkových dat zredukuje, což umožní dosáhnout vyššího výkonu výpočtu při detekování osob v reálném čase. Další výhodou této operace je, že dostaneme hloubková data s přibližně konstantní hustotou, která není závislá na vzdálenosti objektu od snímače. V dalším kroku algoritmu se vychází ze samozřejmé skutečnosti, že se osoba pohybuje po vodorovném povrchu a můžeme tedy rovinu povrchu z hloubkových dat odstranit (Obrázek 9: Odstranění země). Koeficienty roviny, která je z hloubkových dat odstraněna, byly zadány při spuštění algortimu.



Obrázek 8: Podvzorkovaní vstupního obrazu

Obrázek 9: Odstranění země

Další částí algoritmu je provádění shlukové analýzy hloubkových dat z obrázku (Obrázek 9: Odstranění země). Shluková metoda slouží ke klasifikaci objektů, tj. sdružuje naměřené hloubkové body do skupin (shluků) tak, aby si jednotky náležející do stejné skupiny byly svými vlastnostmi nějak podobné (viz [3]). Shluková analýza vychází tedy z principu podobnosti. Vyčíslení této podobnosti je jedním ze základních problémů shlukové analýzy. Existuje mnoho způsobů, jak sestrojit ukazatele této podobnosti, např. ukazatel na principu euklidovské vzdálenosti, který je použit v této práci. Pro každý bod je hledán jeho nejbližší soused. Pokud tento nejbližší sousední bod leží v maximální nastavené euklidovské vzdálenosti r, je přidán do shluku (patří do stejné skupiny). Znázorněno na obrázku (Obrázek 10: Rozdělení bodů do skupin)



Obrázek 10: Rozdělení bodů do skupin, převzato z [4]

Postupné vyhledávání nejbližšího souseda daného bodu znamená vždy projít všechny naměřené body a porovnat je s tímto bodem, což je ale časově náročné. Proto jsou všechny naměřené body organizovány v KD stromu, který vyhledávání nejbližšího souseda zefektivňuje. KD strom je speciálním případem BVS (binární vyhledávácí strom), který provádí dělení scény střídavě podle x-ové a y-ové souřadnice (více ve zdrojích [5; 6]). Horní výpočetní složitost vytvořeného KD-stromu je $O(n\ log_2\ n)$, kde n je počet zadaných bodů.

Další dva parametry, které se nastavují u shlukové analýzy, jsou minimální a maximální počet bodů ve skupině. Předpokládá se, že snímaná osoba zaujímá určitý prostor, a proto musí vytvářet shluk s vyšším počtem bodů. Proto je možné menší shluky neuvažovat a již se s nimi dále nezabývat. Obdobně se nepředpokládá, že by osoba zabírala celou scénu. Z tohoto důvodu lze shluky s vysokým počtem bodů rovněž eliminovat. Velikosti těchto parametrů jsou závislé na nastavené velikosti voxelů. Následující obrázek (Obrázek 11: Výsledek shlukové analýzy) zobrazuje všechny vytvořené shluky po provedení shlukové analýzy na datech z obrázku (Obrázek 9: Odstranění země). Lze si všimnout, že tam chybí obrázek odpovídající levé krajní zdi, což je způsobeno tím, že neprošel požadavkem na maximální počet prvků ve shluku. Tento shluk byl algoritmem vyřazen a dále neuvažován.

(a) Žebřík (b) Osoba (c) Skříň

Obrázek 11: Výsledek shlukové analýzy

Nyní následuje poslední fáze, a to rozhodnout, ve kterém z těchto shluků se nachází osoba. Základem rozpoznání osoby je extrakce příznaků z RGB dat, které poté vytvoří výsledný deskriptor. K tomu je použita metoda založená na histogramu orientovaných gradientů (HOG) [7]. Gradient představuje směr změny intenzity barvy v obraze. Skládá se ze dvou složek, a to směru a velikosti. Tyto gradienty bývají velmi silné hlavně na hranách objektů. Na obrázku (Obrázek 12: Zobrazení gradientů) je vidět, že gradienty jsou především silnější na konturách (obrysech) lidského těla (hlava, ramena a dolní končetiny). Velikost gradientu je zjištěna konvolucí obrázku s derivační maskou $[-1,0,1]$ v horizontálním a vertikálním směru. Vstupní obrázek je upraven do rozlišení 128 x 64 bodů a poté je ještě rozdělen na síť buněk o stejné velikosti. V každé takové buňce je spočítán histogram orientovaných gradientů. Kombinace těchto histogramů tvoří výsledný HOG deskriptor. Posledním

krokem je konečná klasifikace s využitím tohoto HOG deskriptoru. K tomu je použit klasifikátor využívající podpůrné vektory (SVM). Výsledkem je konkrétní hodnota pravděpodobnosti, zda se v daném shluku nachází osoba. Je-li tato hodnota pod nastavenou hodnotou pravděpodobnosti, je tento shluk prohlášen za osobu a poloha této osoby je dále předána sledovací části tvořené Kálmánovým filtrem.

Obrázek 12: Zobrazení gradientů

4 Lokalizace zdroje zvuku

V situacích, kdy se osoba dostane mimo snímaný prostor senzoru, je potřebné definovat následné chování robota. Strategií může být více. Nejjednodušším řešením je, že se robot zastaví a bude čekat, dokud se daná osoba opět nepostaví do snímaného prostoru. Dalším řešením může být jedna otočka robota o 360^{o}, případně z posledních validních dat pozici dané osoby predikovat nebo návrat robota vždy na definované počáteční místo apod. Další možností řešící tyto situace může být použití integrovaného mikrofonového pole Kinectu, kdy se po vygenerování akustického signálu, např. písknutím, osoba pro robota zviditelní. Vyhodnotí se úhel příchodu akustického signálu a podle toho dojde k natočení snímacího senzoru. Princip mnou použité metody spočívá ve vyhodnocení posuvu příchodu signálů na jednotlivé mikrofony (Time Difference of Arrival - TDOA). Tato metoda funguje na jednoduchém principu, a proto nedokáže lokalizovat více akustických zdrojů ve stejném čase.

4.1 Výpočet posuvu signálu mezi mikrofony

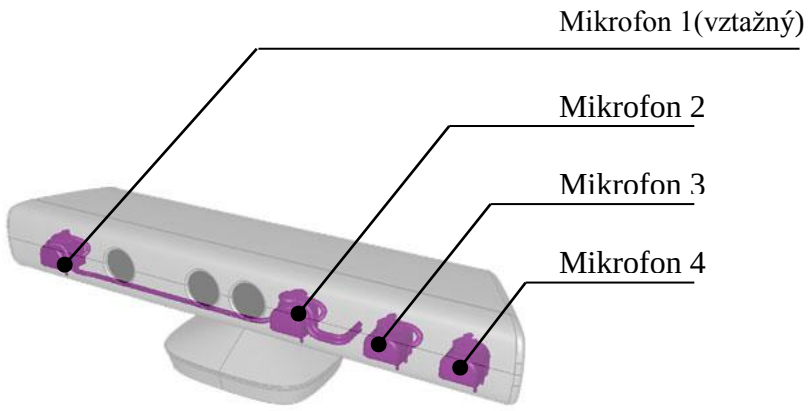
K získání TDOA potřebujeme obdržet signál alespoň ze dvou mikrofonů. Podstatné je, aby vzorkování na mikrofonech probíhalo současně. Pokud máme k dispozici více mikrofonů, zpravidla

z nich určíme jeden mikrofon jako vztažný, vůči kterému se vypočítává posuv ze zbylých mikrofonů. V první části metody se navzorkovaný signál z jednotlivých mikrofonů ukládá do vektorů. Počet vzorků v každém vektoru musí být stejný. Délka těchto vektorů se zpravidla pohybuje v řádu stovek vzorků.

Nechť vektor a označuje naměřené vzorky signálu na vztažném mikrofonu a vektor b označuje vektor vzorků naměřený na druhém mikrofonu. Délka vektorů a a b bude označována písmenem l. Parametr S_{max} vyjadřuje maximální možný posuv mezi mikrofony (v počtu vzorků). Tento parametr je závislý na vzorkovací frekvenci A/D převodníku a vzdálenosti mezi mikrofony. Pokud uvažujeme, že šíření akustické vlny ve vzduchu při 20 C° je 343,71 m.s^{-1}, lze parametr S_{max} vypočítat rovnicí (4).

$$S_{max} = f\frac{D}{343.71} \qquad (4)$$

V rovnici (4) přestavuje D vzdálenost mezi mikrofony a f označuje vzorkovací rychlost A/D převodníku. Konkrétní hodnoty pro zařízení Kinect jsou zobrazeny v následující tabulce 2.



Obrázek 13: Umístění mikrofonů v zařízení Kinect

	Vzdálenosti ke vztažnému mikrofonu (mikrofon 1)	S_{max}
Mikrofon 2	15 cm	7
Mikrofon 3	17 cm	8
Mikrofon 4	19 cm	9

Tabulka 2: Vzdálenosti mikrofonů

Pokud použijeme hodnoty z výše uvedené tabulky a dosadíme je do rovnice (4), dostaneme hodnoty pro každý mikrofon uvedené ve sloupci S_{max}. Vzorkovací rychlost A/D převodníku v zařízení Kinect je 16 KHz. To znamená, že při těchto parametrech nemůže u mikrofonu 2 vůči vztažnému mikrofonu dojít k většímu posuvu (ve vzorcích) signálu než právě o 7 vzorků. Algoritmus porovnání vektorů a a b je podle [8] následující. Ze vztažného signálu na mikrofonu 1 (vektor a) se vybere jeho prostřední část tj. $\left[a_{S_{max}}, a_{S_{max}+1}, a_{S_{max}+2}, \dots, a_{l-S_{max}-1}\right]$ a tím vznikne nový vektor, který je označený jako $\bar{a}$. Délka tohoto nového vektoru je rovna $l - 2S_{max}$ a je označena písmenem d. Indexování vektorů je bráno od nuly. Ze signálu z druhého mikrofonu (vektor b) se vybere část v závislosti na i-tém kroku algoritmu tj.$[b_i, b_{i+1}, b_{i+2}, \dots, b_{l-S_{max}+i}]$, který je označen $\bar{b}(i)\ kde\ i = 0, \dots, 2S_{max}$. Délka vektoru $\bar{b}(i)$ je stejná jako délka vektoru $\bar{a}$. V každém i-tém kroku algoritmu se vypočítá kriteriální funkcí (viz. rovnice 5 nebo 6) shoda vektorů $\bar{a}$ a $\bar{b}(i)$. Hodnota této shody se vypočítá pro všechna zvolená i a vybere se z nich číslo i takové, kde je hodnota shody $k(i)$ nejmenší resp. největší. To je závislé na typu zvolené kriteriální funkce. U rovnice (5) se hledá minimum vzhledem k i a u rovnice (6) se hledá maximum vzhledem k i. Poté $i - S_{max}$ představuje posuv signálů (ve vzorcích) mezi vztažným a druhým zvoleným mikrofonem.

Součet rozdílů vektorů

$$k(i) = \sum_{j=0}^{d} \left|\bar{a}_j - \bar{b}(i)_j\right| \tag{5}$$

Skalární součin

$$k(i) = \sum_{j=0}^{r} \bar{a}_j \bar{b}(i)_j \tag{6}$$

Vypočítaná hodnota posuvu mezi signály se může v případech, kdy je zdroj zvuku daleko od mikrofonů nebo je rušen jiným zdrojem, v krátkém časovém intervalu razantně měnit. Toto chování znamená, že dostáváme invalidní údaje a nemůžeme vypočítaný posuv mezi signály považovat za platný. Tento problém je řešen výpočtem směrodatné odchylky. Pokud je odchylka posledních vypočítaných hodnot za daný čas pod nastaveným limitem, jsou výsledky posuvů zprůměrovány a prohlášeny za platné. Další případ, který může nastat je, že dostáváme na vstup signál o velmi malé amplitudě. Z takovéhoto signálu nelze spolehlivě vypočítat posuv, a proto je prohlášen za neplatný. Problém je řešen nastavením minimální úrovně amplitudy, kterou musí signál dosáhnout, aby bylo možné zahájit výpočet posuvu mezi signály. Tento případ nastává, když není generován žádný zvuk nebo zdroj zvuku se nachází mimo směrovou charakteristiku mikrofonu.

4.2 Nejednoznačnosti při výpočtu posuvu mezi signály.

Při posuzovaní posuvu mezi signály může dojít u harmonických signálů k nejednoznačnostem. Uvažujme čistě harmonický signál o vlnové délce λ. Snímacím zařízením bude Kinect, kde vzdálenost mezi vztažným a druhým mikrofonem je 15 cm (Tabulka 2: Vzdálenosti mikrofonů). Pokud vlnová délka tohoto generovaného harmonického signálu bude mít vlnovou délku menší jak 0,075 m, potom nastávají alespoň tři možnosti posuvu signálu mezi nimi. Při dvojnásobné vlnové délce tj. 0,15 m dostáváme dvě možnosti určení posuvu. Abychom dostali jednoznačné určení posuvu mezi signály, musí platit, že vlnová délka harmonického signálu bude dvakrát delší než vzdálenost mezi mikrofony, tj. bude platit $vzdálenost\ mezi\ mikrofony < \frac{\lambda}{2}$. V praktickém použití se ovšem s těmito problémy téměř nesetkáme, jelikož příchozí zvuky jsou charakteru hlukového nikoliv periodického.

4.3 Výpočet směru zdroje akustických signálů

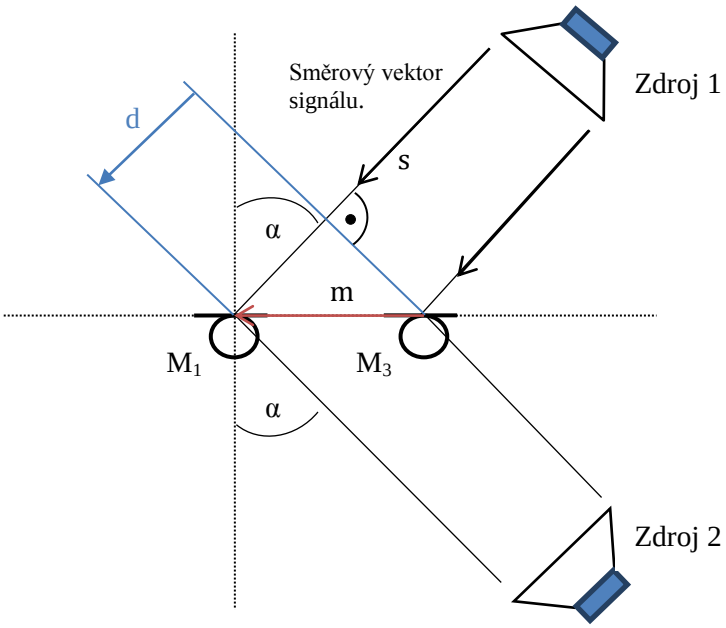
Dříve než začnu blíže popisovat specifikaci zvolené metody pro lokalizaci zdroje akustických signálů, chtěl bych popsat předpoklady, ze kterých jsem vycházel:

- Zdroj signálu leží v rovině senzorů.
- Prostředí je z hlediska šíření signálu homogenní.
- Vyzařované vlnoplochy, které dopadají v blízkosti senzorového pole, mohu aproximovat přímkou (far field situation). Tento předpoklad je ovšem správný, pokud je vzdálenost zdroje zvuku od senzorového pole mnohem delší než vzájemná vzdálenost mezi mikrofony.

Přesnost výpočtu směru akustických zvuků závisí také na zvolené geometrii mikrofonního pole [9]. Základní typy rozmístění mikrofonů jsou následující:

- Lineární – všechny mikrofony se nacházejí v jedné řadě. Mají-li mikrofony mezi sebou konstantní vzdálenost, jedná se o rovnoměrné lineární pole ULA (Uniform Linear Array). U zařízení Kinect se jedná o nerovnoměrné lineární pole, jelikož jsou vzdálenosti mezi mikrofony různé (Obrázek 13: Umístění mikrofonů v zařízení Kinect). Nevýhoda lineárního rozmístění je v tom, že lze lokalizovat zdroj pouze v rozsahu 0° až 180° (Obrázek 14: Prostorová nejednoznačnost lineárního mikrofonového pole).

- Rovinné – v této geometrii mají mikrofony plošné rozmístění. Nejčastěji se používá pravoúhlé. Toto rozmístění dokáže lokalizovat zdroj v rozsahu 0° až 360°.
- Prostorové – V této geometrii jsou mikrofony rozmístěny např. na vrcholech krychle. V tomto uspořádání lze lokalizovat zdroj ve 3D prostoru výpočtem jeho azimutu a elevace.



Obrázek 14: Prostorová nejednoznačnost lineárního mikrofonového pole

V následujícím textu se zabývám pouze lineárním geometrickým mikrofonním polem, které je použito v zařízení Kinect. Úhel příchodu akustického signálu DOA (Direction of Arrival) je vztažen ke kolmici k mikrofonnímu poli. Dle úhlu α a za předpokladu, že mikrofon M_1 je v počátku pravoúhlých souřadnic, lze obrácený jednotkový směrový vektor akustického signálu vyjádřit jako $s = [\sin(\alpha), \cos(\alpha)]$. Vektor spojující mikrofony M_1 a M_2 je označen m. Hledaná velikost vektoru d se určí skalárním součinem vektorů s a m:

$$d = \langle s, m \rangle \tag{7}$$

Časové zpoždění signálu mezi mikrofony M1 a M2 (TDOA) dostaneme vydělením vzdálenosti d rychlostí šíření akustického signálu:

$$TDOA = \frac{1}{c}[\langle s, m \rangle] \tag{8}$$

Při znalosti pozice mikrofonů v zařízení Kinect a vypočítaného TDOA podle kapitoly 4.1 jsme schopni inverzním postupem zjistit azimut zdroje akustických signálů, tj. po vyjádření úhlu α dostaneme z rovnice (8) následují vztah:

$$\alpha = \sin^{-1} \frac{c * TDOA}{0.15} \tag{9}$$

Hodnota 0,15 v rovnici (9) představuje vzdálenost v metrech (Tabulka 2: Vzdálenosti mikrofonů) mezi vztažným mikrofonem a mikrofonem s indexem dva.

5 Řízení pohybu robota

Cílem jednotky pro řízení robota je zajistit jeho plynulý pohyb. Řeší situace, kdy je robot v pohybu a dojde ke ztrátě sledovaného cíle. V této situaci se nesmí robot prudce zastavit, ale musí zpomalovat rovnoměrně. Dalším důležitým parametrem je maximální akcelerace otáčení robota. Při vysoké akceleraci ve směru otáčení dojde k rozmazání scény, která je snímána kamerou, a to způsobí, že algoritmus pro rozpoznávání osob nebude v takto rozmazané scéně schopen osobu detekovat. Důležité je řídit pohyb robota tak, aby k těmto jevům nedocházelo. Prvkem pro řízení přímého a otočného pohybu je PID regulátor.

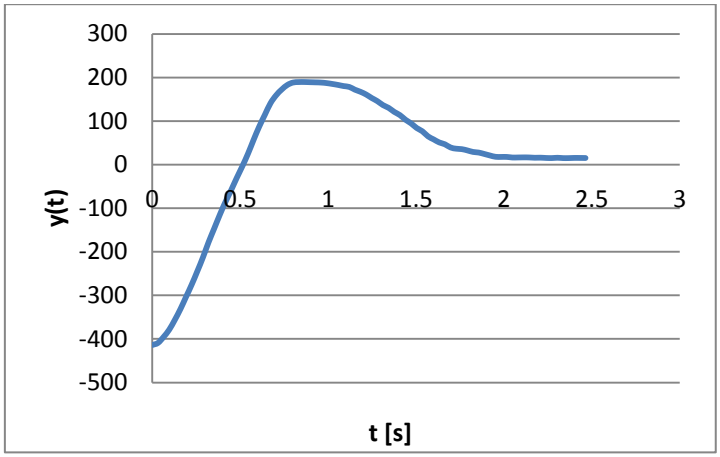
5.1 PID

PID je spojitý regulátor, který patří do kategorie zpětnovazebních regulátorů. Je složen z proporcionální, integrační a derivační složky. Všechny tyto složky se sčítají a tvoří výslednou akční veličinu. Podstatná činnost regulátoru spočívá ve vyhodnocení regulační odchylky $e(t) = w(t) - y(t)$ s cílem vytvořit takový výstupní signál akční veličiny, aby tato odchylka byla eliminována nebo aby byla co nejmenší. Základní rovnice PID regulátoru je následující:

$$u(t) = r_0 e(t) + r_i \int_0^t e(t)dt + r_d \frac{de(t)}{dt} \tag{10}$$

Podle konstant r_0, r_i, r_d, které položíme nule, dostáváme základní druhy regulátorů: P regulátor, I regulátor a kombinované regulátory, kterými jsou PD regulátor, PI regulátor a PID regulátor. K regulování pohybu robota je používán PI regulátor. Jeho vlastnosti pro řízení pohybu robota jsou ideální. V počátku regulačního procesu převládá proporcionální složka, díky které se robot rychle uvede do pohybu. Při přibližování se požadované hodnotě začne převládat vliv integrační složky a ta

zajistí rovnoměrné zpomalení. Bez této integrační složky by došlo při dosažení požadované hodnoty k prudkému zastavení a následné ztrátě sledovaného cíle vlivem prudké změny rychlosti. Toto rovnoměrné zpomalení je doprovázeno jedním překmitem regulované veličiny (Obrázek 15: Průběh regulace při otáčení). Nastavovat regulátor tak, aby k překmitu nedošlo je nežádoucí.



Obrázek 15: Průběh regulace při otáčení

Pro řízení a regulaci v diskrétních systémech se používá PSD regulátor. Jedná se o číslicový regulátor, který je analogický ke spojitému PID regulátoru. PSD regulátor nevyhodnocuje informaci spojitě, ale v diskrétních okamžicích s určitou periodou vzorkovaní. Využívá se několik základních způsobů diskrétních náhrad spojitých algoritmů integrace a derivace. Náhradou integrace je:

- Zpětná obdélníková metoda (ZOBD)

$$I(kT) = T\sum_{i=1}^{k} e(iT) \tag{11}$$

- Stupňovitá náhrada dopředu (DOBD)

$$I(kT) = T\sum_{i=0}^{k-1} e(iT) \tag{12}$$

- Sečná náhrada (LICHO)

$$I(kT) = T\sum_{i=1}^{k} \frac{e(iT) + e[(i-1)T]}{2} \tag{13}$$

Náhrada derivace se provádí zpětnou diferencí:

$$D(kT) = \frac{e(kT) - e[(k-1)T]}{T} \tag{14}$$

Pokud nahradíme v rovnici (10) integrační člen zpětnou obdélníkovou metodou a derivační člen zpětnou diferenci, dostáváme následující polohový algoritmus PSD regulátoru:

$$u(kT) = e(kT) + T\sum_{i=1}^{k} e(iT) + \frac{e(kT) - e[(k-1)T]}{T} \tag{15}$$

Tento polohový algoritmus PSD regulátoru je také použít v mé bakalářské práci pro řízení pohybu robota. Další možností by bylo upravit algoritmus (15) do přírůstkového tvaru, kde se neurčuje celá hodnota akční veličiny v daném okamžiku, ale pouze její přírůstek. Podrobnosti jsou uvedeny v [10].

Pokud se týče nastavení správných parametrů pro PID (PSD) regulátor, existují různá pravidla pro jejich seřizování, jako jsou např. pravidla Zieglera a Nicholse, představující systematický postup návrhu parametrů. Toto je popsáno v [10]. Já jsem postupoval u svého robota metodou pokus – omyl do doby, než jsem docílil regulace o potřebné kvalitě.

5.2 Odometrie

Odometrický systém se používá k zjišťování polohy robota. Nejběžnější princip měření je založen na optické závoře. Na hnacích kolech robota jsou clony s okénky, které střídavě protínají optický paprsek. Dle počtu přerušení optického paprsku lze vyhodnotiti počet otáček hnacího kola a tím určit ujetou vzdálenost. Podvozek robota je vybaven diferenciálním pohonem tvořeným dvěma koly na společné ose. Souřadnice pro určení polohy robota lze vyjádřit vztahy:

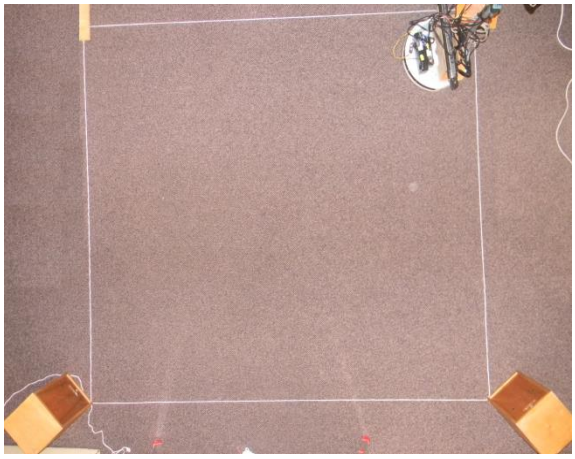
$$\alpha(t) = \alpha(t-1) + (\frac{\Delta s_r - \Delta s_l}{\pi l} * 360) \tag{16}$$

$$x(t) = x(t-1) + \frac{\Delta s_r - \Delta s_l}{2}\cos(\alpha(t) - \alpha(t-1)) \tag{17}$$

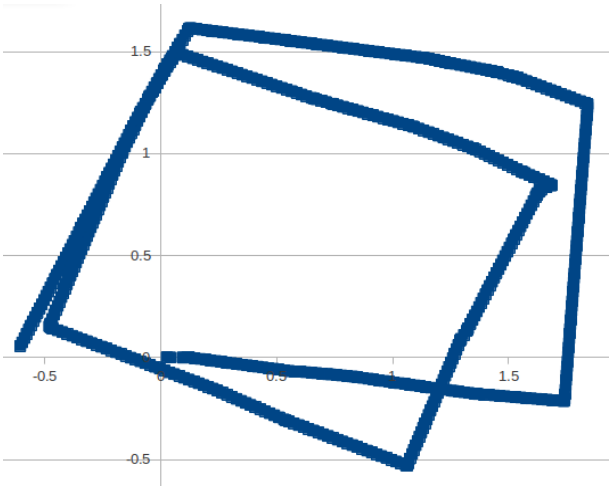
$$y(t) = x(t-1) + \frac{\Delta s_r - \Delta s_l}{2}\cos(\alpha(t) - \alpha(t-1)) \tag{18}$$

Konstanta l ve vzorci (16) je rozteč mezi koly, Δs_r a Δs_l jsou ujeté vzdálenosti na pravém a levém kole vztažené k poslednímu měření. Výsledkem je relativní poloha robota $[x(t),\ y(t)]$ a relativní úhel natočení $\alpha(t)$ v čase t. Tato relativní poloha určuje pouze změnu od poslední platné polohy. Pokud chceme znát absolutní pozici robota, musíme tyto relativní změny sčítat. Nevýhoda tohoto přístupu spočívá v tom, že každé měření relativní polohy je zatíženo určitou chybou v podobě

prokluzu kol vůči povrchu. Pokud tuto chybu nebudeme v průběhu měření korigovat, dostaneme absolutní polohu robota se stále větší chybou. Chyba absolutní polohy u odometrického systému je demonstrovaná dvojitým objezdem čtverce na obrázku (Obrázek 17: Data z odometrického systému).



Obrázek 16: Testovací objezd



Obrázek 17: Data z odometrického systému

Přesnějším měřením absolutní polohy se zabývá bakalářská práce Romana Bedroše [11], kde se za pomocí laserového snímače a ICP algoritmu minimalizuje nepřesnost odometrického systému. Ve své bakalářské práci využívám odometrický systém pouze pro zjišťování relativní polohy, a proto se chybami v absolutní poloze nezabývám.

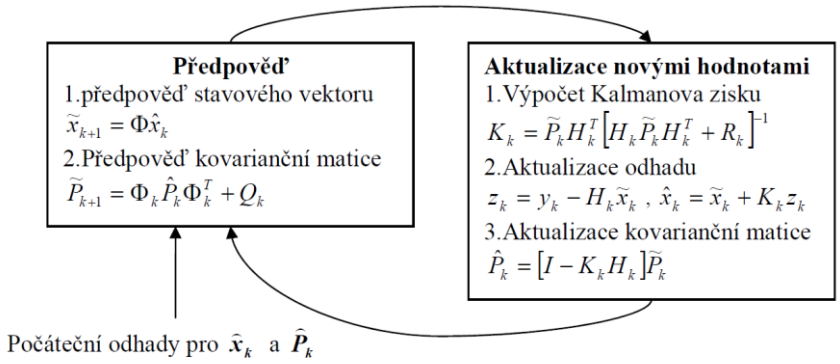
6 Kálmánův filtr

Kálmánův filtr je nástroj pro odhad stavu lineárního systému. Tato technika byla vyvinuta v 60. letech 20. století a našla uplatnění převážně při sledování objektů a v aplikacích počítačového vidění.

Tento filtr byl v mé práci použit pro trasování pohybu osoby k detekční části popsané v kapitole 3.2. Pozice osoby určená v detekční části je zatížena chybami a výpadky měření. Proto je nutné k dosažení dynamiky celého systému tak, aby pokryl tyto výpadky, provádět predikci sledované osoby na základě znalostí předchozího pohybu.

Kálmánův filtr je složen ze dvou kroků: aktualizace filtru a předpovědi nových hodnot dle obrázku Obrázek (18: Diagram rekurze). Jsou-li získány nové hodnoty o poloze osoby, dojde k provedení aktualizace filtru. Poté je provedena následná predikce nového stavového vektoru a nové kovarianční matice. Tento postup se pokaždé s nově příchozími informacemi opakuje. V případě, že informace o poloze osoby neobdržíme, dochází k predikci bez následné korekce. To se projeví zvětšením hodnot prvků kovarianční matice a to znamená zvýšení neurčitosti predikce. Výhoda Kálmánova filtru spočívá v jeho rekurzivní struktuře, a proto kromě aktuálního vstupu a předchozího stavu filtru nejsou žádné další informace z minulosti nutné.

Při vytváření Kálmánova filtru jsem vycházel z diplomové práce Ing. Igora Potůčka, Ph.D. [12].



Obrázek 18: Diagram rekurze převzatý z [12]

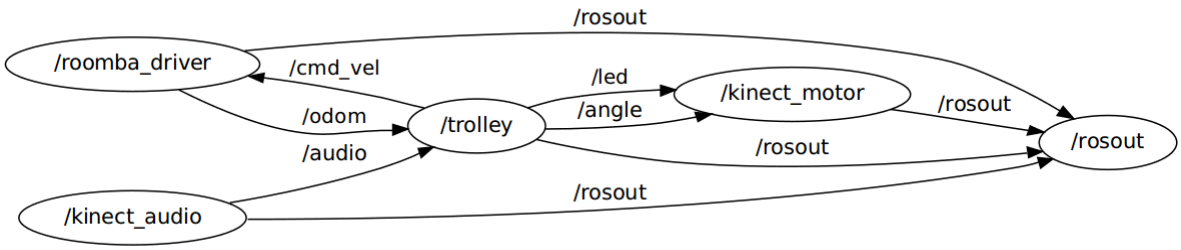
7 Popis softwaru robota

Software robota je implementován v Robotic Operating System (ROS) ve verzi electric. Tato verze byla zvolena proto, aby byla v souladu s ROS používaným na školním robotu. Programy pro ROS je možné vytvářet v několika jazycích jako C++, Python a lisp. Pro svou práci jsem zvolil jazyk C++. Podrobnější informace k ROSu nejsou v mé práci zahrnuty a naleznete je v [13].

Software robota je koncipován do čtyř ROS balíčků: roomba_driver, kinect_motor, kinect_audio a trolley. Každý z těchto balíčků vytváří jeden proces, který je v rámci ROSu nazýván uzlem. Tyto uzly mezi sebou vzájemně komunikují zasíláním zpráv. Každá zpráva je adresovaná jménem komunikačního kanálu a je doručena všem uzlům, které si jméno tohoto komunikačního kanálu zaregistrují. Jedná se tedy o jednosměrný přenos zpráv. Uzlu, který zprávy do komunikačního kanálu odesílá se říká vydavatel (publisher) a tomu, který zprávy přijímá se říká odběratel (subscriber). Na jednom komunikačním kanálu může být zaregistrováno souběžně i více vydavatelů

a odběratelů zpráv. Zprávy mají pevně definovanou datovou strukturu, která se skládá z kombinací jednoduchých datových typů a polí.

Na obrázku (Obrázek 19: Schéma SW robota) jsou graficky zobrazeny všechny používané uzly a jejich vazby.



Obrázek 19: Schéma SW robota

7.1 Konfigurační soubor

Hodnoty nastavitelných parametrů pro robota jsou uloženy v konfiguračním souboru ve formě xml. Cesta k tomuto souboru je předána uzlu `trolley` prostřednictvím vstupního parametru. Tento soubor je rozdělen na čtyř části.

První část:

```
<Control>
    <PID p="0.0005" i="0.00013" d="0.0">Linear</PID>
    <PID p="0.0021" i="0.0011" d="0.0">Angle</PID>
    <Distance>1200</Distance>
</Control>
```

V této části se konfiguruje PSD regulátor pro lineární a úhlový směr robota. Tag *Distance* označuje vzdálenost, kterou si má robot udržovat od sledované osoby.

Druhá část:

```
<Motor>
    <MaxSpeed value="2.0">Linear</MaxSpeed>
    <MaxSpeed value="2.0">Angle</MaxSpeed>
    <MaxAcceleration value="2.0">Linear</MaxAcceleration>
    <MaxAcceleration value="2.0">Angle</MaxAcceleration>
    <MotorTopic>cmd_vel</MotorTopic>
    <PositionTopic>odom</PositionTopic>
</Motor>
```

Tato část konfiguruje vlastnosti podvozku, jeho maximální rychlost a zrychlení, a to v lineárním a úhlovém směru. Dále se zde definují názvy komunikačních kanálů pro zasílání zpráv podvozku a kanál pro příjem odometrických zpráv.

Třetí část:

```
<Audio>
    <Topic>audio</Topic>
</Audio>
```

Zde se konfiguruje název komunikačního kanálu pro příjem audio dat.

Čtvrtá část:

```
<Strategy>
        <Type>1</Type>
</Strategy>
```

V poslední části se volí typ strategie při ztrátě sledované osoby. Je definován rozsah 1-3, kde volba 1 určuje naslouchání z mikrofonů (kapitola 4), volba 2 určuje jednu otočku při hledání ztracené osoby a při volbě 3 se vyčkává, než se osoba vrátí do snímaného prostoru.

7.2 Uzel *trolley*

Jedná se o uzel, který se stará o ovládání celého robota. Pomocí knihovny OpenNI jsou načtena data z Kinectu a zpracována middlewarem NITE. Uzel také obsahuje PSD regulátor pro řízení pohybů robota. Vyhodnocuje úhel příchodu akustického signálu z dat poskytnutých uzlem `kinect_audio` metodou popsanou v kapitole 4.3.

7.3 Uzel *kinect_motor*

Zajišťuje horizontální naklápění Kinectu a zároveň ovládá barvu LED na čelní straně zařízení na základě přijatých zpráv z komunikačních kanálů `/led` a `/angle`. Čelní LED a elektronický kloub Kinectu jsou řízeny knihovnou libfreenect. Význam LED spočívá v signalizaci, že robot detekoval osobu a je připraven ji sledovat.

7.4 Uzel *kinect_audio*

Je to uzel, kterým jsem rozšířil zadání úlohy (vlastní implementace a modifikace). Trvale snímá okolní akustické signály pomocí knihovny libfreenect a zasílá je komunikačním kanálem `/audio` k vyhodnocení do uzlu `trolley`. Zmíněným komunikačním kanálem se přenášejí data ze všech čtyř mikrofonů současně, což má za následek zvýšený přenos dat do uzlu `trolley`. Výhodnější by bylo realizovat výpočetní část již v uzlu `kinect_audio` a komunikačním kanálem přenášet pouze vypočtenou hodnotu o poloze cíle. Bohužel uzel `trolley` jsem programoval jako první uzel systému, takže toto řešení bylo zachováno.

7.5 Uzel *roomba driver*

Tento uzel zajišťuje mechanické ovládání podvozku robota a to pohyb dopředu, dozadu a otáčení v obou směrech. Podrobné pokyny o směru pohybu, lineární a úhlové rychlosti jsou uzlu předávány uzlem `trolley` prostřednictvím komunikačního kanálu `/cmd_vel`. Uzel komunikuje s podvozkem robota komunikačním protokolem výrobce pomocí sériového portu. Tímto portem jsou rovněž načítány informace o poloze robota pomocí odometrie (viz kapitola 5.2) a zasílány dále komunikačního kanálem `/odom` do uzlu `trolley`. Podrobnosti o komunikačním protokolu jsou dostupné na stránkách výrobce[3].

8 Poznatky při testování robota

Jak již bylo popsáno v kapitole 2, není mnou vytvořený robot zcela identický se školním robotem TB2. Hlavní rozdíly mezi nimi jsou v celkové váze a výšce umístění Kinectu. Hmotnost školního robota oproti mému je o několik kg vyšší. Překvapilo mě, jaký vliv to má na celkovou kvalitu regulace jeho pohybu. Při zachovaní stejné proporcionální a sumační složky u PSD regulátoru, kterou jsem měl nastavenou pro svého robota, došlo u školní verze k nestabilní regulaci. Bylo nutné zmenšit proporcionální a sumační složky, čímž došlo ke stabilní regulaci, ale také ke zhoršení regulačního pochodu oproti mému robotu. Dosažení požadované hodnoty tak trvá o něco déle.

Dalším rozdílem mezi oběma roboty je výška umístění Kinectu. U svého robotu jsem se snažil umístit střed vodorovné osy Kinectu do výšky přibližně svého pasu. V případě školní verze je Kinect

[3] http://www.usna.edu/Users/weapsys/esposito/roomba.matlab/Roomba_SCI.pdf

umístěn níž. Svou variantu považuji za výhodnější, neboť v případě sledování osoby nacházející se ve větší vzdálenosti není nutné ve velké míře využívat vertikálního naklápění Kinectu.

Svůj výrobek jsem testoval v různých prostorách. Měl jsem určité problémy v uzavřených místnostech s větším počtem kusů nábytku. Middleware NITE, který jsem použil pro detekci lidí, v některých případech chybně vyhodnotil předmět (nábytek, okno atd.) prostoru jako osobu, zvláště při velmi rychlé změně polohy snímané osoby. Domníval jsem se, že problém bych odstranil nahrazením knihovny PCL middlewarem NITE. Projevily se však pro mé potřeby negativní vlastnosti metody pro detekci lidi obsažené v knihovně PCL (popsal jsem v závěru), a proto jsem middleware NITE ponechal.

Další problémy se projevily při snímání zvuku mikrofony Kinectu. Při zkouškách jsem zjistil, že jejich směrová charakteristika je ledvinová. Jelikož jsou všechny mikrofony umístěny v jedné přímce ve stejném směru orientace, dochází k velmi špatné detekci zvuků přicházejících zezadu.

9 Závěr

Bakalářská práce byla zpracována podle zadání na téma robotický nákupní košík. V rámci tohoto úkolu jsem studoval potřebnou literaturu související s problematikou počítačového vidění, detekce osob a robotického operačního systému ROS.

Úkolem bakalářské práce bylo zhotovení řídícího softwaru pro ovládání školního robotického zařízení. Vlastní zhotovení takového zařízení nebylo součástí zadání práce a školní robot byl samozřejmě pro každého řešitele kdykoliv bez problému k dispozici. Já jsem však považoval pro úspěšnější zvládnutí úkolu za vhodné sestrojit si svoji vlastní verzi robota včetně zajištění všech potřebných mechanických prvků, tj. snímacího a pohybového aparátu včetně nosných a upevňovacích částí. Jelikož můj robot nebyl a ani nemohl být zcela identický s robotem školním (rozdíl v hmotnosti a do jisté míry i v konstrukčním uspořádání), bylo nutné vyřešit možnost operativní úpravy parametrů řídícího softwaru tak, aby vyhovoval oběma zařízením, tedy školnímu i mému. To bylo vyřešeno individuálním nastavením prostřednictvím konfiguračního souboru. Podrobné informace o rozdílech mezi roboty, jejich důsledcích a o konfiguračním souboru jsou uvedeny v příslušných kapitolách bakalářské práce. Navržený způsob ovládání byl na školním robotu testován. Podrobnější a mohu říci, že vyhovující zkoušky však byly prováděny na mém zařízení.

Při řešení problematiky detekování osob jsem zvažoval použití middlewaru NITE nebo knihovny PCL. Při studiu metody pro rozpoznávání osob v knihovně PCL jsem zjistil, že tato

knihovna vyžaduje, aby při snímání osob byla snímána nejen vlastní osoba, ale také rovina, na které stojí. V případech, kdy tato rovina není v záběru (tj. osoba stoji těsně před stěnou či jiným předmětem stojícím na podlaze), tato okolnost znemožňuje detekci. Proto jsem se rozhodl pro middleware NITE, který toto nevyžaduje a navíc je z hlediska výpočtu méně náročný.

Jako vlastní modifikaci jsem navrhl využití čtyř lineárně uložených mikrofonů instalovaných v sestavě Microsoft Kinectu ke snímání a vyhodnocování okolních akustických signálů. To umožní osobě nacházející se mimo zorné pole snímacího senzoru Kinectu upozornit na svoji přítomnost. V uvedeném smyslu je také rozšířen řídící software robota.

9.1 Případné budoucí modifikace

Případné budoucí modifikace by mohly být následující:

- úprava způsobu snímání zvuku za účelem zvětšení úhlu identifikace zdroje zvuku. Tato úprava by znamenala nepoužívat mikrofony Kinectu, ale dodatečně naistalovat všesměrové mikrofony rozmístěné plošně, nikoliv lineárně a změnit algoritmus pro vyhodnocování úhlu příchodu zvuku. Všesměrové mikrofony by byly pevně naistalovány.

- rozšířené ovládání robota. Middleware NITE umožňuje na detekované osobě rozlišovat jednotlivé části těla, čehož by se dalo využít k ovládání robota např. pohybem ruky.

- zlepšení odometrického systému využitím laserového snímače, který je součástí školního robota. Tato problematika je řešena v bakalářské práci Romana Bedroše [11].

10 Bibliografie

[1] **Khoshelham, Kourosh a Elberink, Sander Oude.** *Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications.* [Online] Enschede : Faculty of Geo-Information Science and Earth Observation, 2012. ISSN 1424-8220.

[2] **Hagara, Ladislav.** Kinect pro Xbox 360 a GNU/Linux – Úvod. *abclinuxu.cz.* [Online] 7. 9 2011. http://www.abclinuxu.cz/clanky/kinect-pro-xbox-360-a-gnu-linux-uvod.

[3] Shluková analýza. *Wikipedie.* [Online] 10. 3 2013. [Citace: 4. 5 2013.] http://cs.wikipedia.org/wiki/Shlukov%C3%A1_anal%C3%BDza.

[4] **Jiří Jarkovský, Simona Littnerová.** Masarykova Univerzita. *Vícerozměrné statistické metody.* [Online] [Citace: 2013. 4 5.] http://www.iba.muni.cz/esf/res/file/bimat-prednasky/vicerozmerne-statisticke-metody/VSM-05.pdf.

[5] **Tvrdík, Pavel.** Algoritmy výpočetní geometrie. *ČVUT.* [Online] 2010. [Citace: 4. 5 2013.] https://edux.fit.cvut.cz/oppa/BI-EFA/prednasky/EFA2010-12.pdf.

[6] k-d tree. *Wikipedie.* [Online] 2. 5 2013. [Citace: 4. 5 2013.] http://en.wikipedia.org/wiki/K-d_tree.

[7] Histogram of oriented gradients. *Wikipedia.* [Online] 27. 4 2013. [Citace: 4. 5 2013.] http://en.wikipedia.org/wiki/Histogram_of_oriented_gradients.

[8] **Rajmic, Pavel.** *Metoda časových posunů pro detekci směru přicházejícího zvuku.* [Online] Brno : Vysoké učení technické v Brně, 2002. ISSN: 1213-1539.

[9] **Bezdíček, Martin.** *Lokalizace pohyblivých akustických zdrojů.* [Online] Brno : Vysoké učení technické v Brně, Fakulta elektroniky a komunikačních technologií, 2010.

[10] **Jaroslav, Balátě.** *Automatické Řízení.* Praha : BEN, 2004. ISBN: 80-7300-148-9.

[11] **Bedroš, Roman.** *Modul lokalizace mobilního robotu pro systém Player.* [Online] Praha : České vysoké učení technické v Praze, Fakulta elektronická, 2011.

[12] **Potúček, Igor.** *Sledování pohybu objektů v sekvenci snímků.* [Online] Brno : Fakulta elektrotechniky a komunikačních technologií VUT v Brně, 2003. ISSN 0542-7462.

[13] Robot Operating System. *ROS.* [Online] [Citace: 13. 5 2012.] http://www.ros.org/wiki/.

Seznam příloh

Příloha 1. CD se zdrojovými kódy