

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra kybernetiky

SLAM mapování vnitřního prostoru 3D kamerou

Jakub Nejedlý

Vedoucí: Ing. Jan Chudoba
Obor: Kybernetika a robotika
Studijní program: Robotika
Květen 2017

Poděkování

Na tomto místě bych chtěl poděkovat vedoucímu mé práce, Ing. Janu Chudobovi, za trpělivost, ochotu a podnětné rady. Svým rodičům za jejich podporu při mém studiu, přítelkyni Šárce za její obětavost a toleranci.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 26. května 2017

.....

podpis autora práce

Abstrakt

Práce se zabývá testováním implementací algoritmu iterative closes point (ICP) pro rekonstrukci vnitřního prostředí. Práce se zaměřuje na 2D SLAM a 6D SLAM implementovaný v knihovně MRPT a 6D SLAM implementovaný v nástrojové sadě 3DTK. Pro každou implementaci ICP byl vytvořen program zajišťující dávkové zpracování a export výstupních dat. Data pro registraci jsou čerpána z běžného RGB-D senzoru Asus Xtion Pro. Pro zpracování a konverzi dat byla vytvořena třída použitelná pro kamery využívající knihovnu OpenNI. Práce srovnává rychlosti registrace jednotlivých implementací a optimální nastavení parametrů.

Klíčová slova: mapování vnitřního prostředí, SLAM, ICP, MRPT, 3DTK, RGB-D

Vedoucí: Ing. Jan Chudoba
Certicon,
Evropská 11,
Praha 6

Abstract

This thesis concerns with testing of implementation of iterative closes point algorithm (ICP) for internal environment reconstruction. The thesis focuses on 2D SLAM and 6D SLAM implemented in the MRPT libraries and 6D SLAM implemented in the 3DTK toolkit. For the processing and conversion of data has been implemented a class for cameras using OpenNI libraries. Data for registration are grabbed from the regular RGB-D sensor Asus Xtion Pro. For the processing and conversion of data was implemented a class for cameras using OpenNI libraries. The work compares the registration rate of individual implementations and optimal parameter settings.

Keywords: mapping of internal environment, SLAM, ICP, MRPT, 3DTK, RGB-D

Title translation: Indoor SLAM mapping by 3D camera

Obsah

1 Úvod	1	5 Hardware a implementace	25
2 Mapování a lokalizace	3	5.1 Použitý hardware	25
2.1 Lokalizační metody	3	5.2 Implementované třídy	25
2.1.1 Scan matching lokalizace	4	5.3 Obecné nástroje	26
2.1.2 Kalmanův filtr	4	5.4 Soubory implementované pomocí MRPT	26
2.2 Struktury map	6	5.4.1 2D ranged scanner	27
2.2.1 Bodové mapy	6	5.4.2 Scan matching pomocí ICP	27
2.2.2 Mřížky obsazenosti	8	5.5 Soubory implementované pomocí 3DTK	28
2.2.3 Příznakové mapy	9	5.5.1 Scan matching pomocí 6D SLAM	28
2.2.4 Geometrické mapy	9	5.6 Kompilace a spuštění	28
2.2.5 Topologické mapy	9	6 Experimentální výsledky	31
3 Senzory	11	6.1 2D ranged scanner	31
3.1 Hloubkový RGB-D senzor	11	6.1.1 Registrované snímky	31
3.1.1 Princip práce hloubkové senzoru RGB-D	12	6.1.2 Zhodnocení experimentů	33
3.1.2 Asus Xtion PRO	14	6.1.3 Nastavení parametrů 2D SLAM	34
3.2 Zpracování obrazových dat	15	6.2 ICP scan matching v MRPT	34
3.2.1 Přepočítání vzdáleností na bodovou mapu	15	6.2.1 Registrované snímky	34
3.2.2 Přepočítání dat na dvourozměrný laser scan	16	6.2.2 Zhodnocení experimentů	39
4 SLAM metody	17	6.2.3 Nastavení ICP	39
4.1 ICP a scan-matching	17	6.3 Scan matching pomocí 3DTK	40
4.2 Použité implementace	19	6.3.1 Registrované snímky	40
4.2.1 2D SLAM v MRPT	19	6.3.2 Zhodnocení experimentů	45
4.2.2 3D SLAM v 3DTK	21	6.3.3 Nastavení přepínačů SLAM	45
		7 Závěr	47
		7.1 Možnosti pokračování práce	48

Bibliografie	49
A Struktura přiloženého CD	51
B Zadání práce	53

Obrázky

2.1	Reprezentace robotu v rovině.	5	6.5	Bodové mapy knihovny MRPT před registrací.	35
2.2	Obrázek bodové mapy. Převzato z [18].	7	6.6	Výsledná registrace dat knihovny MRPT pomocí třídy <i>mrpt::slam::CICP</i>	35
2.3	Příklad konstrukce K-D stromu. Převzato z [7].	7	6.7	Bodové mapy televizní stěny před registrací, pohled zpředu.	36
2.4	Příklad mřížky obsazenosti.	8	6.8	Bodové mapy televizní stěny před registrací, pohled shora.	36
2.5	Ilustrační příklad práce octree. Převzato a upraveno z [9].	9	6.9	Registrovaná bodová mapy televizní stěny, pohled zpředu.	37
3.1	Promítané IR body. Převzato z [14] a upraveno.	12	6.10	Registrovaná bodová mapy televizní stěny, pohled shora.	37
3.2	Pole vzdáleností zobrazené ve stupních šedi.	13	6.11	Bodové mapy před registrací.	38
3.3	Ilustrace nepřesnosti měření RGB-D senzoru na větší vzdálenosti. Převzato z [17].	13	6.12	Bodové mapy pokoje po registraci, pohled zpředu.	38
3.4	Kamera Asus Xtion PRO LIVE. Převzato z [28].	14	6.13	Bodové mapy pokoje po registraci, boční pohled.	39
3.5	Schéma pracovního prostoru kamery pro přepočet.	15	6.14	Registrace s použitím přepínačů.	41
3.6	Ilustrace přepočtu na dvourozměrný scan. Převzato z [6] a upraveno.	16	6.15	Registrace s použitím implicitního nastavením parametrů.	41
4.1	Průběh jedné iterace algoritmu ICP.	18	6.16	Fotografie snímaného prostoru vnější fasády.	42
6.1	Mřížka obsazenosti pro snímání pokoje.	32	6.17	Registrovaná mapy fasády domu.	42
6.2	Snímaný prostor dvora.	32	6.18	Registrovaná mapy fasády domu, pohled shora.	42
6.3	Snímaný prostor dvora.	33	6.19	Registrovaná mapy fasády domu bez přepínače <i>--metascan</i>	43
6.4	Mřížka obsazenosti pro snímání dvoru.	33	6.20	Registrovaná mapy dvora bez <i>--metascan</i>	43
			6.21	Registrovaná mapy dvora bez <i>--metascan</i> , pohled shora.	44
			6.22	Registrovaná mapy dvora s parametrem <i>--metascan</i>	44

6.23 Registrovaná mapy dvora s parametrem <code>--metascan</code> , pohled shora.	44
--	----

Tabulky

3.1 Tabulka vlastností Asus Xtion PRO	14
--	----

Kapitola 1

Úvod

Svět mobilní robotiky zažívá velký rozmach. Ten byl umožněn rozvojem potřebné technologie a snížením výrobních cen. Snížení spotřeby přístrojů umožňuje napájení bateriemi stále výkonnějších zařízení. Nárůst výkonu umožnil zpracování většího množství dat v reálném čase. To vše vede ke zvýšení dostupnosti a zájmu o mobilní elektroniku.

Mobilní roboty nacházejí své místo napříč velkým množstvím odvětví. Nachází uplatnění v soukromém, komerčním i armádním sektoru. V soukromém sektoru nalézá mobilní robotika uplatnění jak praktické, tak i rekreační. Praktické využití nacházejí robotické vysavače a sekačky, vyžadující znalost okolního prostředí. Helikoptéry s různým počtem vrtulí získávají oblibu u modelářů. V komerčním sektoru slouží většinou k pořizování záběrů jinak složité dostupných míst, jako je natáčení filmů, kontrola staveb nebo hlídání objektů [25]. Velký rozmach UAV, tedy pilotovatelných nebo autonomních zařízení, ať kolových, podvodních nebo létajících, zažívá vojenská sféra. Tam nacházejí uplatnění průzkumné i jako nosiče bomb. Autonomní rakety či letadla jsou dnes vyráběny profesionálně i amatérsky na obou stranách bojiště. [3]. V současné době armáda Spojených Států Amerických plánuje nasazení autonomních nadzemních a podvodních dronů pro hlídání rozlehlých oceánských oblastí. Pomocí těchto dronů je možné dosáhnout úspory v počtu nasazených vojáku a ušetřit velkou část finančních zdrojů. Dalším využitím autonomních robotů je takzvaná mechanická mula, která slouží pro přepravu zavazadel pro armádní nebo civilní účely.

Autonomní UAV musí být vybavena dostatečným výpočetním výkonem pro zpracování nasbíraných dat a komunikaci s ostatními UAV či lidskou obsluhou. Pro smysluplný provoz UAV je potřeba senzorické vybavení umožňující lokalizaci a mapování zařízení [19]. Během průzkumu je nezbytné, pokud se UAV dostane do pro něj neznámém terénu, aby dokázal prostředí zmapovat a při pohybu v něm znát svojí polohu. UAV musí pro lokalizaci být vybavena vhodnými senzory. Spektrum používaných senzorů pro lokalizaci je velmi široké. Používané jsou ultrazvukové senzory pracující s dobou letu zvuku či laserové měřicí přístroje, např. lidar od firmy Velodyne. Mezi další oblíbené a levné senzory patří 3D kamery realizované stereokamerou od firem Bosch, Zed a Ricoh. Další realizací jsou zařízení typu kinect, kde je prostor snímán

pomocí IR projektoru a kamery.

Senzory bývají na UAV vhodně kombinovány a vzájemně provázány při určování pozice spolu s odometrií. Tam kde je to možné, je také využita GPS. Většina UAV disponuje i klasickou RGB kamerou, které se využívá k mapování, řízení obsluhou, rozeznávání markerů a objektů v okolí.

V rámci této práce se budu zabývat pouze malou částí UAV a to pouze lokalizací pro helikoptéry, quadrokoptéry a octokoptéry. Tedy zařízeními s kolmým vzletem, vybavenými jednou až osmi vrtulemi umístěnými na kostře směrem vzhůru. Směr letu bývá regulován změnou rychlosti otáčení jednotlivých vrtulí řízených centrální jednotkou. V práci je použit běžný zástupce levnějších RGB-D senzorů, a to Asus XtionPro s různými implementacemi lokalizační metody scan matching zprostředkované pomocí ICP. Data jsou registrována bez použití odometrie.

Kapitola 2

Mapování a lokalizace

Mapováním chápeme schopnost robotu vytvářet model okolního prostředí. V závislosti na požadované činnosti robotu je mapa využita jako výstup pro další potřeby operátora nebo použita pro potřeby orientace robotu. Operátor může na základě mapy vyhodnotit nebezpečnost prostředí nebo získat data z jinak nedostupných prostor, jako jsou jeskynní jezera, oceánské dno, budovy po zemětřesení a jiné. Mapa bývá zařízení dodána jako reference nebo je v neznámém prostředí přímo vytvářena. Mapa bývá používána pro nalezení aktuální pozice robotu ze sensorických dat.

Lokalizací rozumíme odhad polohy robotu v daném prostředí. Poloha robotu musí být vztažena k mapě a počátek bývá určen počáteční polohou robotu nebo je definován v externí mapě. Poloha bývá určována v závislosti na vytvářené mapě. Při lokalizaci ve třírozměrném prostoru používáme k popisu polohy robotu obvykle šest souřadnic [x y z yaw pitch roll]. První tři souřadnice obvykle popisují změnu v posunu od počátku souřadnic, druhé tři souřadnice popisují rotaci podél jednotlivých os. Robot bývá obvyčejně substituován svým těžištěm či polohou senzoru a orientací v prostoru. Robot se tedy pohybuje se šesti stupni volnosti [23]. Planární určování polohy robotu se třemi stupni volnosti bývá využíváno kolovými nebo pásovými roboty v prostředích, která se příliš nemění v ose Z. Během planární lokalizace robotu hledáme pozici [x y θ], kde hodnoty x a y jsou vzdálenosti posunu od počátku ve směru dané osy a hodnota θ určuje natočení od osy x. Tato práce je zaměřena na lokalizaci s využitím IR dálkoměru. Princip práce dálkoměru je popsán v kapitole 3.1.1.

2.1 Lokalizační metody

V dnešní době je vyvinuto a implementováno velké množství metod pro lokalizaci. Metody se různí podle druhu použitých senzorů a prostředí ve kterém mají být použity. Ve vnějších prostorách lze s výhodou užít GPS. Ten je dnes velice přesný a hojně využívaný [8]. GPS lokalizaci lze dnes najít ve většině automobilů a je jím vybavena většina mobilních telefonů.

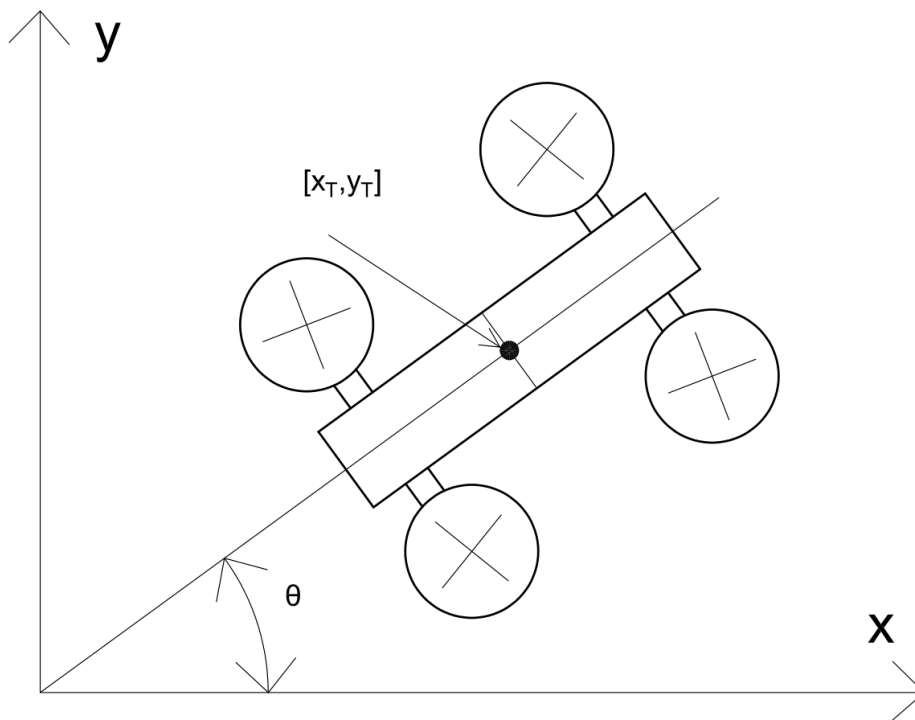
Ve vnitřním prostředí není ve většině případů lokalizace pomocí GPS možná. Je tedy potřeba použít lokálních lokalizačních systémů. Mezi ně patří lokali-

Druhým předpokladem pro takové zjednodušení je, že povrch neobsahuje významné nerovnosti.

V případě pohybu v rovině obsahuje vektor polohy či střední hodnoty polohu v rovině popsanou podle os x a y . Dále obsahuje hodnotu natočení od osy x , popsanou úhlem θ . Popsané v rovnici 2.1.

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (2.1)$$

Reprezentace dronu v rovině je uvedena na obrázku 2.1.



Obrázek 2.1: Reprezentace robotu v rovině.

Popis robotu pohybovými rovnicemi je uveden v rovnicích 2.2, 2.3 a 2.4. Kde v reprezentuje rychlost robotu a natočení robotu vůči počátku souřadnic je popsáno úhlem θ . Za střed dronu považujeme geometrický střed rámu.

$$\dot{x} = v \cos(\theta) \quad (2.2)$$

$$\dot{y} = v \sin(\theta) \quad (2.3)$$

$$\dot{\theta} = \omega \quad (2.4)$$

Vztah jednotlivých stavů k a $k + 1$ je popsán maticí uvedenou v rovnici 2.5. Každému stavu k náleží příslušný čas t . Stav k a $k+1$ jsou měřeny s určitou

časovou prodlevou Δt . [26]

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + \Delta t v \cos(\theta) \\ y_k + \Delta t v \sin(\theta) \\ \theta_{k+1} + t\omega \end{bmatrix} \quad (2.5)$$

Jednotlivá měření charakterizujeme podle rovnice 2.6. Hodnota z obsahuje měření, x_k obsahuje stavy a hodnota D_k popisuje rozptyl měření.

$$z_k = Hx_k + D_k \quad (2.6)$$

Práce Kalmanova filtru poté probíhá ve dvou krocích, časová úprava, nebo také predikce a úprava pozice na základě měření neboli korekce. Rovnice popisující predikci jsou 2.7 a 2.8. Rovnice popisující korekci jsou 2.9, 2.10 a 2.11.

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \quad (2.7)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (2.8)$$

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (2.9)$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (2.10)$$

$$P_k = (I - K_k H)P_k^- \quad (2.11)$$

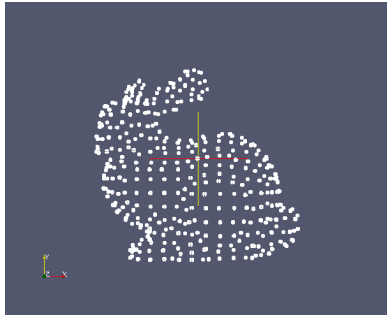
Matice A a B jsou stavové matice o rozměrech $n \times n$ a $n \times 1$. V rámci práce s Kalmanovým filtrem předpokládáme, že jsou konstantní. V rovnici 2.8 vypočítáme počáteční odhad chyby kovariance P_k^- pomocí konečného odhadu chyby kovariance předchozího stavu P_{k-1} a matice Q. Matice Q popisuje šum predikce polohy, tedy například nepřesnost odometrie. Rovnice 2.9 popisuje výpočet Kalmanova zesílení K pomocí matic R, H a matice počátečního odhadu chyby kovariance. Matice R popisuje šum měření polohy pomocí senzorů. Matice H je maticí vztahů mezi stavy x_k a měřeními z_k s rozměry $m \times n$. Konečný výpočet chyby kovariance P_k je popsán v rovnici 2.11 [27].

2.2 Struktury map

Pokud robot provádí mapování je nezbytné vytvářet model okolního světa. Struktura a implementace modelu závisí na použitých senzorech, stupni abstrakce, očekávané velikosti mapy a jejím budoucím použití. Vyjmenuji nejběžněji používané struktury.

2.2.1 Bodové mapy

Bodové mapy můžeme považovat za sensorické mapy. Obsahují množinu nalezených bodů tvořících okolní prostor. Umožňují rychlou lokalizaci pomocí metody scan matching. Bodová mapa má také výhodu v rychlosti zpracování jednotlivých snímků oproti jinému druhu map. Je dostatečně přesná pro vnitřní lokalizaci a je schopná snadné inkrementální aktualizace po získání nového snímku. Běžně používán je binární k-D strom [20].

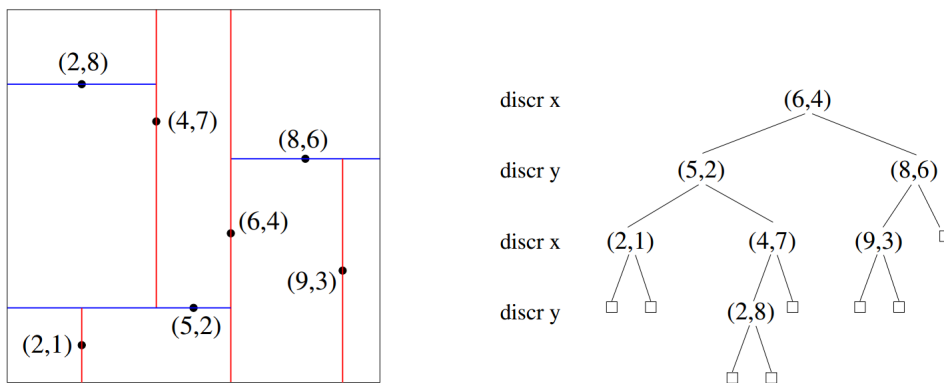


Obrázek 2.2: Obrázek bodové mapy. Převzato z [18].

K-D strom

K-D strom je datová struktura sloužící k ukládání vícerozměrného prostoru. Vstupním stavem je množina bodů v obecně k -dimenzionálním prostoru. Ten je postupně podle implementace algoritmu dělen na podprostory. V případě implementace binárního K-D stromu je prostor dělen na dva podprostory. V každé uzlu daného stromu tedy dochází k dělení podprostoru přímkou či rovinou v závislosti na dimenzi [11].

Uložení dat v K-D stromu z výchozího souboru bodů ukazuje obrázek 2.3. V základní implementaci je pro první dělení vybrán medián souboru. Běžné je také střídání os dělení, toto dělení se však může různit podle implementace. Datový soubor je seříděn podle dělené osy, je podle mediánu rozdělen na dva podprostory a tento postup se opět opakuje pro každý vytvořený podprostor dokud v souboru zůstávají neumístěné prvky. [7]



Obrázek 2.3: Příklad konstrukce K-D stromu. Převzato z [7].

Na obrázku 2.3 je také vidět rozvážený K-D strom. Pokud jsou body do K-D stromu vkládány jednotlivě a nikoliv dávkově, může dojít velice snadno k rozvážení a omezení funkcí stromu. Rozvážení stromu je možné řešit zpětným vyvažováním, ale implementace i provoz je poměrně náročný.

Jednou z největších výhod uchovávání dat v K-D stromu je rychlý přístup k okolním bodům. Body geograficky si blízké jsou ve struktuře uloženy ve společném podstromu. Těto vlastnosti můžeme využít právě u metod scan

matching k urychlení často používané operace vyhledávání sousedních bodů, což vede k úspoře času potřebého pro registraci.

■ 2.2.2 Mřížky obsazenosti

Mřížku chápeme jako dvourozměrné pole buněk popisující okolní prostor. Mřížky obsazenosti jsou definované pro 2D pracovní prostor. Jednotlivé buňky obsahují pravděpodobnost výskytu překážky na dané pozici. Při definování mřížky bývá jedním ze základních údajů délka hrany čtvercové buňky. Jednotlivé snímky sensorický dat jsou vhodně upraveny a poté zařazeny do mřížky podle Bayesových vět. Příklad mřížky je vidět na obrázku 2.4 [15].



Obrázek 2.4: Příklad mřížky obsazenosti.

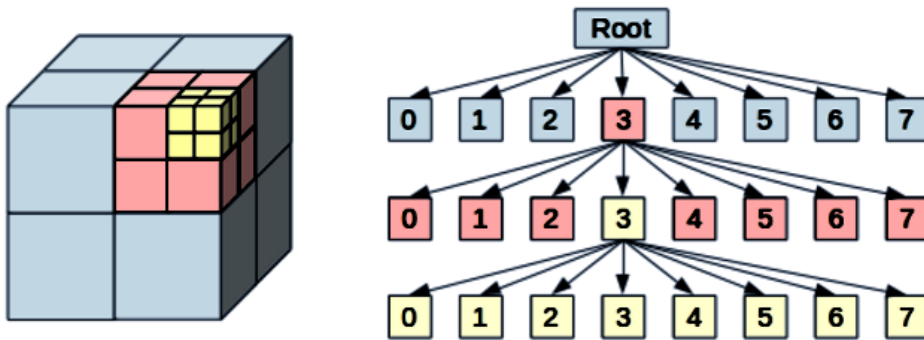
Mřížky obsazenosti jsou poměrně jednoduché na implementaci a použití. Jejich nevýhodou je kvadratický nárůst náročnosti na paměťový prostor, který znemožňuje použití mřížek na velmi rozlehlé oblasti. Tento problém do určité míry kompenzují datové struktury quadtree ve dvourozměrném pracovním prostoru a octree ve třírozměrném.

■ Quadtree a Octree

Quadtree a octree jsou stromové struktury pro reprezentaci vícerozměrného prostoru. Na rozdíl od K-D stromu mají definovanou délku hrany prvku v nejvyšší hloubce. Strom tedy může zjemňovat údaje o prostoru nejvýše do povolené hloubky dané maximální délkou hrany. Pracovní prostor je možné rozšířit, pokud je to nutné, přidáním prvků nad kořen. Každý prvek má buď čtyři potomky pro quadtree nebo osm potomků pro octree, které pravidelně dělí podprostor rodiče podle všech os. Jak je ukázáno na obrázku 2.5, každá buňka obsahuje pravděpodobnostní údaj o výskytu překážky [12].

Při vytváření stromu je každý bod pomocí Bayesových vět přiřazen podle souřadnic do uzlu, pod který spadá. Uzel je vytvořen pouze pokud jsou data pro jeho pracovní prostor naměřena. Stromy nemohou být rozváženy díky nastavení maximální hloubky a data do nich mohou být vkládána snadno po jednotlivých snímcích.

Reprezentace umožňuje úsporu datového prostoru díky tomu, že nezmapovaná či dokonce nezmapovatelná část mapy nemusí být alokována v paměti



Obrázek 2.5: Ilustrační příklad práce octree. Převzato a upraveno z [9].

podrobně.

■ 2.2.3 Příznakové mapy

Příznaková mapa se skládá z význačných bodů v okolním prostředí a z popisu jejich polohy. Mohou uchovávat vzájemné úhly mezi příznaky, úhly pozorovatelnosti příznaků a jistou mírou neurčitosti jednotlivých příznaků [20].

Příznaky bývají brány z RGB obrazu, z kombinace povrchu a RGB obrazu nebo z kombinace povrchu a normály.

■ 2.2.4 Geometrické mapy

Geometrická mapa popisuje pracovní prostor geometrickými primitivami. V případě dvourozměrné mapy se jedná o úsečky, v případě třírozměrné o polygony. Geometrická mapa může být vypočítána z mřížky obsazenosti. Je méně náročná na paměť a umožňuje snadnou lokalizaci robotu s dostatečnou mírou přesnosti.

■ 2.2.5 Topologické mapy

Topologická mapa je reprezentace pracovního prostoru využívající uzly a hrany. Sloužit může k hrubému plánování, tedy že každý uzel reprezentuje určitou oblast, která je detailně popsána například geometrickou mapou. Vyhledávání na úrovni topologické mapy poté může urychlit vyhledávání dlouhé cesty. Topologické mapy lze také využít při plánování cesty uvnitř silničních map. Topologické mapy také mohou být použity v systémech s velmi malými paměťovými prostory nebo při omezeném počtu senzorů [20].

Kapitola 3

Senzory

Senzory jsou nedílnou součástí každého robotického systému. Bez informací o okolním pracovním prostoru není možné vykonávat jakoukoliv smysluplnou činnost. Robot by měl být vybaven alespoň jedním senzorem, ale nejlépe kombinací několika různých sensorů. Důležitými senzory pro řízení robotu bývají senzory regulace otáček pohonů. Mezi další používané sensorické systémy patří odometrie u kolových a pásových robotů, videoodometrie pro létající UAV nebo tlakový senzor u podvodních zařízení. Další senzory pro řízení robotu mohou být například gyroskopy, akcelerometry nebo tenzometry. Senzory používané pro lokalizaci bývají sonary, infračervené senzory a laserové dálkoměry. Infračervené a laserové dálkoměry měří většinou vzdálenost na jedné přímce. Existují však i senzory pracující na principu rozmítaného laserového paprsku například lidar od společnosti Velodyne. Senzory typu kinect od společnosti Microsoft jsou vybavené IR projektorem a kamerou pro hloubkové snímání prostoru.

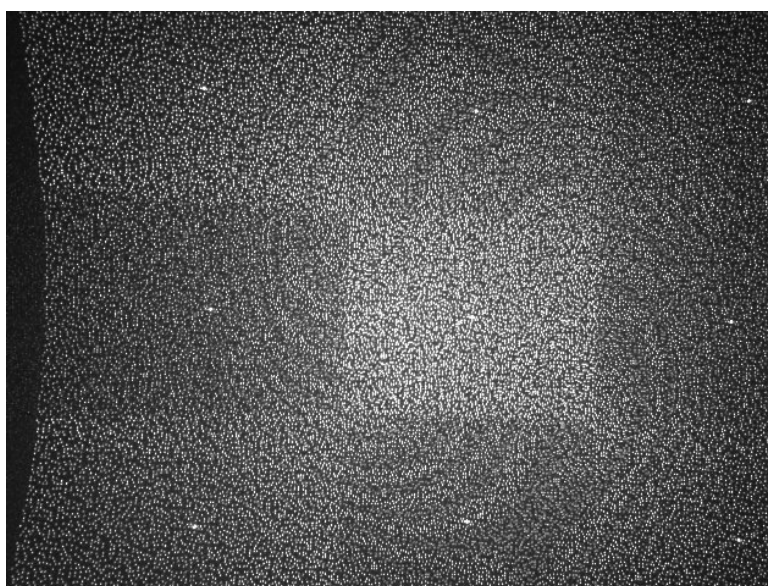
3.1 Hloubkový RGB-D senzor

Hloubkový RGB-D senzor je kombinací barevné kamery a infračerveného dálkoměru. Barevná kamera není pro činnost hloubkového senzoru nezbytná. Hloubkový infračervený senzor je v mobilní robotice poměrně běžně používaný. Senzory se pohybují ve velkém cenovém rozpětí. O velký rozmach RGB-D sensorů se postaral kinect prodávaný od roku 2010 firmou Microsoft. Kinect byl prodáván jako příslušenství pro herní konzoli Xbox 360. Jeho nízká cena a schopnost určit vzdálenost pro každý barevný pixel vzbudily zájem nejen laické veřejnosti. Velkou výhodou Kinect oproti ostatním IR sensorům byla výrazně nižší pořizovací cena, a tím se umožnilo jeho široké použití. Brzy se objevily podobné produkty od konkurenčních firem jako PrimeSense, ASUS či Orbbec [17].

3.1.1 Princip práce hloubkové senzoru RGB-D

Hloubkový RGB-D senzor poskytuje vzdálenostní údaj pro každý pixel. Senzor obvykle nabízí více různých rozlišení s rozdílnými hodnotami fps. Na nižších rozlišeních zvládají kamery zpracovat větší množství snímků za vteřinu. Hodnoty se pohybují u levnějších modelů jako jsou Astra, Asus Xtion nebo Kinect se snímky pohybují mezi 30 až 60 fps.

Senzor je vybaven projektorem a kamerou. Projektor promítá množinu bodů do prostoru snímaného kamerou. Množina bodů je zobrazena na obrázku 3.1. Promítané body je možné zachytit jakoukoliv kamerou, pokud odstraníme infračervený filtr [10]. RGB-D senzory vypočítávají vzdálenost z polohy



Obrázek 3.1: Promítané IR body. Převzato z [14] a upraveno.

velikosti a orientace promítaných bodů. Pokud jsou však body promítány na povrch s vysokou odrazivostí, dochází k jejich lomu a z těchto míst poté není možné odečíst vzdálenost na rozdíl od Time-of-flight senzorů, které počítají s odrazem světla od sledovaného prostoru. Ty bývají dražší, ale jsou odolnější proti vnějšímu rušení.

Výsledná data poskytnutá senzorem je dvourozměrné pole hodnot s rozměry podle snímaného rozlišení. Hodnoty v poli jsou obvykle uloženy v metrech či milimetrech. Zobrazení tohoto pole ve stupních šedi podle vzdálenosti můžeme vidět na obrázku 3.2. Další zpracování dat je uvedeno v kapitole 3.2.

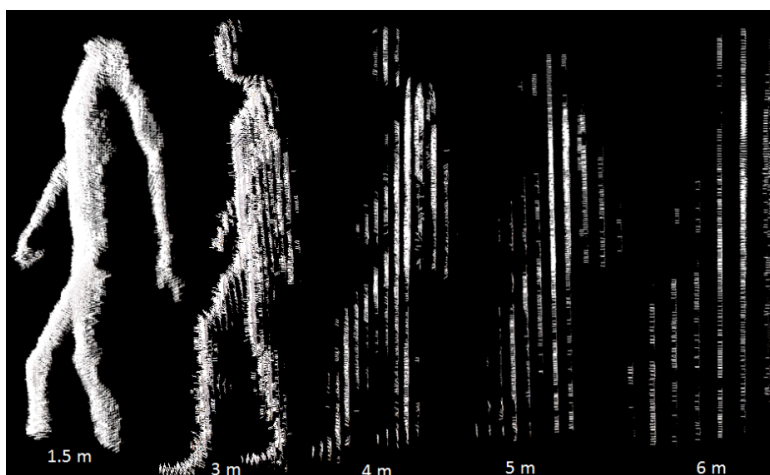
■ Nevýhody RGB-D senzoru

Dálkoměr tohoto typu má několik nevýhod. Senzor vykazuje kvadratickou chybu s rostoucí vzdáleností objektu [20]. Na vzdálenost čtyř metrů již senzor vykazuje o řád větší chybu v mm než specializované IR dálkoměry. Tato chyba také ovlivňuje schopnost dálkoměru přesněji určovat polohu vzdálenějších



Obrázek 3.2: Pole vzdáleností zobrazené ve stupních šedi.

objektů. Tento jev s malým vzorkováním na větší vzdálenost popisuje obrázek 3.3.



Obrázek 3.3: Ilustrace nepřesnosti měření RGB-D senzoru na větší vzdálenosti. Převzato z [17].

Další nevýhodou je snadné ovlivnění měření zdrojem IR záření. Při experimentech ve venkovních prostorách nebylo možné naměřit použitelná data, pokud byla měřená plocha vystavena slunečnímu záření. Snímání bylo možné pouze ve večerních hodinách. Poslední nevýhodou je potřeba přesné kalibrace, protože většina zařízení snímá nepřesně po okraji snímku a rovné stěny zaoblují.

3.1.2 Asus Xtion PRO

Asus Xtion PRO je IR dálkový senzor společnosti Asus. Na trh byl uveden v roce 2012 ve dvou variantách s různými stupni výbavy. Kamera Asus Xtion PRO je vybavena pouze IR projektorem a kamerou. Varianta s vyšší výbavou, Asus Xtion PRO Live, je navíc vybavena RGB kamerou a mikrofony, tedy stejnou výbavou jako konkurenční Kinect. Na obrázku 3.4 je zobrazena kamera Asus Xtion PRO LIVE. Nižší model zachovává tvar serie Xtion, pouze je osazen méně senzory.



Obrázek 3.4: Kamera Asus Xtion PRO LIVE. Převzato z [28].

V tabulce 3.1 jsou uvedeny hodnoty uvedené výrobcem na oficiálních stránkách [13].

Vlastnost	Hodnota
Velikost hloubkého obrazu	VGA : 30fps nebo QVGA : 60fps
Zorný úhel	58° H, 45° V
Požitelná vzdálenost	0.8 až 3.5m
Software	OpenNI SDK
Podporavné rozhraní	USB 2.0

Tabulka 3.1: Tabulka vlastností Asus Xtion PRO

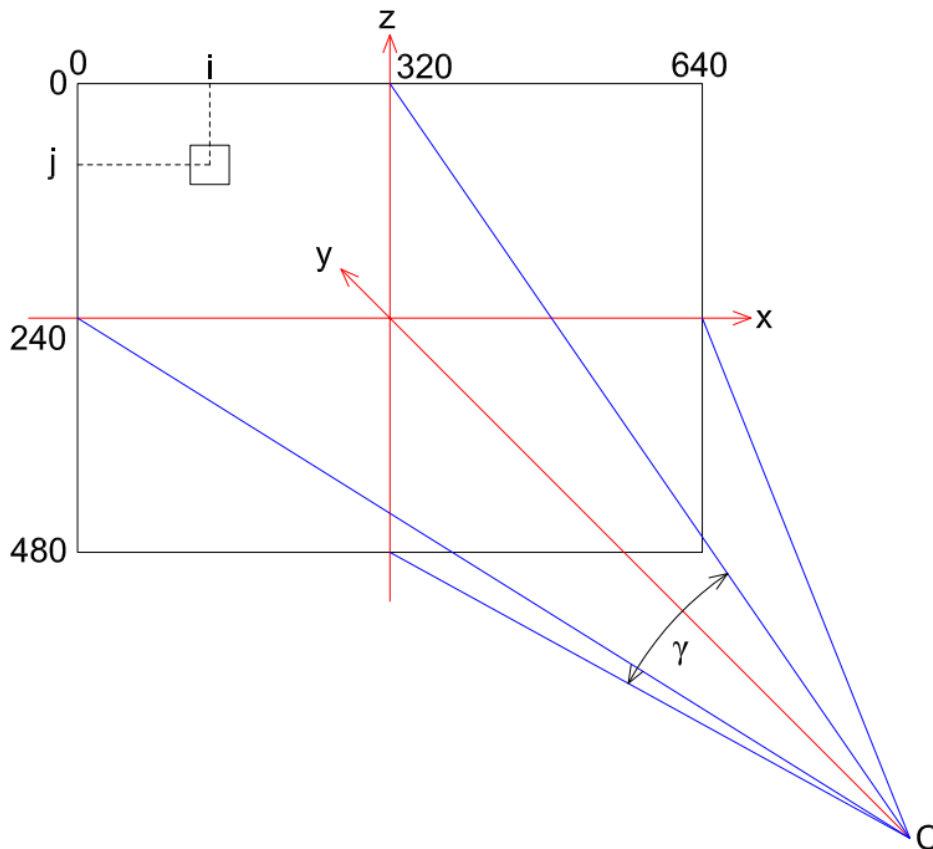
Během měření v experimentální části jsem zjistil, že senzor je schopen identifikovat překážky až do vzdálenost přibližně sedmi metrů, v závislosti na okolních podmínkách. Výrobce však tento údaj neuvádí. Pravděpodobně z toho důvodu, že měření na tuto vzdálenost je již velmi nepřesné. Konkurenční výrobce Orbbec uvádí pro svou kameru Astra Pro pracovní rozmezí 0.6 až 8 metrů [1]. Při testování ve stejných podmínkách byly však snímání prostory i kvalita snímání velmi rovnocenné.

3.2 Zpracování obrazových dat

Senzory obvykle poskytují data v různých formátech a různých stupních vyhodnocení, podle typu a možností senzoru. Sensorická data musejí být obvykle zpracována, než mohou být použita lokalizačním a mapovacím systémem. V této kapitole se zabývám převodem nasnímaných dat na bodovou mapu a dvourozměrný laser scan.

3.2.1 Přepočet vzdáleností na bodovou mapu

Při práci s RGB-D senzory bývá nezbytné převedení vzdálenostní hodnoty každého pixelu na bod v kartézských souřadnicích. Na obrázku 3.5 je náčrtek schématu ilustrujícího pracovní prostor před RGB-D kamerou. Důpočty jednotlivých souřadnic jsou uvedeny v rovnicích 3.1 až 3.4.



Obrázek 3.5: Schéma pracovního prostoru kamery pro přepočet.

$$Hg = \frac{2 \tan(\frac{hfov}{2})}{640} \quad (3.1)$$

$$Vg = \frac{2 \tan(\frac{vfov}{2})}{480} \quad (3.2)$$

$$x = (640 - (i - 319,5)ySH) \quad (3.3)$$

$$z = (480 - (j - 239,5)ySV) \quad (3.4)$$

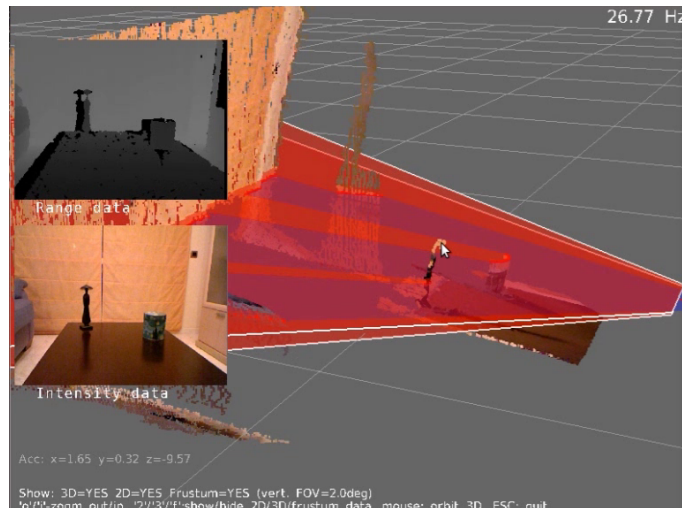
Rovnice 3.1 a 3.2 popisují přepočítání stoupání na pixel a vzdálenost, hodnoty hfov (horizontal field of view) horizontální zorný úhel a vfov (vertical field of view) vertikální zorný úhle jsou vlastností kamery, v našem případě 58° hfov, 45° vfov. Rovnice 3.3 a 3.4 popisují dopočet souřadnic x a z z hodnoty vzdálenosti y . Přepočítání bylo čerpáno z implementace OpenNI grabberu v knihovně Point cloud library [16].

3.2.2 Přepočítání dat na dvourozměrný laser scan

Senzory typu Kinect mohou být převedeny na levný laserový snímač fungující v malých pracovních prostorech. Převedení 3D rozměrového obrazu na 2D scan umožňuje využití velkého množství implementovaných nástrojů a technik pro 2D SLAM [6].

Převod provádí funkce `convertTo2DScan()` ze třídy `mrpt::obs::CObservation3DRangeScan` na `mrpt::obs::CObservation2DRangeScan`. V této třídě je poté nezbytné nastavit horizontální úhel snímání. Pro převod je využívána sada parametrů `mrpt::obs::T3DPointsTo2DScanParams` u které je nutné specifikovat parametr `angle_sup`.

Hodnota `angle_sup` popisuje velikost úhlové odchylky podél osy y ve které budou reflektovány překážky do 2D scanu. Tyto červeně vyznačené roviny jsou demonstrovány na obrázku 3.6.



Obrázek 3.6: Ilustrace přepočítání na dvourozměrný scan. Převzato z [6] a upraveno.

Kapitola 4

SLAM metody

Simultánní lokalizace a mapování, zkráceně SLAM (simultaneous localization and mapping), je spojení dvou problémů, lokalizace, popsané v kapitole 2.1 a mapování, popsaného v kapitole 2.2. Problematika lokalizace a mapování jsou velice úzce svázány. Při pohybu v neznámém prostředí potřebuje robot pro další pohyb znát svou polohu a polohu okolních překážek. Svou polohu robot určuje z odometrických dat, ty ale v čase divergují od skutečné pozice robotu. Proto je v praxi nutné aktualizovat polohu i podle právě nasnímaného okolního prostředí. Pro tuto aktualizaci je tedy nezbytné mít informace o prostoru, ve kterém se robot pohybuje. Simultánní lokalizace a mapování je tedy algoritmus při kterém dochází k pohybu robotu v pro něj neznámém prostředí. Současně s tím je vytvářena mapa okolí podél této trajektorie. Algoritmus se tedy snaží o nejpravděpodobnější odhad skutečné trajektorie robotu.

Robot obvykle začíná v počátku souřadnicového systému. Po posunu robotu je nasnímán okolní prostor a zapracován do vytvářené mapy. Na základně nezbytných úprav orientace dat při registraci je upravena i pozice robotu. Registrace může být prováděna pomocí přímé registrace snímků např. Iterative Closest Point, RANSAC nebo porovnáváním prostorových nebo vizuálních příznaků, např. SIFT nebo NARF. Lokalizace je obvykle prováděna absolutně či pravděpodobnostně. Absolutní lokalizace určuje polohu bez zahrnutí možné nepřesnosti. Pravděpodobnostní lokalizace popisuje polohu robotu pomocí střední hodnoty a rozptylu. V rámci této práce je použita pouze absolutní lokalizace pomocí registrace snímků.

Na začátku této kapitoly popíši SLAM metodu použitou v rámci projektu. Dále uvedu parametry ovlivňující registraci u použitých implementací.

4.1 ICP a scan-matching

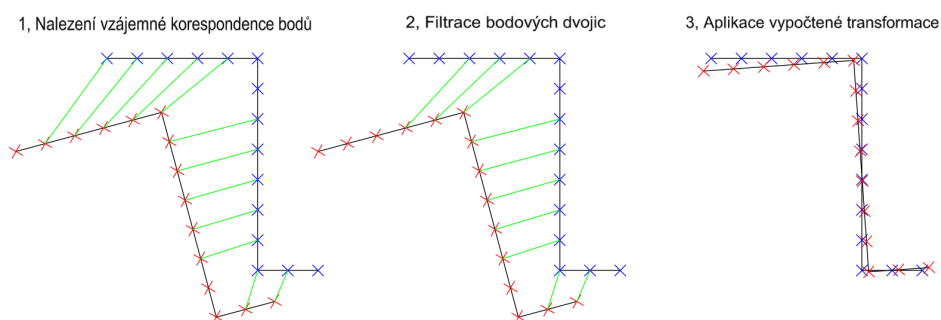
Iterative Closest Point, dále jen ICP, je algoritmus pro hledání transformace bodové mapy vůči referenci tak, aby došlo k překryvu společných částí.

Algoritmus ICP pracuje ve velkém množství iterací, které stále zpřesňují

registraci aktuálního snímku vůči referenci. Činnost algoritmu během jedné iterace je možné shrnout do následujících kroků [4].

- Nalezení vzájemné korespondence bodů mezi referenční a nově nasnímanou bodovou mapou. V případě použití robustních korespondencí jsou páry hledány obousměrně a přijímány pouze páry nalezeny v obou hledáních. V případě rozšiřujících metod, jako je CRF (Condition Random Field), kdy jsou přijímány pouze korespondence na odpovídajících si geometrických primitivech [22].
- Výpočet parametrů transformace minimalizující kritérium vzájemné vzdálenosti $E_{dist}(T, \omega)$ odpovídajících párů. Výpočet vzájemné vzdálenosti je popsán v následující sekci.
- Aplikace transformace na aktuální data.

Tyto kroky demonstruje obrázek 4.1. Iterace probíhají do té doby, dokud není překročena jedna z prahových konstant pro minimální translaci či rotaci nebo není překročen maximální počet iterací. Po poslední transformaci je vrácena celková transformace aktuálního snímku. Tato transformace je aplikována na poslední známou polohu robotu. Ten poté může pokračovat dále v mapování.



Obrázek 4.1: Průběh jedné iterace algoritmu ICP.

Stabilita a konvergence k správnému řešení algoritmu ICP je silně závislá na počátečních podmínkách. Algoritmus ICP může uváznout v lokálním optimu v případě přílišné vzdálenosti odpovídajících si částí bodových map. Je proto vhodné provést výchozí odhad možné polohy a orientace robotu před zahájením registrace [20].

■ Výpočet vzájemné vzdálenosti

Rotaci ve třírozměrném pracovním prostoru můžeme popsat rotacem podél jednotlivých os. Rotace podél osy x je popsána rovnicí 4.1, podél y rovnicí 4.2 a podél z rovnicí 4.3. Jejich vzájemnou kombinací získáme popis rotace

tělesa Eulerovými úhly (ϕ, θ, ψ) .

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \quad (4.1)$$

$$R_y = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (4.2)$$

$$R_z = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.3)$$

Pohyb soustavy podél jednotlivých os popisuje translační matice uvedena v rovnici 4.4.

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \quad (4.4)$$

Během registrace se snažíme o minimalizaci kritéria vzájemné vzdálenosti dat $E_{dist}(T, \omega)$ popsané v rovnici 4.5 [20].

$$E_{dist}(T, \omega) = \sum_{i=1}^n |R_\omega p_i + T - p_i|^2 \quad (4.5)$$

Vektor T je výše popsáný translační vektor, matice R_ω popisuje rotaci aktuální bodové mapy vůči referenci, p_i jsou souřadnice referenční mapy a p_j souřadnice aktuální mapy.

■ 4.2 Použité implementace

V rámci práce jsou použity a testovány dvě implementace SLAM. První testovaný je 2D SLAM implementovaný v knihovně MRPT. Druhý SLAM pro registrování 3D bodových map je implementován v knihovně 3DTK.

■ 4.2.1 2D SLAM v MRPT

První testovaný SLAM je implementován v knihovně MRPT. Jde o dvou-rozměrný SLAM registrovaný pomocí ICP na pravděpodobnostní mřížku obsazenosti. Třída `mrpt::slam::CMetricMapBuilderICP` přijímá senzorická data ve formátu `mrpt::obs::CObservationPtr`. Senzorická data mohou obsahovat odometrické údaje o poloze robotu nebo data okolního prostředí získaná z hloubkových senzorů. Přijímaná data jsou zpracována metodou `processObservation` podle definovaných parametrů a mohou být i zamítnuta pro přílišnou odlišnost od doposud nasnímaných dat, pokud by došlo k chybě senzoru. Implementace poskytuje informace o poslední vypočtené poloze robotu, poskytuje

mapu ve `mrpt::maps::CMultiMetricMap` nebo `mrpt::maps::CSimpleMap` a podporuje základní funkce jako je ukládání, načítání a pozastavení úprav. Mapa obsahuje bodovou mapu a mřížku obsazenosti.

■ Popis parametrů

Parametry v implementaci 2D SLAM lze rozdělit na parametry budované tabulky obsazenosti a parametry SLAM. Parametry lze dále rozdělit na ukončovací a registrační. Registrační parametry obvykle popisují podmínky nalezení korespondence dvou bodů a použité algoritmy při registraci. Ukončovací podmínky omezují počet iterací a posuzují, zda je registrace hotová. Níže jsou uvedené parametry definují mřížku obsazenosti.

- **matchAgainstTheGrid**
Parametr určuje provádění ICP nad mřížkou obsazenosti nebo nad bodovou mapou. Základní hodnota je false, tedy bodová mapa.
- **insertionLinDistance**
- **insertionAngDistance**
Minimální posun nebo rotace robotu pro kterou jsou nová data zahrnuta do mapy.
- **localizationLinDistance**
- **localizationAngDistance**
Minimální posun nebo rotace robotu pro kterou je provedeno ICP.
- **minICPgoodnessToAccept**
Minimální hodnota kvality registrace pro zahrnutí dat do mapy. Může nabývat hodnoty v rozsahu (0,1) základní hodnota je 0.40.

Registrační parametry ICP použitého pro budování mapy.

- **thresholdDist**
- **thresholdAng**
Tyto dva parametry popisují maximální vzdálenost a úhel pro které jsou dva body shledány jako korespondentní. Základní hodnoty je nutné nastavit podle rozměrů pracovního prostoru.
- **ALFA**
Parametr určuje míru konvergence, které se snaží ICP dosáhnout.
- **smallestThresholdDist**
Velikost prahu, pro kterou jsou iterace zastaveny a registrace je shledána jako dostatečně přesná.
- **covariance_varPoints**
Normalizační konstanta σ_p^2 používaná na úpravu hodnot celé 3x3 matice kovariance.

- **doRANSAC**
Provádění RANSAC `mrpt::tfest::se2_l2_robust()`, po určení ICP kovariance pro lepší odhad polohy robotu. Základní volba je neprovádět výpočty.
- **kernel_rho**
Stanovuje hodnotu maximální úpravy pomocí `kernel_rho`. Základní hodnota je 0.07 metru.
- **use_kernel**
Umožňuje používat `kernel_rho` na úpravu vzdáleností nebo používat přesné vzdálenosti. Základní nastavení dovoluje použití zjemnění.
- **skip_cov_calculation**
Vynechání výpočtu kovariance, pro úsporu času. Základní volba je provádění výpočtu.
- **skip_quality_calculation**
Vynechání výpočtu evaluace kvality ICP. Základní hodnota je vynechání výpočtu.
- **corresponding_points_decimation**
Decimace odpovídajících bodů registrovaných na referenční snímek. Základní hodnota je 5.
- **ICP_algorithm**
Volba použitého algoritmu k registraci. Základní algoritmus je `icpClassic`.
- **ICP_covariance_method**
Metoda použitá pro odhad kovariance. Základní metoda je `icpCovFiniteDifferences`.

Ukončovací podmínky ICP algoritmu.

- **maxIterations**
Maximální počet iterací ICP při zpracování jednoho snímku.
- **minAbsStep_trans**
- **minAbsStep_rot**
Tyto dva parametry definují míru přesnosti pro zpracování, pokud bude algoritmem nalezen pohyb či rotace menší než tento práh. Bude ICP ukončeno. Základní hodnota je 10^{-6} metrů a 10^{-6} radiánu.

■ 4.2.2 3D SLAM v 3DTK

Druhý testovaný SLAM je implementován v nástrojové sadě 3DTK. Jde o 6D SLAM registrovaný pomocí ICP na již vytvořenou bodovou mapu. Nástrojová sada vyžaduje datové soubory umístěné v jedné složce. Jeden snímek vyžaduje dva soubory, první s příponou `.3d` obsahuje souřadnice x , y a z v centimetrech

a druhý soubor s příponou `.pose` obsahuje orientaci robotu. Orientace robotu je popsána souřadnicemi na prvním řádku a úhly natočení na druhém. Soubory musí být pojmenovány `scan`, obsahovat pořadové číslo scanu doplněné nulami na tři číslice. Číslování začíná od nuly. Soubory v tomto formátu lze získat z kamery pomocí programů `SaveData` a převést ze souborů `.obs` pomocí programu `ConvertObservationToPointMap` implementovaných v rámci práce.

■ Popis parametrů

Nástroj `6dSlam` používá přepínače pro ICP algoritmus, filtraci bodů a tvorbu bodové mapy. Přepínače jsou při spouštění nástroje zadávány následujícím způsobem `bin/slam6D [options] directory`, tedy například `bin/slam6D -max=500 -r 10.2 -i 20 dat`

- **-a NR**
Slouží k výběru minimalizační metody pro ICP registračního algoritmu. Jako základní volba je Hornova metoda jednotkového quaternionu.
- **-b NR**
Specifikuje velikost listů vytvářeného K-D stromu. Základní hodnota je 20.
- **-c NR**
Parametr určuje minimální vzdálenost smyčky během snímání. Základní hodnotou je 500 cm.
- **-C NR**
Specifikuje minimální počet překrytých snímků pro hledání smyčky. Základní hodnota je 6 snímků.
- **--cache**
Přepínač povoluje cache K-D stromu.
- **--continue**
Přepínač nastavuje výchozí pozici zpracovávaného snímku jako pozici robotu určenou předchozí registrací.
- **-d NR**
Přepínač nastavuje maximální hodnotu pro nalezení párové dvojice, hodnota je udávána v centimetrech. Základní hodnota je 25 cm.
- **-e NR**
Nastavuje pořadové číslo posledního zpracovaného snímku.
- **--exportAllPoints**
Přepínač nastavuje generování výstupu do souboru `points.pts` před ukončením `slam6D`.
- **--epsICP=NR**
Přepínač nastavuje prahovou hodnotu minimální korekce pro ICP. Základní hodnota je 0.00001 cm.

- **-G NR**
Nastavuje minimalizační metodu pro SLAM registraci. Základní nastavení nepoužívá žádnou metodu.
- **-i NR**
Přepínač nastavuje maximální množství iterací ICP při registraci jednoho snímku. Základní hodnota je 50 iterací.
- **-I NR**
Přepínač nastavuje maximální množství iterací graphSLAM. V základním nastavení není graphSLAM využit.
- **-l NR**
Přepínač specifikuje minimální množství snímků ve smyčce.
- **-L NR**
Přepínač určuje metodu pro uzavírání smyček. V základní nastavení není použita žádná metoda.
- **--metascan**
Přepínač zapíná registraci vůči všem doposud zpracovaným snímkům. V základním nastavení probíhá registrace pouze na předchozím snímku.
- **-p**
Přepínač vypíná ICP a důvěřuje pouze souborům pose.
- **-q**
Přepínač zapíná tichý mód.
- **-Q**
Přepínač zapíná super tichý mód.
- **-r NR**
Přepínač zapíná redukci bodů pro registraci pomocí Octree. Určuje minimální délku hrany mřížky obsazenosti.
- **-R NR**
Přepínač nastavuje náhodnou filtraci bodů.
- **-s NR**
Přepínač nastavuje pořadové číslo prvního zpracovávaného snímku.
- **-t NR**
Přepínač nastavuje datovou strukturu pro reprezentaci bodových map. Základní nastavení využívá cache K-D strom.

Kapitola 5

Hardware a implementace

V této kapitole se podrobně seznámíme s použitým hardwarem. Rozebereme implementaci tříd použitých v práci. Popíšeme jednotlivé spustitelné soubory v projektu, jejich účel a nezbytné parametry. Uvedu výsledné programy a podmínky, které předpokládáme při snímání dat. V poslední části se budeme věnovat instalaci nezbytných knihoven, kompilaci práce a testování programů na datech přiložených na CD.

5.1 Použitý hardware

Veškerá implementace a testování byly prováděny na kameře Asus Xtion Pro, detailněji popsané v kapitole 3.1.2. Za účelem srovnání zařízení byla část měření provedena i na kameře Astra S. Kamery byly připojovány k notebooku Asus K52J vybavenému dvoujádrovým procesorem Intel Core i3 350M taktovaný na 2,26 GHz a 4 GB paměti DDR3 s operačním systémem Ubuntu 14.02. Pro správné fungování kamery je u počítače vyžadováno USB 2.0. Offline registrace byla prováděna na počítači N550RC osazeným čtyřjádrovým procesorem i7-6700HQ a 12 GB paměti DDR4.

5.2 Implementované třídy

V rámci práce byly implementovány tři hlavní třídy *XtionGrabber*, *PartialVisualizer* a *MapBuilderWrapper*. Konkrétní popis jednotlivých metod je uveden v hlavičkových souborech implementovaných tříd přiložených na CD. Třída *XtionGrabber* poskytuje snímky ve formátech *mrpt:: maps:: CSimplePointsMap*, *cv::Mat*, *mrpt:: math:: CMatrix*, *mrpt:: obs:: CObservation3DRangeScan* a *mrpt:: obs:: CObservation2DRangeScan*. Třída také umožňuje ukládání souborů s příponou *.obs* a ve formátu umožňujícím zpracování knihovnou 3DTK.

Třída *PartialVisualizer* zprostředkovává vizualizaci jedné bodové mapy, více bodových map v různých barvách nebo živý obraz ve formátu bodové mapy. Třída *MapBuilderWrapper* je obalovou třídou pro *mrpt:: slam:: CMetricMa-*

pBuilderICP. Zprostředkovává vizualizaci dat, zpracování vstupních parametrů, uložení a export mapy.

5.3 Obecné nástroje

Obecné nástroje jsou pomocné spustitelné soubory implementované pro testování, snímání, vizualizaci a konverzi dat. V této kapitole jsou tyto nástroje popsány včetně nezbytných parametrů.

Testování připojení kamery

Pro testování slouží program *TestStream*, který zobrazí obraz snímáný z kamery v nekonečné smyčce. Obraz je zobrazen ve stupních šedi jako dvou-rozměrné pole otevřené v externím panelu.

Snímání a zobrazování dat

Pro jednorázový snímek ze senzoru ve formátu bodové mapy z knihovny MRPT slouží program *TakeScanFromXtion*. Pro běh je nezbytné zadat jméno ukládaného souboru bez přípony. Pro zobrazení bodové mapy slouží program *DrawCloudFromFile*, který také vyžaduje jméno datového souboru.

Dalším programem implementovaným pro snímání v rámci práce je *SaveData*, který přijímá dva parametry. Prvním parametrem je přepínač *-Ob* pro ukládání dat s příponou *.obs* nebo přepínač *-3D* pro ukládání dat určených pro knihovnu 3DTK. Druhým parametrem je počet snímků k uložení.

Konverze dat

Program *ConvertObservationToPointMap*, implementovaný v rámci práce, slouží pro převod nasnímaných dat ve formátu *.obs* do formátu pro knihovnu 3DTK. Datový formát *.obs* je vytvořen jako součást práce a není podporován ani jednou z knihoven. Program přijímá dva argumenty. Prvním argumentem je adresář se soubory *.obs*, druhým parametrem je adresář pro ukládání souborů *.pose* a *.3d*. Soubory *.obs* musí dodržovat stejnou strukturu jako data pro 3DTK.

5.4 Soubory implementované pomocí MRPT

Knihovna Mobile robot programming toolkit, dále jen MRPT, vyvíjena na Univerzitě v Malaga, jejím hlavním autorem je Jose-Luis Blanco-Claraco. Knihovna podporuje velké množství senzorů, zpracování dat a různé konverze. Knihovna obsahuje zpracování odometrických dat a grafické zobrazení zpracovaných údajů. Knihovna je distribuována pod svobodnou BSD licencí, vyjma knihoven třetích stran [5].

■ 5.4.1 2D ranged scanner

Planární scanner využívá pro SLAM třídu `mrpt::slam::CMetricMapBuilderICP`, která zprostředkovává tvorbu mapy z nasnímaných dat ve formátu `mrpt::obs::CObservation`. V implementaci je využívána třída `mrpt::obs::CObservation2DRangeScan` získaná ořezem z `mrpt::obs::CObservation3DRangeScan` pomocí nástroje `obs::T3DPointsTo2DScanParams`. Princip ořezu je popsán v kapitole 3.2.2 `CObservation2DRangeScan` obsahuje jednorozměrné pole nasnímaných vzdáleností v metrech a horizontálním úhlem snímání. Třída `XtionGrabber` poskytuje již ve formátu `mrpt::obs::CObservation2DRangeScan`. Třída `CMetricMapBuilderICP` provádí scan matching pomocí ICP a poskytuje údaj o poslední vypočtené poloze kamery. Popis a nastavení parametrů ICP a mapy je popsán v kapitole 4.2.1.

Spustitelný soubor `2DMapBuilder` jako první parametr přijímá přepínač `-n` pro živé zpracování dat z kamery nebo `-f` pro zpracování nasnímaných dat uložených ve formátu `.obs`. Druhým parametrem je počet snímků ke zpracování při přepínači `-n` nebo složka pro registraci pro `-f` přepínač. Třetím parametrem je jméno souboru, do kterého má být uložena výsledná mapa. Pro správný běh programu musí být ve stejné složce umístěn soubor `MapBuilderConfig.ini`, který je přiložen v příloze B.

■ Omezující podmínky

Při použití senzoru jako ranged scanner předpokládáme následující omezující podmínky.

- Kamera se po celou dobu snímání pohybuje spojitě, bez náhlých změn v rychlosti a směru pohybu.
- Po celou dobu snímání se senzor pohybuje v konstantní výšce a s nulovou či co nejmenší výchylkou v ose Z.
- Jednotlivé snímky na sebe navazují a pracovní prostor se příliš nemění.
- Pracovní prostor neobsahuje dlouhé chodby.

Pokud nejsou tyto podmínky dodrženy, dochází k deformacím snímaného prostoru. V případě snímání dlouhé chodby není identifikován pohyb a vzhledem k absenci odometrických dat je chodba zkrácena na velikost jednoho snímku. V případě rychlých změn v orientaci dojde k nalezení chybné shody. Ta obvykle vede k rychlému zaplnění nasnímaných dat nepřesnostmi, ve snímání poté není možné pokračovat.

■ 5.4.2 Scan matching pomocí ICP

Program `IcpPointMap` využívá třídu `mrpt::slam::CIcp` pro registraci a vizualizaci dvou bodových map ve formátech `mrpt::maps::CSimplePointsMap`. U bodových map je předpokládána vysoká míra vzájemného překryvu. Program

cestu do složek *Redist* a *Include* v souboru *CMakeLists.txt* nebo dodržet již použitou cestu *~/OpenNI/Redist* s *~/OpenNI/Include*. Pokud jsou splněny uvedené požadavky, je možné projekt zkompileovat příkazy 5.1. V případě spuštění programů, které vyžadují *.ini* soubor je nezbytné přesunout tento soubor do složky *bin*.

Listing 5.1: Kompilace práce

```
cd SLAM/  
mkdir bin/  
cd bin  
cmake ..  
make  
./TestStreaming
```

Kapitola 6

Experimentální výsledky

Během práce byly testovány tři programy pro SLAM pracovního prostoru. Prvním testovaným programem je *2DMapBuilder* implementovaný v rámci práce, který pro registraci využívá třídu *mrpt::slam::CMetricMapBuilderICP*. Dalším testovaným programem je *IcpPointMap*, který pro 6D registraci využívá třídu *mrpt::slam::CICP*. Posledním testovaným nástrojem je *slam6D* implementovaný v 3DTK. Pro tento program byla pouze exportována data a nastaveny parametry pro dávkové zpracování.

6.1 2D ranged scanner

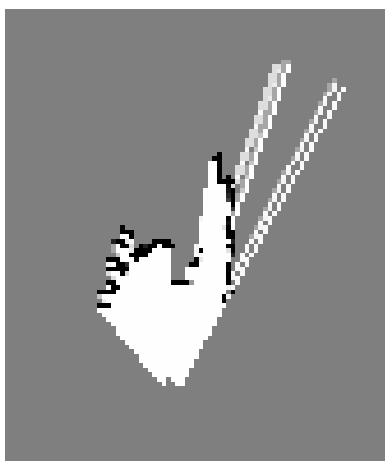
V této části se budeme zabývat testováním programu *2DMapBuilder* využívající třídu *mrpt::slam::CMetricMapBuilderICP* pro planární registraci. Oba uvedené prostory byly nasnímány a později dávkově zpracovány. Soubor *2DMapBuilder* ovšem jako jediný umožňuje i registraci v reálném čase.

6.1.1 Registrované snímky

V rámci testování programu *2DMapBuilder* byly provedeny dva experimenty. První byl proveden v prostorách pokoje s palandou, který je zachycen na obrázcích 6.12 a 6.14. Data pokoje ve formátu *.obs* jsou dostupná na příloženém CD. Druhou registrací byla registrace prostor dvora zachycených na obrázcích 6.2 a 6.3.

Registrace bodových map pokoje

Snímání pokoje proběhlo v malém počtu snímků. Kamera pouze rotovala, byla vychýlena z původní pozice a poté do ní zpět vrácena. Registrovaná mapa obsazenosti je zachycena na obrázku 6.1.



Obrázek 6.1: Mřížka obsazenosti pro snímání pokoje.

■ Registrace bodových map dvora

V rámci snímání dvora bylo zachyceno 150 snímků ve formátu *.obs*. Ty byly poté dávkově převedeny a transformovány do rovinných snímků a registrovány. Registrace snímků vycházela z polohy předchozí registrace a nepoužívala pro zpracování žádná odometrická data. Obrázky zachycují snímaný prostor 6.2, 6.3 a obrázek 6.4 popisuje mřížku obsazenosti.



Obrázek 6.2: Snímaný prostor dvora.



Obrázek 6.3: Snímaný prostor dvora.



Obrázek 6.4: Mřížka obsazenosti pro snímání dvora.

6.1.2 Zhodnocení experimentů

Mapování a lokalizace jsou ve dvou rozměrném pracovním prostoru implementovány velice rychle a umožňují snadné snímání. Implementace umožňuje zpracování až tří snímků za vteřinu při okamžitém zpracování a registraci až deseti snímků za vteřinu při dávkovém zpracování. Senzor RGB-D má vysokou míru šumu a růst nepřesnosti měření s vzdáleností. Tyto nepřesnosti se promítají do algoritmu a znemožňují registraci smyček a uzavřených místností. Nepoužití odometrie způsobuje chybnou registraci rovných chodeb. Program je použitelný pouze v prostorech, které se příliš nemění v ose z . Na obou budovaných mapách jsou jasně patrné vyběhající bílé pruhy, ty jsou způsobeny vadou senzoru v levé části snímací plochy. Této vadě by se dalo předejít oříznutím snímaného prostoru a zmenšením snímaného úhlu.

6.1.3 Nastavení parametrů 2D SLAM

Při optimalizaci parametrů registrace bylo snahou udržení vysoké kvality budované mapy a udržet dobu zpracování snímku pod vteřinu. S tímto cílem byly přenastaveny následující parametry. Konfiguračního souboru *MapBuilder-Config.ini* je přiložen na CD.

- **maxIterations** Maximální počet iterací je možné ponechat vysoký, vzhledem k rychlosti jedné iterace ve dvourozměrném pracovním prostoru. Jako hodnota bylo určeno 80.
- **minAbsStep_trans** Vzhledem k nepřesnostem RGB-D senzoru v řádech desetin milimetru na větší vzdálenosti je možné zvýšit prahovou hodnotu minimální translace až na setiny milimetrů bez ztráty přesnosti registrace. Parametr byl tedy nastaven na 1e-3 mm.
- **minAbsStep_rot** Senzor Asus Xtion pro měří jednotlivé pixely se rozptylem 3.16e-3 radiánu. Prahovou hodnotu minimální rotace je tedy možné zvýšit na 1e-4 radiánu bez ztráty přesnosti registrace.
- **thresholdDist** Hodnota je vzhledem k rychlejším pohybům kamery nastavena na 7cm tedy 0.70.
- **ALFA** Parametr ALFA byl zvýšen až na hodnotu 0.8 pro dosažení co nejpřesnější registrace snímků.

6.2 ICP scan matching v MRPT

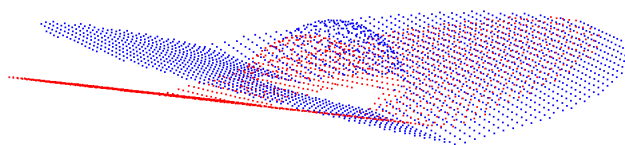
V této části se budeme zabývat testováním programu *IcpPointMap*. Uvedu registrované prostory, výsledky registrace a nalezené optimální nastavení parametrů registrace z hlediska optimalizace doby běhu programu.

6.2.1 Registrované snímky

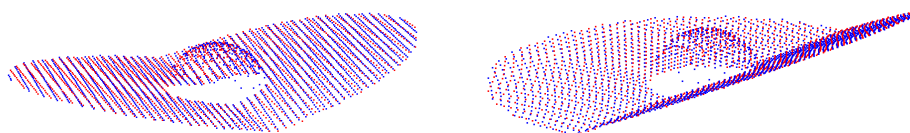
V rámci testování třídy *mrpt::slam::CICP* byla provedeny měření na datech poskytovaných knihovnou MRPT, na stěně s televizorem a poslední registrace probíhala v prostoru pokoje s palandou. Parametry uvedené v sekci 6.2.3 byly použity při všech registracích.

Data poskytovaná knihovnou MRPT

První registrace probíhala na datech, na kterých je třída prezentována v knihovně MRPT. Výchozí poloha bodových map je zachycena na obrázku 6.5. Na obrázcích 6.6 je ukázán výsledek registrace. Data distribuována knihovnou MRPT jsou registrována správně při většině nastavení. Oba snímky obsahují okolo 1700 bodů a jsou registrovány v čase 0.015 sekundy.



Obrázek 6.5: Bodové mapy knihovny MRPT před registrací.

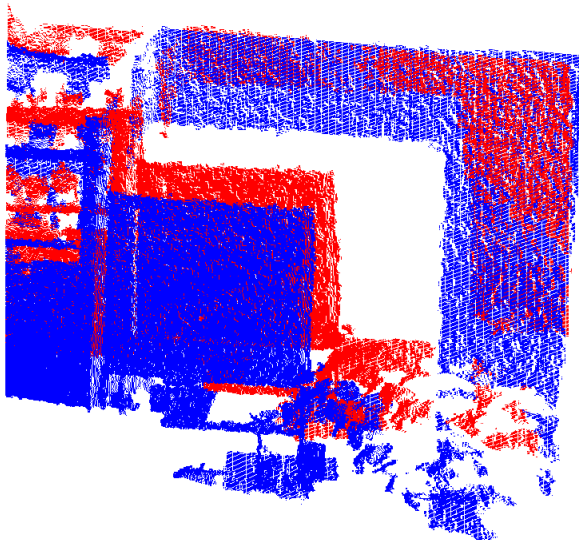


Obrázek 6.6: Výsledná registrace dat knihovny MRPT pomocí třídy `mrpt::slam::CICP`.

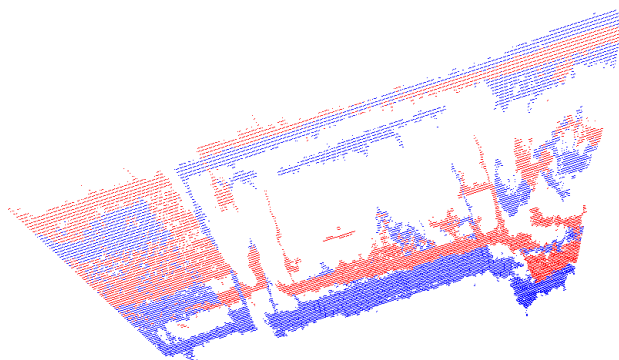
■ Data televizní stěny

Soubory pro registraci byly získány pomocí programu *TakeScanFromXtion* a jsou přiložena na CD. Původní poloha bodových map je ukázána na obrázcích 6.7 a 6.8.

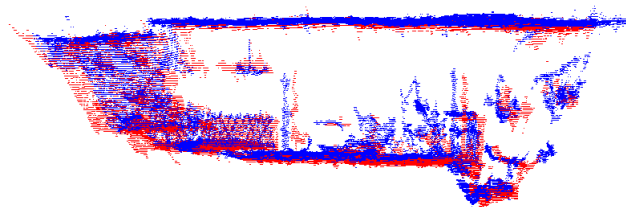
Každý snímek obsahuje 300 000 bodů a registrace je provedena v čase okolo 2.36 sekundy. Tento čas téměř znemožňuje použití implementace ICP pro SLAM na robotu. Bodové mapy jsou v tomto případě velice shodné. Dalším snižováním počtu iterací algoritmu je možné čas dále zkracovat, při rychlé rotaci kamery dochází k větší vzdálenosti bodových map a snímky nejsou správně registrovány. Výsledek registrace při použití parametrů uvedených v kapitole 6.2.3 je popsán na obrázcích 6.9 a 6.10.



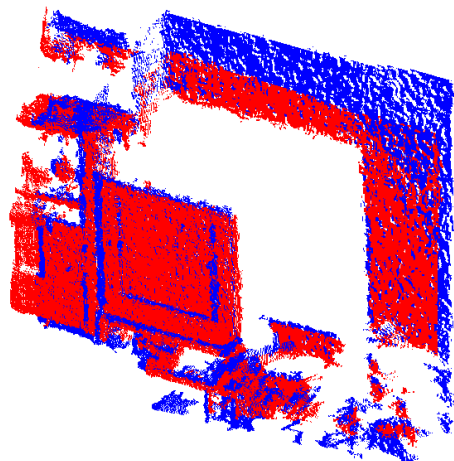
Obrázek 6.7: Bodové mapy televizní stěny před registrací, pohled zředu.



Obrázek 6.8: Bodové mapy televizní stěny před registrací, pohled shora.



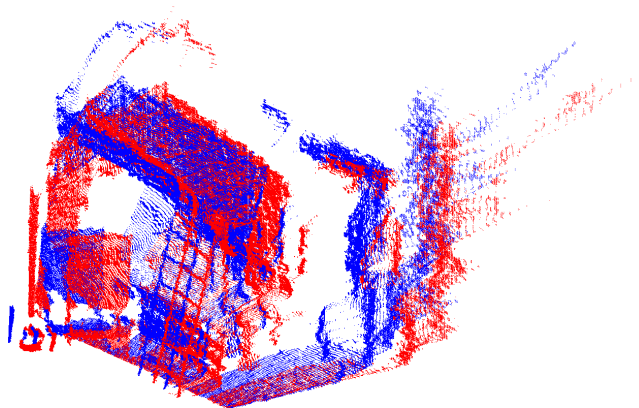
Obrázek 6.9: Registrovaná bodová mapy televizní stěny, pohled zředu.



Obrázek 6.10: Registrovaná bodová mapy televizní stěny, pohled shora.

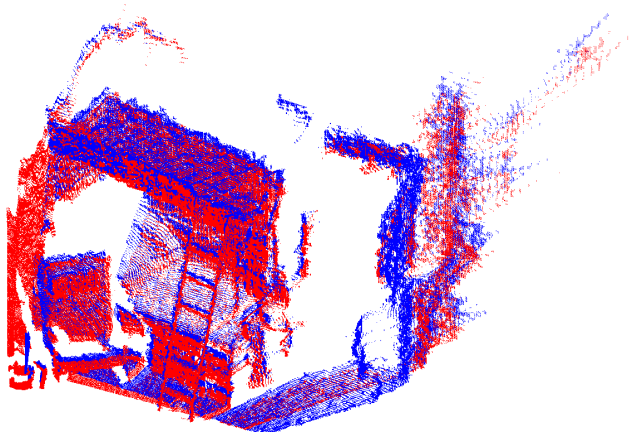
■ Registrace bodových map pokoje

Zpracovávaný prostor je velice členitý a vzájemný posun mezi mapami je v tomto případě největší. Pohyb kamery mezi jednotlivými snímky byl pouze translační. Poloha původních bodových map je popsána na obrázku 6.11.

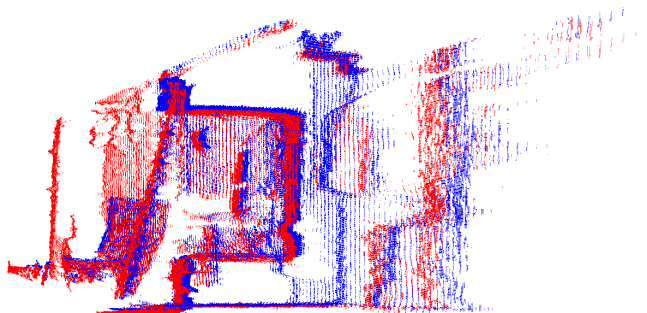


Obrázek 6.11: Bodové mapy před registrací.

Oba snímky obsahují přibližně 300 000 bodů a jejich registrace zabere přibližně 4 sekundy. Vyšší časovou náročnost registrace ve velké míře způsobuje členitost povrchu, která je výrazně větší oproti spíše rovinnému příkladu televizní stěny. Doba registrace vylučuje použití implementace pro SLAM, nicméně vzájemná shoda je po registraci velmi vysoká. Bodové mapy po registraci jsou zobrazeny na obrázcích 6.12 a 6.13.



Obrázek 6.12: Bodové mapy pokoje po registraci, pohled zředu.



Obrázek 6.13: Bodové mapy pokoje po registraci, boční pohled.

6.2.2 Zhodnocení experimentů

Implementace algoritmu ICP pomocí třídy `mrpt::slam::CICP` registruje snímky s vysokou věrohodností. Knihovna MRPT, ani třída `mrpt::slam::CICP` bohužel nenabízejí možnost filtrací bodové mapy. Čas nezbytný pro registraci map prudce roste v závislosti na počtu bodů v mapách a jejich vzájemné vzdálenosti. Oproti ilustračnímu příkladu knihovny MRPT obsahují senzorická data řádově více bodů a časy pro jejich registraci jsou příliš dlouhé a neumožňují zpracování v reálném čase. Čas pro registraci i velice podobných snímků je větší než sekunda. Za této situace není možné použít tuto implementaci pro 6D SLAM bodových map.

6.2.3 Nastavení ICP

Při ICP bylo snahou minimalizovat čas, po který probíhá. Druhým požadavkem je dodržení dostatečné kvality registrace pro lokalizaci. S tímto cílem byly přenastaveny následující parametry. Konfiguračního souboru je přiložen na CD.

- **maxIterations** Snížením počtu iterací ICP je možné dosáhnout výrazné úspory času nezbytného k registraci. Většina registrace je provedena v prvních iteracích. Jako optimální hodnota bylo určeno 50 iterací.
- **minAbsStep_trans** Parametr byl nastaven na hodnotu $1e-3$ mm ze stejných důvodů jako v předchozím nastavení.
- **minAbsStep_rot** Prahová hodnota minimální rotace je nastaven na $1e-4$ radiánu.
- **thresholdDist** Pokud není hodnota parametru dostatečná, neproběhne registrace vzdálenějších bodových map správně. Na testovných datech se osvědčila hodnota 0.50.
- **ALFA** Zvětšením hodnoty ALFA usiluje algoritmus o přesnější registraci. Při příliš vysokém nastavení parametru ALFA má třída `mrpt::slam::CICP`

tendence přesahovat maximální počet iterací až o několik desítek. Jako maximální hodnota ALFA, aby bylo dodrženo předepsané množství iterací, se ukázala hodnota 0.6.

- `kernel_rho` Parametr byl pouze převeden z metrů na milimetry.

6.3 Scan matching pomocí 3DTK

V této kapitole bude popsáno testování nástroje *slam6D* implementovaného v 3DTK. Po náhledu zpracovaných prostor uvedu použité nastavení parametrů. Nastavení parametrů bylo optimalizováno s cílem snížit čas zpracování při dodržení rozumné registrační přesnosti.

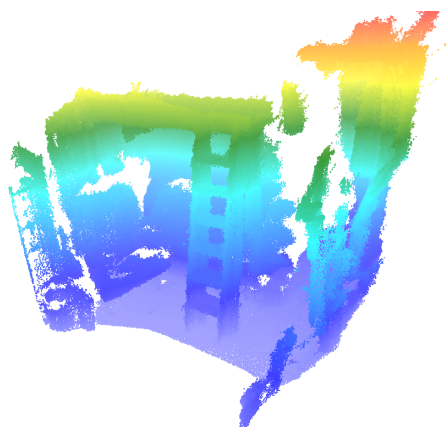
6.3.1 Registrované snímky

V rámci testování nástroje *slam6D* bylo provedeno měření ve vnitřních prostorech a dvě měření v prostorech vnějších. Vnější měření obsahují více snímků a zachycují dvorek a vnější fasádu domu.

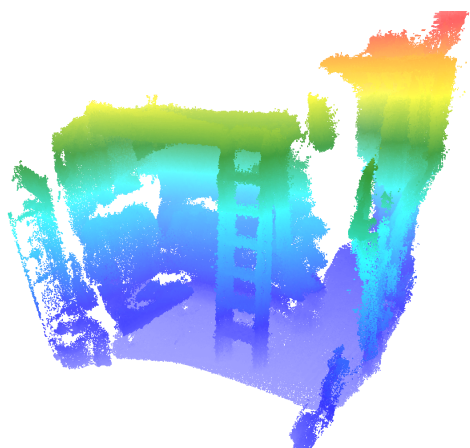
Vnitřní prostory pokoje

Vnitřní SLAM pracuje ve stejném prostoru jako program *IcpPointMap* při třetím experimentu. Datový soubor obsahuje 27 snímků. Snímky před provedením registrace jsou přiloženy na CD. Bodová mapa po registraci je uvedena na obrázku 6.14.

Registrace snímků s použitím implicitních hodnot parametrů trvala 61,1 vteřin a je ukázána na obrázku 6.15. Při použití parametrů uvedených v kapitole 6.3.3 je kompletní registraci, ve srovnatelné kvalitě, možné provést za 13,1 vteřiny. Nejpodstatnější úsporou času je možnost filtrace bodů v OcTree pomocí přepínače `-r`. Použití přepínače `--metascan` umožňuje robustnější registraci, při zpracování 27 snímků nepředstavuje výrazné zpomalení algoritmu. Implementace *slam6D* při registraci pomocí `--metascan` připravává bodovou mapu po každém zpracovaném snímku. Tato aktualizace vede k výraznému zpomalení registrace ve chvíli, kdy bodová mapa obsahuje přibližně 40 milionů bodů.



Obrázek 6.14: Registrace s použitím přepínačů.



Obrázek 6.15: Registrace s použitím implicitního nastavením parametrů.

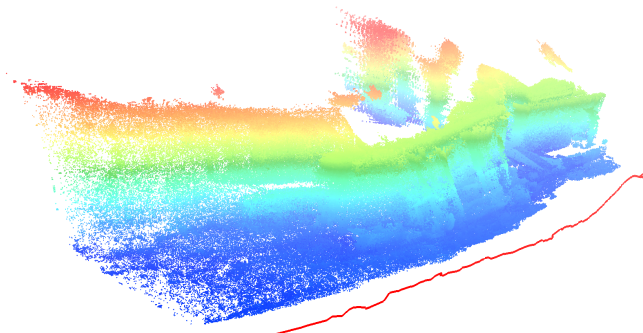
■ Registrace vnější fasády domu

Pro rekonstrukci vnější fasády domu, zachycené na obrázku 6.16, bylo použito 200 obrazových map. Snímání bylo provedeno okolo půlnoci pro minimalizaci IR rušení.

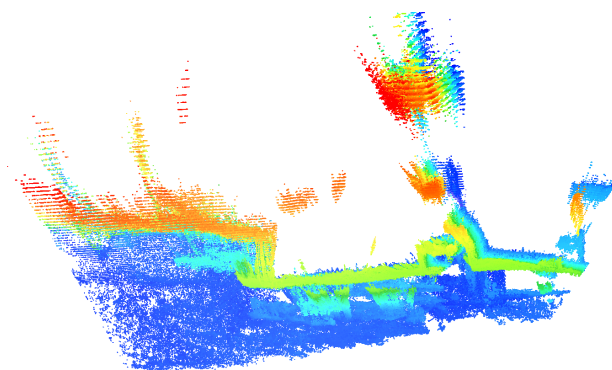
Registrace se zapnutým přepínačem `--metascan` trvá přibližně 1,742 sekund na snímek a výsledná bodová mapa je zobrazena na obrázcích 6.17 a 6.18. Pokud parametr `--metascan` nepoužijeme, dojde k ohybu pracovního prostoru. Čas na registraci se ale výrazně sníží na 0.31 sekund na snímek. Výsledná mapa je zobrazena na obrázku 6.19.



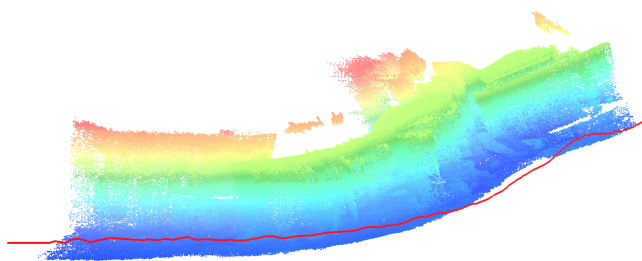
Obrázek 6.16: Fotografie snímaného prostoru vnější fasády.



Obrázek 6.17: Registrovaná mapy fasády domu.



Obrázek 6.18: Registrovaná mapy fasády domu, pohled shora.

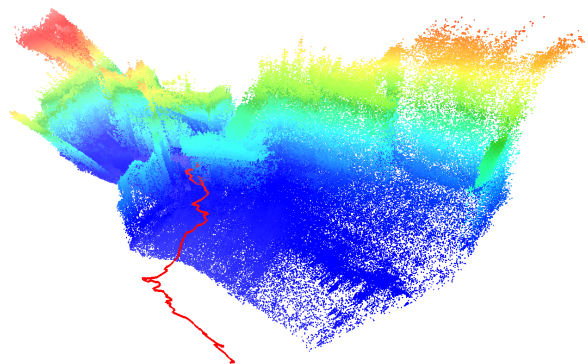


Obrázek 6.19: Registrovaná mapy fasády domu bez přepínače `--metascan`.

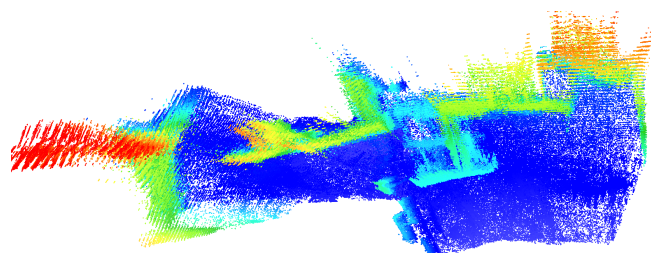
■ Registrace dvora

Posledním mapovaným prostředím je dvůr domu použitý i pro 2D SLAM. Prostor dvora zobrazen na obrázcích 6.2 a 6.3. Prostor je pro SLAM přívětivější než předchozí experiment. Prostor dvora obsahuje velké množství navzájem kolmých rovin a velmi malé množství nesnímatelných ploch. Snímání bylo provedeno v nočních hodinách pro minimalizaci IR rušení.

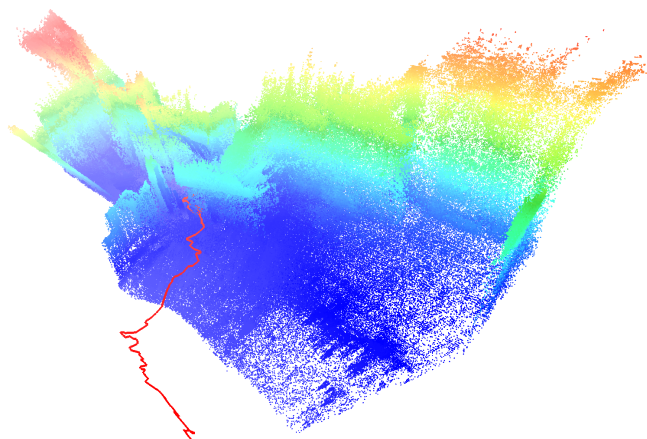
Bodové mapy po registraci bez parametru `--metascan` jsou zobrazeny na obrázcích 6.20 a 6.21, trvání registrace je 351 vteřin. Registrace byla provedena nad stejnými daty i bez parametru `--metascan`. Výsledek je zobrazen na obrázcích 6.22 a 6.23. Doba trvání registrace je 41 vteřin a kvalita registrace je srovnatelná se snímky 6.20 a 6.21.



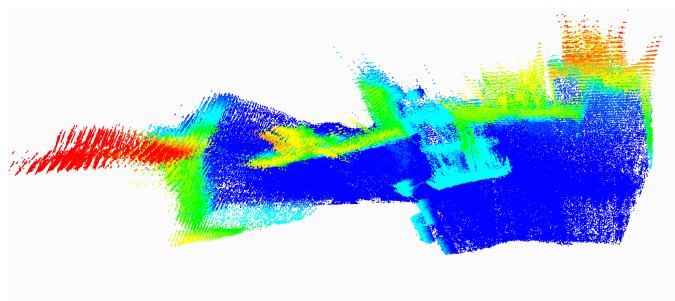
Obrázek 6.20: Registrovaná mapy dvora bez `--metascan`.



Obrázek 6.21: Registrovaná mapy dvora bez `--metascan`, pohled shora.



Obrázek 6.22: Registrovaná mapy dvora s parametrem `--metascan`.



Obrázek 6.23: Registrovaná mapy dvora s parametrem `--metascan`, pohled shora.

■ 6.3.2 Zhodnocení experimentů

Nástroj *slam6D* sady 3DTK nabízí možnost velice rychlého SLAM a robustní registraci snímků i při velkém množství nasnímaných bodů. Použití parametru *metascan* je nezbytné pro správnou registraci velmi členitých prostor. Způsobuje ale velké zpomalení registrace. Při registraci méně členitých prostor není přepínač nezbytný. Ideálním řešením by mohlo být provádění registrace pouze na několika posledních snímcích s konstantním zpožděním.

Sada 3DTK podporuje velké množství rozšíření, např. detekci smyček a použití graph-SLAM. Díky možnosti filtrace bodů je registrace map možná v reálném čase. Sada však není knihovnou a neposkytuje příliš možností pro zpracování nebo navázání na implementaci. Pro použití programu na robotu musejí být vyjmuty používané třídy a zčásti zkompileovány.

■ 6.3.3 Nastavení přepínačů SLAM

Při ladění přepínačů bylo usilováno o registraci snímku pod vteřinu při dodržení rozumné kvality pro registraci. S tímto cílem byly nastaveny následující přepínače.

- **-i** Většina registrace je v programu *slam6D* provedena v prvních iteracích, proto je možné maximální počet iterací snížit s dodržением dobré úrovně přesnosti. Jako optimální hodnota bylo určeno 40 iterací.
- **--epsICP** Podobně jako u předchozího programu je možné zvýšit prahovou hodnotu posunu vzhledem k nepřesnostem RGB-D senzoru. Parametr byl tedy nastaven na 0.001 mm.
- **-r** Při filtraci pomocí OcTree je možné ICP provádět pouze nad pětinou nasnímaných bodů snímku beze ztráty kvality registrace.
- **-d** Hodnota parametru musí být vhodně upravena podle vzdálenosti registrovaných snímků a velikosti pracovního prostoru. Pro sensorická data pohybující se běžně v rozmezí 0.7 až 5.5 metrů se jako optimální hodnota osvědčilo 50 cm při pomalém pohybu kamery.
- **--continue** Tento přepínač je nezbytný při absenci odometrických dat.
- **--metascan** Parametr slouží k robustnější registraci snímků. Při velmi velkém pracovním prostoru a požadavku na rychlost nutné zvážit jeho nezařazení.

Kapitola 7

Závěr

V této práci jsme prověřili práci se třemi implementacemi registračního algoritmu. V souladu se zadáním práce jsme vyhodnotili optimální nastavení parametrů pro budování mapy okolního prostoru v dostatečné kvalitě pro lokalizaci. Pro všechny tři programy jsem provedli optimalizaci parametrů pro zrychlení registrace.

V teoretickém úvodu práce se zabýváme rozborem problematiky lokalizace a mapování včetně popisu činnosti a implementace.

Jako nejlepší SLAM algoritmus se jeví třetí program sady 3DTK, který provádí robustní registraci ve velice solidním čase. Vadou kterou by bylo nezbytné vyřešit při použití na robotu je zpomalení registrace při použití přepínače `--metascan`. Přepínač `--metascan` je v nástrojové sadě implementován velice neefektivně. Jednou z možností jeho reimplementace je provádět registraci vůči konstantnímu počtu předchozích snímků.

Program `2DMapBuilder` pracuje dobře, ale je velice náchylný na měřené prostředí. Vzhledem k absenci odometrických dat špatně registruje pohyb podél rovných zdí. Planarizací senzorických dat dochází k výrazné informační ztrátě a absence odometrie se u tohoto programu projevuje nejvýrazněji. V případě použití této implementace by bylo dobré zvolit planarizaci, která by kompenzovala náklon kamery s využitím kolmých ploch.

Implementace 6D SLAM pomocí třídy `mrpt::slam::CICP` je ze všech použitých řešení nejpomalejší, mnohdy až za únosnou hranici pro lokalizaci robotu. Poskytuje však velice přesnou registraci zpracovávaných dat. Pro její další využití by bylo nezbytné vytvoření datového fitru pro snížení počtu bodů v jednotlivých snímcích.

Všechny tři implementace nasnímaný pracovní prostor zpracovávají do registrované bodové mapy při splnění určitých omezujících podmínek. Těmi je absence přímého slunečního záření, správné nastavení parametrů registrace a plynulost pohybu kamery.

7.1 Možnosti pokračování práce

Práce byla vytvářena za účelem zmapování možných implementací SLAM algoritmů pro autonomní činnost kvadrokoptéry ve vnitřním prostředí. Budoucí postup by měl směřovat k vytvoření SLAM algoritmu na základě výsledků provedených experimentů. Softwarový vývoj by se tedy měl zaměřit na odstranění slabých stránek použité implementace. Vhodným doplňkem by mohlo být zavedení odometrie do mapování.

Co se týče RGB-D senzoru Asus Xtion Pro, by bylo vhodné provést kvalitní kalibraci a kompenzaci nepřesností. Na datech z kamery se v okrajových pixelech velice často objevují prudké ohyby mapované plochy. Druhou možností je provést ořez čerpaných dat o nepřesné okraje snímku.

Pro autonomní činnost kvadrokoptéry je nezbytné vytvořit programy regulující její pohyb a programy plánující bezkolizní pohyb po snímaném prostoru pomocí vytvářené mapy.



Bibliografie

- [1] Orbbec 3D. *Astra & Astra S*. <https://orbbec3d.com/product-astra/>. 2017. (Cit. 25. 04. 2017).
- [2] Orbbec 3D. *Download Orbbec Astra SDK v0.5.0*. <https://orbbec3d.com/develop/>. 2017. (Cit. 16. 05. 2017).
- [3] BBC. *Drones: What are they and how do they work?* <http://www.bbc.com/news/world-south-asia-10713898>. 2012. (Cit. 16. 04. 2017).
- [4] Roman Bedroš. *Modul lokalizace mobilního robotu pro systém Player*. 2011.
- [5] Jose-Luis Blanco-Claraco. *Authors*. <http://www.mrpt.org/Authors/>. 2017. (Cit. 10. 05. 2017).
- [6] Jose-Luis Blanco-Claraco. *Example: kinect-to-2d-laser-demo*. <http://www.mrpt.org/tutorials/mrpt-examples/example-kinect-to-2d-laser-demo/>. 2013. (Cit. 16. 05. 2017).
- [7] Maria Mercè Pons Crespo. “Design, Analysis and Implementation of New Variants of Kd-trees”. Dipl. Polytechnic University of Catalonia, 2010.
- [8] Dabove et al. “GPS & GLONASS Mass-Market Receivers: Positioning Performances and Peculiarities”. In: *Sensors (14248220)* 14 (2014). ISSN: 14248220.
- [9] David. *Advanced Octrees 1: preliminaries, insertion strategies and maximum tree depth*. <https://geidav.wordpress.com/2014/07/18/advanced-octrees-1-preliminaries-insertion-strategies-and-max-tree-depth/>. 2014. (Cit. 22. 04. 2017).
- [10] Felix Endres et al. “3-D Mapping With an RGB-D Camera”. In: *IEEE Transactions on Robotics* 30 (2014).
- [11] Tomáš Foukal. “Demonstrace indexačního algoritmu kD strom a jeho derivátů”. Dipl. Vysoké učení technické v Brně, 2003.
- [12] Armin Hornung et al. “OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees”. In: *Autonomous Robots (2013)* 34 (2013).

Příloha A

Struktura přiloženého CD

```
/
├── Data
│   ├── 3DTK_chodba
│   ├── Obs_pokoj
│   └── PointMap_txt
├── SLAM
│   ├── include
│   │   ├── AstraGrabber.h
│   │   ├── MapBuilderWrapper.h
│   │   ├── SpartialVisualizer.h
│   │   └── XtionGrabber.h
│   ├── src
│   │   ├── includeSrc
│   │   │   ├── AstraGrabber.cpp
│   │   │   ├── MapBuilderWrapper.cpp
│   │   │   ├── SpartialVisualizer.cpp
│   │   │   └── XtionGrabber.cpp
│   │   ├── 2DMapBuilder.cpp
│   │   ├── ConvertObservationToPointMap.cpp
│   │   ├── DrawCloudFromFile.cpp
│   │   ├── IcpPointMap.cpp
│   │   ├── SaveData.cpp
│   │   ├── TakeScanFromXtion.cpp
│   │   └── TestStreaming.cpp
│   ├── IcpConfig.ini
│   ├── MapBuilderConfig.ini
│   └── CMakeLists.txt
├── SLAM mapování vnitřního prostoru 3D kamerou.pdf
└── Autor.txt
```


ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Jakub Nejedlý
Studijní program: Kybernetika a robotika (bakalářský)
Obor: Robotika
Název tématu: SLAM mapování vnitřního prostoru 3D kamerou

Pokyny pro vypracování:

Seznamte se s metodami pro zpracování dat ve formě 3D množin bodů (point cloud) získaných 3D kamerou. Zvolte vhodnou metodu pro vzájemnou registraci jednotlivých měření a implementujte ji pro senzor dodaný pracovištěm vedoucího práce. Metodu volte s ohledem na to, že se předpokládá její použití na robotické helikoptěře pohybující se uvnitř budovy. Vyhodnoťte dosaženou přesnost metody a podmínky, za kterých je vzniklá 3D mapa konzistentní.

Seznam odborné literatury:

- [1] Henry P, Krainin M, Herbst E, Ren X, Fox D.: RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. The International Journal of Robotics Research. 2012.
- [2] Fernández-Madrigal, J.A. and Blanco, J.L.: Simultaneous Localization and Mapping for Mobile Robots: Introduction and Methods, 2012.
- [3] Valenti RG, Dryanovski I, Jaramillo C, Ström DP, Xiao J.: Autonomous quadrotor flight using onboard RGB-D visual odometry. In Robotics and Automation (ICRA), 2014 IEEE International Conference on 2014.

Vedoucí bakalářské práce: Ing. Jan Chudoba

Platnost zadání: do konce letního semestru 2017/2018

L.S.

prof. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 30. 1. 2017