

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE FAKULTA
ELEKTROTECHNIKY A INFORMATIKY**

Ing. Peter Beňo

Autoreferát dizertačnej práce

RGBD kamery v mobilnom robotickom mapovaní

**na získanie akademického titulu
doktor (philosophiae doctor, PhD.)**

v doktorandskom študijnom programe
Robotika a kybernetika

v študijnom odbore
9.2.7. kybernetika

Forma štúdia
Denná forma

Miesto a dátum
Bratislava, August 2018

Dizertačná práca bola vypracovaná

Na	Ústav robotiky a kybernetiky Fakulta elektrotechniky a informatiky Slovenská technická univerzita v Bratislave Ilkovičova 3, 812 19 Bratislava
Predkladateľ	Ing. Peter Beňo Ústav robotiky a kybernetiky Fakulta elektrotechniky a informatiky Slovenská technická univerzita v Bratislave Ilkovičova 3, 812 19 Bratislava
Školiteľ	doc. Ing. František Duchoň, PhD. Ústav robotiky a kybernetiky Fakulta elektrotechniky a informatiky Slovenská technická univerzita v Bratislave Ilkovičova 3, 812 19 Bratislava
Oponenti	prof. Ing. LUDĚK ŽALUD Ph.D. Ústav automatizace a měřicí techniky Fakulta elektrotechniky a komunikačních technologií Technická 3082/12, Královo Pole, 61600, Brno, Česká republika doc. Ing. Ján Vachálek, PhD. Ústav automatizácie, merania a aplikovanej informatiky Strojnícka fakulta Slovenská technická univerzita v Bratislave Ilkovičova 3, 812 19 Bratislava

Autoreferát bol rozoslaný:

Obhajoba dizertačnej práce sa bude konať dňa **o**.....**h**

v miestnosti

Na:
Ústav robotiky a kybernetiky
Fakulta elektrotechniky a informatiky
Slovenská technická univerzita v Bratislave
Ilkovičova 3, 812 19 Bratislava

prof. Dr. Ing. Miloš Oravec
dekan
Fakulta elektrotechniky a informatiky

Abstrakt

Dôležitým prvkom naozaj autonómneho mobilného robotického systému je jeho schopnosť vnímať prostredie, v ktorom sa nachádza a pohybovať sa v ňom. Aby bol robot schopný autonómne prechádzať prostredím a plniť v ňom svoje úlohy, je výhodné, aby disponoval mapou prostredia. Mapu prostredia je možné vytvárať na základe dát zo širokého spektra snímačov. V posledných rokoch výrazne získavajú na popularite vizuálne mapovacie algoritmy používajúce hĺbkové kamery. Hlavným dôvodom je nízka cena a dobrá dostupnosť takýchto zariadení.

Dostupné algoritmy sa zameriavajú najmä na kvalitu výslednej mapy, bez ohľadu na náročnosť spracovania. To výrazne limituje možnosti nasadenia aktuálnych metód na platformách s obmedzeným výpočtovým výkonom.

Práca čitateľovi predstavuje témy hĺbkových kamier, vizuálnej odometrie a robotického mapovania v širšom kontexte. Z početného spektra dostupných postupov identifikuje algoritmy, ktoré je možné optimalizovať pre beh na malom mobilnom robote. Na ich základe navrhuje riešenie vizuálnej odometrie a mapovania pre túto platformu. Pre zvýšenie robustnosti voči zmenám osvetlenia je použitý filter obrazu CLAHE. Za účelom umožnenia behu s frekvenciou 30 Hz je predstavená schéma paralelizácie na výpočtovej jednotke CPU. Na zníženie tempa rastu absolútnej chyby je predstavený mechanizmus extrakcie lokálneho modelu mapy. Ostatné časti riešenia sú dôsledne analyzované s ohľadom na výpočtovú zložitosť a kvalitu výstupu.

Navrhnuté riešenie bolo implementované a jeho vlastnosti overené na reálnom robotickom systéme.

Abstract

One of important aspects of truly autonomous mobile robot is its ability to sense the environment around. For tasks such as navigation, exploration and disaster response, it is significantly beneficial for the robot to know the map of the environment. Maps of environments can be created with data from various sensors. In recent years, mapping algorithms utilizing depth cameras have gained significant popularity, mainly because of their low price and good availability on the market.

Available implementations focus on quality of produced maps, sacrificing the processing speed. This fact limits applications on systems with limited processing power.

This thesis introduces topics of depth cameras, visual odometry and robotic mapping in wider context. From numerous range of available approaches, algorithms suitable for small mobile robot are identified and used for design of proposed solution. To raise the robustness to varying lightning conditions, usage of CLAHE filter is proposed. Schema of parallelization on CPU is introduced which enables the solution to run with frame-rate of 30 Hz. Method of local map extraction is then introduced. This method limits the rate of growth of absolute error and enables camera position recovery after extreme errors. The proposed solution was implemented and its properties were evaluated on real robotic system.

Obsah

Úvod	5
Tézy práce	5
Experimentálna platforma.....	6
Prehľad predchádzajúcej práce, dostupné implementácie	7
Metodika vyhodnotenia	7
Navrhnuté riešenie.....	8
Filter CLAHE.....	10
Paralelné spracovanie	12
Extrakcia lokálneho modelu	14
Mapy prostredí vytvorené pomocou navrhovaného riešenia	17
Záver	19
Prínosy dizertačnej práce	20
Literatúra	21
Publikované práce autora.....	22

Úvod

V posledných rokoch sme boli svedkami nástupu mobilných robotických systémov do našich životov. Vo výrobe automatické vozíky pomáhajú s presunom tovarov a výrobkov. Čistenie podlahy v domácnostiach vykonávajú robotické vysávače, deti sa hrajú s robotickými hračkami, povrch Marsu brázdia ľuďmi vytvorené vozidlá. Pri praktických aplikáciách je veľmi výhodné, keď mobilný robot disponuje určitým stupňom autonómie. Automatický vysávač musí byť schopný samostatne pokryť podlahu celej upratovanej miestnosti, Mars rover Opportunity musí byť schopný pracovať na svojich úlohách aj v čase, keď nie je k dispozícii rádiové spojenie.

Dôležitým prvkom naozaj autonómneho mobilného robotického systému je jeho schopnosť vnímať prostredie, v ktorom sa nachádza a pohybovať sa v ňom. Aby bol robot schopný autonómne prechádzať prostredím a plniť v ňom svoje úlohy, je veľmi výhodné, aby disponoval mapou prostredia. Mapu prostredia je možné vytvárať na základe dát zo širokého spektra snímačov. V posledných rokoch výrazne získavajú na popularite vizuálne mapovacie algoritmy používajúce hĺbkové kamery. Hlavným dôvodom je nízka cena a dobrá dostupnosť takýchto zariadení. Od uvedenia lacných zariadení ako napríklad snímača Microsoft Kinect boli predstavené viaceré techniky, ktoré dokážu vytvárať modely priestoru pomocou tohoto snímača. To láka robotickú komunitu ku využívaniu týchto techník pre vytvorenie mapovacích algoritmov. Problém robotického mapovania nie je však novým problémom a pri návrhu naozaj úspešných nových riešení je potrebné zohľadniť aj historický vývoj tejto problematiky.

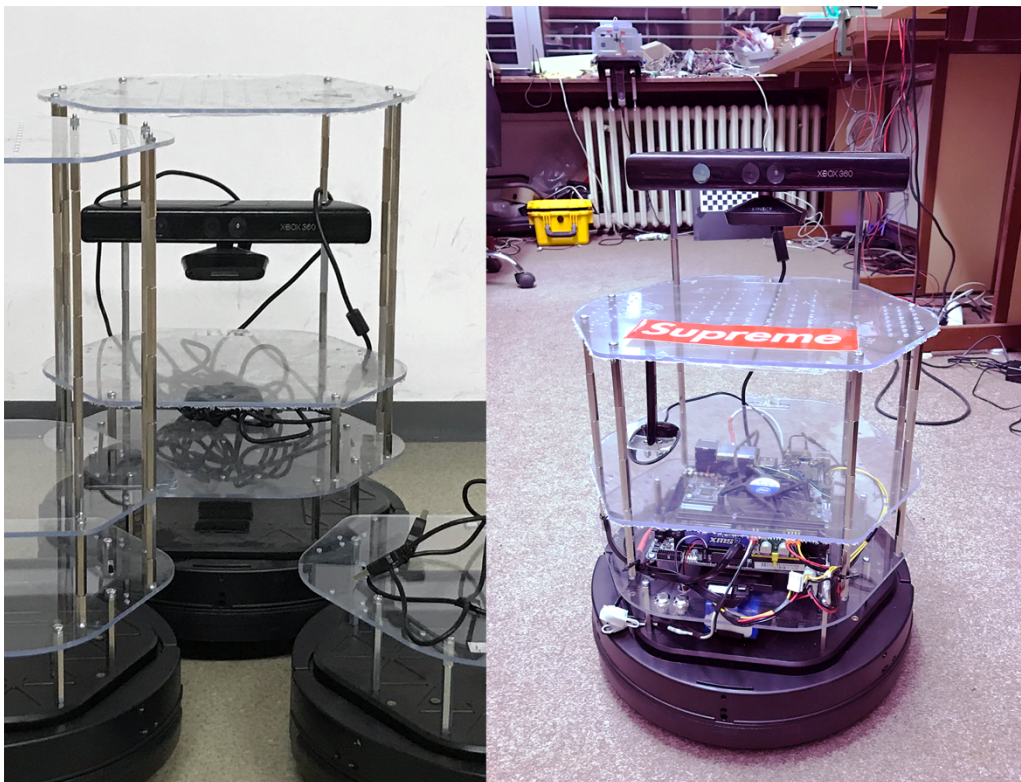
Práca čitateľovi predstavuje témy hĺbkových kamier, vizuálnej odometrie a robotického mapovania v širšom kontexte. Z početného spektra dostupných postupov identifikuje algoritmy, ktoré je možné optimalizovať pre beh na malom mobilnom robote. Na ich základe navrhuje riešenie vizuálnej odometrie a mapovania pre túto platformu. Pre zvýšenie robustnosti voči zmenám osvetlenia je použitý filter obrazu CLAHE. Za účelom umožnenia behu s frekvenciou 30 Hz je predstavená schéma paralelizácie na výpočtovej jednotke CPU. Na zníženie tempa rastu absolútnej chyby je predstavený mechanizmus extrakcie lokálneho modelu mapy. Ostatné časti riešenia sú dôsledne analyzované a optimalizované s ohľadom na výpočtovú zložitosť a kvalitu výstupu. Navrhnuté riešenie bolo implementované a jeho vlastnosti overené na reálnom robotickom systéme.

Tézy práce

1. Navrhnuť nový vizuálny SLAM algoritmus využívajúci RGBD dáta pre malý mobilný robot s čo možno najnižšími nárokmi na hardvérovú konfiguráciu robota.
2. Navrhnuť mapové reprezentácie a spôsob prenosu údajov medzi viacerými robotmi, ktoré by umožňovali ľahký prenos údajov z mapy cez sieť a jej vhodnú distribúciu medzi viaceré roboty.
3. Vytvoriť stratégie multiagentového mapovania priestoru viacerými robotmi a spájania týchto čiastkových máp z viacerých robotov do jednej spoločnej globálnej mapy.
4. Všetky navrhnuté postupy a algoritmy implementovať do reálnych zariadení a verifikovať ich z hľadiska požadovaných vlastností.

Experimentálna platforma

Vytvorené riešenie vizuálnej odometrie a mapovania bolo vyvíjané a experimentálne overované na robote TurtleBot 2, ktorá bola skonštruovaná na pracovisku pre účely tejto práce. **TurtleBot** je svetovo najpopulárnejšia ROS platforma pre vzdelávanie a výskum využívajúca softvér s otvoreným zdrojovým kódom. Vznikla v roku 2010 v spoločnosti Willow Garage a vďaka podpore ROS sa rýchlo rozšírila do celého sveta. Na trh boli doposiaľ uvedené 3 generácie robota TurtleBot a ich viaceré modifikácie [1].



Obrázok 1 - Vytvorený mobilný robot TurtleBot 2. Vľavo robot v skupine robotov rovnakého typu, na ktorý je pre použitie treba umiestniť notebook. Vpravo robot so zvýšeným umiestnením snímača Kinect a osadeným pc v Mini-ITX formáte. Napájanie osadeného PC je zabezpečené samostatnou batériou.

Použitý robot TurtleBot 2 sa skladá z robotického podvozku Kobuki, snímača Kinect a bežného notebooku alebo inej výpočtovej jednotky. Na podvozku je upevnená nosná konštrukcia a hĺbková kamera, na nosnej konštrukcii je umiestnený notebook. Snímač Kinect aj podvozok Kobuki sa ku notebooku pripájajú pomocou zbernice USB 2.0. Potrebná kabeláž bola vyrobená podľa dokumentácie z bežne dostupných elektronických komponentov. Snímač Kinect je umiestnený v zadnej časti robota na zvýšenej platforme, čo mu umožňuje pokryť svojim zorným poľom väčšiu časť prostredia. Podvozok disponuje hardvérovým vybavením pre implementáciu gyro odometrie, ktorá bola použitá pre získanie referenčných trajektórií, voči ktorým bolo naše riešenie vyhodnotené.

V čase dokončenia práce je na pracovisku k dispozícii päť robotických podvozkov Kobuki, ktoré autor prostredníctvom **Programu pre podporu mladých výskumníkov STU** v rámci projektu **Platforma pre výskum multiagentových systémov mobilných robotov** dovybavil nadstavbami TurtleBot 2 a ktoré je možné použiť pre ďalší výskum.

Prehľad predchádzajúcej práce, dostupné implementácie

V našej predchádzajúcej práci boli identifikované [2] a analyzované [3] dostupné riešenia pre tvorbu 3D mapy prostredia vo formáte meshu pomocou snímača Kinect. Analyzované boli dve implementácie – **RGBD-6D-SLAM** [4] a **KinectFusion** [5].

Riešenie **RGBD-6D-SLAM** implementuje systém vizuálnej odometrie postavenej na príznakoch z 2D obrazu (feature-based SLAM). Typ použitého detektora je SURF [6]. Pre zabezpečenie robustnosti je použitá metóda RANSAC. Transformácia medzi snímkami je ďalej korigovaná pomocou metódy ICP [7] [8]. Poloha kamery je lokálne optimalizovaná pomocou algoritmu HOG-man [9]. Výstupný formát mapy je mračno bodov. **Riešenie poskytuje veľmi nízku rýchlosť spracovania 2 s na jeden snímok a vytvorená mapa vo formáte mračna bodov nie je vhodná pre riešenie úloh mobilnej robotiky.**

Kinect fusion algoritmus je metóda 3D povrchovej rekonštrukcie na základe RGBD dát z kamery Microsoft Kinect. Umožňuje konzistentné mapovanie vo vysokej rýchlosti iba na základe hĺbkových dát, takže funguje aj bez viditeľného svetla. Systém v reálnom čase sleduje polohu kamery v šiestich stupňoch voľnosti a integruje nové merania do jedného globálneho 3D modelu pomocou kľzavého priemeru. Rekonštrukcia sa stáva detailnejšou s vyšším počtom spracovaných snímkov danej oblasti. Základným stavebným prvkom Kinect Fusion algoritmu je **Truncated signed distance function (TSFD)** [10].

Po overení dostupných riešení vizuálnej odometrie a získaní máp vo formáte meshu bola navrhnutá vlastná implementácia mapy vo formáte oktálového stromu. Mapy vo formáte oktálového stromu boli potom využité pre riešenie problému navigácie mobilného robota pomocou metódy A* [11]. V práci bolo preukázané, že mapa vo formáte oktálového stromu predstavuje optimálny mapový formát pre návrh mapovacieho algoritmu pre malý mobilný robot. Výhodné sú najmä nízke nároky na úložisko a vysoká rýchlosť pridania nového snímku do už existujúcej mapy. Pri rozlíšení 9 cm bola veľkosť súboru mapy zredukovaná o 99,99%. Na tejto mape bolo vykonané aj experimentálne overenie rýchlosti plánovania globálnej trajektórie pomocou algoritmu A*, ktoré potvrdilo, že v tejto mape je možné časovo efektívne plánovať pohyb mobilného robota [11].

Metodika vyhodnotenia

Pre vyhodnotenie chybovosti nami zrekonštruovanej trajektórie pohybu kamery v priestore od referenčnej trajektórie určenej pomocou gyro odometrie podvozku Kobuki boli analyzované:

- Absolútna chyba polohy a orientácie, tempo rastu v čase
- Chyba polohy a orientácie počas jedného časového kroku, teda jednej iterácie algoritmu
- Rozptyl a stredná hodnota chyby

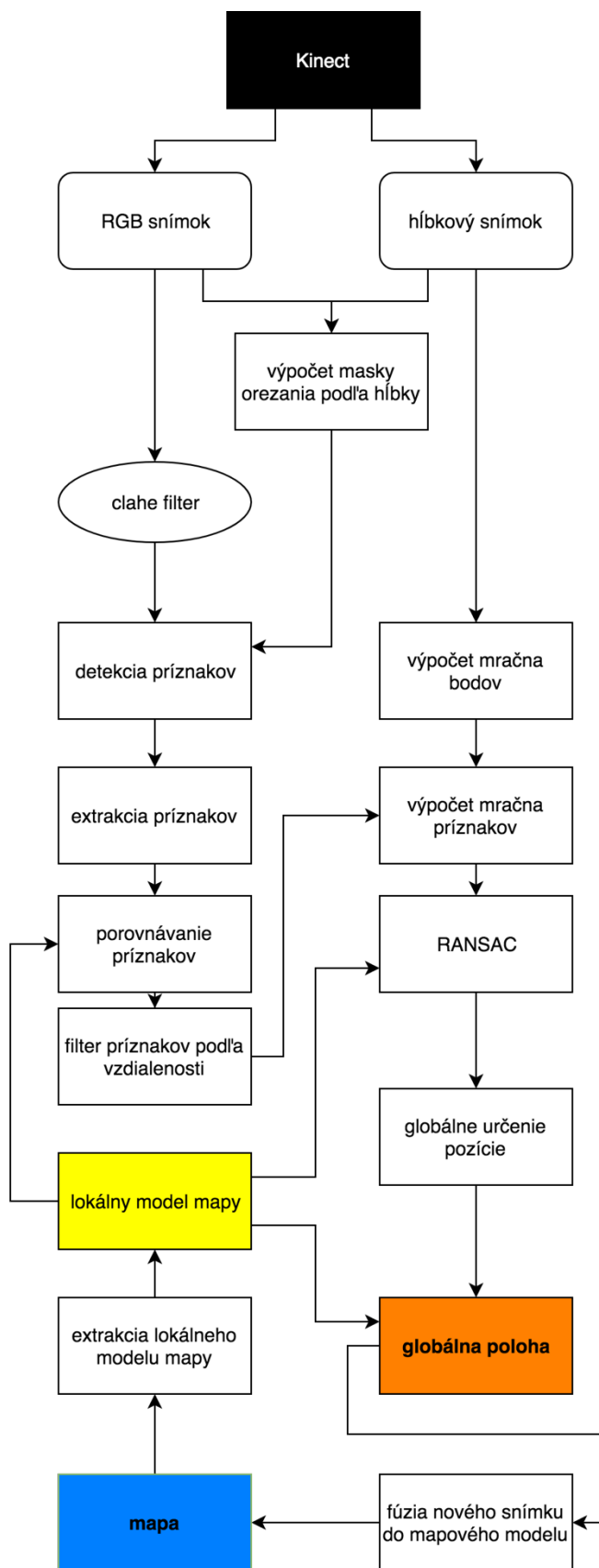
Keďže sa v nameraných dátach nachádzali aj extrémne hodnoty chýb spôsobené vlastnosťami snímaných scén alebo príliš rýchlym pohybom, boli tieto z dát odfiltrované pomocou kritéria 3-sigma a štatistické rozloženia chýb boli určené opakovane. Chyba polohy bola určená ako euklidovská vzdialenosť polohy určenej vizuálnou odometriou a referencie. Chyba orientácie bola vypočítaná ako veľkosť rovinného uhla definovaného smerovými vektormi oboch polôh umiestnenými do počiatku súradnicovej sústavy na rovine preloženej týmito vektormi.

Navrhnuté riešenie

Základnou otázkou, ktorú je potrebné vyriešiť pri návrhu riešenie vizuálnej odometrie a mapovania je výber typu vizuálnej odometrie vhodného pre cieľovú platformu. Na platformách s vysokým výkonom alebo disponujúcimi grafickou kartou je výhodné využiť náročnejšie metódy. To sú napríklad metódy priamej vizuálnej odometrie alebo prístupy postavené na technike Kinect Fusion. Tieto prístupy umožňujú vytváranie máp vo veľmi vysokej kvalite. Situácia je však odlišná na malých mobilných platformách. Pre tieto platformy a teda aj pre naše riešenie, je najvýhodnejšie použiť typ **vizuálnej odometrie postavenej na príznakoch (feature-based VO)**, ktoré:

- Redukujú počet spracovaných dát na minimálnu relevantnú vzorku, čím dosahujú výraznú úsporu času spracovania.
- Sú odolné na väčšie posuny kamery, čo umožňuje konzistentné mapovanie aj pri rýchlejších posunoch kamery a naopak možnosť vynechávania snímok pri pomalom pohybe za účelom zníženia výpočtovej náročnosti riešenia.

Nevýhodou tohoto typu vizuálnej odometrie sú kroky detekcie, extrakcie a porovnávania príznakov, ktoré sú výpočtovo náročné a je potrebné ich vykonávať sekvenčne za sebou. Pre prekonanie tohoto problému bola navrhnutá **schéma paralelného spracovania na CPU**. Zároveň bol vybraný najvhodnejší detektor príznakov z hľadiska výpočtového času. Nevýhodou príznakových metód je aj fakt, že porovnané príznaky obsahujú tzv. outliers, čiže ich nie je možné priamo použiť pre určenie transformácie pohybu kamery. Tento problém bol riešený pomocou **filtra príznakov a metódy RANSAC**. Pre zvýšenie odolnosti na zlé svetelné podmienky a nedostatočnú štruktúru scény bolo navrhnuté využitie filtra **CLAHE**. Za účelom zníženia tempa rastu globálnej chyby boli využité princípy **porovnávania snímku a lokálneho modelu (frame-to-model matching)**. Tento prístup zároveň umožňuje algoritmu obnovu správnej pozície po výskyte extrémnej chyby. Blokovú schému celého navrhovaného riešenia zobrazuje Obrázok 2. Kľúčové časti riešenia sú popísané a vyhodnotené v nasledujúcich kapitolách. Oproti riešeniu RGBD-6D-SLAM [4] nebola pre doladenie transformácie pohybu kamery použitá metóda ICP, keďže ju nebolo možné implementovať ani parametrizovať pri zachovaní požadovanej frekvencie spracovania.

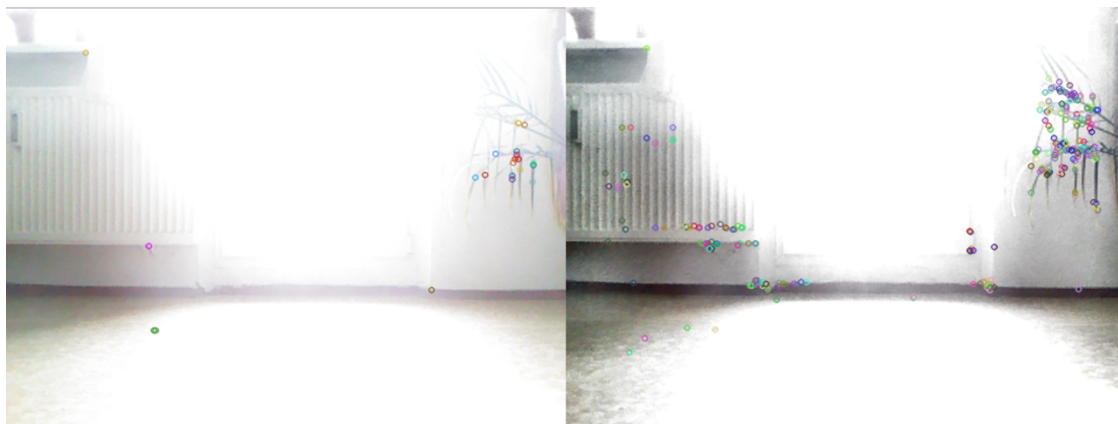


Obrázok 2 - Bloková schéma navrhnutého riešenia

Filter CLAHE

AHE je skratka pre **adaptive histogram equalization** teda **adaptívnu ekvalizáciu histogramu** [12]. Jedná sa o techniku využívanú v počítačovom spracovaní obrazu pre zlepšenie kontrastu v obraze. Odlišuje sa od klasickej ekvalizácie histogramu [13] tak, že pre výpočet používa viacero histogramov, ktoré korešpondujú s lokálnymi časťami obrazu a používa ich pre zlepšenie lokálneho kontrastu v tejto oblasti a teda zvýraznenie rohov a hrán v obraze. Jednou z nevýhod AHE je, že zvýrazňuje šum v relatívne homogénnych častiach obrazu. Metóda CLAHE (contrast limited AHE) [14] oproti AHE navyše obmedzuje silu zvýraznenia v lokálnych častiach a tak úspešne vznik šumu potláča.

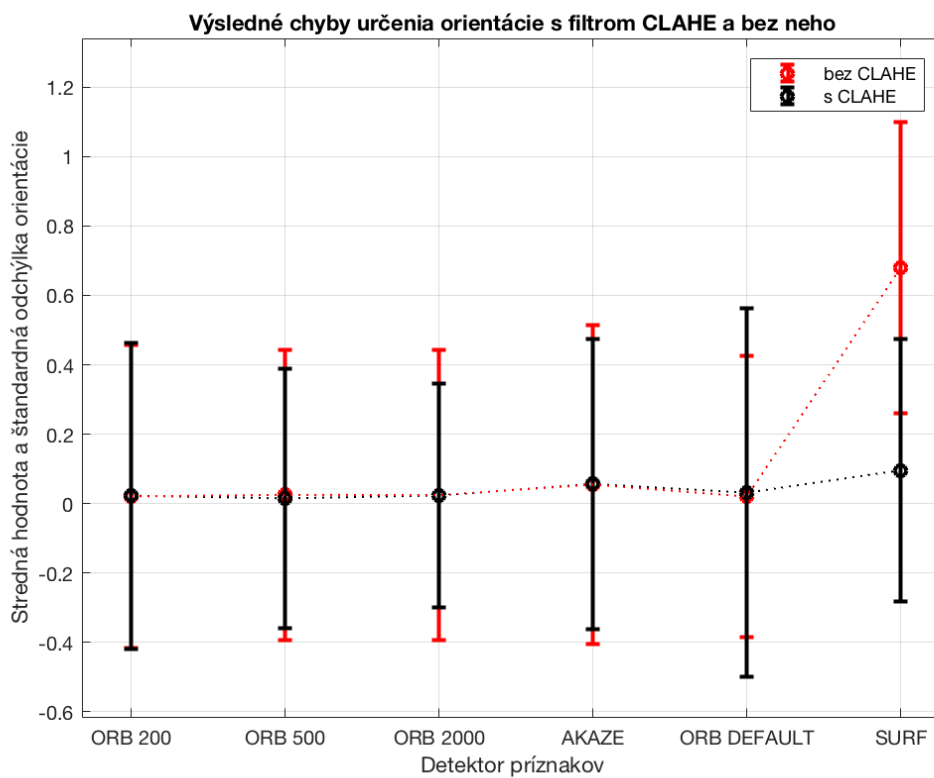
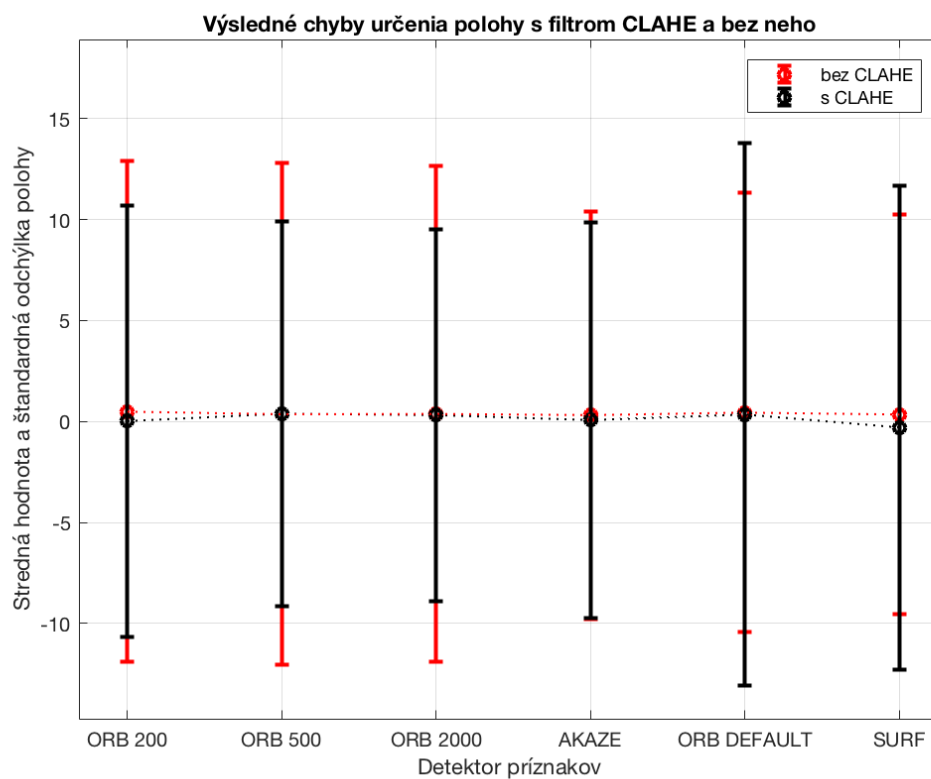
Filter CLAHE je v navrhnutom riešení použitý pre kompenzáciu nevhodného osvetlenia scény. Snahou bolo nájsť filter, ktorý je možné použiť pre spracovanie v reálnom čase, teda s čo najmenšími nárokmi na výpočtovú zložitosť algoritmu. Filter CLAHE má výpočtovú zložitosť $O(N^2)$ kde N je šírka okna použitého pre výpočet v pixeloch. Rýchlosť výpočtu CLAHE filtra pre jeden snímok z RGB kamery snímača Kinect, teda pre rozlíšenie 640x480 pixelov bola priemerne 7 ms pre všetky overované datasety. Keďže cieľom bola úprava svetelnosti scény, obraz bol pred aplikáciou filtra prevedený do farebného modelu CIELAB [15] a filter bol aplikovaný iba na zložku obrazu L , ktorá korešponduje so svetelnosťou obrazu. Vplyv aplikácie filtra CLAHE na RGB snímok zo snímača Kinect obsahujúcom presvetlenú oblasť zobrazuje Obrázok 3.



Obrázok 3 - porovnanie RGB obrazu v presvietenej oblasti. Vľavo obrázok z kamery s detegovanými príznakmi, vpravo obraz po aplikácii filtra CLAHE. Na obraze vidíme vyššie množstvo detegovaných príznakov, výraznejšie hrany a rohy.

Experimentálne overenie ukázalo, že filter CLAHE zvyšuje úspešnosť výpočtu transformácie pohybu kamery v oblastiach trajektórie, kde RGB kamera sníma presvietené, podsvietené, alebo príliš svetelne kontrastné scény. U všetkých testovaných detektorov príznakov dôjde ku zvýšeniu prahu pozorovateľnosti príznaku ako následku lokálnej korekcie kontrastu. Tým pádom sa zvýši počet detegovaných príznakov.

Počas experimentov nebolo pozorované, že by použitie filtra CLAHE s detektormi príznakov ORB a AKAZE so zníženou robustnosťou na zmenu mierky prinášalo nevýhody, ak sa jednalo o spracovanie v reálnom čase, t.j. algoritmus bol schopný spracovať všetky dodané snímky s frekvenciou ~ 30 Hz aj s využitím filtra. Ak však použitie filtra prináša spomalenie celkového výpočtu vizuálnej odometrie a tým pádom aj pokles frekvencie spracovania snímok, je potrebné jeho použitie zvážiť. Do úvahy pripadá aj jeho cielená aktivácia a deaktivácia vo zvolených úsekoch scén, keď si to osvetlenie scény vyžaduje alebo na surových snímkach z RGB kamery nie je dostatočné množstvo detegovaných príznakov.



Obrázok 4 - Chyby polohy a orientácie s filtrom CLAHE a bez neho pre rôzne typy a konfigurácie detektorov príznakov

Paralelné spracovanie

Vyžitie viacerých výpočtových jadier pri implementácii operácii v počítačovom spracovaní obrazu je tradičnou technikou, ako zvýšiť rýchlosť spracovania dátovej vzorky. V praxi sa je možné stretnúť s dvoma druhmi paralelizmu – paralelizmus na grafickej karte počítača (GPU) a paralelizmus na výpočtovej jednotke CPU. Oba postupy využívajú odlišnú hardvérovú jednotku a ku ich návrhu je potrebné pristupovať odlišne. Pri paralelizme na GPU sú implementované algoritmy s vysokým prietokom dát, kde je vysoké množstvo dátových vzoriek spracovávaných súčasne jedným algoritmom na veľkom počte jadier. Pri paralelizme na CPU odlišujeme úlohy, ktoré môžu prebiehať súčasne od tých, ktoré musia prebiehať sekvenčne, organizujeme ich simultánny beh a tým je dosiahnutá úspora času spracovania.

Keďže neuvažujeme, že malý mobilný robot bude vybavený GPU, vytvorený systém vizuálnej odometrie bol paralelizovaný na CPU. Táto kapitola opisuje návrh distribúcie výpočtu vizuálnej odometrie na viacero jadier, ktorý bol implementovaný v našom riešení.

Po úspešnej implementácii algoritmu schopného spracovávať všetky zozbierané datasety na jednom vlákne boli zamerané priemerné časy behu všetkých jeho kľúčových častí na všetkých zozbieraných datasetoch (16 000 snímok). Ako výpočtovo najzložitejšie sa ukázali – výpočet filtra CLAHE (9 ms), výpočet mračna bodov (8 ms) detekcia príznakov a extrakcia príznakov (15 – 200 ms), porovnávanie príznakov (15-40 ms) a samotná metóda RANSAC (2 – 200 ms). Keďže stredná hodnota spracovania jedného snímku na jednom výpočtovom jadre je vždy viac ako 30 ms, nebolo možné dosiahnuť spracovanie v reálnom čase na jednom jadre.

Pri našej implementácii boli využité 4 simultánne bežiace vlákna, ktoré spolu zdieľajú 3 zdieľané prostriedky:

1. **Vlákno pre prístup ku snímaču Kinect** – vlákno potrebuje v priemere 1 ms pre získanie snímku zo snímača Kinect a 5 ms pre prečítanie uloženého snímku z disku pri behu zo záznamu. Vlákno aktuálne získaný snímok uloží do zdieľanej premennej pre ďalšie vlákno a spí. Čas spania je nastavený tak, aby vlákno generovalo nový snímok približne každých 30 ms, ako snímač Kinect. Pomocou zmeny hodnoty spania je možné simulovať iné snímkovacie frekvencie snímača.
2. **Vlákno filtrácie a detekcie príznakov** – toto vlákno si preberie snímok zo zdieľanej premennej z vlákna 1 a vykoná nasledovné operácie:
 - **Aplikuje CLAHE filter**
 - **Vypočíta masku orezania podľa hĺbky**
 - **Vypočíta mračno bodov pre aktuálne spracovávaný snímok**
 - **Deteguje príznaky**
 - **Extrahuje príznaky**

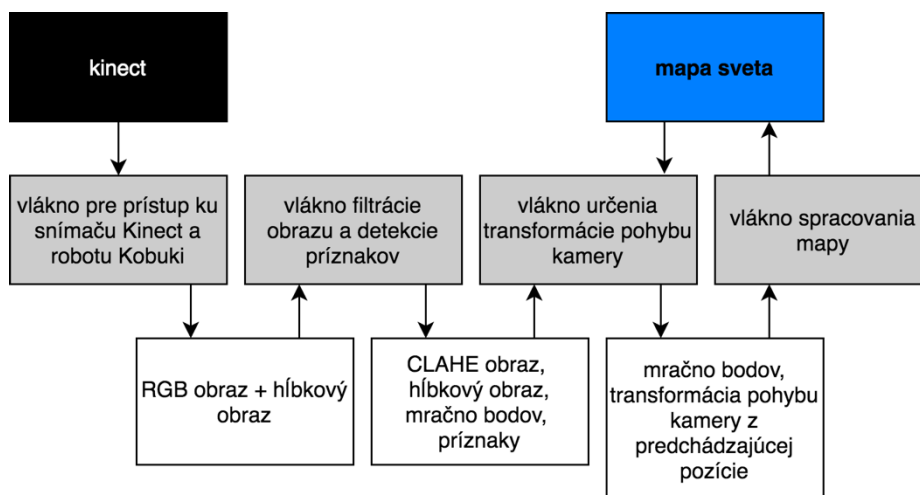
Vypočítané dáta o snímku vlákno uloží spolu so vstupnými dátami do zdieľanej premennej pre vlákno 3 a spí, pokiaľ sa v premennej zdieľanej s vláknom 1 neobjaví nový, nespracovaný snímok. Cieľom bolo implementovať všetky potrebné operácie v tomto vlákne pod 30 ms, keďže snímkovacia frekvencia snímača Kinect je ~30 Hz a snímky teda prichádzajú každých 30 ms.

3. **Vlákno určenia transformácie pohybu kamery** – vlákno porovná deskriptory z predchádzajúcej snímky so snímkom v zdieľanej premennej s vláknom 2, ak už ho nespracovalo v predchádzajúcej iterácii. Deskriptory sú následne filtrované podľa vzdialenosti z porovnávača príznakov. Potom sa skonštruujú mračná bodov z úspešne porovnaných príznakov a tie spracuje metóda RANSAC. Po robustnom určení transformácie sa transformácia uloží do zdieľaného prostriedku s vláknom

4 a vlákno spí, pokiaľ sa na jeho vstupe neobjaví nový snímok. Ak určenie transformácie zlyhá (napríklad v aktuálnom snímku nie je k dispozícii aspoň 5 príznakov a teda nie je možné určiť transformáciu pohybu kamery), vlákno na svoj výstup nastaví jednotkovú maticu 4x4, predstavujúcu nulový pohyb.

4. **Vlákno spracovania mapy** – v tomto vlákne sa aktuálny posun kamery integruje s predchádzajúcimi transformáciami pre získanie globálnej polohy. Mračno bodov z aktuálneho snímku je následne transformované pomocou tejto polohy do globálnych súradníc a zlúčené s mapou prostredia.

Implementácia takéhoto systému predpokladá použitie korektného synchronizačného mechanizmu prístupu ku zdieľaným prostriedkom. V teórii softvérového inžinierstva je možné implementovanú schému klasifikovať ako konkurentnú (concurrent) bežiacu paralelne [16]. Viaceré odlišné úlohy navrhutej schémy totiž prebiehajú konkuretné (súčasne) a zároveň na viacerých jadrách procesora, disponujú zdieľanými prostriedkami a organizovane ich využívajú. V ideálnom prípade by malo byť možné pri takomto modeli umiestniť do každého vlákna algoritmus trvajúci maximálne 30 ms a aplikácia by mala byť schopná spracovávať snímky v reálnom čase. V praxi však algoritmy nie je možné zrýchliť lineárne pridávaním výpočtových CPU jadier, keďže obsahujú aj časti, ktoré musia bežať sekvenčne za sebou. V tomto prípade sú takýmito časťami najmä časti prístupu ku zdieľaným prostriedkom. Každý spracovávaný snímok je totiž nutné spracovať štyrmi vláknami v poradí presne od prvého po posledné a vlákna teda k tejto úlohe pristupujú sekvenčne. Zároveň, pri spracovaní na bežnom PC je možné, že operačný systém sa rozhodne niektorú z úloh pozastaviť na väčší ako požadovaný čas, keďže bežné implementácie paralelizmu na CPU negarantujú presný dostupný procesorový čas. Schéma distribúcie výpočtu vizuálnej odometrie na viaceré vlákna je zobrazená na Obrázok 5.



Obrázok 5 - Schéma distribúcie výpočtu vizuálnej odometrie vo vytvorenom riešení medzi viacerými výpočtovými vláknami. Šedé bloky predstavujú vlákna, biele bloky predstavujú zdieľané prostriedky.

Experimentálne overenie bolo realizované pomocou spúšťania algoritmu s inou konfiguráciou detektora príznakov ORB tak, aby vždy vykonal detekciu za rozdielny čas. Tento detektor je pre tento experiment vhodný najmä z dôvodu, že umožňuje definíciu požadovaného množstva detegovaných príznakov. V experimente bol zvyšovaný počet požadovaných vzoriek aj parametre ovplyvňujúce úspešnosť detekcie pri zmene mierky, ktoré sa u množstva detektorov ukázali ako jednoduchý spôsob zrýchlenia

behu detekcie za cenu straty istej miery robustnosti. Pre ďalšie predĺženie spracovania bol využitý aj filter CLAHE. **V praxi bolo možné využívať 2 jadrá procesora s dobou spracovania max 25 ms pre jednu iteráciu a zvyšné 2 jadrá s dobou spracovania max 10 ms a dosahovať spracovanie snímok frekvenciou 30 Hz.** Ak sa algoritmus dostane do situácie, kde nestíha snímky spracovávať s frekvenciou 30 Hz, začne snímky tzv. odhadzovať - nebude ich akumulovať vo vstupnej fronte, ale odhodí ich.

Extrakcia lokálneho modelu

V riešení RGBD-6D-SLAM [4], je poloha aktuálne určovaného snímku určovaná voči podmnožine predchádzajúcich snímok. Následne sú vypočítané transformácie lokálne optimalizované, čo výrazne redukuje drift, teda nárast globálnej chyby určenia pohybu kamery. Zároveň, tento postup výrazne zvyšuje lokálnu stabilitu určenia polohy. Ak kamera sleduje segment scény bez pohybu, chyba rastie veľmi pomaly. Porovnávanie príznakov a určovanie transformácie pohybu kamery je však výpočtovo náročná operácia, ktorej výpočtové nároky rastú s počtom spracovávaných snímok, a na malom mobilnom robote nemusí byť možné ju implementovať v reálnom čase, ak aktuálnu snímku porovnáваме viacnásobne.

Ak je k dispozícii mapa, z ktorej je možné spätne získať príznaky a využiť ich pre ďalšie určenia polohy, je výhodnou stratégiou vyberať príznaky, ktoré majú dobrú opakovateľnosť v rámci aktuálne spracovávaného prostredia. Preto je výhodné evidovať, koľkokrát bol jednotlivý príznak detegovaný a úspešne porovnaný v minulosti.

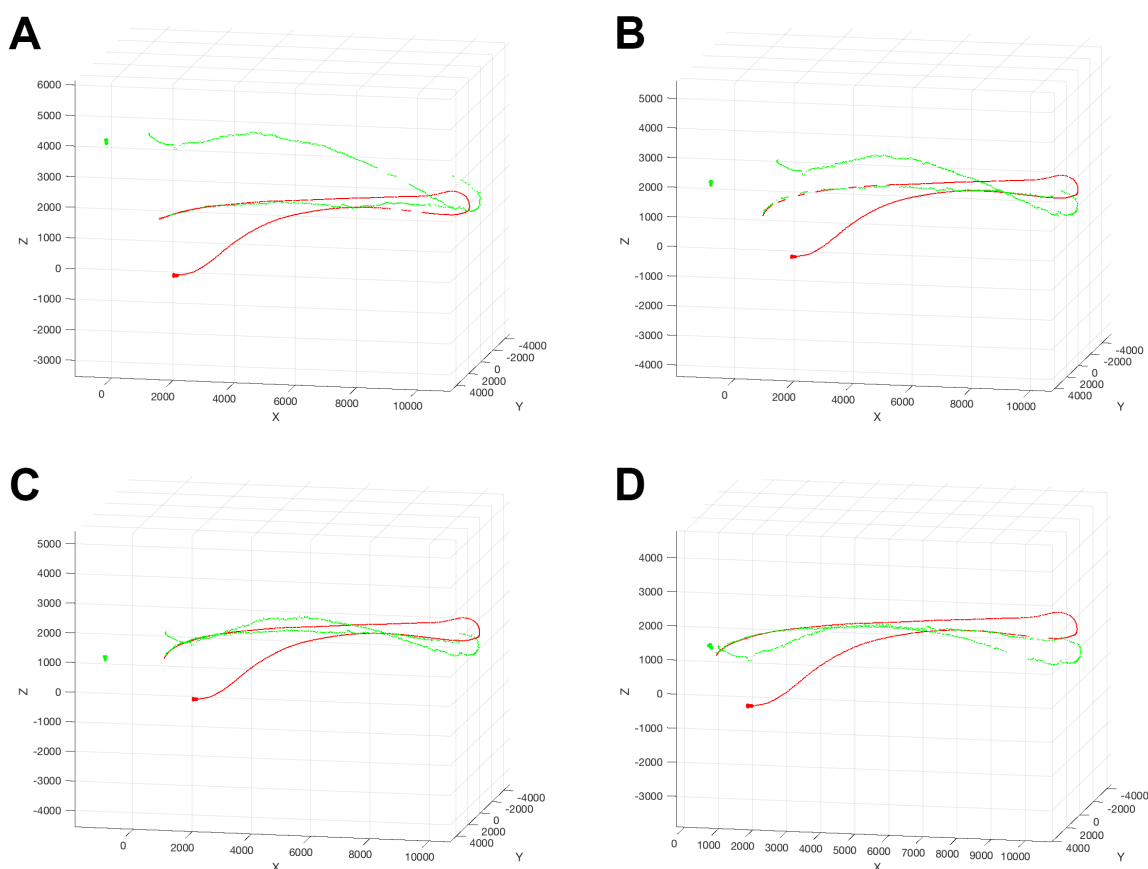
V riešení publikovanom v knihe Hacking the Kinect [17] sa preto pre určenie aktuálnej polohy používa miesto množiny príznakov z predchádzajúceho snímku množina príznakov a ich globálnych 3D súradníc, ktoré boli aktualizované počas posledných piatich iterácií algoritmu. Algoritmus tak vytvára lokálnu mapu príznakov, vďaka ktorej zvyšuje robustnosť zvyšovaním počtu referenčných príznakov oproti samotnému predchádzajúcemu snímku. Týmto postupom bol vzorom aj pri návrhu nášho riešenia.

V navrhnutom riešení pre určenie referenčnej množiny príznakov, ktorá bude porovnávaná s aktuálne spracovávanou snímkom, sa použije **lokálny model mapy**. Lokálny, pretože je generovaný z globálnej mapy zo sledu n snímok, ktoré boli nasnímané bezprostredne pred aktuálnym snímkom. Tento proces nazývame **extrakciou lokálneho modelu mapy**. Týmto spôsobom je možné kompenzovať situácie, v ktorých je aktuálne spracúvaný snímok poškodený alebo neobsahuje dostatočné množstvo príznakov, ale predchádzajúce pozície sú pre určenie pohybu kamery vhodné. Nevýhodou tohoto prístupu je, že jednoduchým akumulovaním všetkých spracovaných príznakov z predchádzajúcich snímok sa zvyšuje počet vzoriek, ktoré vstupujú do porovnávaní a metódy RANSAC, a tým sa predlžuje výpočtový čas. Ďalšou nevýhodou je, že sa zanedbáva fakt, že príznaky z viacerých snímok boli detegované viacnásobne. To znamená, že v lokálnom modeli mapy sa môže každý najlepší príznak nachádzať viackrát, len s inou priestorovou polohou. Preto je v našom riešení priestorová poloha viacnásobne detegovaných príznakov nahradená strednou hodnotou ich polohy, čo je realizované pomocou určenia každej súradnice ako strednej hodnoty polohy príznaku v danej osi. **Tento postup výrazne redukuje počet spracúvaných príznakov, v priemere o 45%, čo výrazne znižuje čas potrebný pre porovnávanie príznakov aktuálne spracovávaného snímku a referenčnej lokálnej mapy.**

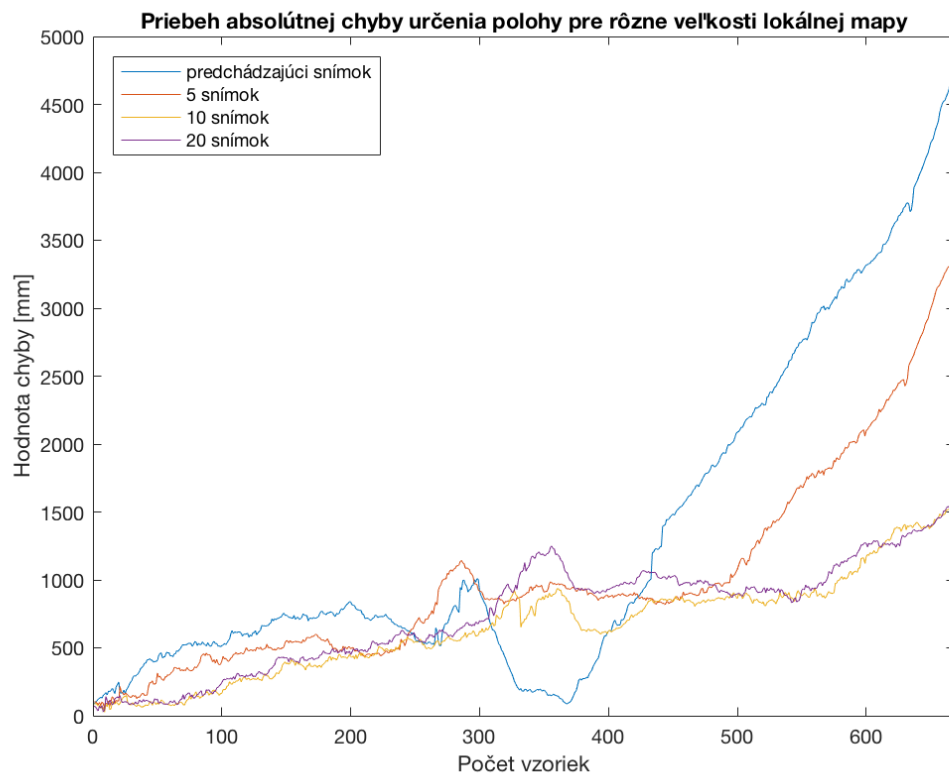
Experimentálne overenie prinieslo predpokladané výsledky. Pomocou použitia viacerých predchádzajúcich snímok pre výpočet správnej transformácie pohybu kamery dochádza k úspešnému potlačeniu driftu a teda zníženiu tempa rastu absolútnej chyby polohy. Navyše, algoritmus je schopný sa lepšie vysporiadať so skenovanými oblasťami, kde je krátkodobo snímaná nevhodná scéna, čo zvyšuje

celkovú konzistentnosť mapy a limituje výskyt extrémnych hodnôt chyby. Závislosť absolútnej chyby určenia polohy a orientácie od veľkosti lokálneho mapového modelu zobrazujú obrázky

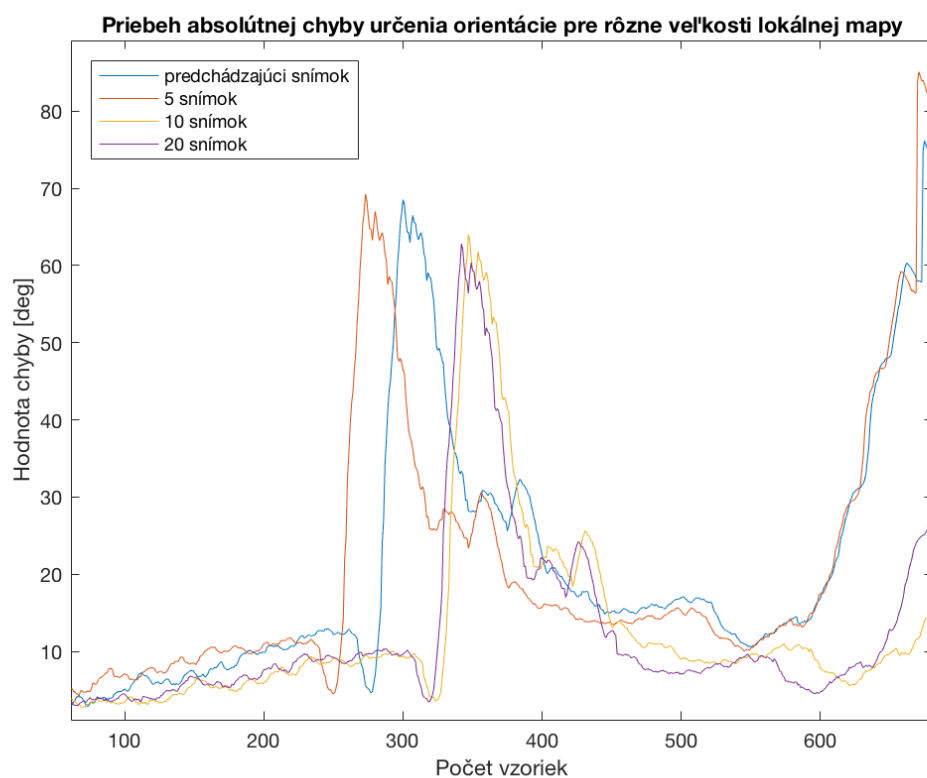
Obrázok 7 a Obrázok 8. Obrázok 6 zobrazuje vplyv veľkosti mapového modelu na výslednú zrekonštruovanú trajektóriu v jednom z našich experimentov. Ako vidíme, použitie už 5 snímkov výrazne znižuje tempo rastu globálnej chyby. Použitie 10 snímkov redukuje tempo rastu chyby ešte výraznejšie, ale zvýšenie počtu na 20 snímkov neprinieslo už žiadne zlepšenie. Chyby orientácie tento prístup ovplyvňuje v menšej miere ako chyby polohy, z dôvodu nízkej miery vzájomného prekrytia lokálneho mapového modelu a aktuálne spracovávaného snímku. V rámci prezentovaného riešenia je prípustné, pre dodržanie výstupnej frekvencie 30 Hz, použiť sekvenciu maximálne posledných 35 snímkov, z ktorých sa vyberie každá druhá, teda 17 snímkov.



Obrázok 6 - Porovnanie zrekonštruovaných trajektórií pohybu kamery pri rôznych veľkostiach lokálneho modelu mapy. Červená trajektória reprezentuje gyro odometriu podvozku Kobuki, zelená trajektória predstavuje trajektóriu určenú pomocou vizuálnej odometrie. A.) Iba predchádzajúci snímok, B.) predchádzajúcich 5 snímkov, C.) predchádzajúcich 10 snímkov, D.) predchádzajúcich 20 snímkov.

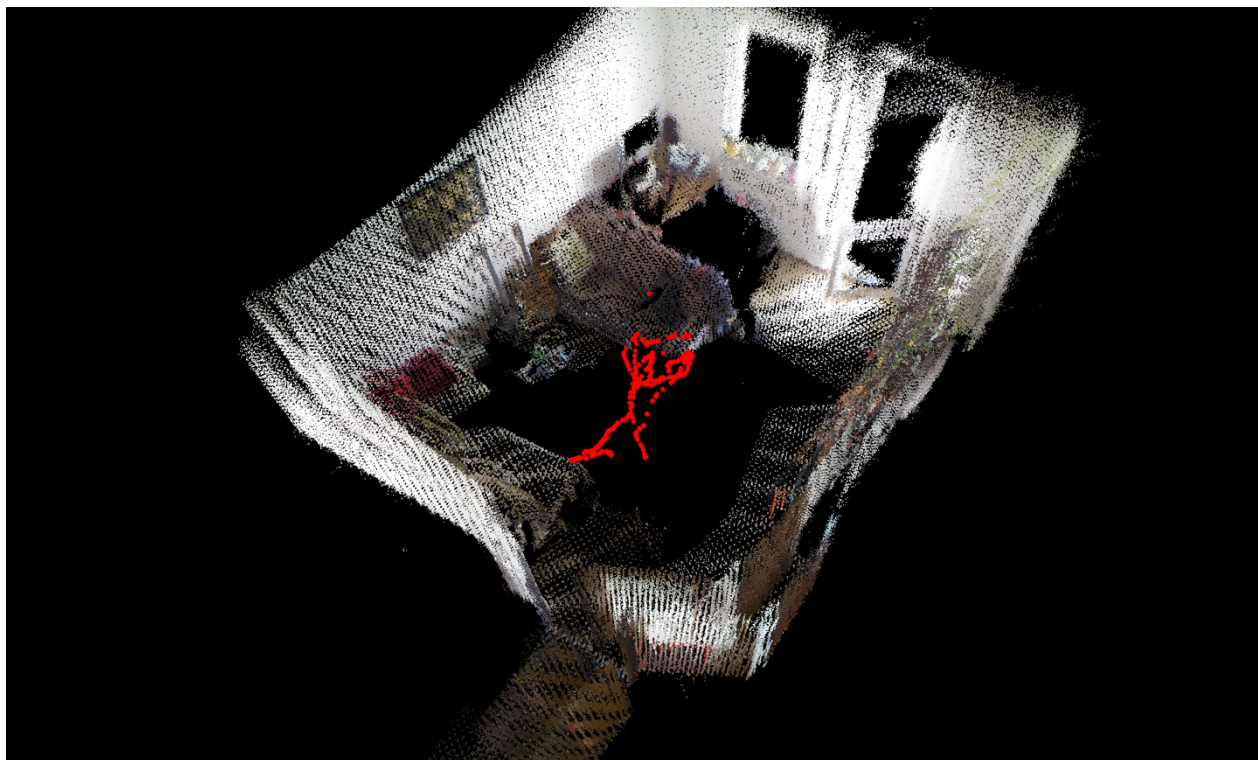


Obrázok 7 - Priebeh absolútnej chyby určenia polohy v závislosti od veľkosti lokálnej mapy.

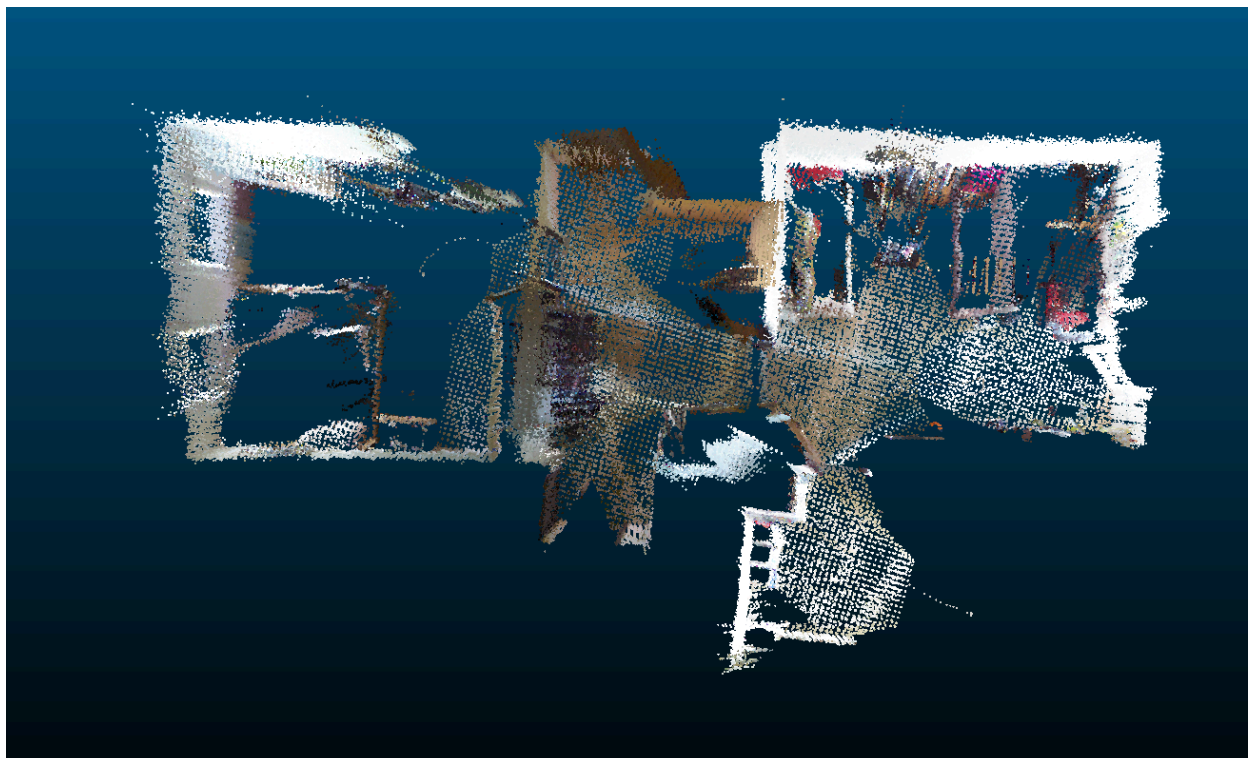


Obrázok 8 - Priebeh absolútnej chyby orientácie v závislosti od veľkosti lokálneho modelu mapy

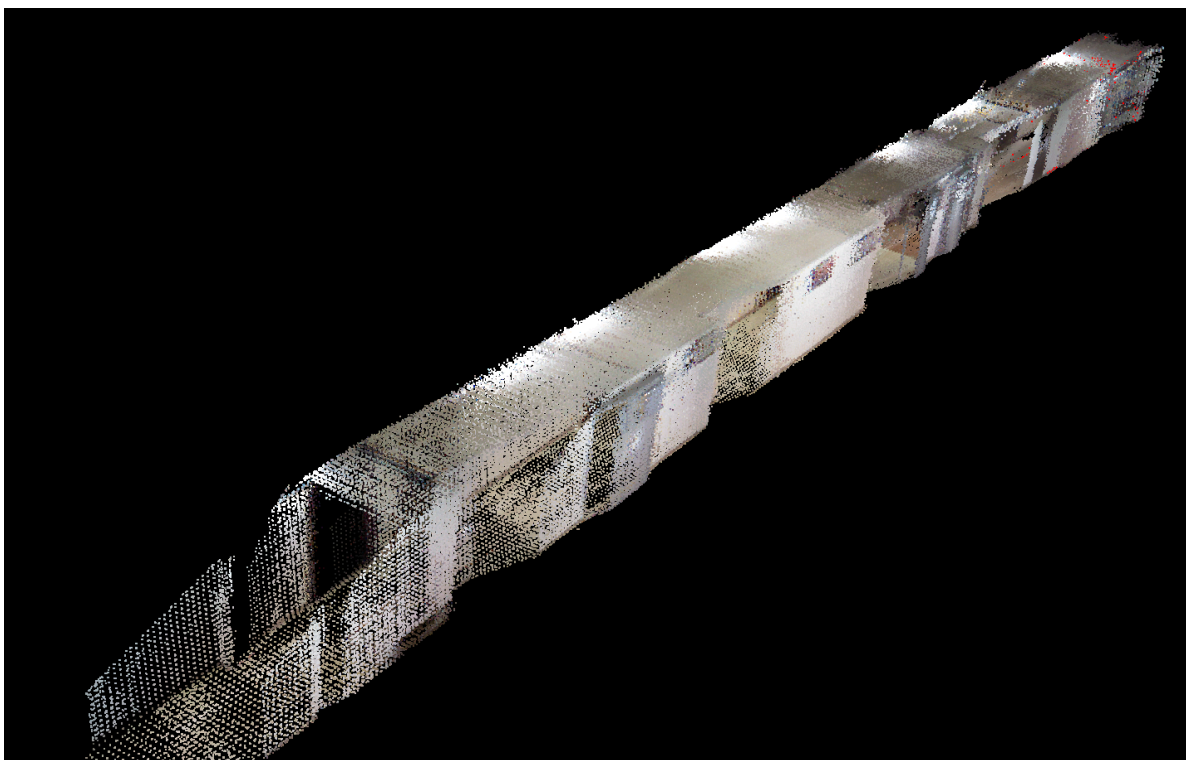
Mapy prostredí vytvorené pomocou navrhovaného riešenia



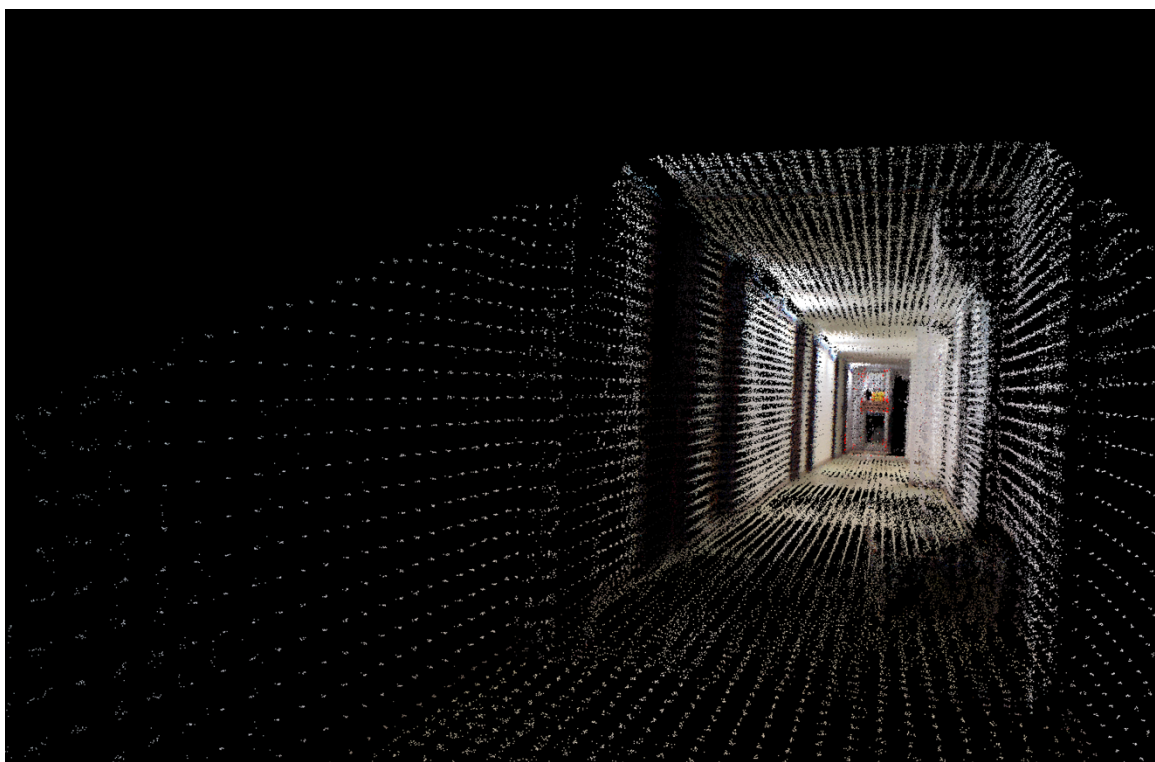
Obrázok 9 - Mapa obývacej izby vytvorená pomocou skenovania z voľnej ruky. Červenou sú vyznačené jednotlivé pozície kamery a v trajektórii pohybu.



Obrázok 10 - Mapa bytu vytvorená pomocou nášho riešenia jedným rýchlym prechodom cez byt panelákového domu



Obrázok 11 - Mapa chodby pri Národnom centre robotiky (NCR), pohľad zhora



Obrázok 12 - Mapa pri NCR, pohľad zvnútra

Záver

Dizertačná práca čitateľa uviedla do problematiky robotického mapovania pomocou relatívne nového typu snímača – hĺbkovej kamery. Práca poskytuje komplexný taxonomický prehľad metód robotického mapovania a lokalizácie a ich pridružených problémov. Boli predstavené priestorové mapové reprezentácie a zhodnotená ich vhodnosť pre riešenie štandardných úloh robotiky. V práci boli predstavené a vzájomne porovnané hlavné postupy, ktoré môžu byť použité riešenie problému vizuálnej odometrie pomocou hĺbkovej kamery. Dôraz bol kladený na príznakové metódy, ktoré boli použité aj v prezentovanom riešení.

Vybrané dostupné riešenia boli analyzované a boli identifikované problémy, ktorými tieto implementácie trpia. Bola skonštruovaná robotická platforma po vzore robota TurtleBot 2, ktorá bola použitá pre záznam referenčných trajektórií. Následne bola navrhnutá metodika vyhodnotenia kvality rekonštrukcie pohybu kamery (algoritmu vizuálnej odometrie) voči takto získanej referencii. Táto metodika bola použitá pre vyhodnotenie vlastností navrhovaného riešenia.

Navrhnuté riešenie umožňuje beh algoritmu vizuálnej odometrie v reálnom čase, teda s výstupnou frekvenciou rovnou frekvencii snímača, čiže ~30 Hz. Toto bolo umožnené pomocou implementácie paralelizmu na výpočtovej jednotke CPU a viacerým mechanizmom, ktoré redukovávajú množstvo spracovávaných dát v jednotlivých krokoch algoritmu na minimálnu relevantnú vzorku. Pre dosiahnutie maximálnej rýchlosti spracovania bol na základe experimentov vybraný detektor a deskriptor ORB. Pre zvýšenie robustnosti voči zmenám osvetlenia a nedostatočnej štruktúre scény bol využitý filter CLAHE a jeho dopad na ovplyvnené časti riešenia bol dôkladne analyzovaný. V rámci riešenia bola navrhnutá modifikovaná schéma metódy RANSAC, pre ktorú boli popísané postupy stanovenia všetkých parametrov na optimálne hodnoty z hľadiska rýchlosti spracovania. Navrhnutá mapová reprezentácia sa skladá z viacerých individuálnych máp, ktoré umožňujú implementáciu ďalších metód pre vytvorenie globálne konzistentného SLAM riešenia v budúcnosti. Pre zníženie tempa rastu globálnej chyby bola implementovaná metóda extrakcie lokálnej mapy, ktorá je použitá ako referencia pre určenie transformácie voči aktuálne spracúvanému snímku. Riešenie je tak schopné zotaviť sa po extrémnych náhodných chybách určenia polohy.

Riešenie funguje vo všetkých šiestich stupňoch voľnosti (6 DoF), čo umožňuje jeho nasadenie ako na pozemných, tak aj na lietajúcich robotoch. Riešenie je implementované v jazyku C++ s využitím multiplatformných knižníc, čo umožňuje jeho beh pod operačnými systémami Windows aj platformách na architektúre Unix. Vytvorený systém je tak možné použiť ako riešenie lokalizácie a mapovania, alebo doplnok existujúcich metód na širokom spektre mobilných robotických platforiem. Uvádzané postupy a teoretický prehľad je možné využiť aj pri riešení problémov 3D rekonštrukcie a 3D skenovania. Zdrojové kódy implementácie sú voľne dostupné v [18]. Vytvorené riešenie bolo implementované a jeho vlastnosti overené na reálnej robotickej platforme.

Prínosy dizertačnej práce

1. Návrh, implementácia a overenie algoritmu vizuálnej odometrie a mapovania s nízkou výpočtovou náročnosťou vhodného pre malý mobilný robot.
2. Popis a experimentálne vyhodnotenie schémy paralelizácie výpočtu príznakov vizuálnej odometrie pomocou viacerých výpočtových jadier CPU.
3. Návrh použitia a vyhodnotenie dopadu filtra CLAHE pre kompenzáciu nevhodného osvetlenia scény. S použitím tohoto filtra v kontexte vizuálnej odometrie sa autor doposiaľ nestretol.
4. Návrh a implementácia mechanizmu tvorby a extrakcie lokálneho modelu mapy za účelom zníženia tempa rastu globálnej chyby.
5. Overenie efektívnosti viacerých detektorov príznakov na reálnych datasetoch. Návrh filtrácie porovnaných príznakov za účelom zvýšenia kvality riešenia pre detektory a deskriptory ORB a AKAZE.
6. Návrh metodiky vyhodnotenia kvality riešenia vizuálnej odometrie voči referenčnej trajektórii.
7. Široký teoretický prehľad komplexne spracúvajúci problematiku vizuálnej odometrie a jej aplikácii pre robotické mapovanie. Podľa vedomostí autora sa jedná o jednu z prvých publikácií tohoto typu v SR.

Literatúra

- [1] W. Garage, "Turtlebot," 2011. [Online]. Available: <http://turtlebot.com>. [Accessed 23 Jún 2018].
- [2] B. Peter, "SPRACOVANIE DÁT ZO SNÍMAČA KINECT ZA ÚČELOM TVORBY 3D MAPY PROSTREDIA," Bratislava, 2013.
- [3] D. F. T. M. H. P. M. Beňo Peter, "3d map reconstruction with sensor kinect: Searching for solution applicable to small mobile robots," in *In Robotics in Alpe-Adria-Danube Region (RAAD), 23rd International Conference on IEEE.*, Smolenice, Slovakia, 2014.
- [4] N. E. a. F. E. a. J. H. a. J. S. a. W. Burgard, "Real-time 3D visual SLAM with a hand-held RGB-D camera," in *Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum*, Vasteras, Sweden, 2011.
- [5] D. K. O. H. D. M. R. N. P. K. J. S. S. H. D. F. A. D. A. F. Shahram Izadi, "KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera".
- [6] T. T. L. V. G. H. B. Andreas Ess, "Speeded-Up Robust Features (SURF)," *Comput. Vis. Image Underst.*, no. 110, pp. 346-359, June 2008.
- [7] P. J. & M. N. D. Besl, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 12(2), p. 239-256, 1992.
- [8] Y. & M. G. Chen, "Object modelling by registration of multiple range images.," *Image and Vision Computing*, vol. 3, no. 10, p. 145-155, 1992.
- [9] K. R. C. F. U. H. C. Grisetti G., "Hierarchical optimization on manifolds for online 3d and 3d mapping."
- [10] B. CURLESS and M. LEVOY, "A volumetric method for building complex models from range images.," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques.*, 1996.
- [11] B. Peter, P. Vladimír, D. Frantiek and D. Martin, "Using Octree Maps and RGBD Cameras to Perform Mapping and A* Navigation," in *2016 International Conference on Intelligent Networking and Collaborative Systems (INCoS)*, Ostrava, Czech Republic, 2016.
- [12] D. J. R. W. L. a. J. W. W. Ketcham, "Image enhancement techniques for cockpit displays.," CULVER CITY CA, 1974.
- [13] M. a. C. P. Petrou, *Image processing: the fundamentals*, John Wiley & Sons, 2010.
- [14] K. Zuiderveld, "Contrast limited adaptive histogram equalization.," in *Graphics gems*, 1994, pp. 474-485.
- [15] I. C. Consortium, *Image technology colour management - Architecture, profile format, and data structure, Specification ICC.1:2004-10*, 2004.
- [16] G. R. Andrews, *Concurrent programming: principles and practice*, Benjamin/Cummings Publishing Company, 1991.
- [17] J. B. N. E. F. D. H. C. & P. M. Kramer, *Hacking the kinect*, vol. 268, New York, NY: Apress, 2012.
- [18] I. B. Peter, "Kinect visual odometry mapping," Slovak University of Technology, 30 6 2018. [Online]. Available: <https://github.com/najlepsiwebdesigner/kinect-vo-mapping>.
- [19] P. Beňo, "Kinect visual odometry mapping," Slovak University of Technology, 30 6 2018. [Online].

Available: <https://github.com/najlepsiwebdesigner/kinect-vo-mapping>.

- [20] S. D. Roth, "Ray casting for modeling solids.," *Computer graphics and image processing*, vol. 18, no. 2, pp. 109-144, 1982.
- [21] M. Pirovano, "KinFu - an open source implementation of Kinect Fusion + case study: implementing a #D scanner with PCL," 2012.
- [22] M. V. Henry Roth, "Moving Volume KinectFusion," *BMVC*, pp. 1-11, 2012.

Publikované práce autora

ADC Vedecké práce v zahraničných karentovaných časopisoch

- ADC01 KONIAR, Dušan - HARGAŠ, Libor - LONCOVÁ, Zuzana - DUCHOŇ, František - BEŇO, Peter. Machine vision application in animal trajectory tracking. In *Computer Methods and Programs in Biomedicine*. Vol. 127, (2016), s. 258-272. ISSN 0169-2607. V databáze: CC: 000372521500022.
- ADC02 KONIAR, Dušan - HARGAŠ, Libor - LONCOVÁ, Zuzana - SIMONOVÁ, Anna - DUCHOŇ, František - BEŇO, Peter. Visual system-based object tracking using image segmentation for biomedical applications. In *Electrical Engineering*. Vol. 99, Iss. 4 (2017), s. 1349-1366. ISSN 0948-7921. V databáze: CC: 000415972700022 ; SCOPUS: 2-s2.0-85026909750.

ADE Vedecké práce v ostatných zahraničných časopisoch

- ADE01 DUCHOŇ, František - BUČKA, Peter - SZABOVÁ, Martina - DEKAN, Martin - BEŇO, Peter - TÖLGYESSY, Michal. Image processing of motion for security applications. In *European Scientific Journal*. Vol. 13, No. 27 (2017), s. 44-58. ISSN 1857-7881.
- ADE02 KONIAR, Dušan - HARGAŠ, Libor - LONCOVÁ, Zuzana - DUCHOŇ, František - BEŇO, Peter. Usage of video analysis for tracking of laboratory animals. In *Acta Technica Corviniensis - Bulletin of Engineering [elektronický zdroj]*. Tom. 10, fasc. 2 (2017), online, s. 147-152. ISSN 2067-3809.

ADF Vedecké práce v ostatných domácich časopisoch

- ADF01 DUCHOŇ, František - BEŇO, Peter - BABINEC, Andrej - DEKAN, Martin - DÚBRAVSKÝ, Jozef - JURNICKÝ, Tomáš - MUSIČ, Josip. Use of children's games in robotics. In *Acta Mechatronica [elektronický zdroj]*. Vol. 1, Iss. 4 (2016), online s. 7-11. ISSN 2453-7306.

AFC Publikované príspevky na zahraničných vedeckých konferenciách

- AFC01 BEŇO, Peter - PAVELKA, Vladimír - DUCHOŇ, František - DEKAN, Martin. Using octree maps and RGBD cameras to perform mapping and A* navigation. In *IEEE INCoS 2016 : International conference on intelligent networking and collaborative systems. Ostrava, Czech Republic. 7-9 September 2016*. Danvers : IEEE, 2016, S. 66-72. ISBN 978-1-5090-4123-7. V databáze: IEEE ; WOS: 000386596100013.

AFD Publikované príspevky na domácich vedeckých konferenciách

- AFD01 BEŇO, Peter - DUCHOŇ, František - TÖLGYESSY, Michal - HUBINSKÝ, Peter - KAJAN, Martin. 3D Map Reconstruction with Sensor Kinect. In *RAAD 2014 [elektronický zdroj] : 23rd International Conference on Robotics in Alpe-Adria-Danube Region. Conference Proceedings. September 3-5,*

2014, Smolenice Castle, Slovakia. 1. vyd. Bratislava : Publishing House of Slovak University of Technology, 2014, CD-ROM, [6] p. ISBN 978-80-227-4219-1.

- AFD02 DUCHOŇ, František - BABINEC, Andrej - KAJAN, Martin - BEŇO, Peter - FLOREK, Martin - FICO, Tomáš - JURIŠICA, Ladislav. Path planning with modified A star algorithm for a mobile robot. In *Procedia Engineering [elektronický zdroj] : The 6th International Conference on Modelling of Mechanical and Mechatronic Systems MMaMS 2014, 25-27 November 2014, Vysoké Tatry, Slovakia*. Vol. 96, (2014), p. 59-69. ISSN 1877-7058.
- AFD03 DUCHOŇ, František - VONDRÁČEK, Martin - DEKAN, Martin - BABINEC, Andrej - SPIELMANN, Róbert - SZABOVÁ, Martina - MIKULOVÁ, Zuzana - BEŇO, Peter - DÚBRAVSKÝ, Jozef. Homogenous multi-robot system for mapping of unknown environment. In *SAMI 2016 : IEEE 14th international conference Symposium on Applied Machine, Intelligence and Informatics. Herľany, Slovakia, January 21-23, 2016*. 1. vyd. [S. l.] : IEEE, 2016, S. 17-22. ISBN 978-1-4673-8739-2. V databáze: IEEE ; WOS: 000381795100002 ; SCOPUS: 2-s2.0-84964607617.
- AFD04 KAJAN, Martin - ŠOVČÍK, Ján - DUCHOŇ, František - MRAFKO, Leo - FLOREK, Martin - BEŇO, Peter. Sensoric subsystem of Automated Guided Vehicle. In *RAAD 2014 [elektronický zdroj] : 23rd International Conference on Robotics in Alpe-Adria-Danube Region. Conference Proceedings. September 3-5, 2014, Smolenice Castle, Slovakia*. 1. vyd. Bratislava : Publishing House of Slovak University of Technology, 2014, CD-ROM, [6] p. ISBN 978-80-227-4219-1.
- AFD05 KONIAR, Dušan - HARGAŠ, Libor - LONCOVÁ, Zuzana - DUCHOŇ, František - BEŇO, Peter. Laboratory animals tracking in videosequences. In *ELEKTRO 2016 : 11th International conference. Štrbské Pleso, Slovak Republic. May 16-18, 2016*. Žilina : University of Žilina, 2016, S. 537-542. ISBN 978-1-4673-8698-2. V databáze: IEEE ; SCOPUS: 2-s2.0-84981343555 ; WOS: 000386959800108.