

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

**INTERAKTÍVNE SPRACOVANIE ÚDAJOV V PROSTREDÍ
ZMIEŠANEJ REALITY**

Diplomová práca

2016

Jakub Čech, Bc.

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

**INTERAKTÍVNE SPRACOVANIE ÚDAJOV V PROSTREDÍ
ZMIEŠANEJ REALITY**

Diplomová práca

Študijný program: Informatika
Študijný odbor: 9.2.1 Informatika
Školiace pracovisko: Katedra počítačov a informatiky
Školiteľ: Doc. Ing. Branislav Sobota, PhD.
Konzultant:

2016 Košice

Jakub Čech, Bc.

Abstrakt v SJ

Cieľom tejto práce je poukávanie na možnosť, vhodnosť a správnosť riešenia problému určenia priestorovej pozície užívateľa s dátovou prilbou pre virtuálnu realitu (HMD) bez referenčného bodu pomocou dostupných optických SLAM metód a vypracovanie funkčného, správne nastaveného a vyladeného systému, na ktorom je koncept možné priamo reálne otestovať. Sústreďenie je kladené na dostupnosť poukázaných riešení, funkčnosť celku a dokázanie správnosti a funkčnosti takéhoto konceptu. Ako hlavný problém je riešený problém určenia pozície v priestore a mapovanie priestoru, teda problém SLAM. Z množstva dostupných riešení problému SLAM je vybrané práve riešenie založené na spracovaní obrazu z kamery. Vypracované riešenie používa Oculus Rift DK2 ako HMD a kameru pripevnenú na ňom a teda je k dispozícii aj snímač IMU. Dáta sú spracované na počítači s operačným systémom Linux, využívajúc robotický operačný systém ROS na ktorom sú spustené algoritmy PTAM a LSD-SLAM. Okrem navrhutej a implementovanej fúzie údajov z Oculusu a SLAM algoritmov sú spracovávané aj dáta z inerčných IMU snímačov Oculusu pomocou Matlabu dvojitou integráciou akcelerácie na polohu. Vstupom SLAM algoritmov je neupravený tok obrazu z kamery a preto sú otestované aj rôzne možnosti predfiltrácie obrazu algoritmi ako Sobelové derivácie či Gaussovský filter.

Kľúčové slova v SJ

SLAM, PTAM, ROS, Oculus, HMD, pozícia v priestore, 3D pozícia, IMU, Sobelové derivácie, Gaussovský filter, LSD-SLAM

Abstrakt v AJ

The aim of this thesis is to point at option, suitability and correctness of solving the problem of positional tracking of head-mounted display without reference point using methods and algorithms of visual SLAM. Another aim is to create functional, well setup and tuned system providing such functionality, where the whole concept can be tested and promoted. The thesis concentrates on availability of proposed solutions, functionality of the system and proving of such concept. The main problem being solved is a problem of simultaneous mapping and localisation, a problem SLAM. Among many fields of SLAM, visual SLAM is chosen, where the main sensor is being a camera. Proposed system uses Oculus Rift DK2 as a head-mounted device, thus providing

availability of IMU unit, and a camera attached to the Oculus. Data flow is being processed on computer running Linux operating system and ROS operating system using two visual SLAM algorithms, PTAM and LSD-SLAM. For the purpose of sensor fusion, IMU data are being processed by simple double integration in Matlab. An input of visual SLAM algorithms is video stream and therefor, pre processing filtration is implemented providing many filters to be used such as Sobel derivatives and Gaussian filter.

Klíčové slova v AJ

SLAM, PTAM, ROS, Oculus, HMD, positional tracking, 3D position, IMU, Sobel derivatives, Gaussian filter, LSD-SLAM

Zadanie práce

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
Katedra počítačov a informatiky

ZADANIE DIPLOMOVEJ PRÁCE

Študijný odbor: 9.2.1 Informatika

Študijný program: Informatika

Názov práce:

Spracovanie údajov v prostredí zmiešanej reality

Data processing in mixed reality environment

Študent: **Bc. Jakub Čech**

Školiteľ: **doc. Ing. Branislav Sobota, PhD.**

Školiace pracovisko: **Katedra počítačov a informatiky**

Konzultant práce:

Pracovisko konzultanta:


Pokyny na vypracovanie diplomovej práce:

1. Naštudovať problematiku virtuálnej reality, zmiešanej reality a spracovania obrazu.
2. Navrhnuť systém spracovania obrazu pre technológiu zmiešanej reality s orientáciou na monokulárnu rekonštrukciu snímaného obrazu z pohľadu pozorovateľa.
3. Implementovať systém navrhnutý v bode 2 vrátane prípadnej možnosti zmeny algoritmu spracovania scény resp. filtrácie snímaného obrazu.
4. Vykonať experimentálne práce s implementovaným systémom vrátane spracovania alebo dotvorenia potrebných modelov a zhodnotiť dosiahnuté výsledky.
5. Vypracovať dokumentáciu podľa pokynov vedúceho práce

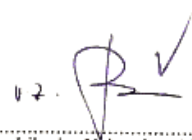
Jazyk, v ktorom sa práca vypracuje: slovenský

Termín pre odovzdanie práce: 29.04.2016

Dátum zadania diplomovej práce: 31.10.2015


.....
doc. Ing. Jaroslav Porubán, PhD.
vedúci garantujúceho pracoviska



17. 
.....
prof. Ing. Liberios Vokorokos, PhD.
dekan fakulty

Čestné vyhlásenie

Vyhlasujem, že som celú diplomovú prácu vypracoval samostatne s použitím uvedenej odbornej literatúry.

Košice, 24. apríla 2016

.....

PodĎakovanie

Chcel by som sa veľmi pekne poďakovať vedúcemu práce, Doc. Ing. Branislavovi Sobotovi, PhD. za predané vedomosti, skvelé podmienky pri vypracovávaní práce, vloženie dôvery a voľnosť pri vypracovávaní toho, čo ma zaujíma. Takisto by som sa chcel poďakovať Michalovi Titkovi za dlhodobé zapožičanie Oculus Rift DK2.

Obsah

Zoznam obrázkov	11
Zoznam tabuliek	13
Zoznam symbolov a skratiek	14
Úvod	15
Formulácia úlohy a cieľ práce.....	18
Celkový pohľad na systém.....	19
1. Teoretický rozbor SLAM	21
1.1. Rozdelenie podľa snímačov.....	22
1.1.1. Exteroceptívne	22
1.1.1. Proprioeceptívne.....	23
1.2. Známe všeobecné metódy riešenia SLAM – Pravdepodobnostné filtre	23
1.2.1. Kalman Filter	24
1.2.2. Časticový filter	24
1.2.3. Metódy založené na algoritme Expectation maximization	25
1.3. Reprezentácia mapy.....	25
1.3.1. Metrické mapy	25
1.3.2. Topologické mapy	25
1.4. Asociácia dát.....	26
2. Analýza SLAM pre použitie v práci a opis vybraných algoritmov.....	27
2.1. Výber zariadení a metód	27
2.2. Optický SLAM	27
2.2.1. Kamera ako hlavný snímač.....	28
2.2.2. Techniky využívajúce Structure from Motion	29
2.2.3. Body záujmu	30
2.2.4. Metódy založené na intenzite pixelov.....	31
2.2.5. Problémy párovania obrazu a asociácie dát.....	31
2.3. IMU a inerčný SLAM.....	32

2.3.1.	Snímač IMU	33
2.3.2.	Inerčný SLAM.....	33
2.4.	Výber prostredia a systémov.....	34
2.4.1.	Robotický operačný systém ROS.....	34
2.4.2.	Rviz a tf	35
2.5.	Vybrané algoritmy pre optický a inerčný SLAM	36
2.5.1.	PTAM	37
2.5.2.	LSD-SLAM	45
2.5.3.	Algoritmus spracovania údajov z inerčných snímačov.....	53
2.6.	Algoritmy pred-filtrácie	53
2.6.1.	Gaussovský filter	53
2.6.2.	Sobelové derivácie	54
2.6.3.	Laplacov operátor	54
3.	Dodefinovanie, testovanie a výber už konkrétnych zariadení	55
3.1.	Výber dátovej helmy pre virtuálnu realitu a jej vlastnosti	55
3.2.	Výber monokulárnej kamery.....	56
3.3.	Výber stereokulárnej kamery.....	59
4.	Návrh, implementácia a testovanie podčastí systému	60
4.1.	Návrh	60
4.2.	Komunikácia medzi monokulárnou kamerou a ROS.....	62
4.3.	Komunikácia medzi stereokulárnou kamerou a ROS.....	62
4.4.	Implementácia predfiltrácie obrazu z kamery	63
4.4.1.	Interpretácia stereo kanálov v ROS.....	64
4.5.	Výber metód testovania algoritmov a zhotovovanie materiálov	65
4.6.	Testovanie a nastavenie Ethzasl-PTAM.....	67
4.6.1.	Kalibrácia kamery	67
4.6.2.	Testovanie bez filtrácie	67
4.6.3.	Testovanie s filtráciou	72

4.6.4.	Testovanie v rôznych prípadoch.....	74
4.6.5.	Vybrané nastavenia Ethzasl-PTAM a záver testovania	79
4.7.	Testovanie a nastavenie LSD-SLAM.....	79
4.7.1.	Testovanie bez filtrácie	80
4.7.1.	Testovanie s filtráciou	83
4.7.1.	Testovanie v rôznych prípadoch.....	84
4.7.2.	Vybrané nastavenia LSD-SLAM a záver testovania	89
4.8.	Testovanie a nastavenie S-PTAM	89
4.8.1.	Kalibrácia kamery	89
4.8.2.	Zlyhanie používania	90
4.9.	Komunikácia medzi Oculus a ROS	90
4.9.1.	Vykresľovanie	90
4.9.2.	Získavanie surových dát z Oculusu	90
4.10.	Komunikácia medzi ostatnými modulmi	90
4.11.	Testovanie dvojitej integrácie akcelerácie	91
5.	Návrh a implementácia fúzie dát a ladenie celku	92
5.1.	Návrh a implementácia fúzie dát	92
5.2.	Testovacia RVIZ TUKE scéna.....	94
5.3.	Vykresľovanie na X screen a ladenie rýchlosti	95
5.4.	Testovanie celku.....	95
6.	Záver.....	96
	Zoznam použitej literatúry	98
	Prílohy	101

Zoznam obrázkov

Obr. 1 – Fyzicko-hárdverová schéma celého systému.	19
Obr. 2 – Softvérová schéma celého systému.	20
Obr. 3 – Príklad zmapovaného prostredia vďaka pohybu užívateľa prostredím a lokalizácia užívateľa	21
Obr. 4 – Snímač lidar a mapa vytvorená mapovaním pomocou lidar.	22
Obr. 5 – Veľká chyba dead reckoningu alebo odometrie z údajov z IMU. Prevzaté z [3].	23
Obr. 6 – Čip IMU.	23
Obr. 7 - Nájdene body záujmu v obraze.	31
Obr. 8 - Pred uzavretím a po detekcii a uzavretí slučky	32
Obr. 9 – Ukážka prostredia pluginu Rviz v operačnom systéme ROS.	35
Obr. 10 – Bežiaci PTAM algoritmus.	38
Obr. 11 – Schéma mapovacieho vlákna algoritmu PTAM. Prebrané z [14].	43
Obr. 12 – Príklad zmapovaného prostredia pomocou LSD-SLAM aj s pozami a trajektóriou kamery.	46
Obr. 13 – Porovnanie množstva využitých informácií z obrazu. Prevzaté z SD-VO [21].	46
Obr. 14 – Aplikácia Gaussianovho filtra na 2D obraz. Pred a Po.	54
Obr. 15 – Aplikácia Sobelových derivácií na 2D obraz. Pred a Po.	54
Obr. 16 - Obraz z kamery Genius WideCam F100.	57
Obr. 17 – Obraz z kamery Logitech Webcam C930e.	57
Obr. 18 - Obraz z kamery SJCAM SJ4000.	57
Obr. 19 – Spojenie Oculus Rift DK2 a SJCAM SJ400.	59
Obr. 20 – Schéma celého systému.	60
Obr. 21 - Schéma toku dát v okolí podmodulu Rviz.	61
Obr. 22 – Viackroková procedúra konvertovania 3D záznamu na použiteľný formát.	62
Obrázok 23 – Predfiltračný systém.	63
Obr. 24 – Konštrukcia sprostredkujúca simultánne natáčanie pomocou monokulárnej aj stereokulárnej kamery.	65
Obr. 25 – Príklad grafu z testu Test 1_3_PTAM. Pád algoritmu v čase 1160 s následnou obnovou po určitej dobe čakania ktorá nie je nijako znázornená.	66
Obr. 26 – Schéma testovania.	67
Obr. 27 – Náhľad obrázkov z testovacieho videa Test_1_Monocular.mov z testu Test 1.	67
Obr. 28 – Graf správneho pohybu kamery z Test 1_1_PTAM.	69
Obr. 29 – Príklad screenshotu z Test 1_1_PTAM.	69

Obr. 30 – Graf k Test 2_4_PTAM. Graf predstavuje najlepšie dosiahnuté prirodzené správanie Ethzasl-PTAM zvýšením Max Patches na 650.	72
Obr. 31 – Náhľad obrázkov z testovacieho videa Test_4_Monocular.mov z testu Test 4.	74
Obr. 32 – Výsledný najsprávnejší graf odhadovanej pozície kamery pomocou Ethzasl-PTAM. Test 4_2_PTAM.	75
Obr. 33 - Výsledný graf pozície kamery LSD-SLAM v Test 2_11_LSD.	82
Obr. 34 – Výsledný graf pozície kamery LSD-SLAM v Test 3_2_LSD.	84
Obr. 35 – Výsledný najsprávnejší graf odhadovanej pozície kamery pomocou LSD-SLAM. Test 4_2_LSD.	85
Obr. 36 – Vizúálne testovanie inerčného SLAM.	91
Obr. 37 – Problém fúzie dát.	92
Obr. 38 – Štruktúra stromu tf koordinačných rámcov fúzie dát.	93
Obr. 39 – Testovacia TUKE Rviz scéna.	94
Obr. 40 – Náhľad demonštračného videa.	95

Zoznam tabuliek

Tab. 1 – Porovnanie vlastností kamier.....	58
Tab. 2 – Tabuľka k Test 1_PTAM.....	68
Tab. 3 – Tabuľka k Test 2_PTAM.....	70
Tab. 4 - Tabuľka k Test 3_PTAM.....	73
Tab. 5 - Tabuľka k Test 4_PTAM.....	74
Tab. 6 - Tabuľka k Test 5_PTAM.....	76
Tab. 7 – Tabuľka k Test 6_PTAM.....	77
Tab. 8 – Tabuľka k Test 7_PTAM.....	78
Tab. 9 – Tabuľka k Test 2_LSD.....	81
Tab. 10 – Tabuľka k Test 3_LSD.....	83
Tab. 11 - Tabuľka k Test 4_LSD.....	84
Tab. 12 - Tabuľka k Test 5_LSD.....	86
Tab. 13 – Tabuľka k Test 6_LSD.....	88
Tab. 14 – Tabuľka k Test 7_LSD.....	89

Zoznam symbolov a skratiek

SLAM	Simultaneous localization and mapping
PTAM	Parallel tracking and mapping
LSD-SLAM	Large-scale direct monocular SLAM
RANSAC	Random sample consensus
KF	Kalman filter
EKF	Extended kalman filter
HMD	Head mounted display
IMU	Inertial measurement unit
BA	Bundle adjustment
AHRS	Attitude and heading reference systems
ROS	Robotic operating system
CML	Concurrent mapping and localization
SFM	Structure from motion
SMC	Sequential monte-carlo
API	Application programming interface
KR	Klíčový rámeček

Úvod

Interaktívne spracovanie údajov v prostredí zmiešanej reality je obrovský súbor metód a postupov ktoré sú v dnešnej dobe viac ako aktuálne. Sú hlavným pilierom nastupujúcich alebo teda už reálne ustanovených trendov ktoré udávajú inovácie a rozmach zmiešanej a virtuálnej reality. Je treba povedať, že zmiešaná a virtuálna realita sa často prekrývajú a osobne si myslím, že je vhodné a veľmi správne pristupovať k obom a uvažovať v rovine kde sú obe reality. Zmiešaná a virtuálna realita majú obrovské využitie v oblastiach ako výučba, vizualizácia, zdravotníctvo ale hlavne vo veľkých masovo rozšírených oblastiach ako sú hry, filmy a celkovo interaktívna zábava.

Na trhu je množstvo zariadení od masovo rozšírených, komerčných a cenovo dostupných zariadení a sad pre zmiešanú a virtuálnu realitu až po drahé a ťažko dostupné zariadenia určené pre vývoj či iné špecializovanejšie zameranie. Je vhodné povedať, že zmiešaná realita obohacuje realitu ktorú užívateľ vníma o umelé prvky či objekty ktoré sú do vnímanej reality doplnené výpočtovým systémom. V tomto segmente prevládajú rôzne dátové prilby alebo okuliare s priehľadnými displejmi v ktorých je realita obohatená, napr. Google Glass. V úplnej (immersive) virtuálnej realite je realita užívateľa vizuálne, zvukovo prípadne ešte inak plne zamenená za virtuálnu. V tomto segmente prevládajú tiež rôzne dátové prilby a okuliare ako napr. Oculus Rift. Obe reality môžu využívať podsystemy ako vizuálny, zvukový, hmatový prípadne podsystem pre rôzne iné vnemy. Hlavným zdrojom vstupných údajov je človek a bezprostredné prostredie v ktorom sa práve nachádza a preto tieto reality môžu snímať rôzne vstupy od užívateľa a prostredia ako je natočenie hlavy, očí, snímanie pohybu, zvuku atď.. Samozrejme ako vstup môže slúžiť aj klávesnica, myš a rôzne iné zariadenia ale je potrebné povedať že, pre človeka je vždy najprirodzenejšie používať len to, čo mu je od prírody dané. V dnešnej dobe je pre najrozšírenejšie dátové prilby a okuliare pre virtuálnu a zmiešanú realitu bežné snímanie natočenia hlavy užívateľa, polohy užívateľa v rámci prostredia, snímanie rúk užívateľ a snímanie hlasového vstupu.

Výstupom alebo výsledkom použitia zmiešanej či virtuálnej reality na človeka je obohatenie alebo zmena reality ktorú človek vníma. To aké údaje sú potrebné je definované účelom a formou konkrétneho zariadenia virtuálnej a zmiešanej reality. Množstvo dneska dostupných dátových prilb ponúka skvelé snímanie natočenia hlavy užívateľa a v drvivej väčšine týchto zariadení sa pracuje aj na snímaní polohy užívateľa. Snímanie polohy užívateľa resp. hlavy užívateľa v priestore je ale vo väčšine prípadov vyriešené pomocným snímačom či kamerou ktorá nie je súčasťou náhlavnej sady ale je položená mimo, je to referenčný bod.

Táto skutočnosť obmedzuje použitie konkrétneho zariadenia na oblasť len v blízkosti referenčného bodu. Práve tu vzniká výzva, vytvorenie systému pre snímanie polohy užívateľa bez referenčného bodu.

Snímanie polohy užívateľa v svojom bezprostrednom prostredí bez referenčného bodu má obrovský potenciál, význam a využitie do budúcnosti. Už dnes existujú zariadenia zmiešanej a virtuálnej reality ktoré sú plne mobilné ako napr. Samsung Gear VR či Google Glass a obohatenie o takýto systém by bol veľkým prínosom. Využitie je rozsiahle v zmiešanej či virtuálnej realite ale aj v množstve rôznych iných odvetví a to hlavne v robotike.

Cieľom je teda poukázať na možnosť, vhodnosť a správnosť riešenia problému určenia priestorovej pozície užívateľa s HMD bez referenčného bodu pomocou dostupných optických SLAM metód a vypracovanie funkčného, správne nastaveného a vyladeného systému na ktorom je koncept možné priamo reálne otestovať čo predstavuje prácu nad ohromným súborom možností.

Riešenie tohto problému je vlastne riešenie problému SLAM teda určovanie neznámej polohy užívateľa či kamery v neznámom prostredí pričom je k dispozícii veľký súbor metód riešenia. Tento súbor obsahuje veľké množstvo použiteľných snímačov, filtrov, algoritmov, fúzií, zariadení a celkovo postupov z ktorých je urobený prienik. Vďaka popularite, jednoduchosti, efektívnosti a dostupnosti sú ako zariadenia vybrané Oculus Rift DK2 ako náhlavná prilba pre virtuálnu realitu a kamera, po dôslednom rešerši vhodných SLAM snímačov, ako základný snímač pre SLAM algoritmy ktorá je pripevnená na prednej časti Oculusu. Pre dôsledný výber už konkrétnej kamery sú otestované tri kamery z ktorých je vybraná najvhodnejšia pre splnenie podmienok optického SLAM. Táto kombinácia je veľmi dostupnou a má veľký potenciál byť v budúcnosti často využívaná vďaka stupňujúcej sa popularite virtuálnych sad a jednoduchosti dostupnosti kamery.

Celý systém je spustený na počítači s operačným systémom Linux na ktorom je ako jadro práce spustený robotický operačný systém ROS ktorý je pre práce založený na robotickom videní veľmi vhodný.

Pre takto skonštruovaný systém sú možnosti optického SLAM a fúzie dát veľmi rozsiahle. Pre otvorenosť, dostupnosť kódu a všeobecnú popularitu sú vybrané SLAM algoritmy PTAM a LSD-SLAM pričom je vhodné že oba fungujú na rôznom princípe získavania informácie z obrazu. PTAM je založený na rozpoznávaní bodov záujmu v obraze, párovaní bodov v rôznych rámoch a upravovaním vytvorenej mapy pomocou BA. LSD-SLAM využíva viac informácií z obrazu keďže nepoužíva body záujmu ale funguje priamo na spracovaní intenzity pixelov a gradientov. Vyhotovený systém je ďalej obohatený o spracovanie dát z inerciálnych snímačov Oculusu ktoré sú

spracované v Matlabe jednoduchou dvojitou integráciou akcelerácie na polohu. Výsledné dáta sú spojené vlastnou implementáciou fúzie dát.

Takto navrhnutý systém je kompletný no keďže dáta tečúce z kamery nie sú nijak upravované pred vstupom do SLAM algoritmov, je implementovaná predfiltrácia obrazu pre otestovanie vhodnosti. Sú testované rôzne algoritmy spracovania obrazu ako Sobelove derivácie či Gaussovský filter.

Je vhodné povedať že táto práca je prienik ohromného množstva rozsiahlej tematiky. Nie je zameraním práca na jednej menšej dielčej časti no skôr práca s celým obrovským celkom a uvažovaním nad ním pričom veľké množstvo použitých algoritmov a zariadení je potrebné pre sfunkčnenie a využiteľnosť upraviť a vyladiť.

Pre dobré pochopenie štruktúry práce sú opísané jednotlivé kapitoly. V prvej kapitole je uvedený rozbor problému SLAM ako všeobecného problému. V druhej kapitole, ktorá tvorí teoretický rozbor aj teoretickú analýzu, sú vybrané snímače a zariadenia teda aj užšia oblasť SLAM ktorá je popísaná, ako aj výber algoritmov a ich popis. Keďže vybrané snímače sú kamera a IMU (inerčné snímače), následne sú opísané témy optický SLAM a inerčný SLAM. Poslednou časťou druhej kapitoly je výber prostredia, teda ROS a opis vybraných algoritmov teda PTAM, LSD-SLAM a dvojitej integrácie akcelerácie na polohu. Tretia kapitola predstavuje testovanie, výber a špecifikáciu už konkrétnych zariadení, teda HMD sa stáva Oculus Rift DK2 a kamerou je SJCAM SJ5000. V štvrtej kapitole nasleduje návrh, implementácia a testovanie jednotlivých častí systému. Popísaná je štruktúra systému, komunikácia medzi jednotlivými časťami, úpravy kódu, úpravy algoritmov, menšie implementácie. Ďalej sú testované algoritmy, správne nastavené, je implementovaná a testovaná predfiltrácia a výstupom kapitoly sú funkčné podčasti systému. Piata kapitola je zameraná na fúziu všetkých dát a spojenie všetkých častí do funkčného celku. Fúzia je implementovaná spojením orientácie získanej z Oculusu a posunu kamery získanej z optických SLAM algoritmov. Šiestou kapitolou je záver.

Formulácia úlohy a cieľ práce

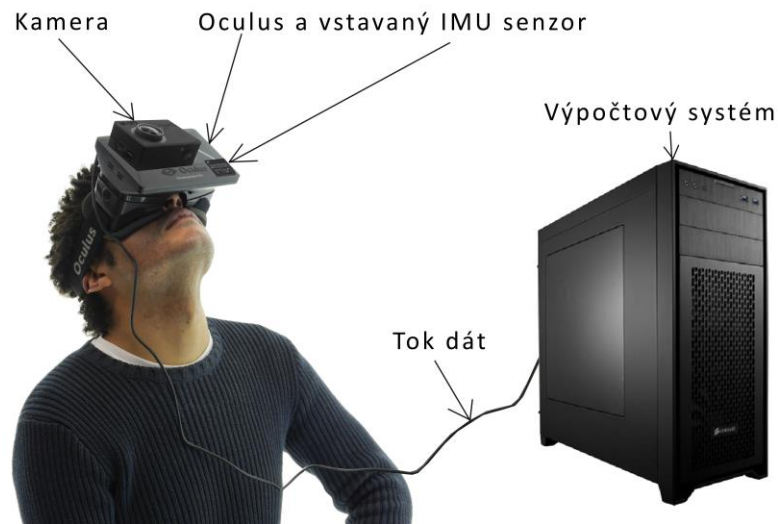
Pri formulácii úlohy a cieľov práce je potrebné vyjadrenie sa k jednotlivým bodom zadávacieho listu keďže je téma pomerne voľná a body zadávacieho listu všeobecne zadané. Dodefinované sú iba body, ktoré sú všeobecnejšie a potrebujú dodefinovať.

Navrhnuť systém spracovania obrazu pre technológiu zmiešanej reality s orientáciou na monokulárnu rekonštrukciu snímaného obrazu z pohľadu pozorovateľa – je treba dodefinovať, že rekonštrukcia je formou vytvárania mapy, kedy pri pridávaní bodov alebo elementov je známa pozícia kamery. Z toho je zjavné, že pre vytýčený cieľ je dôležitejšia odhadovaná pozícia tejto kamery a rekonštrukcia je menej dôležitá. Preto je vhodné zameranie sa na odhad pozície kamery a problém rekonštrukcie ako takej nerozvíjať.

Čo do podmienok, cieľový koncept a systém by mal zohľadňovať, zahŕňať, približovať sa, opisovať a riešiť viacero podmienok a problémov ako spracovanie údajov v reálnom čase, prácu v rôznych svetelných podmienkach ako je dnu alebo vonku, prechod z vnútorného prostredia do vonkajšieho a naopak, neobmedzenosť pohybu, toleranciu pohybov osôb alebo objektov v okolí, rýchle pohyby, zotavenie sa z chyby, jednoduchosť a pohodlnosť pre užívateľa, prostredie bez umelých značiek.

Celkový pohľad na systém

Pred vstupom do jadra práce je kvôli rozsiahlosti a pomernej zložitosti pochopenia celkov vhodne uviesť štruktúru aká je použitá naprieč prácou. Za týmto účelom je potrebné uvedenie nasledujúcich pomocných celkových schém.

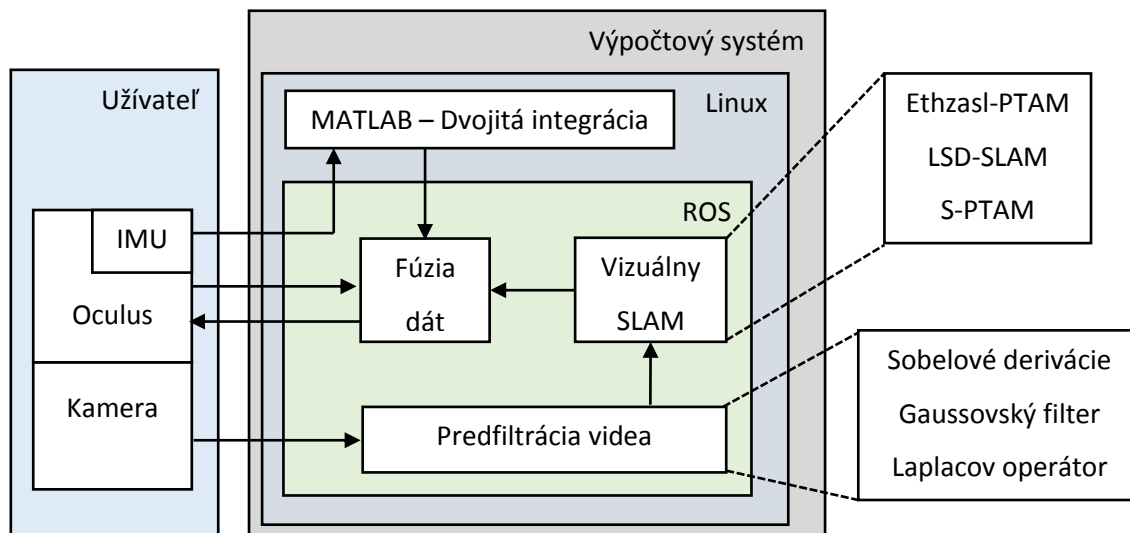


Obr. 1 – Fyzicko-hárdverová schéma celého systému.

Na vyššie uvedenom obrázku je znázornená celková fyzicko-hárdverová schéma. Zobrazuje jednotlivé hárdverové prvky systému, teda Oculus, ktorého súčasťou je IMU snímač, na Oculuse pripevnenú kameru a tieto komponenty zapojené do výpočtového systému, teda počítača.

Zosnímané informácie tečú z Oculusu (aj IMU) a kamery do výpočtového systému. V tomto systéme sú spracované a výsledok je vyslaný a zobrazený späť v Oculuse.

Pre úplné pochopenie systému je takisto potrebné uviesť celkovú softvérovú schému systému.



Obr. 2 –Softvérová schéma celého systému.

Zo softvérovej schémy je zrejmé, ako je systém zostavený. Na výpočtovom systéme, teda počítači, je spozajznený operačný systém Linux a na ňom robotický operačný systém ROS. V tomto ROS systéme je implementovaný modul predfiltrácie videa ako aj modul vizuálneho SLAM. Modul predfiltrácia videa predstavuje rôzne algoritmy pre spracovanie videa ako sú Sobelové derivácie, Gaussovský filter atď.. Vizuálny SLAM predstavuje algoritmy vizuálneho SLAM ako sú Ethzasl-PTAM, LSD-SLAM atď.. Vstupom vizuálneho SLAM je predfiltrované video z kamery ktoré je týmto SLAM spracované a výstupom SLAM je odhad pózy kamery v 3D priestore. Na Linuxe je takisto spozajznený program MATLAB ktorý dvojitou integráciou dát z IMU predstavuje inerčný SLAM.

Dáta teda tečú z kamery do predfiltrácie, odkiaľ ďalej putujú do vizuálneho SLAM a ďalej do fúzie dát. V rovnakom čase dáta z IMU putujú do MATLABu a odtiaľ následne do fúzie dát. Takisto v rovnakom čase dáta z Oculusu putujú do fúzie dát. Tieto všetky tri toky sú fúziou dát spojené a výsledok je vyslaný do Oculusu.

Takto abstraktne a všeobecne opísaný a predstavený celkový systém by mal čitateľovi značne uľahčiť pochopenie štruktúry jadra a celkovej následnosti celkov a kapitol. Je teda vhodné pokračovať jadrom práce.

1. Teoretický rozbor SLAM

SLAM, teda súbežná lokalizácia a mapovanie predstavuje úlohu lokalizácie užívateľa, čiže určenie jeho polohy, v neznámom prostredí pri súbežnom mapovaní tohto neznámeho prostredia. Teda pre lepšie chápanie, robot idúci neznámym prostredím je schopný prostredie mapovať, čiže vykresľovať jeho mapu a zároveň lokalizovať kde sa v nej nachádza v 2D či 3D priestore. SLAM je takisto známy pod pojmom CML [8].



Obr. 3 - Príklad zmapovaného prostredia vďaka pohybu užívateľa prostredím a lokalizácia užívateľa

Na Obr. 3 je znázornená mapa ktorá je výsledkom fungujúceho systému SLAM na užívateľovi využívajúcom laser ako snímač. SLAM je veľmi komplexná množina pretože je to iba názov problému a jeho riešenie nie je presne dané vzhľadom na použité snímače, postupy ani filtre a existuje množstvo kombinácií. Je potrebné povedať že všeobecne sú vstupom pre SLAM dáta zo snímačov nejakého druhu ktoré sú v SLAM spracované a vyhodnotené. Takisto pre naše použitie, kde je hlavným cieľom získanie pozície, nie je mapovanie podmienkou ale výhodou, pretože pre správne fungovanie postačí zmena polohy bez záznamu o mape, no možnosť mapovania môže výrazne pomôcť pri správnosti a hlavne presnosti určovania polohy.

Nasleduje rozdelenie problému SLAM podľa použitých snímačov v kapitole 1.1. Rozdelenie podľa snímačov. V kapitole 1.2. Známe všeobecné metódy sú opísané hlavné metódy riešenia a v kapitole 1.3. Reprezentácia mapy sú opísané formy reprezentácie mapy.

Je ešte treba povedať, že pre všeobecné pochopenie štruktúry SLAM je treba zdefinovať hlavné konceptuálne problémy SLAM ktorými sú:

- lokalizácia,
- mapovanie,

- asociácia dát

Lokalizácia a mapovanie sú zahrnuté v už uvedených kapitolách a problému asociácie dát je venovaná osobitná kapitola neskôr a z toho dôvodu jej vo všeobecnom rozbere SLAM nie je venovaná pozornosť.

1.1. Rozdelenie podľa snímačov

1.1.1. Exteroceptívne

Exteroceptívne (exteroceptive) snímače snímajú okolitý svet a nie vlastnosti užívateľa. Patria sem snímače ako sonar, laser, kamera, GPS [1]. Všetky tieto snímače majú šum či chybu. Prvé tri zaznamenávajú len lokálny okolitý svet pričom ale dosť presne a fungujú pri podmienkach špecifických pre každú technológiu. GPS nefunguje všade a má takisto väčšiu chybu či nepresnosť.

Z pohľadu množstva informácií je najviac množstva informácií snímaných kamerou ktorá je takisto finančne dostupná. Množstvo informácií sa takisto odzrkadľuje na väčšej výpočtovej náročnosti. V súčasnosti je kamera populárnym snímačom pre SLAM a takisto je dostupných množstvo publikácií vzhľadom na túto tému.

Technológie využívajúce sonar alebo laser používajú vo väčšine prípadov metódu čas letu a teda je meraný čas od vyslania signálu k jeho opätovnému prijatiu a tým vypočítaná vzdialenosť. Pre sonar je signál zvuk a pre laser zas lúč zo spektra vlnových dĺžok a je treba povedať, že aj tieto snímače sú vhodné snímače použité za účelom tejto práce. Existujú viaceré publikované práce riešenia SLAM laserom ktoré boli zvažované pre túto prácu ako napr. práca Ji Zhang [2] ktorá využíva lidar, teda laserový radar. Takéto lidary sú ale pomerne drahé. Je vhodné poukázať aj na ďalšie publikácie Ji Zhang [5,6,7] ktoré sa venujú riešeniu SLAM a to lidarom, kamerou, GPS alebo rôznymi kombináciami týchto snímačov a to so skvelými výsledkami. GPS je pre svoju veľkú nepresnosť pre túto prácu nepoužiteľný snímač.



Obr. 4 – Snímač lidar a mapa vytvorená mapovaním pomocou lidarů.

1.1.1. Proprioceptívne

Proprioceptívne (proprioceptive) snímače snímajú vlastnosti užívateľa ako sú rýchlosť, zmena pozície, zmena natočenia kolies, akcelerácia, rotácia kolesa atď., teda nesnímajú vlastnosti okolitého sveta. Snímačmi ktoré tieto údaje dokážu poskytnúť sú gyroskop, akcelerometer, magnetometer počítadlá atď.. Bežne dostupné moduly ktoré troma snímačmi gyroskop, magnetometer a akcelerometer disponujú sa nazývajú IMU (inertial measurement unit). Inerčnej SLAM je venované viac priestoru v druhej kapitole pretože je pre využitie v tejto práci vhodná a dostupná. Metóda určovania polohy vďaka údajom výlučne z vlastností užívateľa ako sú zrýchlenie či natočenie sa nazýva dead reckoning [1]. Dead reckoning metódy sú bežne využívané v námorníctve či letectve kde sa prechádzajú stovky míľ no pre využitie v menšej mierke je ich chyba či šum pomerne veľký. Rovnaký princíp a vlastnosti ako dead reckoning je spájaný s pojmom odometria a jediným skutočným rozdielom je, že odometria sa odohráva zväčša na súši, nie na mori, a je často spájaná s tematikou robotiky. Pri odometrii aj dead reckoningu sa pozícia zisťuje inkrementačným akumulovaním zmeny polohy čo vedie k nepresnostiam kvôli akumulácii chyby.



Obr. 6 – Čip IMU.



Obr. 5 - Veľká chyba dead reckoningu alebo odometrie z údajov z IMU.

Prevzaté z [3].

Vstupné údaje môžu pochádzať od rôznych zdrojov resp. snímačov. Pre príklad odometrie je vhodné uviesť aj prípad, kde robot sleduje natočenie a rotáciu svojich kolies. Pri takomto pohybe po rovine je teda z natočenia a otáčania kolies možné vypočítať zmenu polohy. Pre túto prácu sú takéto mechanické postupy nevhodné kvôli zložitosti a poklese pohodlia.

1.2. Známe všeobecné metódy riešenia SLAM – Pravdepodobnostné filtre

V posledných rokoch sa stali najznámejšími a najbežnejšími metódami riešenia SLAM metódy založené na pravdepodobnostných filtroch [8]. Popísané sú tri z ktorých každý predstavuje matematickú deriváciu Bayesovho pravidla.

Každý z uvedených filtrov je vhodný alebo menej vhodný na riešenie určitého konkrétneho problému. Filtre sú všeobecné matematické metódy a neviažu sa na riešenie len danej oblasti či problému.

1.2.1. Kalman Filter

Ako pravdepodobnostný filter je veľmi známy a používaný Kalman Filter (KF) ako aj jeho dve hlavné obmeny – Extended Kalman Filter (EKF) a Information Filter (IF) známy tiež ako Extended IF (EIF) [8]. Vstupom KF filtra sú dáta zo snímačov pričom výstup pri splnení podmienok konverguje k správne optimálnemu odhadu. Podmienky zahŕňajú, aby bolo možné systém popísať ako lineárny pričom systémové chyby a chyby zo snímačov majú Gausovo rozdelenie. Filter pracuje rekurzívne s aktuálnymi hodnotami ako aj s predchádzajúcimi hodnotami čo má za následok vyhladenie. Filter odhaduje pravdepodobnosť súčasného stavu ako aj stavu budúceho. Váha určená pre vážený priemer je daná kovariančnou maticou ktorá udáva chybu odhadovaného stavu [3]. Vylepšeným KF je EKF ktorý sa používa v prípade, že je systém nelineárny alebo je číselne nestabilný. EKF je založený na využívaní stavového vektora ktorý sa skladá z polohy entity alebo užívateľa a elementov mapy ktoré sú rekurzívne obnovované a upravované podľa nelineárneho modelu pozorovania a transformácie [1]. Neurčitost' je reprezentovaná pravdepodobnostnými funkciami hustoty. Zložitosť filtra je kvadratická, je nevhodný pre väčšie priestranstvá a je náchylný na chyby. Existuje viacero pokusov o zníženie zložitosti napr. CEKF (Compressed EKF), SEIF (Sparse extended information filter), Divide and Conquire, CI-Submaps (Conditionally independent Subamaps). Filter IF je založený na inverzii stavového chybového vektora kovariančnej matice a má takisto viaceré výhody oproti čistému KF. K výhodám patrí filtrovanie dát súčtom informačných matic a vektora čím je dosiahnutá vyššia presnosť. K nevýhodám patrí nutnosť odhadu stavu v aktualizáčnom kroku pri nelineárnych systémoch a takisto nevýhodou je nutnosť inverzie matic v predikčnom kroku čo výrazne zvyšuje výpočtovú zložitosť IF. Problémom Kalmanových filtrov je menšia využiteľnosť pri rozsiahlych mapách [1].

1.2.2. Časticový filter

Časticový filter, teda Particle filter (PF) je rekurzívny Bayesov filter ktorý je založený na metóde Monte Carlo a takisto sa nazýva aj Sequential Monte-Carlo (SMC). Reprezentácia hustoty pravdepodobnosti je v metóde Monte Carlo predstavená ako množina vzoriek ktorých váhy a rozloženie v stavovom priestore určuje pravdepodobnosť jednotlivých polôh užívateľa. Časticový filter je schopný spracovať aj veľmi nelineárne dáta zo snímačov a takisto nie-Gausovský šum vďaka reprezentácii distribúcie ktorá je reprezentovaná ako množina vzoriek, či častíc, ktoré pochádzajú práve z tejto distribúcie. Metóda nie je vhodná pre nasadenia v reálnom čase pretože so vzrastajúcim počtom bodov záujmu veľmi rastie výpočtová náročnosť [8]. Kvôli tejto náročnosti

sa PF používa hlavne na riešenie problému lokalizácie a v mnohých prácach sa pre vyriešenie celého problému SLAM kombinuje s inými metódami. Príkladom takýchto známych prác sú fastSLAM [34] a fastSLAM2.0 [35]. FastSLAM zachytáva distribučnú pózu užívateľa ako sadu Rao-Blackwellized častíc, kde každá častica reprezentuje trajektóriu užívateľa, a takisto zachováva svoju vlastnú mapu využívajúc EKF [1]. Takisto táto častica či vzorka obsahuje hypotézu o asociácii svojich dát a prežije len s určitou pravdepodobnosťou. Algoritmus sa skladá z procesu generovania častíc a procesu re-samplingu. Zložitosť je logaritmická pričom čím viac častíc, tým dlhší výpočtový čas no je ťažké definovať aký počet častíc je dostatočný pre správny a dostatočný výpočet.

1.2.3. Metódy založené na algoritme Expectation maximization

Expectation maximization (EM) je štatistický algoritmus vyvinutý v súvislosti s estimátorom Maximum Likelihood (ML) vhodný a používaný prevažne pre riešenie problému mapovania. Metóda sa skladá z dvoch krokov: predikčný krok (E-krok) a krok maximalizácie (M-krok). V prvom kroku metóda spracúvava údaje nad množinou polôh užívateľa, teda lokalizuje užívateľa v mape a v druhom kroku metóda počíta odhadovanú najpravdepodobnejšiu mapu. Výsledkom takéhoto procesu je časom sa zvyšujúca presnosť mapy. Metóda je veľmi vhodná pre riešenie problému asociácie dát. Nevhodnosť metódy pre použitie v reálnom čase je odôvodnená nutnosťou spracovania tých istých údajov viacnásobne pre získanie správnej mapy. Odhad pozície robota sa stáva exponenciálne zložitejší rastúcou veľkosťou mapy čo má za následok nestabilitu. Tento problém môže byť vyriešený použitím inej metódy alebo filtra v E-kroku.

1.3. Reprezentácia mapy

Mapovanú mapu je možné ukladať či reprezentovať rôznym spôsobom.

1.3.1. Metrické mapy

Prevládajú mriežkové alebo bodové mapy. V mriežkových môže byť súradnica obsadená alebo nemusí byť obsadená a každá bunka môže obsahovať 2D, 2.5D alebo 3D údaje [1]. Bodové mapy zachovávajú 3D polohu bodu záujmu. Riedke bodové mapy nie sú vhodné pre použitie vyhýbania sa prekážok pretože nemajú dostatočné husté mračno.

1.3.2. Topologické mapy

Topologické mapy reprezentujú prostredie ako zoznam významných miest, napríklad spád alebo stena, spojených čiarami. Reprezentácia prostredia ako graf je vhodnou voľbou pri snímaní veľkým priestranstiev.

1.4. Asociácia dát

Asociácia dát je problém popisujúci neurčitosť vstupnej množiny dát. Je to problém určenia a hlavne identifikácie spojitosti dát. Pre názorný príklad je vhodné uviesť príklad kde je na dvoch obrázkoch rovnaká budova no na jednom počas dňa a na druhom počas noci. Pre človeka by riešením tohto problému bolo určenie, že táto budova je jedna a tá istá.

2. Analýza SLAM pre použitie v práci a opis vybraných algoritmov

Po všeobecnom opise SLAM problému je vhodné určiť konkrétnejšie smer ktorým je vhodné sa uberať čo do snímačov, zariadení a detailnejšieho popisu relevantného podproblému SLAM.

2.1. Výber zariadení a metód

Pre podrobné preskúmanie užšej oblasti problému SLAM a vyberanie algoritmov je potrebné zvoliť zariadenia a snímače. Je známe, že je použitá dátová prilba pre virtuálnu realitu, teda HMD. Je vhodné sa zamyslieť akými snímačmi takéto bežné prilby disponujú, či sú vhodne použiteľné a akými snímačmi je ich možné obohatiť aby boli splnené podmienky práce. Väčšina HMD je v dnešnej dobe osadená mikrofónom a inerciálnymi snímačmi IMU. Prvým snímačom ktorý je teda možné testovať alebo používať pre riešenie problému SLAM je IMU z HMD no z rozboru je zjavné že takýto snímač pravdepodobne nie je dostačujúci pre dostatočne funkčný SLAM. Po zvážení podmienok ako efektívnosť, cena, pohodlie, výkon, dostupnosť, platforma, potenciál je zvolený ako druhý snímač kamera. Užšími oblasťami SLAM ktorými je teda potrebné sa zaoberať je optický a inerčný SLAM. Pri riešení problému je takisto možné oba riešenia vhodne kombinovať a je potrebné povedať, že hlavným výberom je teda optický SLAM s ľubovoľným výpomocným použitím inerčného SLAM.

2.2. Optický SLAM

Optický či vizuálny SLAM (visual SLAM) je podoblasť problému SLAM pri ktorého riešení sa ako hlavný snímač používa kamera. Existuje veľké množstvo metód, algoritmov a postupov ktoré sa používajú a treba povedať, že rovnako existuje aj veľké množstvo druhov kamier ktoré sa používajú. Dostupné sú kamery s rôznymi vlastnosťami, určeniami a spektrami ktoré snímajú a kamerám ako takým je venovaná kapitola 2.2.1. Kamera ako hlavný snímač.

Čo sa týka metód riešenia vizuálneho SLAM, je možné používať všetky filtre uvedené vo všeobecnom rozbere SLAM, no existujú aj špecifické metódy určené iba pre vizuálny SLAM ako sú techniky využívajúce Structure from Motion opísané v kapitole 2.2.2. Techniky využívajúce Structure from Motion.

Opísané metódy pracujú s informáciami získanými zo vstupnej sady obrazov. Existujú dva hlavné metódy získavania týchto informácií a rozdelené sú do dvoch kategórií podľa množstva využitých informácií z jednotlivých obrazov:

- riedke sledovanie (sparse tracking),
- husté sledovanie (dense tracking).

Riedke sledovanie využíva iba určitú časť informácií z obrazu, a to napríklad body, úsečky, menšie oblasti. Tejto oblasti je venovaná kapitola 2.2.3. Body záujmu.

Husté sledovanie využíva celý obraz a týmto metódam je venovaná kapitola 2.2.4. Metódy založené na intenzite pixelov.

Poslednou kapitolou optického SLAM je kapitola 2.2.5. ktorá je venovaná asociácii dát.

2.2.1. Kamera ako hlavný snímač

Podľa použitého snímania sú známe viaceré druhy vizuálneho SLAM [1]:

- visual SLAM – pre snímanie je použitá kamera alebo kamery,
- visual-inertial SLAM – riešenie SLAM pomocou kamery aj IMU s následnou fúziou týchto prístupov nejakou formou,
- vision-only alebo camera-only SLAM – pre snímanie je použitá výlučne iba kamera alebo kamery,
- single-camera alebo mono-SLAM – pre snímanie je použitá výlučne iba jedna kamera.

Je treba povedať, že pri využívaní kamier ako snímačov je potreba kalibrácie obrazu kvôli prítomnosti skreslenia spôsobeného objektívom a optickými vlastnosťami kamery. Pre potreby kalibrácie existuje obrovské množstvo algoritmov.

Ďalej je vhodné povedať že existujú tri hlavné problémy pri snímaní scény kamerou [1]:

- Rozmazaný obraz. Pri nosení kamery človekom je obraz z kamery neostrý kvôli rýchlim pohybom. Tento problém je vyriešený lepším algoritmom alebo je takisto vhodné ho riešiť znížením expozičného času kamery.
- Pohyb v prostredí. V prostredí sa môžu hýbať autá, ľudia. Viaceré systémy a algoritmy s tým rátajú a veľmi spoľahlivo filtrujú resp. nezachytávajú.
- Málo textúry alebo opakujúca sa textúra v snímanom obraze.

Každý z uvedených problémov môže nastať aj v prípade bežného nosenia dátovej prilby a preto je ich potrebné ošetriť a brať do úvahy.

Pri využívaní kamery ako hlavného snímača existujú prístupy s použitím:

- jednej kamery - monokulárne prístupy,
- dvoch kamier - stereokulárne prístupy,
- viacerých kamier,
- RGB-D kamier – kamier ktoré zaznamenávajú aj hĺbku scény, napr. Kinect.

2.2.1.1. Monokulárne prístupy

Pri použití jednej kamery nastáva problém mierky a inicializácie. Problém inicializácie spočíva v tom, že nie je možné z jedného prvého obrazu z kamery vypočítať perspektívu. Tento problém je v súčasnosti vyriešený rôznymi algoritmami ktoré sa delia na algoritmy s omeškaním a bez omeškania.

2.2.1.2. Stereokulárne prístupy a prístupy využívajúce viac kamier

Pri dvoch a viacerých kamerách ani jeden z problémov jednej kamery nie je relevantný pretože hneď od začiatku je zjavné, že sú k dispozícii minimálne dva obrazy a mierka je určená takisto pretože je známa vzdialenosť medzi kamerami. Pri využívaní dvoch kamier ale vznikajú viaceré iné problémy ako ich synchronizácia či odlišný obraz ktorý je viazaný na vlastnosti kamery ako automatická expozícia, vyváženie bielej, čiže celkovo údaje zo svetelného snímača.

2.2.1.3. RGB-D kamery

RGB-D kamery sú kamery ktoré okrem obrazu poskytujú informácie o hĺbke pomocou hĺbkovej kamery. Tieto kamery nefungujú za bežných podmienok cez deň v otvorených priestranstvách.

2.2.2. Techniky využívajúce Structure from Motion

Snímajúc scénu kamerou je vytvorená množina obrazov ktoré zobrazujú neznámu scénu či štruktúru. Techniky Structure from Motion (SfM) sú založené na vypočítavaní 3D štruktúru scény z vstupnej sady týchto obrazov. Štandardná procedúra spočíva v získaní bodov záujmu z obrazov aby mohlo dôjsť k spárovaniu obrazov a následnej nelineárnej optimalizácii nazývanej Bundle Adjustment (BA) za cieľom minimalizovať reprojekčnú chybu [1]. Existuje viacero prístupov

k riešeniu tohto problému. Je treba ale povedať že problém SfM ako taký sa nezaobera pozíciou kamery ale vytváraním 3D modelu scény.

2.2.2.1. Vizuálna odometria

Jednou z metód riešiacich inkrementálny SfM je vizuálna odometria. Metóda dokáže v reálnom čase určovať polohu kamery a pozíciu bodov v priestore z tradičnej kamery resp. záznamu. Pre správne pochopenie rozdielu medzi SfM a vizuálnou odometriou je treba povedať, že vizuálna odometria má práve za hlavnú úlohu určovanie pozície kamery a štruktúra scény je vedľajšia. Vylepšené metódy vizuálnej odometrie používajú lokálny BA. Tieto metódy dokážu pracovať s tisícami bodov v reálnom čase.

2.2.2.2. Bundle adjustment

Metóda ktorá takisto spadá pod problém SfM. BA je súbor metód riešiacich problém postupného vyhladzovania a vylepšovania vypočítanej štruktúry scény postupom času s pribúdajúcimi bodmi, elementami a informáciami.

2.2.3. Body záujmu

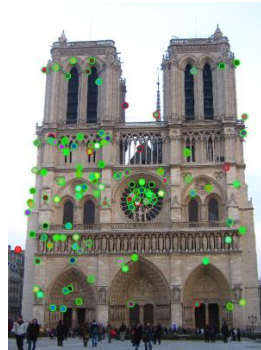
Opísané metódy fungujú na základe spracovania informácií z vstupnej sady obrazov ktoré je potrebné nejako získať. Jedným z týchto postupov je extrakcia bodov záujmu. Mnoho algoritmov v oblasti optického SLAM používa pre riešenie problému body záujmu. Tieto body využíva aj vyššie spomenutá metóda SfM. Body záujmu (salient features) predstavujú body získané z obrazu, sú to body v 2D obraze. Existuje viacero postupov a treba povedať že body záujmu nemusia byť body, môžu to byť oblasti či úsečky. Pre optický SLAM založený na hľadaní úsečiek v obraze je vhodné poukázať na prácu Guoxuan Zhang [20]. Orientačné body môžu byť:

- Umelé – napr. papier so vzorom pripevnený na podlahe. Uľahčuje orientáciu a mapovanie priestoru a je potrebné ich pripraviť dopredu.
- Prirodzené – čiže prirodzené prostredie kde sa človek pohybuje, teda strom, textúra, kraj steny.

S ohľadom na podmienky a požiadavky práce je pozornosť venovaná prirodzeným orientačným bodom. Proces získavania bodov záujmu z obrazu sa skladá z dvoch fáz:

1. Detekcia (detection) – nájdenie bodu v obraze
2. Popis (description) – popisanie bodu napr. jeho okolím alebo ináč

Je treba povedať že stabilný bod záujmu sa vyznačuje ľahkosťou extrakcie, presnosťou a stabilnosťou voči natočeniu, posunu, zmeny veľkosti a zmeny osvetlenia scény [1].



Obr. 7 - Nájdené body záujmu v obraze.

2.2.3.1. Detektory

Existuje veľké množstvo detektorov medzi ktoré patrí Harris corners detector, Harris-Laplaca Hessian-Laplace points detectors, Harris-Affine a Hessian-Affine, Difference of Gaussians použité v SIFT, Maximally Stable Extremal Regions, Features from Accelerated Segment Test (FAST), Fast-Hessian použité v SURF atď.. Rôzne detektory sú rôzne vhodné pre rôzne podmienky.

2.2.3.2. Popisovače

Najznámejším popisovačom je SIFT popisovač ktorý priradí bodu vektor zložený z 128 komponentov [1]. Tieto sú vypočítané z okolia bodu. Existuje variácia SIFT nazývaná PCA-SIFT. Popisovače sa takisto líšia a rôzne popisovače majú lepší výkon v rôznych oblastiach. Medzi ďalšie známe popisovače patrí SURF a rôzne iné ako napr. ASIFT, BRIEF teda Robust Independent Elementary Features, ORB, PIRF.

Je treba podotknúť že existujú nové metódy ktoré pracujú priamo v rovine intenzity pixelov a nepotrebujú popisovače ani extrakciu bodov. Tieto metódy sú v súčasnosti populárne.

2.2.4. Metódy založené na intenzite pixelov

Metódy založené na intenzite pixelov nevyužívajú body záujmu. Obrazy prichádzajúce z kamery sú porovnávané medzi sebou a vďaka zmene polohy pixelov je možné určiť hĺbku scény.

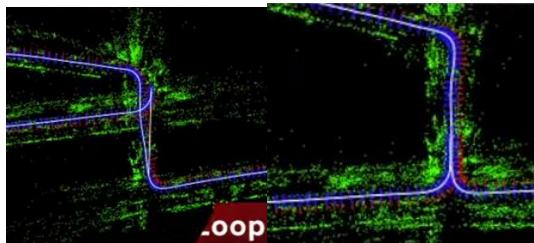
2.2.5. Problémy párovania obrazu a asociácie dát

Dáta, teda získané body záujmu zo za sebou idúcich snímkou z kamery alebo kamier majú vzájomnú spojitosť. Disciplína ktorá túto oblasť rieši sa nazýva párovanie obrazu a existujú dva kategórie techník a to short baseline a long baseline. Algoritmov a postupov pre oba techniky je mnoho. Asociáciu dát je možné popísať ako hľadanie súvisu konkrétnych dát s dátami z celej mapy

alebo množiny. Táto disciplína nie je viazaná iba na problematiku optického SLAM ale je braná ako všeobecný problém SLAM. Asociácia dát má viacero podproblémov.

2.2.5.1. Detekcia uzavretia slučky

Detekcia uzavretia slučky – je rozpoznanie miesta ktoré bolo užívateľom už navštívené s následnou úpravou mapy a polôh užívateľa porovnaním súčasnej polohy a polohy korešpondujúcej s rozpoznaným miestom. Detekcia musí byť ale určite správna pretože prípad kedy by bolo miesto rozpoznané ako rovnaké a pritom by to nebolo to isté reálne miesto, čo sa kvôli opakovateľnosti textúr v reálnom svete môže stať, by mal katastrofálne následky. Najčastejšie používané techniky sú GraphSLAM, Bag of Visual Words (BoVW) či Vocabulary. Správna detekcia slučiek je hlavným prínosom pre robustný SLAM, teda SLAM používaný vo veľkých priestranstvách. Pre použitie v práci je táto detekcia nežiadúca.



Obr. 8 - Pred uzavretím a po detekcii a uzavretí slučky

2.2.5.2. Unesený robot

Unesený robot (Kidnapped robot) je problém ktorý popisuje znovu sa nájdenie v mape. Pre lepšie pochopenie robot či užívateľ už pozná mapu a nastane prípad kedy sa zrazu ocitne na inom mieste, napríklad vypnutím zariadenia a prenesením na iné miesto. Problém spočíva v rozpoznaní daného miesta a správnom určení polohy v už naučenej mape. Vyriešenie tohto problému v rámci tejto práce je výhodou a môže napomôcť stabilite algoritmu.

2.2.5.3. Mapovanie a lokalizácia viacerých užívateľov v jednej mape

Mapovanie a lokalizácia viacerých užívateľov v jednej mape (Multi-session and cooperative mapping) – je kooperatívne pracovanie na jednej mape viacerými užívateľmi.

2.3. IMU a inerčný SLAM

Keďže je ako pomocný snímač zvolený snímač IMU, je vhodné popísať presnejšie čo to IMU je a možnosti inerčného SLAM.

2.3.1. Snímač IMU

Inertial Measurement Unit (IMU) predstavuje inerčný snímač. Je to elektronické zariadenie ktoré sa používa na snímanie vlastností užívateľa ako akcelerácia, uhlová rýchlosť a poprípade aj magnetické pole okolo užívateľa pomocou snímačov ako akcelerometer, gyroskop poprípade magnetometer [24]. Snímač je veľmi používaný v širokom spektre zariadení od lietadiel a lodí až po mobilné zariadenia.

Gyroskop sa používa pre snímanie natočenia zariadenia vďaka neustálemu snímaniu gravitácie Zeme. Dáta poskytované gyroskopom reprezentujú natočenie zariadenia okolo určitej osi.

Akcelerometer sa používa pre snímanie negravitačného zrýchlenia zariadenia v smere určitej osi no bez znalosti natočenia zariadenia.

Magnetometer reprezentuje dáta určujúce polohy pólů Zeme čo je v konečnom dôsledku bežne používané pre kalibráciu a úpravu chýb gyroskopu a akcelerometra.

Existujú snímače s rôznym počtom snímaných osí no v dnešnej dobe sú najviac rozšírené triosé gyroskopy, akcelerometre a magnetometre ktoré fungujú na elektronickom princípe nazývanom MEMS. Každý zo snímačov teda poskytuje tri hodnoty v každom čase pre každú z troch osí X, Y, Z ktoré sú snímané.

2.3.2. Inerčný SLAM

Inerčný SLAM je podproblém SLAM v ktorom hlavný senzor predstavuje inerčný senzor IMU. Ako bolo spomenuté, táto forma odometrie je veľmi nepresná a pre použitie v práci nie je dostačujúca. Preto je inerčný SLAM použitý iba ako výpomocný alebo voliteľný a v tejto kapitole mu nie je venované veľa priestoru.

Je treba povedať že, problému inerčného SLAM nie je v literatúre venovaný samotný priestor pretože samotný inerčný SLAM ako plnohodnotný SLAM nemôže fungovať. Už po jednoduchej úvahe je treba povedať, že takýto SLAM neposkytuje mapovanie a preto je vhodnejšie nazvať tento problém skôr inerčná odometria.

Prvým predpokladom úspešnej inerčnej odometrie je správne určenie natočenia zariadenia. Tomuto problému nie je venovaných veľa algoritmov no už dnes dostupné aplikácie a API disponujú veľmi presným a správnym už implementovaným výpočtom natočenia vďaka

snímačom IMU. Tieto algoritmy majú za úlohu fúziu dát z gyrokopu, akcelerometra a magnetometra pre poskytnutie natočenia.

Ďalším možným výpočtom pre ktorý je možné IMU použiť je výpočet zmeny polohy vďaka dvojitej integrácii akcelerácie v smere natočenia. Existujú takisto postupy a prístupy kde v tomto kroku výpočtu je zahrnutý aj nejaký filter, napríklad KF. Takisto je vhodné vyzdvihnúť a zdôrazniť prácu Seb Madwick [18,27] kde sú výsledky spracovania inerčných dát naozaj pozoruhodné hoci naviazané na konkrétny typ pohybu, napríklad chôdzu [18] alebo oscilačný pohyb [27].

2.4. Výber prostredia a systémov

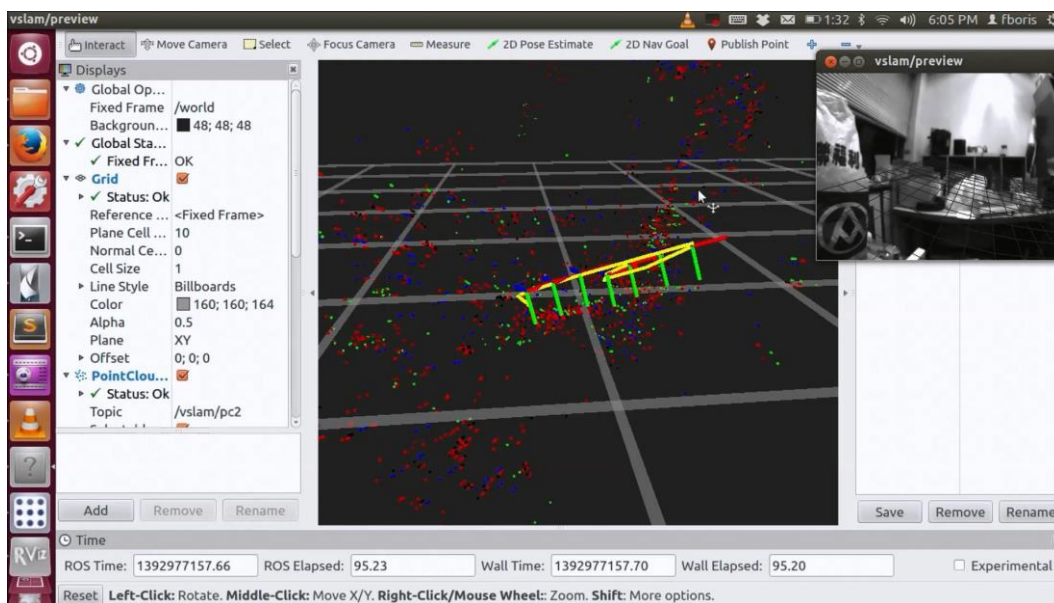
Po podrobnom rozbere dostupných SLAM riešení pre nami zvolené snímače je potrebné zvoliť systémy, prostredia a programovacie jazyky ktoré treba použiť a takisto je vhodné sa zamyslieť do akej miery je potrebné riešenia preberať a do akej miery implementovať vlastné. Pre ohromnú rozsiahlosť a náročnosť témy je dôraz kladený na použitie už dostupných riešení, ich úpravu a zameranie sa na koncept a vytvorenie funkčného príkladu. Ako prostredie a operačný systém je zvolený operačný systém Linux a ako podprostredie a hlavné jadro štruktúry je zvolený robotický operačný systém ROS. V akademickej sfére, ako aj vývojárskej, kde je množstvo takýchto prístupov vyvíjaných a implementovaných sa práve táto kombinácia používa najčastejšie a preto je veľmi vhodné zvoliť si práve túto kombináciu. Je treba povedať že problém SLAM je vlastne problém veľmi často spájaný s oblasťou robotického videnia a aj to je jedným z hlavných dôvodov a opodstatnení tohto výberu.

Keďže je ROS v informatickej sfére nepoznaný systém, je vhodné systém popísať a preto je v nasledujúcej kapitole popísaný. Ešte pred popisom je vhodné povedať, že pre implementáciu algoritmu riešiaceho inerčný SLAM je zvolený program MATLAB ktorý je pre spracovanie takého to typu dát veľmi vhodný, a pri spojzdení je jednoduché experimentovať s rôznymi postupmi algoritmami pre spracovanie inerčných dát.

2.4.1. Robotický operačný systém ROS

Robotic Operating System (ROS), teda robotický operačný systém je veľmi rozšírený a používaný systém v robotike ktorý sa skladá z ohromného množstva knižníc, nástrojov a podprogramov. Je to voľne dostupný softvér ktorý je možné stiahnuť na internetových stránkach [9]. Táto rozsiahla štruktúra obsahuje vlastné kanály, správy, štruktúry, komunikáciu a je naozaj mohutná, preto sa aj nazýva operačný systém. Obsahuje množstvo podprogramov ako je napr.

Rviz pre zobrazovanie a manipuláciu s dátovým mračnom, vizualizáciu pohybu robota, vykresľovanie 3D priestoru atď.. Systém podporuje spoluprácu a komunikáciu s mnohými známymi aplikáciami či knižnicami ako je napr. OpenCV, Matlab a Ogre3D. K vlastným štruktúram patrí napríklad rosnode čo predstavuje bežiaci podprogram, rostopic čo predstavuje kanál cez ktorý podprogram vysiela dáta, rosmmsg čo predstavuje formát správy ktorú podprogram vysiela atď.. Takisto je pre ROS dostupné veľké množstvo akademických projektov ktoré sú priamo udržiavané a podporované pre ROS v akomkoľvek smere vo forme pluginov a softvérov či už ide o oblasť robotického videnia alebo čisto matematické algoritmy pracujúce nad súborom dát.



Obr. 9 – Ukážka prostredia pluginu Rviz v operačnom systéme ROS.

2.4.2. Rviz a tf

Keďže Rviz predstavuje modul v ktorom je možné zobrazovať rôzne prvky, je treba zdefinovať vzťah medzi Rviz a prvkom tf. Prvok tf predstavuje koordinačný rámec ktorý sa môže nachádzať v strome štruktúry iných koordinačných rámcov. Teda pre správne pochopenie je dobré predstaviť si koordinačný rámec svet, ktorý je počiatkom stromu. V tomto rámci sa môže nachádzať koordinačný rámec lietadlo. No a v koordinačnom rámci lietadlo sa môže nachádzať koordinačný rámec pilot atď.. Fungovanie tejto štruktúry je opísané v publikácii [32]. Takýmto spôsobom je možné teda v prostredí ROS počítať transformácie medzi rôznymi rámcami. Pre výpočet transformácie medzi koordinačnými rámcami c a a , teda transformáciu T_a^c , medzi ktorými je v strome koordinačný rámec b je:

$$T_a^c = T_a^b * T_b^c$$

Pre lepšie pochopenie je možné predstaviť si a ako svet, b ako lietadlo a c ako pilota.

Každý tf koordinačný rámeček sa skladá z polohy a orientácie opísanými vektorom súradníc a quaternionom vzhľadom na materský koordinačný systém, teda:

$$T_a^b = [p_a^b, Q_a^b]$$

kde p_a^b predstavuje polohu tf koordinačného rámca b v koordinačnom systéme tf koordinačného rámca a a Q_a^b predstavuje orientáciu alebo quaternion koordinačného rámca b v koordinačnom systéme tf koordinačného rámca a . Poloha p predstavuje vektor s tromi údajmi a to polohou v každej z osí súradnicového systému, teda X , Y a Z .

2.5. Vybrané algoritmy pre optický a inerčný SLAM

Zariadenia, metódy, prostredia, systémy a spôsob SLAM je vybraný. Ďalším krokom ktorý je potrebné urobiť, je vhodne zvoliť a vybrať už konkrétne algoritmy SLAM ktoré je potrebné použiť, testovať a upravovať. Pre dobrý prínos práce a hlavne správne a reálne funkčný systém je treba povedať, že najvhodnejšie je vyberať iba state-of-the-art algoritmy, čiže algoritmy ktoré sú vo svojej oblasti najlepšie a takisto je podmienkou otvorenosť kódu. Pri hľadaní najlepšieho algoritmu by bolo najvhodnejšie porovnanie obrovského množstva algoritmov nad súborom rovnakých vstupných dát a porovnanie výsledkov.

Práve takýmto objektívnym porovnávaním algoritmov a prístupov v oblasti robotického videnia sa zaoberá Inštitút technológií v Karlsruhe [11]. Vytvorený porovnávací systém [10] poskytuje ohromné množstvo dát z obrovského množstva snímačov pre použitie v ľubovoľnom vyvinutom systéme v jednej z oblastí stereometrie, odometrie, rozoznávania objektov a mnoho ďalších, a výsledok je zaradení do tabuľky medzi výsledky iných systémov. Takýmto spôsobom je teda vytvorené poradie systémov s podrobnými výsledkami a takisto aj základným popisom kde patrí aj informácia o otvorenosti kódu. Je treba ale povedať že systém neponúka porovnávanie inerčných SLAM algoritmov a pre použitie v práci je vhodný pre vyberanie algoritmov iba pre vizuálny SLAM.

Po podrobnom preštudovaní týchto voľne dostupných algoritmov v oblasti vizuálneho SLAM pri zohľadnení nutnosti funkčnosti v prostredí ROS, sú vybrané algoritmy S-PTAM [13] a S-LSD-SLAM [12]. Algoritmus S-PTAM je stereokulárny a je postavený a založený na veľmi známom pôvodnom algoritme PTAM [14] od autorov Klein a Murray ktorý je monokulárny. Existuje mnoho variácií a úprav tohto známeho algoritmu. Pre prínos práce je vhodné otestovať stereokulárne aj

monokulárne metódy a práve preto je použitá okrem S-PTAM aj variácia PTAM a to s názvom Ethzasl-PTAM [15,16] ktorá je monokulárna, vylepšená a upravená pre ROS.

Algoritmus S-LSD-SLAM je stereokulárny algoritmus no voľne dostupný kód je dostupný len k jeho originálnej monokulárnej verzii LSD-SLAM [17]. Pre optický SLAM sú teda vybrané tri metódy alebo algoritmy a to S-PTAM, Ethzasl-PTAM a LSD-SLAM.

Ďalším krokom je výber algoritmov pre inerčný SLAM. Je treba povedať že inerčný SLAM ako oblasť nie je veľmi zaužívaný pojem a v oveľa väčšej forme sa používa v spojení s vizuálnym SLAM ako vizuálne-inerčný SLAM. Používaný skôr ako dopomoc. Po podrobnom preštudovaní dostupných algoritmov je zvolená jednoduchá dvojitá integrácia akcelerácie čím je možné získať vektor zmeny polohy. Algoritmus používa prostredie Matlab ktoré je s ROS a Linuxom plne kompatibilné. Vybraný algoritmus pre inerčný SLAM je teda jednoduchá dvojitá integrácia akcelerácie.

Je treba povedať, že navrhnuté algoritmy je potrebné patrične upraviť, nastaviť, vyladiť a takisto nie je vybraný žiadny algoritmus pre fúziu týchto dát čo je teda potrebné navrhnúť a vyhotoviť tiež.

Nasleduje opis jednotlivých algoritmov.

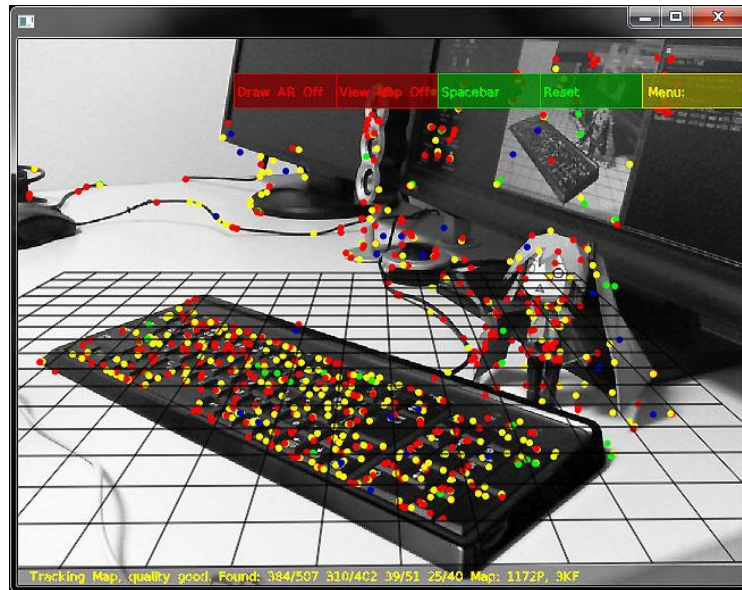
2.5.1. PTAM

Kedže oba algoritmy S-PTAM a Ethzasl-PTAM sú založené na pôvodnom PTAM algoritme, je vhodné opísať práve tento pôvodný algoritmus a neskôr zdefinovať v čom sa vybrané algoritmy od pôvodného líšia.

Algoritmus PTAM je monokulárny algoritmus riešiaci problém optického SLAM využívajúci body záujmu v obraze. Teda používa riedke sledovanie.

Metóda odhaduje pózu kamery a mapu prostredia v neznámej scéne. Mapovanie a sledovanie je rozdelené do separátnych vlákien čo umožňuje používať metódy BA v reálnom čase.

Všeobecne sa algoritmus správa tak, že vytvára bodovú mapu prostredia snímaného kamerou, teda formu riedkej monokulárnej rekonštrukcie, a takisto v reálnom čase zobrazuje kde sa kamera v tejto mape nachádza. Je treba povedať, že algoritmus je navrhnutý pre prácu v menších priestoroch a nie je vhodný pre prostredia s opakujúcou sa textúrou.



Obr. 10 – Bežiaci PTAM algoritmus.

Metódu je možné zhrnúť do nasledujúcich bodov:

1. Do algoritmu vstupujú dáta vo forme obrazov z kamery.
2. Sledovanie (tracking) a mapovanie sú rozdelené do separátnych vlákien ktoré sú vykonávané paralelne
3. mapovanie je založené na kľúčových rámcoch (keyframes) ktoré sú spracovávané pomocou BA,
4. pri inicializácii algoritmu je použitá inicializácia mapy prostredníctvom päť bodového algoritmu 5-Point Algorithm,
5. nové body sú inicializované pomocou epipolárneho vyhľadávania,
6. je možné mapovať tisíce bodov v reálnom čase.

Zotavenie z mapy je plusom pretože minimalizuje akumulovanú chybu. Algoritmus sa zotaví s malým skokom čo nemá za dôsledok zhoršený zážitok.

Metóda je všeobecne založená na postupnosti krokov. Prvým je inicializácia ktorá predstavuje inicializáciu pomocou päť bodového algoritmu pracujúceho s dvoma obrazmi 5-Point Algorithm [19] ktorým sú vytvorené prvé body v 3D mape. Následne sú nové body pridávané postupne do mapy pričom je celá množina spolu s kamerou v reálnom čase spracovávaná lokálnym alebo globálnym BA. Kamera si v mape uchováva pózy len v určitých úsekoch ktoré sa nazývajú kľúčové rámce (KR). Algoritmus obsahuje aj zotavenie sa po chybe, alebo lepšie povedané znovu nájdenie sa pri stratení. Je treba povedať že algoritmus je takto veľmi všeobecne opísaný a nasledujú podrobné opisy dielčích častí.

2.5.1.1. Mapa

Mapa je zostavená zo súboru bodov M v poli W nazývanej svet. Každý bod predstavuje lokálne plochú oblasť s textúrou. Bod p_j , kde j predstavuje poradie, je vo svete W definovaný ako $p_{jW} = (x_{jW} \ y_{jW} \ z_{jW} \ 1)^T$. Každý bod obsahuje takisto normálu oblasti n_j a referenciu na zdrojové pixely danej oblasti.

Mapa takisto obsahuje N kľúčových rámcov. Kľúčový rámec je snímka obrazu v určitý čas. Každý kľúčový rámec má priradený koordinačné pole so stredom určeným kamerou, označené K_i pre i -tý kľúčový rámec. Transformácia medzi týmto poľom a poľom sveta je potom E_{K_iW} . Každý kľúčový rámec obsahuje takisto štyri levely pyramídy čiernobieleho obrazu vo formáte 8bpp a to nasledovne. Nultý level pyramídy obsahuje obraz v plnom rozlíšení 640x480, každý ďalší level je zmenšený až na level tri v ktorom je rozlíšenie 80x60.

Pixely ktoré tvoria oblasť nie sú ukladané individuálne ale obsahujú údaj o KR v ktorom sa zobrazili ako prvé. To znamená, že každý bod mapy má uloženú referenciu na jediný KR, na jediný level pyramídy v tomto KR a na jedinou oblasť pixelov v tomto levelu pyramídy.

2.5.1.2. Sledovanie

Sledovanie je založené na sledovaní významných bodov a je predpoklad že v procese sledovania je už mapa inicializovaná takže sa v nej už nachádzajú body. Algoritmus používa FAST-10 body vyhľadávané na každom levely pyramídy bez použitia non-maximal supression čo má za výsledok body vo forme oblastí. Je možné algoritmus sledovania popísať v bodoch:

- Nový obraz je načítaný z kamery a k dispozícii je póza kamery založená na predchádzajúcom obraze,
- body mapy založené na predchádzajúcom obraze sú premietnuté na aktuálny obraz,
- v aktuálnom obraze je hľadané malé množstvo (50) hrubozrných bodov,
- póza kamery je aktualizovaná na základe týchto výsledkov,
- v aktuálnom obraze je premietnuté a hľadané veľké množstvo (1000) jemnozrných bodov,
- finálna póza je vypočítaná zo všetkých nájdených zhôd.

2.5.1.3. Póza kamery a premietanie

Aby bolo možné body premietnuť na rovinu obrazu, je potrebné ich transformáciou transformovať z poľa sveta do poľa stredu kamery C . Táto transformácia je tvorená ľavým násobením s maticou E_{CW} , ktorá reprezentuje pózu kamery,

$$p_{jC} = E_{CW} p_{jW} \quad (1)$$

Obrazec C vo svete W je zapísaný ako CW . Matica E_{CW} obsahuje rotačnú aj pozičnú zložku a patrí do Lieho algebry Lie $SE(3)$ kde sú 3D transformácie s pevným telesom.

Aby bolo možné body z obrazu kamery spracovať, kvôli skresleniu, je zadaný projekčný model $CamProj()$:

$$\begin{pmatrix} u_i \\ v_i \end{pmatrix} = CamProj(E_{CW} p_{iW}) \quad (2)$$

Model používa projekciu dierkovej komory ktorá podporuje barelové radiálne skreslenie. Model transformuje $r \rightarrow r'$. Je predpokladom že parametre ohnisková vzdialenosť (f_u, f_v) , p body (u_0, v_0) a skreslenie (ω) sú známe, potom:

$$CamProj \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} + \begin{bmatrix} f_u & 0 \\ 0 & f_v \end{bmatrix} \frac{r'}{r} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (3)$$

$$r = \sqrt{\frac{x^2 + y^2}{z^2}} \quad (4)$$

$$r' = \frac{1}{\omega} \arctan(2r \tan \frac{\omega}{2}) \quad (5)$$

Základnou požiadavkou pre správne sledovanie a mapovanie je rozlíšenie rovnice (2) od zmien v pozícii a natočení kamery E_{CW} čo predstavuje pózu kamery. Zmeny v póze kamery sú reprezentované ľavým násobením E_{CW} s maticou M o rozmeroch 4x4 predstavujúcu pohyb kamery:

$$E'_{CW} = ME_{CW} = \exp(\mu)E_{CW} \quad (6)$$

Pohyb kamery M je členom $SE(3)$ a je možné ho parametrizovať na minimum pomocou šesť členného μ vektora. Prvé tri členy vektora μ zvyčajne predstavujú transláciu (posun) a ďalšie tri predstavujú rotáciu a rozsah. Z rovnice (6) je možné triviálnym spôsobom získať $\frac{\partial u}{\partial \mu_i}, \frac{\partial v}{\partial \mu_i}$.

2.5.1.4. Vyhľadávanie v oblasti

Pri hľadaní každého bodu p v práve spracúvanom obraze je vyhľadávanie použité len v úzkom okolí alebo úzkej oblasti kde je predpoklad že sa bod bude nachádzať. Pred vyhľadávaním v tejto oblasti je potrebné oblasť posunúť a prispôbiť novej perspektíve ako aj skresleniu kamery. Teda:

$$A = \begin{bmatrix} \frac{\partial u_c}{\partial u_s} & \frac{\partial u_c}{\partial v_s} \\ \frac{\partial v_c}{\partial u_s} & \frac{\partial v_c}{\partial v_s} \end{bmatrix} \quad (7)$$

Množina $\{u_s, v_s\}$ predstavuje horizontálne a vertikálne presunutie pixelov v danom levely pyramídy danej oblasti daného obrazu a množina $\{u_c, v_c\}$ predstavuje rovnaké presunutie no v nultom levely pyramídy. Táto matica je vytvorená spätným premietaním presunutí v zdrojovom levely pyramídy kľúčového rámca na rovinu oblasti a opätovným premietaním týchto presunutí do aktuálneho obrazu. Determinant matice A predstavuje level pyramídy v ktorom je hľadanie uskutočnené. Tento determinant predstavuje takisto plochu vo forme pixel x pixel ktorý by jediný zdrojový pixel pokrýl v obraze v plnom rozlíšení; $\det(A) / 4$ je prislúchajúca plocha v pyramídovom levely jedna, atď.. Cieľový pyramídový level l je vybraný tak aby $\det(A) / 4^l$ sa blížilo k jednote, teda je vyhľadávaný level v ktorom sa oblasť najviac zhoduje s mierkou.

Šablóna oblasti vyhľadávania rozmerov 8×8 je vytvorená zo zdrojového levelu použitím skreslenia $A/2^l$ takisto bilineárnou interpoláciou. Priemerná intenzita pixelov je odčítaná z intenzít jednotlivých pixelov za účelom stabilnosti pri zmene osvetlenia. Ďalším krokom je nájdenie najlepšej zhody so šablónou ktorá je hľadaná vo fixne danom rozsahu okolo predpokladanej pozície v cieľovom pyramídovom levely. Toto nájdenie je tvorené vyhodnocovaním SSD jadier blížiacim sa nule nad súborom všetkých lokácií FAST bodov v kruhovom rozsahu a vyberaní lokácií s najmenším rozdielom vyhodnotenia. Ak vyhodnotenie je menšie ako nastavený prah, oblasť je považovaná za nájdenú.

Pre hľadšie sledovanie je vykonávaná takisto minimalizácia chyby podrobnejšie opísaná v samotnom [14].

2.5.1.5. Aktualizačný krok pózy

Súbor S úspešne ohodnotených oblastí je predpokladom aby bolo možné aktualizovať pózu kamery. Pozícia oblasti je daná ako $(\hat{u}\hat{v})^T$ v levely nula a šum je popísaný ako $\sigma^2 = 2^{2l}$ vynásobené maticou identity o rozmere 2×2 . Aktualizácia pózy kamery je počítaná iteračne ako minimalizácia robustnej funkcie projekčnej chyby:

$$\mu' = \underset{\mu}{\operatorname{argmin}} \sum_{j \in S} \operatorname{Obj} \left(\frac{|e_j|}{\sigma_j}, \sigma_T \right) \quad (8)$$

kde e_j je vektor reprojekčnej chyby:

$$e_j = \begin{pmatrix} \hat{u}_j \\ \hat{v}_j \end{pmatrix} - \operatorname{CamProj}(\exp(\mu) E_{CW} p_j) \quad (9)$$

$\operatorname{Obj}(\cdot, \sigma_T)$ je Tukeyova biváhová objektová funkcia a σ_T je robustný mediánový odhad odchýlky odvodený od všetkých zvyškov. Použitých je takisto desať iterácií metódy najmenších štvorcov aby bolo umožnené M-estimátoru konvergovať k správne odhadu.

2.5.1.6. Dvojfázové hrubé-na-jemné sledovanie

Za účelom zlepšenia stability pri rýchlom pohybe kamerou alebo rýchlejšej akcelerácií, vyhľadávanie oblastí a aktualizácia pózy kamery je urobená dva krát. Prvým krokom je hrubé hľadanie kde je vyhľadávaných 50 oblastí z najvyššieho levelu pyramídy a hľadaný kruhový okruh je veľký. Z týchto výsledkov je následne aktualizovaná póza kamery. Následne, vysoké množstvo oblastí, až do tisíc, je spätne premietnutých a hľadaných už ale v oveľa užšom kruhovom rozsahu. Vyhľadovanie na subpixeloch je povolené iba na najvyšších leveloch pyramídy. Finálna póza kamery je následne vypočítaná za pomoci hrubých (50) aj jemných (1000) výpočtov.

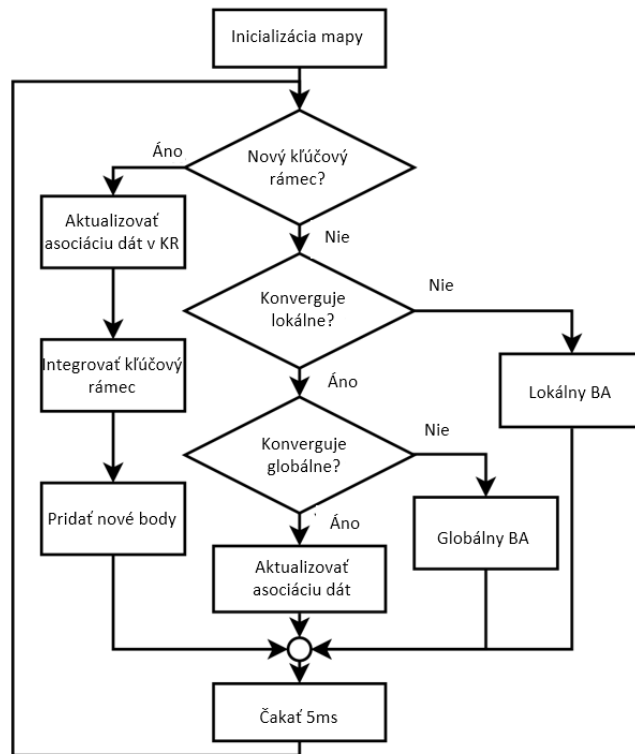
2.5.1.7. Kvalita sledovania a zotavenie po chybe

Kvalita sledovania je určená pomerom úspešne znovu nájdených bodov, teda bodov ktoré už sú nájdené a hľadajú sa v aktuálnom obraze. Ak pomer klesne pod určitý prah, sledovanie prejde do stavu, kedy nevytvára nové kľúčové rámce pretože existuje predpoklad, že obraz je rozmazaný alebo nesprávny.

Ak pomer klesne ešte pod nižší prah, sledovanie je považované za stratené a algoritmus prejde do stavu zotavovania sa po chybe. Viac o zotavovaní sa v [23].

2.5.1.8. Mapovanie

Mapovanie je proces vytvárania 3D bodovej mapy. Vytváranie mapy je použité dva krát. Prvý krát pri inicializácii algoritmu, teda na začiatku, kedy je inicializovaná mapa a následne opakovane vždy keď je pridaný nový kľúčový rámec (KR). Schéma mapovacieho vlákna je znázornená nižšie na Obr. 9.



Obr. 11 – Schéma mapovacieho vlákna algoritmu PTAM. Prebrané z [14].

2.5.1.9. Inicializácia mapy

Na inicializáciu mapy je použitý päť bodový algoritmus ktorý ale vyžaduje spoluprácu používateľa pre správnu inicializáciu. Je potrebné aby užívateľ spustil tlačidlom algoritmus a následne pohybom, plynule prešiel o určitú vzdialenosť, vhodne 10 cm a následne opäť stlačil tlačidlo. Týmto spôsobom sú vytvorené dva kľúčové rámce kde je z vyhľadávania tisícky bodov nájdená korešpondencia a následne vytvorená matica pomocou algoritmu RANSAC ktorá je následne trianguláciou pretvorená na prvotnú inicializačnú mapu. Tá je ešte na konci inicializácia vyhladená pomocou BA.

Vygenerovaná mapa má ale ľubovoľnú mierku, čo je známy problém monokulárnych techník, a preto je mierka daná predpokladom že pri inicializácii bola kamera premiestnená o spomenutých 10 cm. Podstava mapy, alebo jej stred a orientácia je daná plochou ktorá je pomocou RANSAC hypotéz vytvorená tak, aby pretínala čím väčšie množstvo nasnímaných bodov pri inicializácii.

2.5.1.10. Nový kľúčový rámec a epipolárne hľadane

Mapa obsahuje po inicializácii dva kľúčové rámce. Postupným prehliadaním scény kamerou sú postupne pridávané nové kľúčové rámce a body mapy aby bolo umožnené mape rásť. Kľúčové rámce sú pridávané len za určitých podmienok a to aby bolo sledovanie v úspešnom stave, aby bol súčasný obraz vzdialený aspoň 20 obrazov od posledného obrazu kedy bol pridaný KR, a takisto aby bola kamera v určitej minimálnej vzdialenosti od posledného kľúčového bodu. Podmienky sú nastavené tak aby boli KR redšie pridávané vo väčšej scéne a hustejšie pri približovaní sa detailom.

Obraz je spracúvaný vláknom sledovania. Prv sú v obraze nájdené body záujmu FAST v každom levely pyramídy. Následné sú použité algoritmy non-maximal suppression a filter prahu založený na algoritme Shi-Tomasi aby bolo možné nájsť najviac vyhovujúce, teda najviac charakteristické body v každom levely pyramídy. Body nájdené v blízkosti oblastí ktoré už boli nájdené sú odhodnené. Všetky ostatné body sú považované za kandidátov na pridanie do mapy.

Body sú nájdené no nie je možné priradiť hĺbkovú informáciu z len jedného KR. Preto je vybraný najbližší ďalší KR vďaka ktorému je možné pomocou epipolárneho vyhľadávania a triangulácie určiť bodovú hĺbku. Pri tomto hľadaní sa používajú porovnanie s SSD jadrami blížiacimi sa nule. Zhody sú hľadané len v rovnakom levely pyramídy a treba povedať že epipolárna priamka nie je nekonečná ale je použitá hypotéza o hĺbke bodu.

2.5.1.11. Bundle adjustment

S každým i -tým kľúčovým rámcom je pridružený súbor meraní S_i . Pre znázornenie, j -tý bod v mape v i -tom kľúčovom rámci by bol opísaný ako $(\hat{u}_{ji} \ \hat{v}_{ji})^T$ s odchýlkou σ_{ji} pixelov. Daným stavom mapy $\{E_{K_1W}, \dots, E_{K_NW}\}$ a $\{p_1, \dots, p_M\}$, každé meranie nad obrazom má takisto priradenú reprojekčnú chybu e_{ji} vypočítanú podľa (9). BA v každej iterácii upravuje mapu a to minimalizáciou robustnej funkcie:

$$\{\{\mu_2 \dots \mu_N\}, \{p'_1 \dots p'_M\}\} = \underset{\{\{\mu\}, \{p\}\}}{\operatorname{argmin}} \sum_{i=1}^N \sum_{j \in S_i} \operatorname{Obj} \left(\frac{|e_{ji}|}{\sigma_{ji}}, \sigma_T \right) \quad (10)$$

Opísaný BA je takmer úplne zhodný s implementáciou Levenberg-Marquardt s pridaným Tukeyovým M-estimátorom. Uvedeným BA sú upravené všetky kľúčové rámce okrem prvého ktorý je pevný a takisto sú upravené všetky pozície bodov v mape. Tento globálny BA je v takomto prípade kvadraticky náročný problém a pri vzrastajúcom počte KR vzrastá prudko aj potrebný

výpočtový čas. Kvôli tomu je implementovaný takisto lokálny BA ktorý upravuje iba určitý počet kľúčových rámcov X , v ďalšom kroku viac kľúčových rámcov Y a určitý počet bodov mapy Z .

$$\{\{\mu_x \in X\}, \{p'_z \in Z\}\} = \operatorname{argmin}_{\{\{\mu\}, \{p\}\}} \sum_{i \in X \cup Y} \sum_{j \in Z \cap S_i} \operatorname{Obj}(i, j) \quad (11)$$

Množina X predstavuje aktuálny KR a takisto niekoľko najbližších KR v blízkosti aktuálneho. Množina Z predstavuje všetky body ktoré rámce z množiny X môžu vidieť a množina Y predstavuje všetky ostatné rámce ktoré body Z dokážu vidieť. To znamená že lokálny BA upravuje pózu aktuálneho KR a jeho najbližších susedov ako aj všetky body nimi videné používajúc všetky výpočty vypočítané nad týmito bodmi od vzniku každého bodu.

Lokálny BA sa používa vždy keď je pridaný nový KR, prebiehajúci globálny BA je pozastavený až po dokončenie lokálneho pre možnosť kontinuálneho skúmania mapy.

2.5.1.12. Odlišnosti v S-PTAM a Ethzasl-PTAM

Ethzasl-PTAM je vylepšený PTAM implementovaný pre prácu v ROS. Vylepšený je vo viacerých smeroch ale hlavne vylepšený pre lepšiu prácu v prostrediach s opakujúcou sa textúrou pomocou výmeny algoritmu pre hľadanie bodov FAST za AGAST a zmenou pôsobenia na iných leveloch pyramíd. Sú pridané viaceré rozšírené možnosti ako maximálny limit KR, automatická inicializácia po chybe, zmena vnútornej štruktúry filtrovania bodov atď., ktoré okrem iného docieľujú beh aj na mobilných platformách čo je veľký bonus pretože mnoho HMD používa ako hlavné zariadenie v dnešnej dobe smartfón.

S-PTAM je takisto vylepšený PTAM určený pre stereokulárny SLAM. Algoritmus nie je úpravou ale je postavený od základov celkovo konceptuálne ale napodobujúci PTAM. Ako algoritmus pre hľadanie bodov v obraze je použitý Shi-Tomasi a ako deskriptor je použitý BRIEF. Je treba ale upozorniť, že aj napriek skvelým výsledkom tohto algoritmu je algoritmus pomerne nový a nie je známe ako veľmi je stabilný a robustný a aké môže byť jeho správanie v daných reálnych podmienkach.

2.5.2. LSD-SLAM

LSD-SLAM je pokračovaním pôvodného prvého obdobného prístupu SD-VO [21]. LSD-SLAM je oproti pôvodnému prístupu obohatený o vytváranie mapy ktorá je v pôvodnom SD-VO nevytváraná celkovo, len dočasne pre aktuálny obraz. Keďže pre využitie v práci sú dôležitejšie

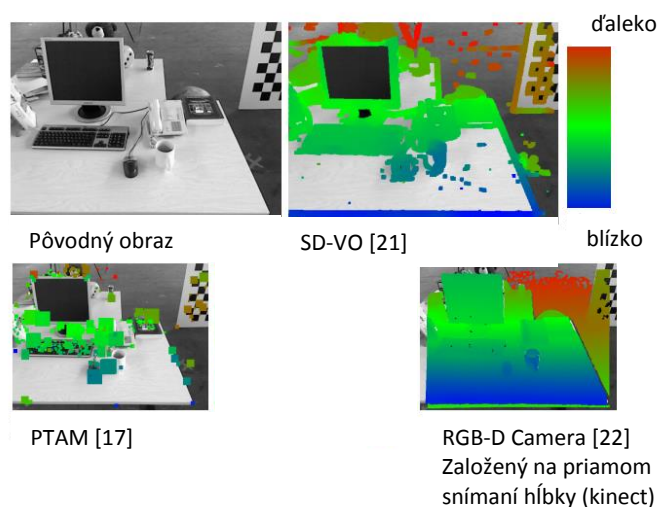
poznatky z pôvodnej SD-SLAM, je opísaná práve táto prvá obdoba metódy. Takisto je vhodné opísať čím je použitá LSD-SLAM obohatená, čo je uvedené po opise SD-VO.



Obr. 12 – Príklad zmapovaného prostredia pomocou LSD-SLAM aj s pozami a trajektóriou kamery.

SD-VO znamená Semi-Dense Visual Odometry. Tento prístup predstavuje monokulárnu vizuálnu odometriu v reálnom čase. V tomto prístupe nebol ešte použitý pojem SLAM pretože metóda nevytvára celkovú mapu, teda nemapuje prostredie čo je podmienkou SLAM a preto je označovaná iba ako vizuálna odometria. SD-VO patrí k metódam hustého sledovania. Metóda je založená na počítaní polo-hustej (semi-dense) invertovanej hĺbkovej mapy, teda formy polo hustej monokulárnej rekonštrukcie, aktuálneho obrazu ktorá je ďalej použitá na počítanie pózy kamery pomocou zarovnanania hustého obrazu (dense image alignment). Vyhodnocované sú pixely ktoré majú nejaký gradient. Každý výpočet je reprezentovaný ako Gaussovská distribúcia pravdepodobnosti nad invertovanou hĺbkovou mapou.

Základom metódy je maximalizácia využitia informácií poskytnutých v obraze.



Obr. 13 – Porovnanie množstva využitých informácií z obrazu. Prevzaté z SD-VO [21].

Ako je vidieť, husté sledovanie, použité v SD-VO, využíva omnoho viac informácií ako sledovanie bodov záujmu PTAM no stále menej informácií ako hĺbkové kamery RGB-D a práve preto sa nazýva polo-husté.

Vypočítaná hĺbková mapa je šírená z obrazu na obraz a aktualizovaná pomocou porovnávania dvoch obrazov. Vhodný referenčný obraz je vybraný pre každý pixel na základe poznania informácie o hĺbke bodu a na rovnakom základe je aj limitovaný rozsah hľadanie disparity.

2.5.2.1. Polo-hustý odhad hĺbkovej mapy

V tejto sekcii je opísaný základný princíp a myšlienka algoritmu a to odhad a výpočet polo-hustej invertovanej hĺbkovej mapy pre aktuálny obraz z ktorej je možné ďalej odhadnúť pózu kamery. Hĺbková mapa je počítaná s každým novým obrazom a vyhladzovaná s každým novým stereo výpočtom hĺbky založenom na stereo porovnaní každého pixelu s adaptívnou vzdialenosťou porovnávaných obrazov (adaptive baseline) čo umožňuje odhad hĺbky blízkych aj vzdialených pixelov. Každý pixel udržiava práve jednu invertovanú hĺbkovú informáciu reprezentovanú gaussovskou distribúciou pravdepodobnosti.

Je potrebné povedať že d označuje invertovanú hĺbku pixelu.

2.5.2.2. Aktualizácia hĺbkovej mapy na základe sterea

Metóda využíva prístup založený na pravdepodobnosti pre volenie vzdialenosti porovnaných obrazov. Celková aktualizácia hĺbkovej mapy, ktorá je vykonávaná vždy keď príde nový obraz, pozostáva z nasledujúcich krokov:

1. Zvolené sú podmnožiny pixelov s vysokou presnosťou vyhľadania disparity.
2. Tri kritéria sú ďalej použité na tieto podmnožiny opísané v kapitole 2.5.2.4. Metóda párovania stereo obrazov.
3. Vhodný referenčný obraz je potom vybraný pre každý pixel a takisto je vykonané jednorozmerné vyhľadávanie disparity.

Získaný invertovaný hĺbkový odhad je potom spojený do hĺbkovej mapy fúziou.

2.5.2.3. Výber referenčného obrazu

Referenčný obraz je vybraný tak, aby najlepšie spĺňal požiadavky a to maximalizáciu stereo presnosti pri udržiavaní rozsahu hľadania disparity a udržiavanie malého uhlu pohľadu. Keďže presnosť sterea je závislá na viacerých faktoroch a výber obrazu je separátny proces pre

každý pixel je potrebné zdefinovať pravidlá výberu. Pre pixel je použitý najposlednejší obraz kedy rozsah hľadania disparity a uhol pohľadu spadal pod nejakú určenú hranicu. Ak je hľadanie disparity neúspešné, „vek“ pixelu je navýšený tak aby následné hľadania disparity používali nové obrazy kde je predpoklad a možnosť že daný pixel je ešte stále možné vidieť.

2.5.2.4. Metóda párovania stereo obrazov

Pre nájdenie zhody je použitá metóda epipolárneho vyhľadávania rovnakej intenzity až do hĺbky subpixelov. Ak je predom k dispozícii hypotéza invertovanej hĺbky, interval vyhľadávania je limitovaný na $d \pm 2\sigma_d$ kde d a σ_d znamenajú štandardnú odchýlku hypotézy inak je prehľadávaný celý rozsah disparity.

Použitá je SSD chyba naprieč piatimi rovnako vzdialenými bodmi na epipolárnej priamke čo vedie k veľkej robustnosti v oblastiach s vysokou frekvenciou pixelov a takisto je zachovaná jednorozmernosť tohto vyhľadávania. Je treba povedať že je takisto výpočtovo efektívna keďže štyri z piatich interpolovaných hodnôt obrazu je možné znovu použiť ďalšom výpočte SSD.

2.5.2.5. Vyhodnotenie neurčitosti

V sekcii vyhodnotenie neurčitosti (uncertainty estimation) je propagáciou neurčitosti odvodený výraz pre vyjadrenie chybovej odchýlky σ_d^2 v invertovanej hĺbke d . Všeobecne je možné výraz odvodiť vyjadrením optimálnej invertovanej hĺbky d^* ako funkcie vstupov s určitým šumom. Obrazy sú označené ako I_0, I_1 , relatívna orientácia ako ξ a kalibrácia kamery ako projekcia funkcie π

$$d^* = d(I_0, I_1, \xi, \pi) \quad (1)$$

Chybová odchýlka pre d^* je potom

$$\sigma_d^2 = J_d \sum J_d^T \quad (2)$$

kde J_d predstavuje Jacobian z d a \sum predstavuje kovarianciu vstupnej chyby. Pre jednoduchosť analýza je vykonaná iba na základe bodov nie oblastí vyhľadáváním rovnakej intenzity na epipolárnej priamke.

Za týmto účelom je výpočet rozdelený do troch krokov:

1. Epipolárna priamka v referenčnom obraze je vypočítaná.
2. Najlepšia zhodná pozícia $\lambda^* \in \mathbb{R}$ naprieč priamkou je zistená.

3. Invertovaná hĺbka d^* je vypočítaná z disparity λ^* .

Prvé dva kroky zahrňujú dva nezávislé zdroje chyby a to geometrickú chybu, ktorá vzniká zo šumu na ξ a π a objavuje sa v kroku jeden a fotometrickú chybu ktorá vzniká zo šumu na obrazoch I_0 a I_1 a objavuje sa v kroku dva. V tretom kroku sú tieto chyby zvýraznené alebo utlmené v závislosti na vzdialenosti obrazov (baseline).

2.5.2.6. Vyhodnotenie neurčitosti – chyba geometrickej disparity

Táto kapitola sa viaže k prvému kroku. Geometrická chyba ε_λ na disparite λ^* je spôsobená šumom ξ a π . Je použitá aproximácia:

Nech je zamýšľaný úsek epipolárnej priamky $L \subset \mathbb{R}^2$ definovaný ako

$$L := \left\{ l_0 + \begin{pmatrix} l_x \\ l_y \end{pmatrix} \mid \lambda \in S \right\} \quad (3)$$

kde λ je disparita v intervale hľadania S , ďalej $(l_x, l_y)^T$ je normalizovaný smer epipolárnej priamky a l_0 bod reprezentujúci nekonečnú hĺbku. Po tomto predpoklade takisto existuje predpoklad že iba absolútna pozícia tohto úseku priamky, teda l_0 je ovplyvnený izotropným Gaussovským šumom ε_l . Keďže v reálnej prevádzke sú úseky priamky zvyčajne malé, aproximácia poskytuje stabilitu pri rotácii.

Pozičná chyba ε_l na epipolárnej priamke má za následok malú disparitnú chybu ε_λ ak je epipolárna priamka paralelná s gradientom obrazu a to hlavne veľkého. Obraz spôsobuje že optimála disparita λ^* je položená na určitej izočiare s rovnakou intenzitou. Táto izočiar je aproximovaná tak aby bola lokálne lineárna teda aby bol gradient lokálne konštantný:

$$l_0 + \lambda^* \begin{pmatrix} l_x \\ l_y \end{pmatrix} \stackrel{!}{=} g_0 + \gamma \begin{pmatrix} -g_x \\ g_y \end{pmatrix}, \quad \gamma \in \mathbb{R} \quad (4)$$

kde $g := (g_x, g_y)$ je gradient obrazu a g_0 bod na izočiare. Pôsobenie šumu na hodnoty obrazu sú odvodené v ďalšej sekcii a preto sú body g a g_0 v tejto sekcii považované za bezšumové.

Vyjadrovanie λ tvorí disparitu λ^* s pôsobením šumového vstupu obrazu l_0 :

$$\lambda^*(l_0) = \frac{\langle g, g_0 - l_0 \rangle}{\langle g, l \rangle} \quad (5)$$

Analogicky ako v (2), odchýlku geometrickej chyby disparity je možné vyjadriť:

$$\sigma_{\lambda(\xi, \pi)}^2 = J_{\lambda^*(l_0)} \begin{pmatrix} \sigma_\xi^2 & 0 \\ 0 & \sigma_\pi^2 \end{pmatrix} J_{\lambda^*(l_0)}^T = \frac{\sigma_l^2}{\langle g, l \rangle^2} \quad (6)$$

kde g je normalizovaný gradient obrazu, l je normalizovaný smer epipolárnej priamky a σ_l^2 je odchýlka ε_l . Je treba povedať že tento šum je šumom z relatívnej orientácie kamery ξ a kalibrácie kamery π , teda je nezávislý od šumu z intenzity obrazu.

2.5.2.1. Vyhodnotenie neurčitosti – Fotometrická chyba disparity

Táto kapitola sa viaže k druhému kroku. Malé chyby v intenzite obrazu majú veľký dopad na vypočítanú disparitu ak je gradient obrazu malý a malý efekt ak je gradient obrazu veľký. Matematicky je tento vzťah vyjadrený nasledovne, a to tak, že hľadaným členom je disparita λ^* ktorá minimalizuje rozdiel medzi intenzitami:

$$\lambda^* = \min_{\lambda} (i_{ref} - I_p(\lambda))^2 \quad (7)$$

kde i_{ref} je referenčná intenzita a $I_p(\lambda)$ je intenzita obrazu na epipolárnej priamke pri disparite λ . Predpokladom je dobrá inicializácia λ_0 ktorá je výsledkom vyhľadávania. Ďalej používajúc Taylorov rád prvého stupňa pre I_p :

$$\lambda^*(I) = \lambda_0 + (i_{ref} - I_p(\lambda_0))g_p^{-1} \quad (8)$$

kde g_p je gradient I_p ktorý predstavuje gradient obrazu naprieč epipolárnou priamkou. Pre ujasnenie je treba povedať, že je uvažovaný iba šum i_{ref} a $I_p(\lambda_0)$. Rovnaký výsledok je možné dosiahnuť všeobecne uvažujúc šum vytvárajúci sa hodnotách obrazu zahrnutých vo výpočte g_p .

Odchýlka fotometrickej chyby disparity je teda:

$$\sigma_{\lambda(I)}^2 = J_{\lambda^*(I)} \begin{pmatrix} \sigma_i^2 & 0 \\ 0 & \sigma_i^2 \end{pmatrix} J_{\lambda^*(I)} = \frac{2\sigma_i^2}{g_p^2} \quad (9)$$

kde σ_i^2 je odchýlka šumu intezity obrazu. Táto chyba je závislá iba na šume z hodnôt intenzity obrazu a je úplne nezávislá na geometrickej chybe disparity.

2.5.2.2. Konverzia pixel na invertovanú hĺbku

Pri malých rotáciách kamery je invertovaná hĺbka d približne proporčná k disparite λ a pohľadová odchýlka invertovanej hĺbky $\sigma_{d,obs}^2$ je potom:

$$\sigma_{d,obs}^2 = \alpha^2 (\sigma_{\lambda(\xi,\pi)}^2 + \sigma_{\lambda(I)}^2) \quad (10)$$

kde konštanta proporcionality α je rozdielna pre každý pixel a predstavuje:

$$\alpha := \frac{\delta_d}{\delta_\lambda} \quad (11)$$

kde δ_d je dĺžka prehľadavaného intervalu invertovanej hĺbky a δ_λ je dĺžka prehľadávaného rozsahu epipolárnej priamky. Konštanta α je inverzne lineárna v dĺžke pohybu kamery a takisto aj závisí na smere pohybu kamery a lokácii pixelov v obraze.

Pri používaní chyby SSD nad viacerými bodmi naprieč epipolárnou priamkou je vhodné určiť hornú hranicu neurčitosti zhody:

$$\sigma_{\tilde{d},obs-SSD}^2 \leq \alpha^2 (\min\{\sigma_{\lambda(\xi,\pi)}^2\} + \min\{\sigma_{\lambda(l)}^2\}) \quad (12)$$

kde min prechádza všetkými bodmi zahrnutými v chybe SSD.

2.5.2.3. Fúzia pre vyhodnotenie hĺbky

Po získaní hĺbky pre pixel je inicializovaná ak daný pixel nemá žiadnu hypotézu. Inak ak hypotéza existuje, obe distribúcie sú vynásobené čo predstavuje aktualizáciu krok v Kalmanovom filtri.

Daná je už existujúca distribúcia $N(d_p, \sigma_p^2)$ a pohľadový šum $N(d_o, \sigma_o^2)$, potom výsledná hĺbka je

$$N = \left(\frac{\sigma_p^2 d_o + \sigma_o^2 d_p}{\sigma_p^2 + \sigma_o^2}, \frac{\sigma_p^2 \sigma_o^2}{\sigma_p^2 + \sigma_o^2} \right) \quad (13)$$

2.5.2.4. Propagácia hĺbkovej mapy

Po aktualizácii pózy kamery je d_0 premietnutý do nového obrazu ako d_1 , potom:

$$d_1(d_0) = (d_0^{-1} - t_z)^{-1} \quad (14)$$

kde t_z je pohyb kamery v smere pohľadu kamery. Odchýlka d_1 je potom:

$$\sigma_{d_1}^2 = J_{d_1} \sigma_{d_0}^2 J_{d_1}^T + \sigma_p^2 = \left(\frac{d_1}{d_0}\right)^4 \sigma_{d_0}^2 + \sigma_p^2 \quad (15)$$

kde σ_p^2 je predpoklad neurčitosti ktorý priamo predstavuje aktualizáciu krok Kalmanovho filtra.

Ak sú pre pixel dostupné dve hypotézy a sú štatisticky podobné, teda v hraniciach 2σ , tak je s nimi nakladané ako so separátnymi hĺbkami a ak sú veľmi odlišné, sú zahodené.

2.5.2.5. Regularizácia hĺbkovej mapy

Po tom čo pixely obsahujú svoju hĺbkovú mapu, je táto mapa aproximovaná hĺbkovými mapami okolitých bodov. Ak je skos medzi susednými bodmi príliš veľký (nad 2σ) tak sa body neovplyvňujú aby bola zachovaná ostrosť.

Algoritmus obsahuje takisto zaobchádzanie s výchyškami – teda pixelmi s nesprávnou hypotézou. Podľa pravdepodobnosti správnosti hypotézy je buď pravdepodobnosť správnosti zvýšená keď je hypotéza v novom obraze nájdená a potvrdená alebo znížená ak nájdená nie je prípadne ak sa veľmi líši okolitým hypotézam.

2.5.2.6. Husté sledovanie

Husté sledovanie (dense tracking) je založené na priamej minimalizácii fotometrickej chyby:

$$r_i(\xi) := (I_2(\omega(x_i, d_i, \xi)) - I_1(x_i))^2 \quad (16)$$

kde funkcia $\omega: \Omega_1 \times \mathbb{R} \times \mathbb{R}^6 \rightarrow \Omega_2$ mapuje každý bod $x_i \in \Omega_1$ v referenčnom obraze I_1 do príslušného bodu $\omega(x_i, d_i, \xi) \in \Omega_2$ v novom obraze I_2 . Ako vstup je potrebná 3D póza kamery $\xi \in \mathbb{R}^6$ a odhadovaná invertovaná hĺbka $d_i \in \mathbb{R}$ pre pixel v I_1 . Pre zvýšenie stability pri pohybujúcich sa objektoch a zmene svetla je pridaná váhová schéma (weighting scheme). Takisto je pridaná odchýlka invertovanej hĺbky σ_d^2 pre zvýšenie stability algoritmu v začiatkoch kedy je inicializovaný. Výsledná minimalizovaná energie teda predstavuje:

$$E(\xi) := \sum_i \frac{\alpha(r_i(\xi))}{\sigma_{d_i}^2} r_i(\xi) \quad (17)$$

kde $\alpha: \mathbb{R} \rightarrow \mathbb{R}$ definuje váhu pre daný zvyšok. Minimalizáciu tejto chyby je možné interpretovať ako počítanie maximálneho odhadu pravdepodobnosti pre ξ predpokladajúc nezávislé šum na hodnotách intenzity obrazu. Výsledný váhový problém najmenších štvorcov je vyriešený efektívne používajúc iteratívne počítaný Gussov-Newtonov algoritmus spojený s hrubý-na-jemný (coarse-to-fine) prístupom.

2.5.2.7. Odlišnosť s LSD-SLAM

Metóda LSD-SLAM používa jadro z SD-VO no je obohatená o tvorbu veľkej mapy preskúmaného prostredia, teda je ju už možné nazývať SLAM. Mapovanie je založené na algoritme Pose-Graph. Takisto je implementovaný algoritmus pre správne určovanie mierky v rámci mapy, keď sa kamera vzdáľuje od detailného pohľadu po viac obšírny pomocou určovania podobnosti obrazov pomocou priameho odhadu transformácie podobnosti $\xi \in sim(3)$

a pravdepodobnostne konzistentnej integrácii neurčitosti hĺbkovej mapy do procesu sledovania. Tento algoritmus pre určovanie mierky mapy nehovorí o mierke v rámci reálneho sveta, iba v rámci mapy samotnej a jej častí, teda všeobecný problém mierky pri monokulárnych metódach nerieši.

2.5.3. Algoritmus spracovania údajov z inerčných snímačov

Ako algoritmus spracovania dát z inerčných snímačov je zvolená jednoduchá dvojité integrácia akcelerácie čím je získaný polohový vektor.

$$\Delta p = \Delta a * \Delta t^2$$

teda zmena vektora polohy Δp je rovná zmene vektora akcelerácie Δa za štvorec zmeny času Δt^2 .

2.6. Algoritmy pred-filtrácie

Vstupným súborom algoritmov optického SLAM je množina obrazov, teda živé video. Algoritmy pracujú nad súborom týchto dát a práve preto je veľmi vhodné otestovanie rôznych predfiltrácií tohto videa. Táto téma je veľmi zaujímavá a preto je vhodné otestovať algoritmy pre spracovanie obrazu ktoré by mohli napomôcť.

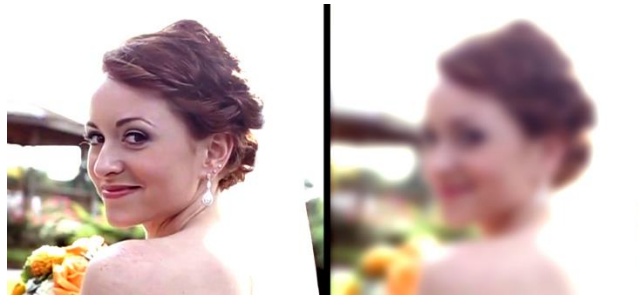
Po úvahách boli zvolené algoritmy opísané v nasledujúcich kapitolách pričom je dané, pri každom algoritme, aký prínos je možné očakávať aj s odôvodnením prečo.

2.6.1. Gaussovský filter

Aplikáciou Gaussovského filtra na maticu obrazu je výsledkom rozostrenie obrazu kde každý bod je aproximovaný okolitými bodmi do určitej vzdialenosti. Je to veľmi jednoduchý spôsob filtrácie obrazu ktorý bol zvolený pre preskúmanie užitočnosti predspracovania obrazu z dôvodu odstránenia nadbytočného šumu, a zameranie sa na prominentejšie a výraznejšie oblasti. Pre 2D obraz platí:

$$G_0(x, y) = A e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2} - \frac{(y-\mu_y)^2}{2\sigma_y^2}}$$

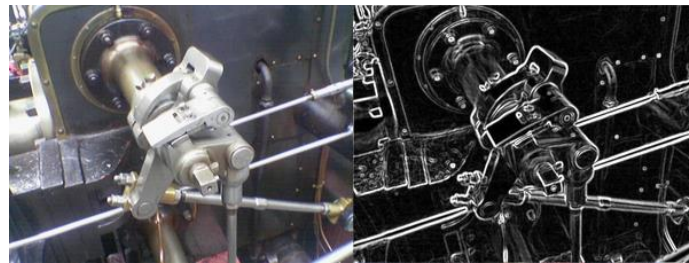
kde μ predstavuje vrchol a σ predstavuje odchýlku pri neznámych x a y .



Obr. 14 – Aplikácia Gaussovho filtra na 2D obraz. Pred a Po.

2.6.2. Sobelové derivácie

Aplikáciou sobelových derivácií na maticu obrazu je v končnom dôsledku zobrazenie hrán v obraze.



Obr. 15 – Aplikácia Sobelových derivácií na 2D obraz. Pred a Po.

Ako je vidieť na Obr. 15, Sobelové derivácie umožňujú vďaka derivácii nad maticou zobrazovať hrany veľmi dobre, v takej forme ktorá je konzistentná a stabilná. Derivácie boli zvolené pre predpoklad výpomoci algoritmu zameriavať sa na oblasti s najviac cennými informáciami, teda na oblasti s hranami. Po úvahe, že aj človek rozpoznáva prostredie hlavne vďaka hranám je vyskúšanie takejto filtrácie viac než dôležité. Je treba povedať, že pri Sobelových deriváciách je bežné použitie Gaussovho filtra ešte pred samotnými deriváciami.

2.6.3. Laplacov operátor

Laplacov operátor je zvolený z rovnakého dôvodu možnosti celkovej výpomoci ako Sobelové derivácie. Operátor je vlastne pokračovaním Sobelových derivácií a to tak, že je celá matica zderivovaná znova.

3. Dodefinovanie, testovanie a výber už konkrétnych zariadení

Všeobecný výber zariadení je definovaný. Ako hlavné zariadenie je použitá dátová helma pre virtuálnu realitu obsahujúca inerčný snímač a na nej pripevnená kamera. Pred ďalším postupom je potrebné tieto zariadenia dodefinovať, otestovať a vybrať už konkrétne zariadenia pre použitie.

3.1. Výber dátovej helmy pre virtuálnu realitu a jej vlastnosti

Dostupnosť dátových prilb je ešte obmedzená a nie je dostupných viacero možností pre testovanie. Z dôvodov dostupnosti, obľúbenosti a všeobecnej veľkej podpore je zvolená dátová prilba Oculus Rift DK2 ktorá obsahuje inerčný snímač Invensense MPU-6500 [25]. Tento inerčný snímač obsahuje trojosí gyroskop a takisto trojosí akcelerometer na technológii MEMS.

K ďalšiemu prvku balíka Oculus Rift DK2 patrí kamera pre pozičné sledovanie ktorá dokáže veľmi presne a jemne vykazovať polohu prilby v priestore. Táto kamera aj s už implementovaným algoritmom predstavuje dokonale fungujúci SLAM ktorý je pre túto prácu príkladný, no je to SLAM založený na referenčnom bode, ktorým je kamera a rozsah fungovania je veľmi úzky a to 0,5 až 2,5 metra v priestore pred kamerou.

Je veľmi dôležité povedať, že Oculus Rift DK2 ako taký, ma skvelé API s možnosťou získavania rôznych surových dát až po dáta spracované a to hlavne:

- Surové dáta z IMU
 - Uhlová rýchlosť (gyroskop)
 - Zrýchlenie (akcelerometer)
 - Magnetické pole (magnetometer)
- Spracované dáta
 - Orientácia
 - Poloha (iba s kamerou)
 - Rýchlosť (iba s kamerou)
- Atd.
- Rôzne meta dáta

Z dát spracovávaných je veľmi dôležité podotknúť, že API poskytuje veľmi presnú a správne vypočítanú orientáciu alebo natočenie prilby v priestore, čo je veľmi vhodné využiť. Vypísané dáta nie sú všetky ktorými disponuje API, no sú absolútne najhlavnejším prienikom.

3.2. Výber monokulárnej kamery

Výber kamery je veľmi dôležitý a podstatný krok. Keďže kamera poskytuje vstupný súbor dát pre optický SLAM, je správny výber kamier kľúčový. Existuje veľké množstvo dostupných kamier s rôznymi vlastnosťami a preto je vhodné sa zamerať na vlastnosti kamery odporúčané autormi využívaných algoritmov a teda zamerať sa na odporúčania uvedené v [26] ktoré opisujú vhodné vlastnosti kamery:

- Globálna uzávierka. Rolovacia uzávierka zhoršuje výsledky,
- široké zorné pole, aspoň 130 stupňov,
- vysoká snímkovacia rýchlosť, aspoň 30 snímkov za sekundu,
- odporúčané rozlíšenie 640x480.

Ďalším bodom, ktorý nie je uvedený, no z celkového prehľadu je zjavný a logický je používanie bezstratovej kompresie prenosu dát, teda využívanie obrazu v surovom obrazovom formáte YUYV na čo je teda naviazané aj použité rozhranie keďže je potrebné aby pri danom rozlíšení a formáte bol poskytnutý dostatočne rýchly prenos dát. Takisto je vhodné povedať, že nevyhnutnou vlastnosťou je takisto možnosť živého vysielania, teda priameho prenosu dát pre spracovanie do počítača namiesto ukladania dát na médium. Všeobecne žiadanými vlastnosťami sú čím lepšie obrazové vlastnosti, teda kvalita obrazu, dynamický rozsah, vysoká citlivosť pri slabom osvetlení atď.. Je treba povedať, že veľmi žiadanou vlastnosťou je čím širšia hĺbka ostrosti za účelom obsiahnutia čím viac informácií bez rozostrenia.

Kamery s vybranými vlastnosťami sú bohužiaľ dosť nedostupné a aj po viac týždňovej komunikácii s viacerými dovozcami týchto kamier nebolo možné takúto kameru zapožičať alebo zohnať na dostatočne dlhé obdobie. Hlavnými vlastnosťami ktoré takéto profesionálne kamery oproti bežne dostupným disponujú je globálna clona a priame živé vysielanie cez vysoko rýchlostné rozhranie ako je napríklad USB 3.0.

Výber je teda zúžený na bežne dostupné kamery a pre správny výber boli otestované tri kamery, a to Genius WideCam F100, Logitech Webcam C930e a SJCAM SJ4000. Všetky tri kamery disponujú funkciou živého vysielania, rozhraním USB 2.0 a snímkovacou frekvenciou aspoň 30

snímkou za sekundu. Pre porovnanie kamier je uvedená tabuľka vlastností a takisto obraz vyhotovený každou z kamier. Na konci porovnania je uvedený konkrétny výber.



Obr. 16 - Obraz z kamery Genius WideCam F100.



Obr. 17 – Obraz z kamery Logitech Webcam C930e.



Obr. 18 - Obraz z kamery SJCAM SJ4000.

Nasleduje tabuľka porovnania relevantných vlastností kamery. Obraz kamery je posúdený vizuálne. Pri poli „podpora YUYV“ je takisto uvedené maximálne rozlíšenie pri ktorom je možné prenos uskutočňovať vzhľadom na prenosovú rýchlosť rozhrania USB 2.0 a požiadavku snímkovacej frekvencie aspoň 30 snímkou za sekundu. Hĺbka ostrosti je posúdená z reálneho testu a zo šírky šošovky.

Kamera/Parameter	Genius WideCam F100	Logitech Webcam C930e	SJCAM SJ4000
Kvalita obrazu	Veľmi slabá	Skvelá	Skvelá
Šírka zorného poľa (°)	120	90	170
Dostupné rozlíšenia / snímková frekvencia pri prenose stratovou kompresiou	1920x1080 / 30fps 1280x720 / 30fps	1920x1080 / 30fps 1280x720 / 30fps	1280x720 / 30fps 640x480 / 60fps
Podpora YUYV	Nie	Áno, 640x480	Nie
Globálna clona	Nie	Nie	Nie
Hĺbka ostrosti	Stredná	Stredná	Široká
Zaostrovanie	Manuálne	Automatické	Pevné/Nezmeniteľné

Tab. 1 – Porovnanie vlastností kamier.

Po podrobnom otestovaní kamier je zvolená kamera SJCAM SJ4000. Žiadna z kamier nedisponuje globálnou clonou. Genius Widecam je nepoužiteľná kvôli veľmi zlej kvalite obrazu. Finálne rozhodnutie nastáva medzi Logitech Webcam a SJCAM. Výhodou Logitech je podpora YUYV no obrovskou nevýhodou veľmi úzke zorné pole. Práve to je dôvodom voľby SJCAM, ktorá disponuje veľmi širokým zorným poľom, skvelou kvalitou obrazu a širokou hĺbkou ostrosti s pevným zaostrením čo je výhodou. Zvolená kamera nepodporuje bezstratový prenos dát a preto je použitý jediný kamerou podporovaný a veľmi rozšírený stratový prenos obrazových dát formátom JPEG.



Obr. 19 – Spojenie Oculus Rift DK2 a SJCAM SJ400.

3.3. Výber stereokulárnej kamery

Keďže je použitie stereokulárnej kamery a stereokulárneho SLAM algoritmu druhoradé, je použitá školou dostupná stereokulárna kamera SONY Handycam HDR-TD30VE. Táto kamera nedisponuje funkciou živého vysielania, disponuje takisto veľmi úzkym zorným poľom a pomalším automatickým zaostrovaním s stredne širokou hĺbkou ostrosti.

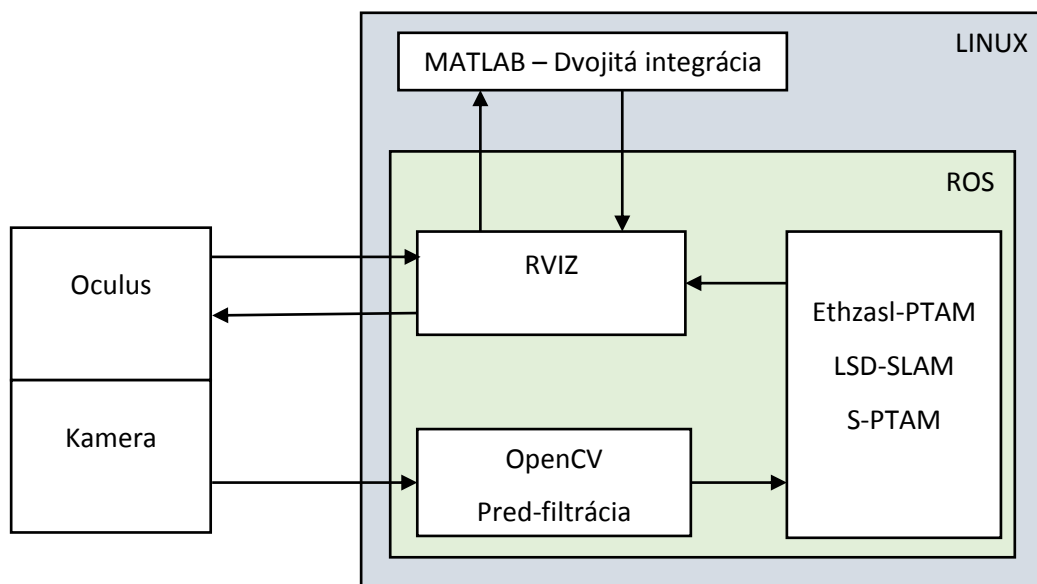
Z týchto dôvodov nie je algoritmus možné testovať v reálnom čase a je otestovaný iba formou natočenia materiálu či videa a následným spustením algoritmov na týchto materiáloch až po samotnom natáčaní.

4. Návrh, implementácia a testovanie podčastí systému

Keďže všetky zariadenia, algoritmy, rozhrania sú zvolené a skonkretizované, je vhodné pristúpiť k samotnému návrhu a implementácii systému. Tento návrh a implementácie predstavuje spojenie a zlepenie vybraných a navrhnutých dielov a postupov do väčšieho celku.

4.1. Návrh

V nasledujúcej schéme je zobrazený schematický návrh celého systému skladajúci sa z podčastí.



Obr. 20 – Schéma celého systému.

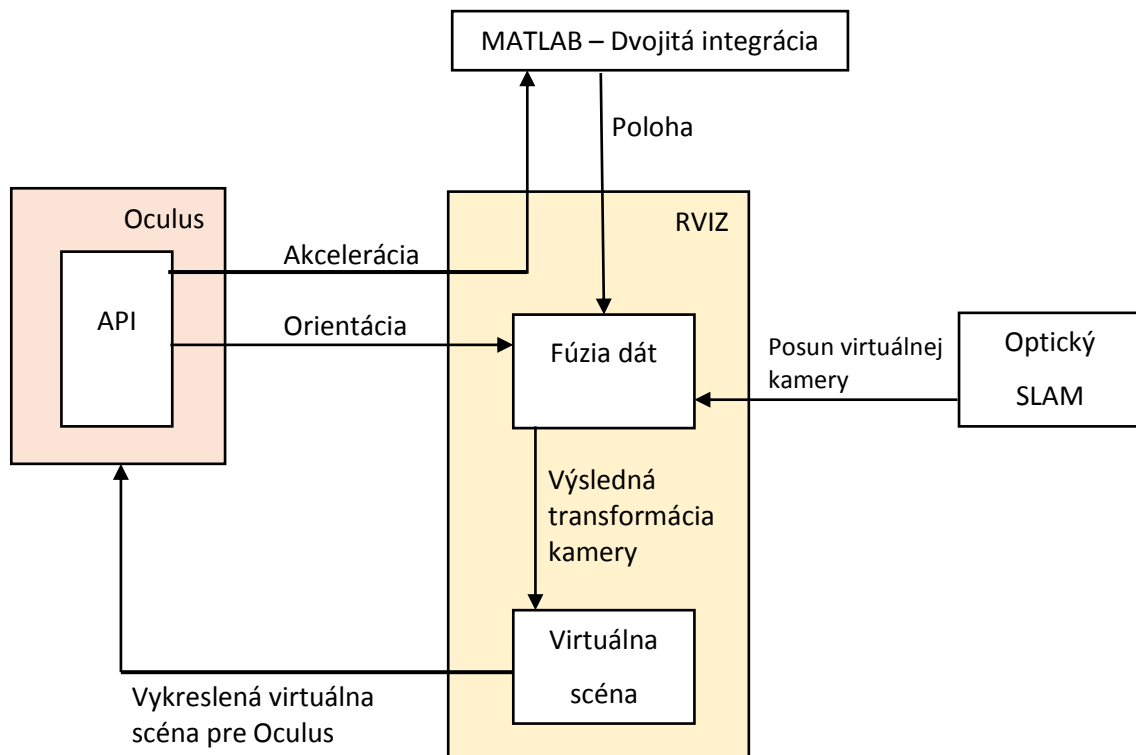
V schéme sú zmiešané zariadenia s rozhraniami a algoritmi z dôvodu správnej predstavy fungovania systému. Je zjavné, že šípky predstavujú toky dát. Pre správne pochopenie schémy, tok dát začína v kamere. Obraz z kamery je vstupom predfiltrácie kde je obraz patrične upravený. Predfiltrácia je opísaná v kapitole 4.4. Implementácia predfiltrácie obrazu z kamery. Ďalej z predfiltrácie putuje upravený obraz ako vstup do jedného z algoritmov optického SLAM odkiaľ už nie je výstupom obraz, ale pohyb kamery vo virtuálnom priestore, vo forme tf koordinačného rámca ktorý vstupuje do podmodulu Rviz. Rviz predstavuje podmodul ROS pre zobrazovanie virtuálnej scény, pre vizualizáciu, zobrazovanie rôznych iných prvkov, o ktorom je viac napísané aj s obrázkom v kapitole 2.4.1. Robotický operačný systém ROS a v 2.4.2. Rviz a tf.

V rovnakom čase sú okrem virtuálnej pozície kamery vstupom podmodulu Rviz aj dáta získané prostredníctvom API z Oculusu. Z dát Oculusu sú ďalej vybrané potrebné inerčné dáta a posielané do Matlabu na spracovanie odkiaľ putujú naspäť do Rviz už spracované inerčné dáta. Rviz teda predstavuje pomyselné jadro v ktorom sa všetky dáta stretávajú a nastáva fúzia dát.

Je veľmi dôležité povedať, že Oculus ako taký disponuje už spomínaným veľmi presným vykazovaním orientácie. Z toho dôvodu, je pri fúzii pre transformáciu kamery použitá práve táto orientácia vo forme tf koordinačného rámca a z optického SLAM je odzrkadlený iba pohyb kamery v smere svojho vlastného súradnicového systému. Fúzia dát z Oculusu, v konečnom dôsledku orientácie, a dát z virtuálnej pozície kamery algoritmu optického SLAM, teda v konečnom dôsledku posunu virtuálnej kamery, je opísaná v kapitole 5.1. Návrh a implementácia fúzie dát.

Výsledná transformácia kamery je v poslednom kroku premietnutá do virtuálnej scény v Rviz a obraz vykreslený v Oculusu.

Pre lepšie pochopenie toku dát v okolí modulu Rviz, teda nejakého pomyselného jadra, je opísaná nasledujúca schéma.



Obr. 21 - Schéma toku dát v okolí podmodulu Rviz.

Schéma opäť mieša rôzne formy zariadení a tokov dát pre správnu predstavu fungovania tokov dát a celkovej komunikácie. Modul API je iba abstraktný a je umiestnený v module Oculus len pre ilustráciu a lepšiu predstavu. Nad šípkami je uvedený druh posielaných dát.

4.2. Komunikácia medzi monokulárnou kamerou a ROS

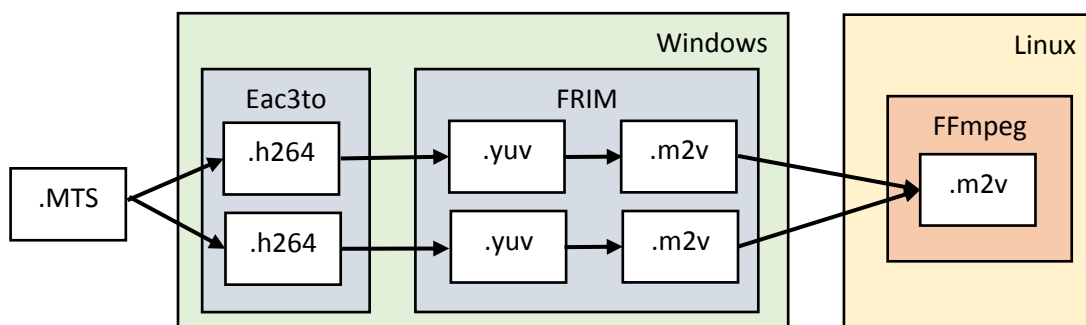
Pre spozajzdnenie komunikácie medzi monokulárnou kamerou a ROS je treba spozajzdnenie živého prenosu a vysielanie tohto prenosu priamo do prostredia ROS prostredníctvom špecifického kanála určeného pre obrazové dáta. Za týmto účelom bolo otestovaných množstvo ovládačov a použitý je ovládač `usb_cam` z knižnice softvérov ROS.

Pri prehrávaní už zhotoveného videa z tejto kamery je použitý modul pre prehrávanie `gscam` z knižnice softvérov ROS ktorý bolo takisto nutné správne vyladiť a spozajzdniť.

4.3. Komunikácia medzi stereokulárnou kamerou a ROS

Ako bolo spomenuté, použitá stereokulárna kamera nepodporuje živé vysielanie a preto je potrebné implementovať spôsob ako z uložených médií sprostredkovať dáta do prostredia ROS v správnej forme. Kamera SONY zhotovuje záznam vo formáte `.MTS` ktorý obsahuje dva prúdy dát a to ľavý a pravý obrazový tok dát. Tieto prúdy je potrebné správne prehrať a pustiť v obrazových kanáloch prostredia ROS. Tieto dva kanály je potrebné takisto správne časovo synchronizovať a poskytnúť potrebné dáta.

Za týmto účelom je navrhnutá nasledujúca viackroková procedúra pretože uvedené prehrávanie je veľmi zložitá a náročná požiadavka.



Obr. 22 – Viackroková procedúra konvertovania 3D záznamu na použiteľný formát.

Viacroková procedúra v sebe zahŕňa použitie dvoch operačných systémov, a to Windows a Linux, a využitie celkom troch voľne dostupných knižníc, a to Eac3to, FRIM a FFmpeg. MTS súbor je prv rozdelený na ľavý a pravý tok dát v pôvodnom stratovom kódovaní pomocou Eac3to, kde sú

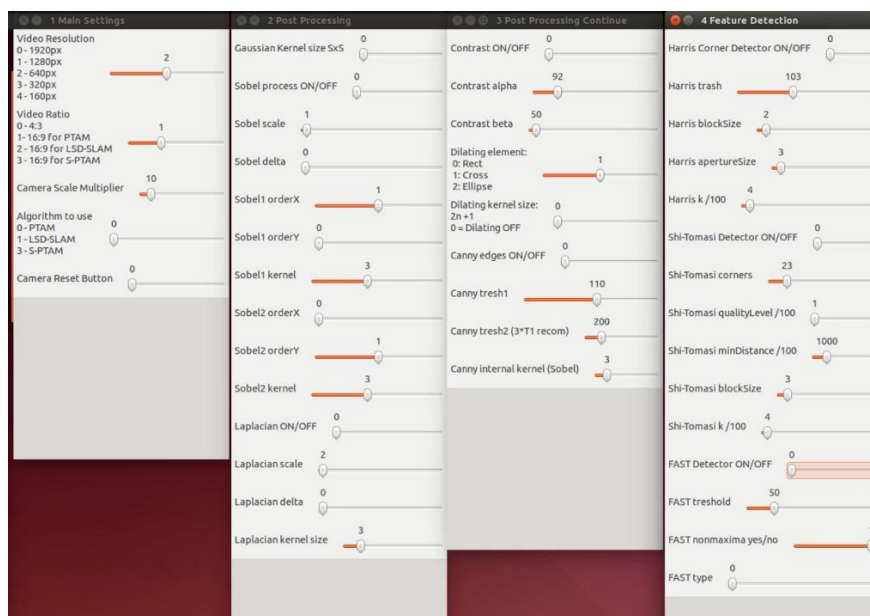
výsledné toky vo formáte .h264. Následne sú tieto toky opäť konvertované na súbory .yuv bez prekódovania kde zaberajú ohromné množstvo priestoru. Kvôli tomu sú následne konvertované stratovo na súbory formátu .m2v pomocou kódovania mpeg2. Pre posledné dve konverzie je použitá knižnica FRIM. Výsledné súbory sú následne spojené do jedného kde sa v jednom obraze nachádzajú oba, jeden vedľa druhého. K tejto konverzii je použitá knižnica FFmpeg ktorá je následne použitá aj k uskutočneniu posledného kroku, a to odstráneniu prekladania. K viacerým krokom bolo potrebné správne nastavenie kvality kódovania a konvertovania pre zachovanie všetkých informácií poskytnutých v pôvodnom súbore. Celý proces je zrýchlený a zautomatizovaný pomocou dvoch naprogramovaných skriptov ktoré dokážu spracovať naraz aj veľký počet vstupných súborov. Použité nastavenia kódovania a konvertovania ako aj skripty sú opísané v užívateľskej a systémov príručke.

Výsledný .m2v súbor je možné prehrávať pomocou už spomenutého modulu gscam rovnako ako v prípade monokulárnej kamery.

4.4. Implementácia predfiltrácie obrazu z kamery

Po prijatí vstupného toku obrazu zo živého vysielania alebo z prehrávania video súboru je potrebné dáta správne modifikovať a filtrovať. Predfiltrácia obrazu nie je iba aplikovanie filtrov ale celková príprava obrazového toku pre spracovanie. Táto príprava zahŕňa zmenu rozlíšenia, pomeru strán či zadanie či je vstup monokulárny alebo stereokulárny.

Za cieľom splnenia týchto požiadaviek je implementovaný predfiltráčný systém zobrazený na obrázku.



Obrázok 23 – Predfiltráčný systém.

Zobrazený systém ponúka funkcie modifikácie a filtrácie obrazu:

- Základné modifikačné nástroje
 - Zmena rozlíšenia
 - Zmena pomeru strán
- Nástroje pre neskoršie ovládanie algoritmov
 - Mierka kamery
 - Výber algoritmu
 - Tlačidlo pre zresetovanie pozície kamery
- Filtre
 - Gaussovský filter
 - Sobelové derivácie
 - Laplacianov filter
 - Kontrast
 - Dilatácia
 - Cannyho hrany
- Detektory bodov
 - Harrisov detektor hrán
 - Detektor hrán Shi-Tomasi
 - Detektor hrán FAST

Systém je implementovaný v prostredí ROS využívajúc knižnicu OpenCV. Je treba povedať že systém ponúka širšie spektrum filtrov ako bolo uvedené v syntéze za cieľom testovania a zvedavosti. Takisto sú implementované rôzne detektory bodov a hrán za cieľom demonštrácie a živého príkladu ako dobre dané detektory pracujú. Všetky funkcie sú implementované za cieľom hľadania nových možností a správneho zoznamovania sa s tematikou. Nástroje pre neskoršie ovládanie algoritmov predstavujú sadu nástrojov ktorými sú ovládané niektoré vlastnosti celého systému a práca sa na ne odvoláva v neskorších kapitolách.

4.4.1. Interpretácia stereo kanálov v ROS

Neviditeľnou súčasťou predfiltrácie je takisto implementácia ďalšieho spracovania obrazu pre prípad spracovania stereo dát. Monokulárne dáta nie je potrebné špeciálne spracovať no to isté neplatí pre dáta stereokulárne. Keďže je kvôli absencii živého vysielania stereokulárnej kamery využívané prehrávanie z video súboru .m2v kde sú obe obrázky vedľa seba, je potrebné rozdelenie tohto videa na dve obrazové kanály v prostredí ROS a takisto publikácia informačných

kanálov pre tieto obrazové toky. Rozdelenie obrazu na dve obrazové toky je implementované priamo v predfiltrácii.

Pre potrebu publikácie informačných kanálov je implementovaný systém s názvom `image_transport_tutorial`. Systém poskytuje možnosť publikácie informačných kanálov pre oba obrazové toky pričom je načítaný kalibračný súbor pre každú kameru a takisto je pridaná pre oba informačné kanály rovnaká časová stopa čo zabezpečuje synchronnosť a celkovo systém zabezpečuje správnu funkcionality a využitie stereo prehrávania v prostredí ROS. Systém je podrobnejšie opísaný v užívateľskej a systémov príručke.

4.5. Výber metód testovania algoritmov a zhotovovanie materiálov

Pred prístupím k testovaniu algoritmov optického SLAM je vhodné opísať metodológiu testovania ako aj zobraziť schematické zobrazenie.

Aby bolo testovanie objektívne, je potrebné aby vstupnými dátami boli už zhotovené súbory a nie živé vysielanie. Ďalšou požiadavkou je zahrnúť množstvo rôznych prípadov ktoré môžu reálne nastať, a to od prípadu ideálneho pre každý algoritmus, cez prípady pohľadu do oblohy, odlesku zrkadla, jednoliatej textúry, veľkej zmeny expozície, tieňov, príliš opakujúcej sa textúry až po rýchle pohyby či rotačné pohyby bez posunu. Tieto prípady nadväzujú na správne pochopenie funkcie algoritmov a spôsobu akým fungujú a aké sú ich hraničné podmienky.

Dôležitým atribútom testovania je odchýlka a presnosť a preto každý testovací súbor začína na rovnakom mieste a takisto na rovnakom mieste končí. Pre objektívne posúdenie je takisto potrebné aby boli mono aj stereo materiály totožné a preto je zhotovená konštrukcia a vybavenie zobrazené na nasledujúcom obrázku.

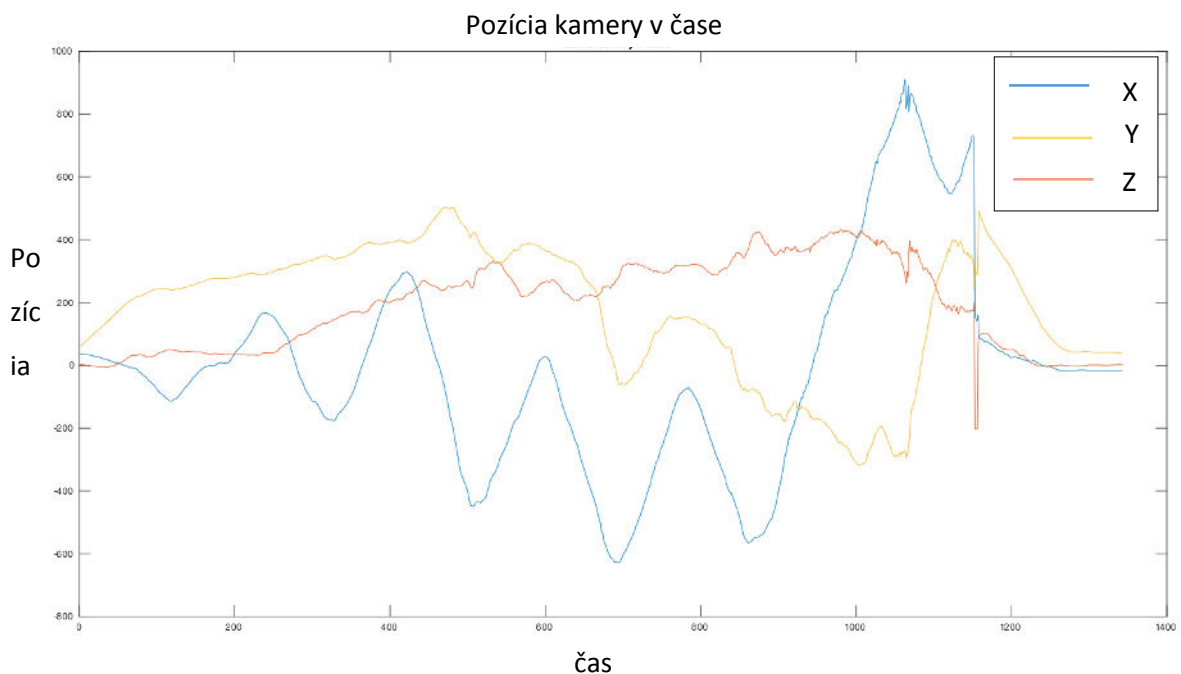


Obr. 24 – Konštrukcia sprostredkujúca simultánne natáčanie pomocou monokulárnej aj stereokulárnej kamery.

Pre testovanie sú zvolené nasledujúce atribúty:

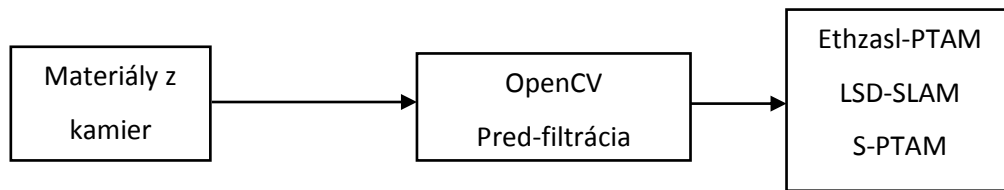
- Vizualne posúdenie,
- odchýlka konečnej pozície od začiatkovej,
- vypočítaná pozícia virtuálnej kamery XYZ formou tabuľky,
- graf priebehu XYZ atribútov.

Rotácia vypočítanej virtuálnej kamery nie je zaznamenávaná keďže orientácia kamery v celkovom systéme je získaná z Oculusu. Je treba povedať, že testovanie je prevažne vizuálne kde je hlavný posudok trochu viac subjektívny pretože nie je k dispozícii reálna trajektória kamery, teda nie je možné vyhotoviť matematicky presné posúdenie trajektórie, a hlavné posudzovanie vyplýva zo správneho uváženia chovania sa algoritmov pri sledovaní okien s prebiehajúcim algoritmom. Toto správanie je možné zopakovať spustením algoritmov nad súborom videí znova a bližší popis a postup je dostupný v užívateľskej a systémovej príručke. Ku každému testu sú takisto vyhotovené viaceré screenshoty z priebehu a všetky materiály sú k dispozícii na prílohovom DVD. Je vhodné poznamenať že pád algoritmu je v grafoch znázornený prepádom no nie rovnou čiarou po prepade, keďže padnutý algoritmus nevykazuje žiadne hodnoty, takže je graf v stave čakania nových hodnôt ktoré po zotavení môžu ďalej prichádzať.



Obr. 25 – Príklad grafu z testu Test 1_3_PTAM. Pád algoritmu v čase 1160 s následnou obnovou po určitej dobe čakania ktorá nie je nijako znázornená.

Schéma testovania je nasledujúca:



Obr. 26 – Schéma testovania.

4.6. Testovanie a nastavenie Ethzasl-PTAM

Prvým algoritmom pre testovanie je algoritmus Ethzasl-PTAM. Je vhodné povedať, že daný algoritmus, čo vyplýva zo syntézy a štúdia, pracuje veľmi presne hlavne pri posuvnom pohybe a je náchylný k rozpadu pri čisto rotačných pohyboch. Vyžaduje veľkú mieru pohybu a obsahuje veľmi robustný systém znovu nájdenia sa. Použitá dáta pochádzajú z monokulárnej kamery keďže je tento algoritmus monokulárny.

4.6.1. Kalibrácia kamery

Pred samotnými testami je potrebná správna kalibrácia kamery. Za týmto účelom je kamera kalibrovaná pomocou šachovnice pomocou Ethzasl-PTAM kalibračného nástroja [28].

4.6.2. Testovanie bez filtrácie

Prvým krokom testovania je testovanie bez použitia filtrácie. Hlavným cieľom tohto testovania je prv vhodné nastavenie záznamových vlastností kamery ako ostrosť, kvalita videa a ako druhý krok, vyladenie algoritmu k najlepšej funkcionalite. Pre testovanie bez filtrácie je vyhotovené video ktoré odzrkadľuje dokonalý prípad pre tento algoritmus, teda prípad kde je posun kamery veľký, rotácia minimálna a priestor predstavuje pomerne malá miestnosť s veľkým množstvom záchytných bodov. Všetky testy menom korešpondujú so štruktúrou zložiek na priloženom DVD a obsahujú takisto použité videá.


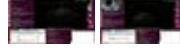

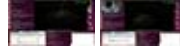

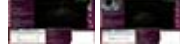






Obr. 27 – Náhľad obrázkov z testovacieho videa Test_1_Monocular.mov z testu Test 1.

Ak nie je dané inak, pri všetkých testoch sú použité východzie nastavenia algoritmov, kamier, predfiltrácie a spracovanie algoritmi je funkčné v reálnom čase so snímkovacou frekvenciou aspoň 25 snímkou za sekundu.

4.6.2.1. Test 1

V nasledujúcom teste sú porovnané rôzne nastavenia záznamu monokulárnej kamery. Označenie testu predstavuje pomenovanie testu ako aj formu štruktúry pomenovania zložiek na priloženom prílohovom DVD.

Označenie	Nastavenia kamery			Graf	Screenshoty	Odchýlka (dielikov)
	HDR	Kvalita	Ostrosť			
Test 1_1_PTAM	Off	Normal	Normal			8
Test 1_2_PTAM	On	Normal	Normal			7
Test 1_3_PTAM	Off	Fine	Normal			0
Test 1_4_PTAM	Off	Fine	Soft			-
Test 1_5_PTAM	Off	Normal	Soft			-

Tab. 2 – Tabuľka k Test 1_PTAM.

Test 1_1_PTAM – Počas celého priebehu jemné drmácanie kamery, čo je odzrkadlené aj na grafe ktorý nie je hladký. Algoritmus celkovo veľmi stabilný úplne bez pádu. Veľmi dobré.

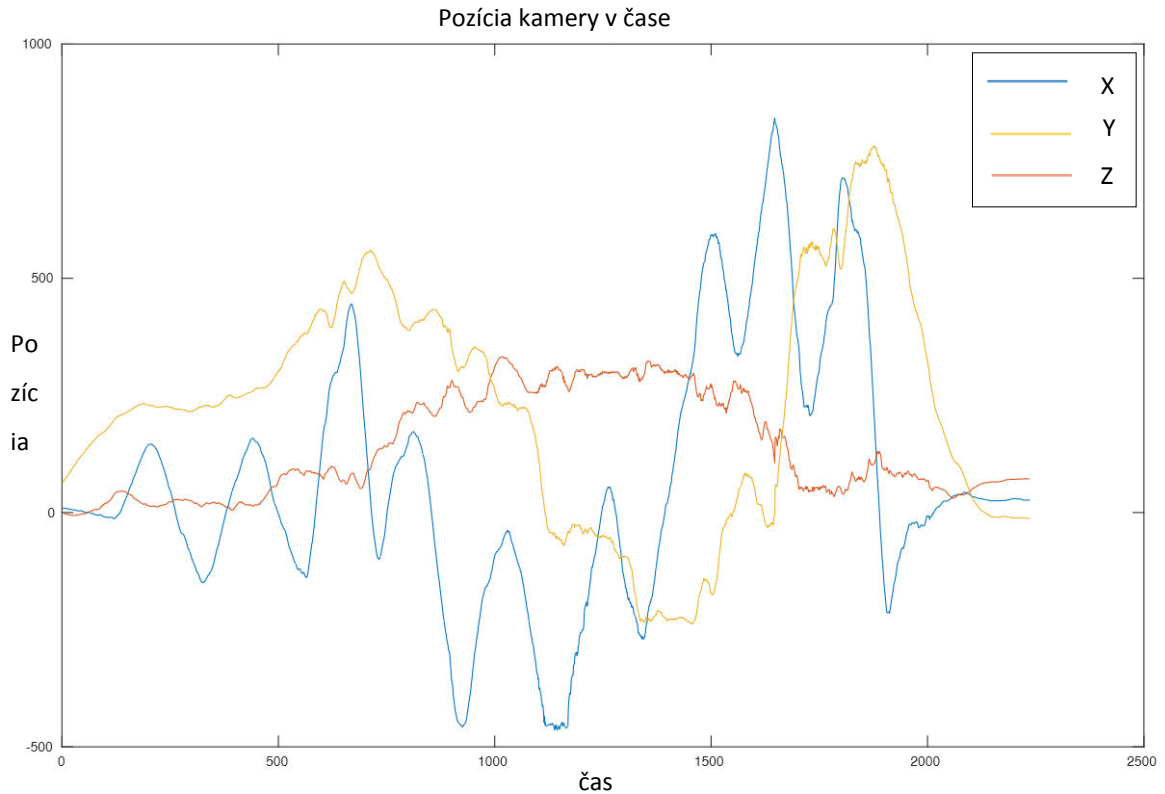
Test 1_2_PTAM – Rovnako jemné drmácanie, niekedy miestami aj väčšie čo je vidieť takisto aj na grafe. Veľmi podobné správanie no menej zachytených bodov.

Test 1_3_PTAM – Jemné drmácanie miestami veľké skoky. V priebehu algoritmu pád a obnovenie čo je vidieť na grafe pri veľkom prepade v čase 1160. Pád a následná obnova je dôsledkom nulovej odchýlky posunu základne pri zaparkovaní kamery čo je teda vlastne už nerelevantné meranie.

Test 1_4_PTAM – Obrovská nestabilita spôsobená zvýšením šumu. Nie je známe prečo pri kvalite Fine daná kamera vykazuje takúto funkcionality. Pád algoritmu v polovici bez zotavenia. Nepoužiteľné.

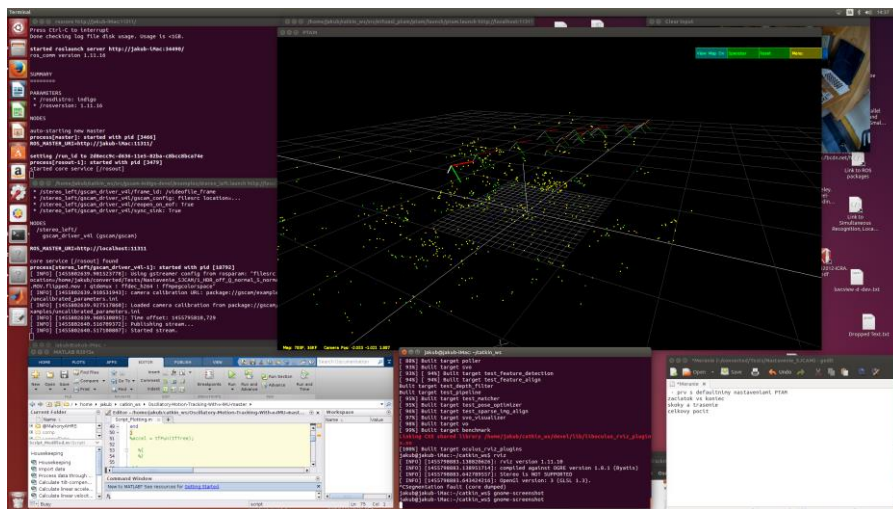
Test 1_5_PTAM – Výsledok rovnaký ako v teste 1_4 no pád ešte skôr. Nepoužiteľné.

Z testov je teda zjavné, že najvhodnejšími nastaveniami sú nastavenia pôvodné, teda funkcia HDR vypnutá, kvalita normálna a ostrosť takisto normálna. Pri daných nastaveniach bola scéna pomerne veľmi stabilná a správna. Je vhodné uviesť výsledný graf tohto pohybu na ktorom je zreteľný pohyb posuvný hlavne v osi X kde užívateľ je v neustálom pohybe zo strany na stranu.



Obr. 28 – Graf správneho pohybu kamery z Test 1_1_PTAM.

Je vhodné a dôležité všimnúť si začiatok a koniec grafu. Pri reálnom natáčaní kamera začína aj končí na úplne rovnakom mieste no v grafe je zreteľný posun, teda chyba a odchýlka kde jednotlivé osy nekončia na rovnakých hodnotách ako začínajú. Drmácanie je najviac zreteľné na osy Z ktorá je veľmi kostrbatá aj napriek pomerne hladkému natáčaniu.



Obr. 29 – Príklad screenshotu z Test 1_1_PTAM.

4.6.2.2. Test 2















V Teste 2 sú testované a ladené samotné nastavenia Ethzasl-PTAM pre najlepšie vyladenie. Nastavenia a možnosti Ethzasl-PTAM sú naozaj veľmi rozsiahle a preto je pozornosť kladená na vybrané parametre ktoré majú potenciál najviac ovplyvniť výsledok a naopak ponechané sú nastavenia ktoré sú dlhodobo zaužívané, pevné, dostatočne podložené a nemajú potenciál meniť konečný výsledok. Pri hľadaní vhodných parametrov pre zmenu je potrebné správne pochopenie fungovania, pochopenie publikácie a takisto je aj pozornosť upriamená na odporúčania samotných autorov v príspevku [29]. Postup nastavovania parametrov Ethzasl-PTAM, ktorých je viac ako 20, je opísaný v užívateľskej a systémovej príručke.

Vybrané nastavené parametre aj s východzími nastaveniami sú:

- Rozlíšenie: 640x360
- Max Patches: 300
- Max KFs: 10

Parameter Max patches predstavuje maximálne množstvo zachytených bodov záujmu. Je predpokladateľné že zvýšenie by mohlo pomôcť stabilite algoritmu no takisto povolí zachytávanie menej stabilných bodov. Parameter Max KFs predstavuje maximálny počet kľúčových rámcov udržiavaných v pamäti počítača a teda zvýšením je možné predpokladať zvýšenie zapamätávania si mapy no aj vyššiu výpočtovú náročnosť.

Test je spustený na rovnakom videu ako Test 1, teda na dokonalom prípade pre tento algoritmus. V nasledujúcej tabuľke, zmenený parameter je jediným zmeneným parametrom pri danom teste a zvyšné parametre sú nastavené na pôvodné východzie hodnoty pre separátne meranie dopadu daného parametra.

Označenie	Zmenený parameter	Graf	Screenshoty
Test 2_1_PTAM = 1_1_PTAM	Plne východzie		
Test 2_2_PTAM	Rozlíšenie 1280x720		
Test 2_3_PTAM	Rozlíšenie 320x180		
Test 2_4_PTAM	Max Patches 650		
Test 2_5_PTAM	Max Patches 1000		
Test 2_6_PTAM	Max KFs Unlimited		
Test 2_7_PTAM	Max KFs 4		

Tab. 3 – Tabuľka k Test 2_PTAM.

Test 2_1_PTAM – Tento test je totožný resp. prebratý z Test 1_1_PTAM a beží pekne a stabilne v reálnom čase.

Test 2_2_PTAM – Výsledky sú podobné s Test 2_1_PTAM no algoritmus dva krát padol čo je vidieť aj na grafe. Bolo nájdených viac bodov na najnižších pyramídach čo ale vôbec neprispelo k stabilite. Celkovo oveľa nižšia stabilita a viaceré pády.

Test 2_3_PTAM – Veľmi málo stabilné, po krátkom čase úplný rozpad bez zotavenia sa. Nepoužiteľné.

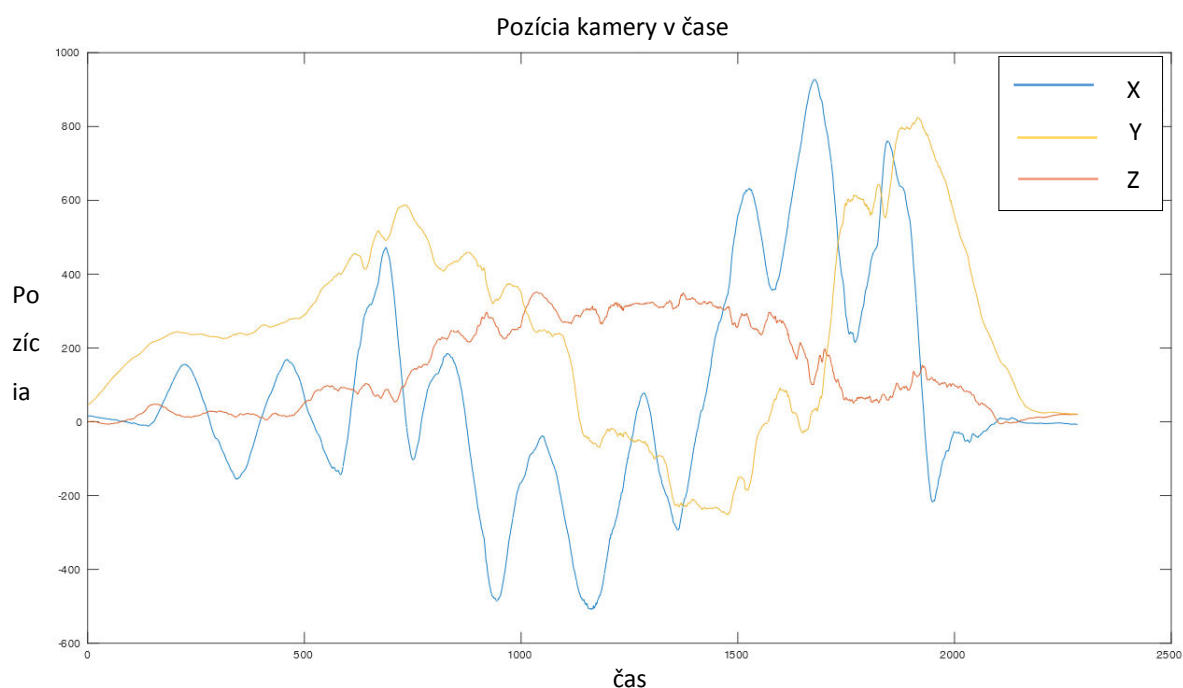
Test 2_4_PTAM – Veľmi stabilné, menej natriasania, hladší stabilný priebeh, menšia konečná odchýlka čo je vidieť na začiatku a konci grafu. Zvýšenie počtu bodov pomohlo, krivka je omnoho hladšia a celkovo presnejšia. Takto zvýšený počet zachytených bodov pomohol.

Test 2_5_PTAM – Stabilné, no vykazuje veľké zakolísania čo je vidieť na špičkách grafu na osi X. O dosť väčšia odchýlka a nepresnosti, náhodné posuny. Príliš vysoké množstvo bodov uškodilo zrejme kvôli zachytávaniu už aj málo stabilných bodov záujmu.

Test 2_6_PTAM – Celý priebeh v poriadku no v čase 1630 rozpad a neskoršie zotavenie sa. Nekonečný počet kľúčových rámcov nezlepšil výsledky.

Test 2_7_PTAM – Test prebehol hladko a v poriadku. Je vidieť jemné trhlinky v priebehu osy X. Celkovo parameter neuškodil ani nepomohol.

Z testov teda vyplýva, že najvhodnejším rozlíšením je rozlíšenie 640x360, veľmi vhodným krokom je zvýšenie Max Patches na 650 a takisto ponechanie Max KFs na 10. Tieto nastavenia sú teda kvôli vhodnosti vymenované a použité za nové východzie nastavenia algoritmu Ethzasl-PTAM.



Obr. 30 – Graf k Test 2_4_PTAM. Graf predstavuje najlepšie dosiahnuté prirodzené správanie Ethzasl-PTAM zvýšením Max Patches na 650.



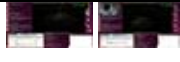




4.6.3. Testovanie s filtráciou

Pred testovaním s filtráciou je treba povedať, že cieľom je nájsť parametre ktoré majú potenciál zlepšiť funkčnosť algoritmov. Za týmto účelom je zvolená množina len určitých testovaných filtrov. Sobelové derivácie sú testované z osobnej zaujímavosti preskúmania dopadu, keďže veľmi presne opisujú hrany v obraze. Gaussovský filter je testovaný z dôvodu filtrácie menších, menej zreteľných bodov. Laplacian z dôvodu zvedavosti, no bez predpokladu alebo očakávania pomoci. Ďalšie filtre ako kontrast atď. testované nie sú pretože v nich nie je zreteľný potenciál a možnosť výpomoci hlavného algoritmu.

4.6.3.1. Test 3

Je treba povedať že ak nie je dané inak, všetky parametre predfiltrácie sú nastavené na východzie. Tieto východzie parametre sú bližšie opísané v užívateľskej a systémovej príručke. Pre rýchlu predstavu východzie nastavení predfiltrácie vid' Obr. 21.

Označenie	Zmenený parameter	Graf	Screenshoty
Test 3_1_PTAM = 2_4_PTAM	Nové východzie		
Test 3_2_PTAM	Sobel zapnutý		
Test 3_3_PTAM	Gauss Kernel 6 + Sobel zapnutý		

Test 3_4_PTAM	Gauss Kernel 15 + Sobel zapnutý		
Test 3_5_PTAM	Gauss Kernel 25 + Sobel zapnutý		
Test 3_6_PTAM	Gauss Kernel 3		
Test 3_7_PTAM	Gauss Kernel 9		
Test 3_8_PTAM	Gauss Kernel 6 + Laplacian zapnutý		
Test 3_9_PTAM	Gauss Kernel 6 + Sobel zapnutý + Dilatácia zapnuta + Delating element elipse + Laplacian zapnutý + Laplacian delta 200		

Tab. 4 - Tabuľka k Test 3_PTAM.

Test 3_1_PTAM – Tento test je totožný respektíve prebratý z Test 2_4_PTAM a beží pekne a stabilne v reálnom čase s novými východzími nastaveniami.

Test 3_2_PTAM – Rozpad hneď po začiatku, nepoužiteľné.

Test 3_3_PTAM – Držanie v poriadku, stabilné celý čas no ku koncu rozpad, nezlepšuje celkovo. Je zreteľné že pomocným prvkom je Gauss a nie Sobel.

Test 3_4_PTAM – Celkom stabilne, udržanie aj ku koncu, nedochádza k rozpadu, veľmi podobné správanie ako v Test 2_4_PTAM. Takisto ale nezlepšuje celkovo, na konci trošku väčšia odchýlka čo je vidieť na grafe.

Test 3_5_PTAM – Okamžitý rozpad. Gauss 25 je príliš veľký kernel a dochádza k prílišnej strate informácií. Je zjavné že použiteľný Gauss je v rozmedzí 0 až 12.

Test 3_6_PTAM – Celé správne a v podstate totožné, nedošlo k nejakej väčšej zmene, viac stabilné ako Test 2_4_PTAM.

Test 3_7_PTAM – Je vidieť že algoritmus sa správa stabilnejšie no nachádza menej bodov. Je možné povedať že Gauss spôsobuje teda zameranie sa algoritmu na významnejšie body a tým napomáha. Na konci ale rozpad. Celkovo viac stabilné ale ku koncu rozpad bez zotavenia.

Test 3_8_PTAM - Nestabilné, veľmi nestabilné s obrovským trasením čo je vidieť aj na grafe. Laplacian nevypomáha.

Test 3_9_PTAM – Tento test predstavuje kombináciu mnohých predfiltrácií a hneď po začiatku sa rozpadá. Je nepoužiteľný no poukazuje na možnosť kombinovania filtrov.

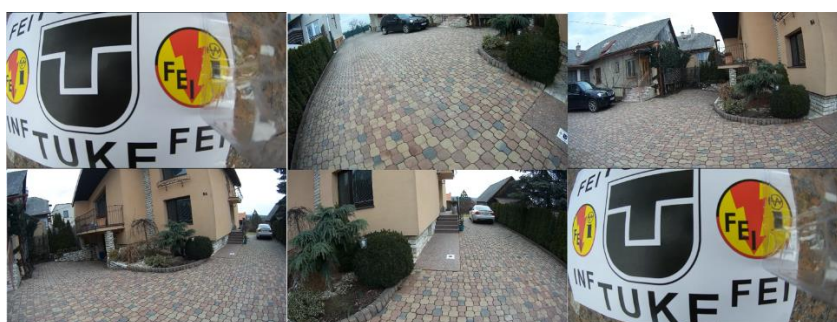
Z testov je zjavné že Sobel nenapomáha algoritmu k správnejšiemu priebehu no Gauss áno. Gaussov filter spôsobuje väčšiu stabilitu.

4.6.4. Testovanie v rôznych prípadoch

Kedže je algoritmus vyladený a je takisto otestovaná predfiltrácia so znalosťou aký filter má potenciál napomôcť, je vhodné pristúpiť k testovaniu správania sa algoritmu v rôznych reálnych prípadoch.

4.6.4.1. Test 4

Hlavným prvým rozdielnym prípadom od doteraz testovaného ideálneho je zmena priestoru z malej miestnosti na otvorené priestranstvo pri zachovaní charakteru pohybu a takisto podmienok vhodných pre Ethzasl-PTAM ako je dostatok záchytných bodov.



Obr. 31 – Náhľad obrázkov z testovacieho videa Test_4_Monocular.mov z testu Test 4.

Označenie	Zmenený parameter	Graf	Screenshoty
Test 4_1_PTAM	Nové východzie		
Test 4_2_PTAM	Gauss Kernel 5		
Test 4_3_PTAM	Gauss Kernel 9		
Test 4_4_PTAM	Gauss Kernel 15		
Test 4_5_PTAM	Gauss Kernel 5 + Sobel zapnutý		
Test 4_6_PTAM	Gauss Kernel 10 + Sobel zapnutý		

Tab. 5 - Tabuľka k Test 4_PTAM.

Test 4_1_PTAM – Rozpad hneď pri zábere na celú dlažbu. Je zrejmé, že dlažba je pre Ethzasl-PTAM veľmi zložitý element pretože má hustú kontrastnú textúru ktorá sa opakuje. Nájdených veľa

nestabilných šumových bodov na dlažbe. Nepoužiteľné, je potreba filtrácie týchto nestabilných šumových bodov z dlažby.

Test 4_2_PTAM – Drmácanie no drží a takisto aj udržané počas celého videa. Gauss pomohol a algoritmus sa nerozpadol a dokonca hoci s drmácaním bolo udržané celé sledovanie. Je zrejme z grafu, že poloha kamery bola udržaná keďže osy končia tam kde začali s jemnou odchýlkou.

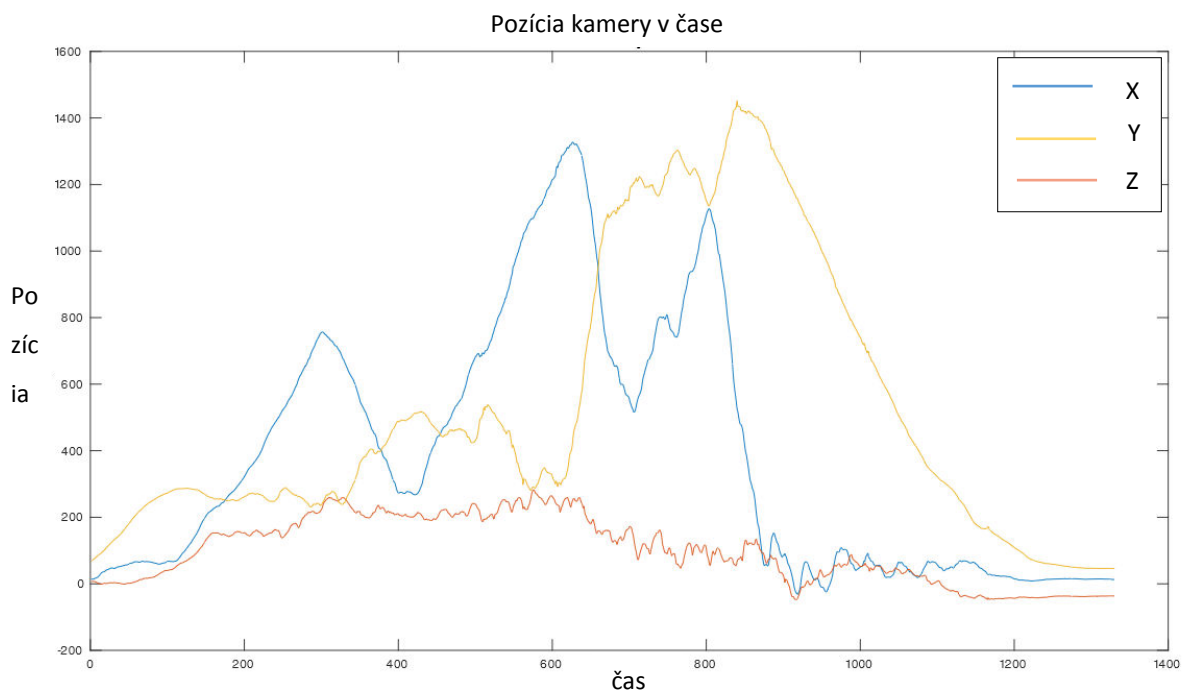
Test 4_3_PTAM – To isté ako 4_2_PTAM, aj graf je totožný, úplne.

Test 4_4_PTAM – Príliš nestabilné, rozpad, nepoužiteľné. Už príliš vysoký Gauss.

Test 4_5_PTAM – Veľa šumových bodov, hneď rozpad. Nepoužiteľné.

Test 4_6_PTAM – Stabilnejšie no veľké drmácanie a obrovské nesprávne posuny, ku koncu rozpad. Nepoužiteľné.




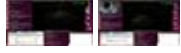



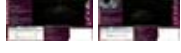




















Z testov je zjavné, že Sobel je naozaj nepoužiteľný a nepomocný filter pre Ethzasl-PTAM. Naopak Gaussov filter spôsobuje väčšiu stabilitu a preto je pre tento algoritmus považovaný za jedinú možnú napomáhajúcu predfiltráciu.



Obr. 32 – Výsledný najsprávnejší graf odhadovanej pozície kamery pomocou Ethzasl-PTAM. Test 4_2_PTAM.

4.6.4.2. Test 5

Nasleduje testovanie správania sa algoritmu v rôznych prípadoch na otvorenom priestranstve.

Označenie	Prípád	Zmenený parameter	Graf	Screenshoty
Test 5_1_1_PTAM	Prechádzka po dvore	Nové východzie		
Test 5_1_2_PTAM	Prechádzka po dvore	Gauss Kernel 6		
Test 5_2_1_PTAM	Detail k zemi	Nové východzie		
Test 5_2_2_PTAM	Detail k zemi	Gauss Kernel 6		
Test 5_3_1_PTAM	Záber na oblohu	Nové východzie		
Test 5_3_2_PTAM	Záber na oblohu	Gauss Kernel 6		
Test 5_4_1_PTAM	Odlesk auta	Nové východzie		
Test 5_4_2_PTAM	Odlesk auta	Gauss Kernel 6		
Test 5_5_1_PTAM	Jednoliata stena	Nové východzie		
Test 5_5_2_PTAM	Jednoliata stena	Gauss Kernel 6		
Test 5_6_1_PTAM	Čistá rotácia	Nové východzie		
Test 5_7_1_PTAM	Rýchly posuvný pohyb	Nové východzie		
Test 5_7_2_PTAM	Rýchly posuvný pohyb	Gauss Kernel 6		
Test 5_8_1_PTAM	Realita	Nové východzie		

Tab. 6 - Tabuľka k Test 5_PTAM.

Test 5_1_1_PTAM – Veľa šumu a hneď rozpad. Správanie je možné predpokladať keďže je dlažba problémovým elementom pre PTAM s novými východzími nastaveniami.

Test 5_1_2_PTAM – Dlhšia stabilita ako v prvom prípade. Po nejakom čase rozpad. Nezvláda väčšie priestory.

Test 5_2_1_PTAM – Rýchlo stratené a nenájdene sledovanie.

Test 5_2_2_PTAM – Nie stabilné ale dokončené správne. Obe detaily zvládlo. Konečná jemná odchýlka. Dokazuje správnosť použitia Gauss pre filtráciu mäťúcich textúr.

Test 5_3_1_PTAM – Zo začiatku v poriadku. Po prechode kamery na oblohu a zmene expozície rozpad.

Test 5_3_2_PTAM – To isté ako 5_3_1_PTAM. Obloha je pre PTAM problémová a nie je možné ju sledovať ani s použitým Gauss.

Test 5_4_1_PTAM – Pri prichádzaní k autu drží. Pri zábere na auto a pohybe pri ňom rozpad. Predpokladateľné.

Test 5_4_2_PTAM – Rovnaký výsledok ako pri 5_4_1. Odlesky sú pre PTAM nežiaduce a spôsobujú pád.

Test 5_5_1_PTAM – Pri prechádzaní k stene všetko drží. Pri pristúpení k stene a zábere celej jednoliatej steny nastáva rozpad algoritmu, predpokladateľné, žiadne body k zachyteniu.

Test 5_5_2_PTAM – Rovnaký výsledok. Gauss nepomáha pretože už bez Gauss je málo bodov a s použitím ešte menej.

Test 5_6_1_PTAM – Pri rotácii pád. Je to predpokladateľné keďže pre PTAM je dôležitý dostatočný posun. Gauss nemá zmysel testovať.

Test 5_7_1_PTAM – Hneď pri začatí rýchleho pohybu je možné pozorovať rozpad.


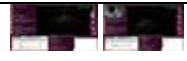
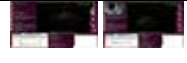

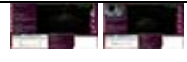


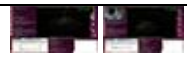
Test 5_7_2_PTAM – Rovnaký výsledok. Rýchle pohyby spôsobujú rozpad.

Test 5_8_1_PTAM – Test predstavuje mix rýchleho pohybu, rotácie, detailov, atď.. Hneď nastáva rozbitie a je zrejme že algoritmus nie je dostatočne robustný pre takéto zaobchádzanie.

Z testov je zrejme, že algoritmus je veľmi krehký. Jeho funkcia je veľmi slabá v otvorených priestranstvách, pri povrchoch s opakujúcou sa textúrou, s jednoliatou textúrou, pri rýchlej zmene expozície a pohľade na oblohu, pri rýchlych pohyboch, čistej rotácii, odleskoch či zameraní sa na detail. Je to ohromné množstvo obmedzení v reálnych prípadoch veľmi možných.

4.6.4.3. Test 6

Ďalšie testovanie rôznych prípadov no vo vnútorných priestoroch.

Označenie	Prípad	Zmenený parameter	Graf	Screenshoty
Test 6_1_1_PTAM	Prechádzka vnútri	Nové východzie		
Test 6_1_2_PTAM	Prechádzka vnútri	Gauss Kernel 5		
Test 6_2_1_PTAM	Jednoliata textúra	Nové východzie		
Test 6_3_1_PTAM	Rýchly pohyb	Nové východzie		
Test 6_4_1_PTAM	Prechádzka vnútri 2	Nové východzie		
Test 6_4_1_PTAM	Prechádzka vnútri 2	Gauss Kernel 6		

Tab. 7 – Tabuľka k Test 6_PTAM.

Test 6_1_1_PTAM – Začiatok je úplne stabilný. Pri prechode medzi izbami kde je trochu menej bodov hneď rozpad a rozpadnuté až do konca.

Test 6_1_2_PTAM – Rozpadnutie ešte skôr. Je zjavné a zrejmé že Gauss nie je vhodné používať v prípadoch kde je rozpad spôsobený nedostatkom bodov. Prípade takejto prechádzky vo vnútri by mal byť pre tento algoritmus zvládnuteľný a je sklamaním že nie je.

Test 6_2_1_PTAM – Pri začiatku všetko stabilné. Pri prechode na textúru rozpad a zotavenie sa po následnom opustení textúry. Potvrdenie nezvládnutia jednoliatych textúr.

Test 6_3_1_PTAM – Strata pri rýchlom pohybe no rýchle znovu nájdenie sa a zotavenie pri stabilizácii. Pri rýchlom pohybe do nového prostredia ostáva rozpadnutý.

Test 6_4_1_PTAM – Táto prechádzka predstavuje prechádzku vo vnútri kde sú miesta iba s veľa bodmi pre zamedzenie pádu z nedostatku bodov. Stabilné veľmi dlho no neskôr pád kvôli nedostatočnému posunu.

Test 6_4_1_PTAM – Rovnaký výsledok ako v prípade 6_4_1.

Z testov je zjavné že aj v malých priestoroch je algoritmus veľmi nestabilný a náchylný na pád už pri menšej čistej rotácii a je neschopný získavať dostatočný počet bodov už pri miestach s trochu menším počtom bodov.

4.6.4.4. Test 7

Posledné testovanie rôznych prípadov.

Označenie	Prípade	Zmenený parameter	Graf	Screenshoty
Test 7_1_1_PTAM	Slnko oproti, tieň, expozičný prechod	Nové východzie		
Test 7_1_2_PTAM	Slnko oproti, tieň, expozičný prechod	Gauss Kernel 5		
Test 7_2_1_PTAM	Slnko oproti, tieň, expozičný prechod 2	Nové východzie		
Test 7_2_2_PTAM	Slnko oproti, tieň, expozičný prechod 2	Gauss Kernel 6		
Test 7_3_1_PTAM	Prechod von do dnu	Nové východzie		
Test 7_3_2_PTAM	Prechod von do dnu	Gauss Kernel 6		
Test 7_4_1_PTAM	Prechod osoby	Nové východzie		

Tab. 8 – Tabuľka k Test 7_PTAM.

Test 7_1_1_PTAM – Zo začiatku stabilné no už pri vychádzaní z tieňa pád. K slnku ani nedošlo.

Test 7_1_2_PTAM – Stabilnejšie a dlhšie držanie, aj napriek tomu obdobný výsledok.

Test 7_2_1_PTAM – Na slnku drží stabilne. Po expozičnom prechode a tieni rozpad.

Test 7_2_2_PTAM – Oveľa lepšie stabilnejšie držanie, obdobný výsledok.

Test 7_3_1_PTAM – Pre prechode malo zaznamenaných bodov a rozpad. Z časti chyba zmeny expozície, z časti málo nájdených bodov.

Test 7_3_2_PTAM – To isté ako 7_3_1 ale ešte menej bodov.

Test 7_4_1_PTAM – Pri prechode rozpad no hneď znovu nájdenie.

Z vykonaných testov je zrejmé, že algoritmus je málo stabilný pre expozičné prechody a takisto čistý posun vpred a vzad. Takisto je málo robustný voči pohybujúcim sa osobám a tieňom.

4.6.5. Vybrané nastavenia Ethzasl-PTAM a záver testovania

Z vykonaného obrovského množstva testov je zrejmé, že je vhodné pri natáčaní monokulárnou kamerou používať nastavenia východzie, teda kvalitu aj ostrosť na nastavenie normal. Ďalej je zrejmé, že je vhodné zvýšenie maximálneho množstva povolených Ethzasl-PTAM bodov záujmu z 300 na 650 a takisto používať rozlíšenie 640x360. Pre prípad predfiltrácie je jediný vhodný použiteľný filter filter Gaussov a to v rozmedzí 0 až 10 pri potrebe väčšej stability no nie je vhodný pre prípady s nedostatočným množstvom nájdených bodov ale pre prípady veľkého množstva zle určených šumových bodov.

Z testov takisto vyplýva že algoritmus je veľmi krehký a náchylný k rozpadu pre veľkom množstve reálne dejúcich sa prípadov ako je tieň, expozičný prechod, odlesk, čistá rotácia, rýchle pohyby, čistý posun vpred vzad, pohľad do oblohy, otvorenejšie priestory, opakujúca sa textúra atď..

4.7. Testovanie a nastavenie LSD-SLAM

Pri testovaní algoritmu LSD-SLAM je vhodné povedať, že použitý algoritmus je tiež monokulárny a preto je vstupom testovania množina totožných obrazových súborov použitá v predchádzajúcom testovaní Ethzasl-PTAM čo je v konečnom dôsledku veľmi potrebné pre porovnanie algoritmov navzájom. Označovanie a pomenovania sú obdobné ako v testovaní Ethzasl-PTAM.

Test 1 je preskočený pretože kamera je už správne nastavená.

4.7.1. Testovanie bez filtrácie

4.7.1.1. Test 2

Aj v prípade LSD-SLAM je k dispozícii väčšie množstvo nastaviteľných parametrov no je vybraná množina testovaných parametrov u ktorej je predpoklad možnosti veľkého dopadu na výkon a funkčnosť. Podrobnosti a postup nastavovania sú obdobne opísané v užívateľskej a systémovej príručke. Parametre hľadania slučky nie sú testované pretože uzatváranie slučky je nežiadúce. Východzie relevantné testované parametre:

- Rozlíšenie: 640x352
- minusedGrad: 5.0
- cameraPixelNoise: 4.0
- KF Usage: 4
- KF Dist: 3
- doSLAM: áno


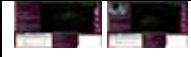




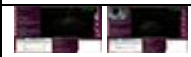



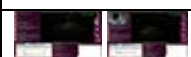





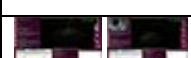




Parametre minusedGrad a cameraPixelNoise predstavujú citlivosť algoritmu. Nižšie hodnoty spôsobujú snímanie aj slabších gradientov. Parametre KF Usage a KF Dist predstavujú množstvo vytváraných kľúčových rámcov a sú obdobou parametra Max KF z Ethzasl-PTAM, väčšia hodnota spôsobuje viac vytvorených kľúčových rámcov. Parameter doSLAM je veľmi dôležitý a predstavuje prepínač ktorým je možné vypnúť alebo zapnúť neustále upravovanie polohy kamery vzhľadom na mapu. Pri vypnutí doSLAM sa algoritmus správa viac ako odometria, mapa síce vytváraná je no robot sa ňou neorientuje a neupravuje.

Vstupným obrazom tohto algoritmu je obraz s takým rozlíšením, že každý rozmer je deliteľný číslom 16, čo je podmienkou funkčnosti algoritmu.

Veľmi dôležité je porovnávanie výsledkov tohto testu s grafom 2_4_PTAM, pretože vstupný súbor je totožný obrazový súbor a algoritmus Ethzasl-PTAM vykázal v tomto teste skvelé výsledky o ktoré je možné sa pomerne dôverne opierať a snažiť sa doceliť podobné výsledky.

Spracovanie v reálnom čase predstavuje násobič spomalenia pričom 1.0x predstavuje spracovanie v reálnom čase, teda 30 snímkou za sekundu, 0.5x predstavuje dvojnásobné spomalenie, teda 15 snímkou za sekundu atď.. Toto pozorovanie je uskutočnené vďaka skutočnosti, že v čase, keď algoritmus nestíha, spracovanie videa je spomalené a je možné sledovať pokles a presnú hodnotu snímkovacej frekvencie.

Označenie	Zmenený parameter	Spracovanie v reálnom	Graf	Screenshoty
-----------	-------------------	-----------------------	------	-------------

Test 2_1_LSD	Vychodzie	1.0x		
Test 2_2_LSD	Rozlíšenie 1280x720	0.39x		
Test 2_3_LSD	Rozlíšenie 320x176	1.0x		
Test 2_4_LSD	minusedGrad 2.0	1.0x		
Test 2_5_LSD	minusedGrad 10.0	1.0x		
Test 2_6_LSD	cameraPixelNoise 1.5	1.0x		
Test 2_7_LSD	cameraPixelNoise 7.0	1.0x		
Test 2_8_LSD	KF Usage 10	1.0x		
Test 2_9_LSD	KF Dist 10	1.0x		
Test 2_10_LSD	KF Usage 20 Dist 15	1.0x		
Test 2_11_LSD	doSLAM vypnutý	1.0x		

Tab. 9 – Tabuľka k Test 2_LSD

Test 2_1_LSD – Priebeh je pomerne nestabilný a kamera je pohybovo upravovaná v každom čase čo spôsobuje dojem neustáleho zlého pohybu. Mnohé zlé odhady je možné pozorovať hlavne na grafe pri porovnaní s grafom 2_4_PTAM. K obrovskému rozdielu dochádza pri posledných troch kopcoch osi X. Posledný kopec je v LSD-SLAM príliš nízko a takisto celkový tvar priebehu X je od polovice pomerne skreslený. Takisto je možné pozorovať od polovice grafu nepresnú Y-ovú os ktorá je v niektorých miestach naozaj veľmi zle pokrivená. Najväčšie nepresnosti sú ale viditeľné na tretom kopci osi X. Algoritmus vykázal takisto jednu veľkú nájdenú slučku podľa čoho sa kamera asi v čase 2150 upravila. Celkovo algoritmu vykazuje veľké úpravy polohy kamery v reálnom čase a je menej presný ako Ethzasl-PTAM v tomto prípade. Neustála úprava kamery a nájdenie slučky je spôsobené zapnutým parametrom doSLAM.

Test 2_2_LSD – Algoritmus je oveľa stabilnejší ale aj príliš pomalý. Spomalenie je na hranici nepoužiteľnosti, až 0.39x. Výsledky sú omnoho lepšie kde je vidieť na grafe, že takmer úplne a plne kopíruje graf Ethzasl-PTAM. V čase 1400 nastáva pád ktorý je spôsobený nestíhaním algoritmu. Celkovo je toto nastavenie výsledkami skvelé, kde je vidieť že aj tretí kopec osi X je správne okopírovaný, no nepoužiteľné z dôvodu nedostatočnej rýchlosti výpočtu. Je možné, že algoritmus by dané nastavenia zvládol pri väčšom výpočtovom výkone.

Test 2_3_LSD – Veľká nepresnosť a nestabilita čo je vidieť aj na grafe. Najvhodnejším rozlíšením teda ostáva 640x352.

Test 2_4_LSD – Zhoršené výsledky, čo je odzrkadlené aj na grafe.

Test 2_5_LSD – Takisto zhoršené výsledky, oveľa kostrbatejší graf. Parameter minusedGrad je najvhodnejšie ponechať na východzíh 5.0.

Test 2_6_LSD – Celkom stabilné, obrovská uzavretá slučka. Žiadne všeobecné zlepšenie.

Test 2_7_LSD – Totožné s 1_6. Nie je potrebné meniť cameraPixelNoise.

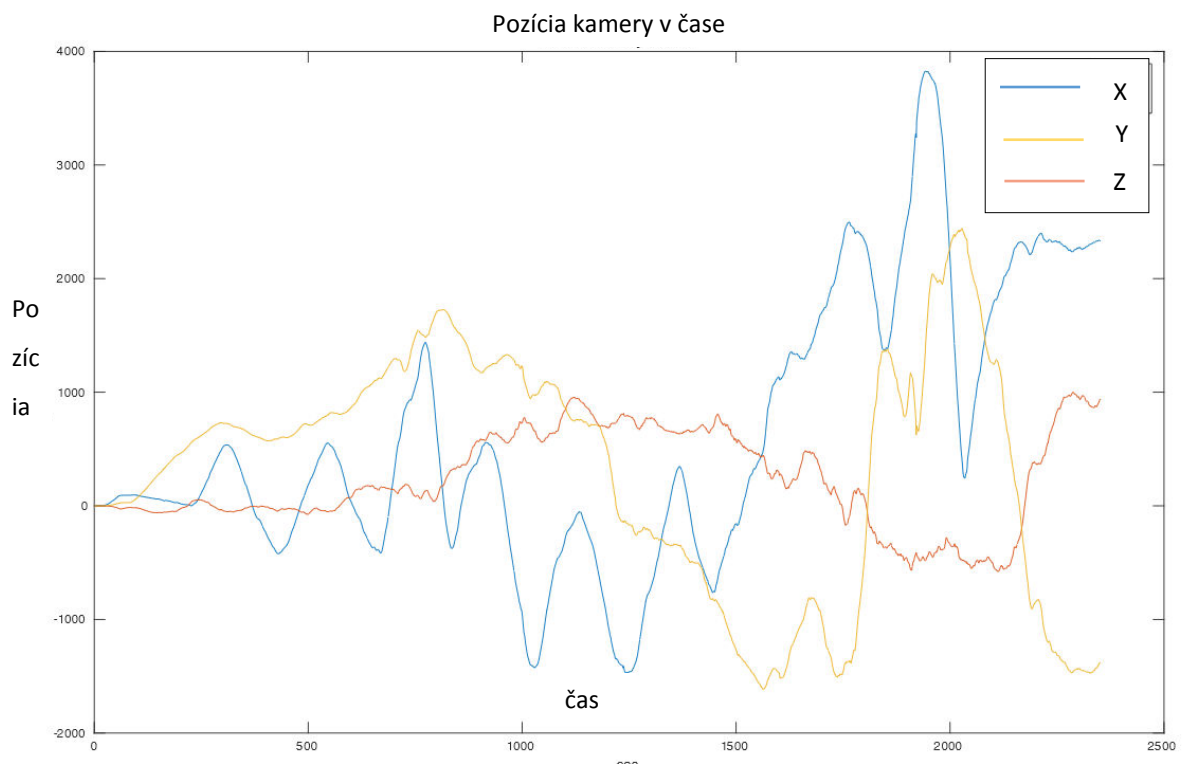
Test 2_8_LSD – Žiadne viditeľné zlepšenie podľa grafu.

Test 2_9_LSD – Viac kostrbatý a nesprávny priebeh.

Test 2_10_LSD – Viac kostrbatý a nesprávny priebeh. Nie je potrebné meniť KF parametre.

Test 2_11_LSD – Vo všetkých LSD SLAM testoch doteraz bola pozícia kamery v reálnom čase veľmi upravovaná algoritmom čo značne prispelo k samovoľnému pohybu kamery. S vypnutým doSLAM je pohyb kamery omnoho prirodzenejší. Celková chyba je obrovská, čo je vidieť na grafe, no pre potreby využitia algoritmu v reálnom čase je práve funkcia doSLAM vypnutá správnym krokom.

Celkovo je z testov zreteľné že pôvodné hodnoty sú nastavené správne a nie je potrebné ich meniť. Pre presné výsledky je vhodné použiť vyššie rozlíšenie no kvôli výpočtovej náročnosti je zvolené rozlíšenie 640x352.




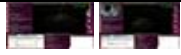



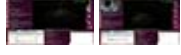










Obr. 33 - Výsledný graf pozície kamery LSD-SLAM v Test 2_11_LSD.

Zobrazený graf je vhodné porovnať s Test 2_4_PTAM.

4.7.1. Testovanie s filtráciou

Ďalším krokom po vyladení samotného algoritmu je hľadanie vhodnej filtrácie ktorá by mohla napomôcť výsledkom algoritmu.

4.7.1.1. Test 3

Označenie	Zmenený parameter	Graf	Screenshoty
Test 3_1_LSD = Test 2_1_LSD	Vychodzie		
Test 3_2_LSD	Sobel zapnutý		
Test 3_3_LSD	Gauss Kernel 6 + Sobel zapnutý		
Test 3_4_LSD	Gauss Kernel 15 + Sobel zapnutý		
Test 3_5_LSD	Gauss Kernel 3		
Test 3_6_LSD	Gauss Kernel 9		
Test 3_7_LSD	Gauss Kernel 5 + Laplacian zapnutý		
Test 3_8_LSD	Gauss Kernel 6 + Sobel zapnutý + Dilatácia zapnuta + Delating element ellipse + Laplacian zapnutý + Laplacian delta 200		

Tab. 10 – Tabuľka k Test 3_LSD.

Test 3_1_LSD – Predstavuje totožný test aj výsledok ako doposiaľ najlepší dosiahnutý, teda ako Test 3_1_LSD.

Test 3_2_LSD – Oveľa stabilnejšie a presnejšie riešenie. Krivky sú oveľa presnejšie a absolútne sa podobajú krivkám Ethzasl-PTAM. Sobel priniesol zlepšenie ktoré je zrejme spôsobené zameraním sa algoritmu viac na hrany ako na gradienty. Celkovo je to veľké zlepšenie. Menej presné ukončenie vo forme malej odchýlky.

Test 3_3_LSD – Totožný výsledok, teda zlepšený, ako v prípade 2_2_LSD no s väčšou odchýlkou na konci.

Test 3_4_LSD – Veľké zhoršenie kriviek, zväčšená odchýlka. Už príliš veľký Gauss.

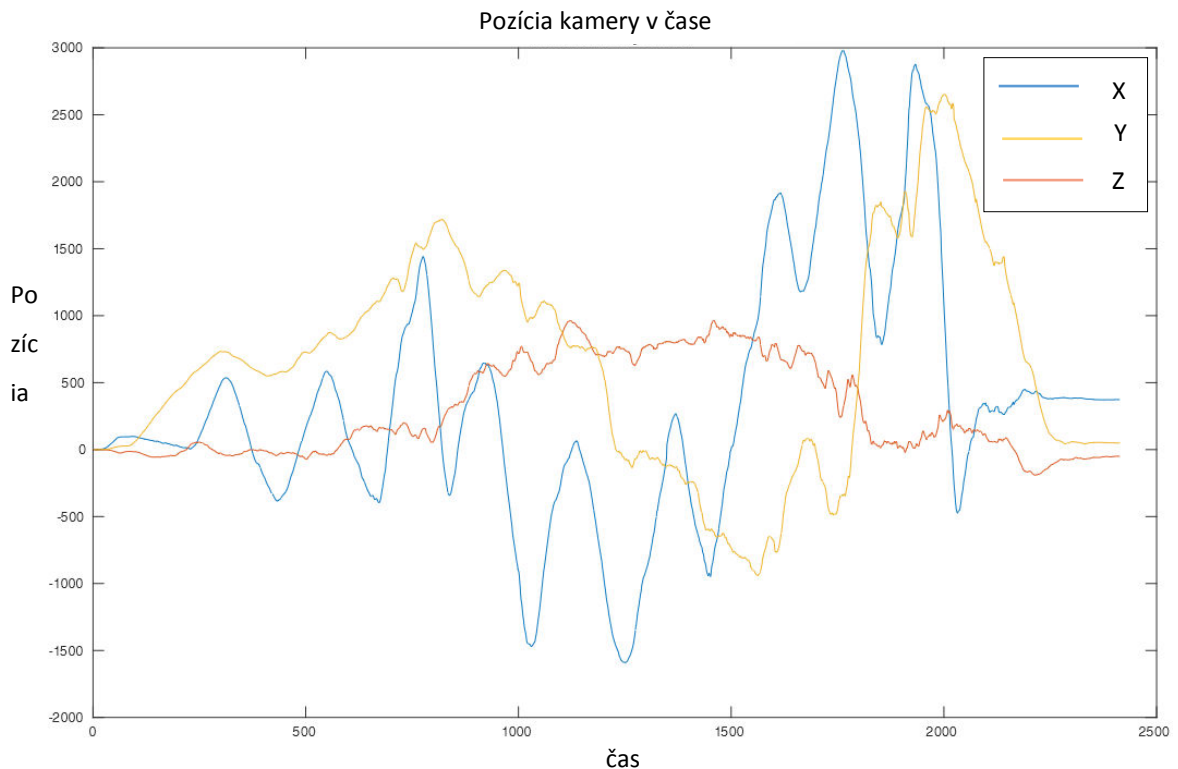
Test 3_5_LSD – Nepoužiteľné. Krivky sú neskutočne mimo, roztraseené.

Test 3_6_LSD – Takisto veľmi zle. Je zrejme že Gauss pôsobí na algoritmus práve opačne ako v prípade Ethzasl-PTAM a to úplne zle.

Test 3_7_LSD – Podobné zlepšenie ako v prípade Sobela. Väčšia konečná odchýlka.

Test 3_8_LSD – Nepoužiteľné.

Z testov je zrejme, že obrovským prínosom je práve použitie Sobelovho filtra a práve tento filter je považovaný za jediný nápomocný pre tento algoritmus.



Obr. 34 – Výsledný graf pozície kamery LSD-SLAM v Test 3_2_LSD.

Zobrazený graf je vhodné porovnať s Test 2_4_PTAM.

4.7.1. Testovanie v rôznych prípadoch

Keďže je algoritmus vyladený a je takisto otestovaná predfiltrácia so znalosťou aký filter má potenciál napomôcť, je vhodné pristúpiť k testovaniu správania sa algoritmu v rôznych reálnych prípadoch.

4.7.1.1. Test 4

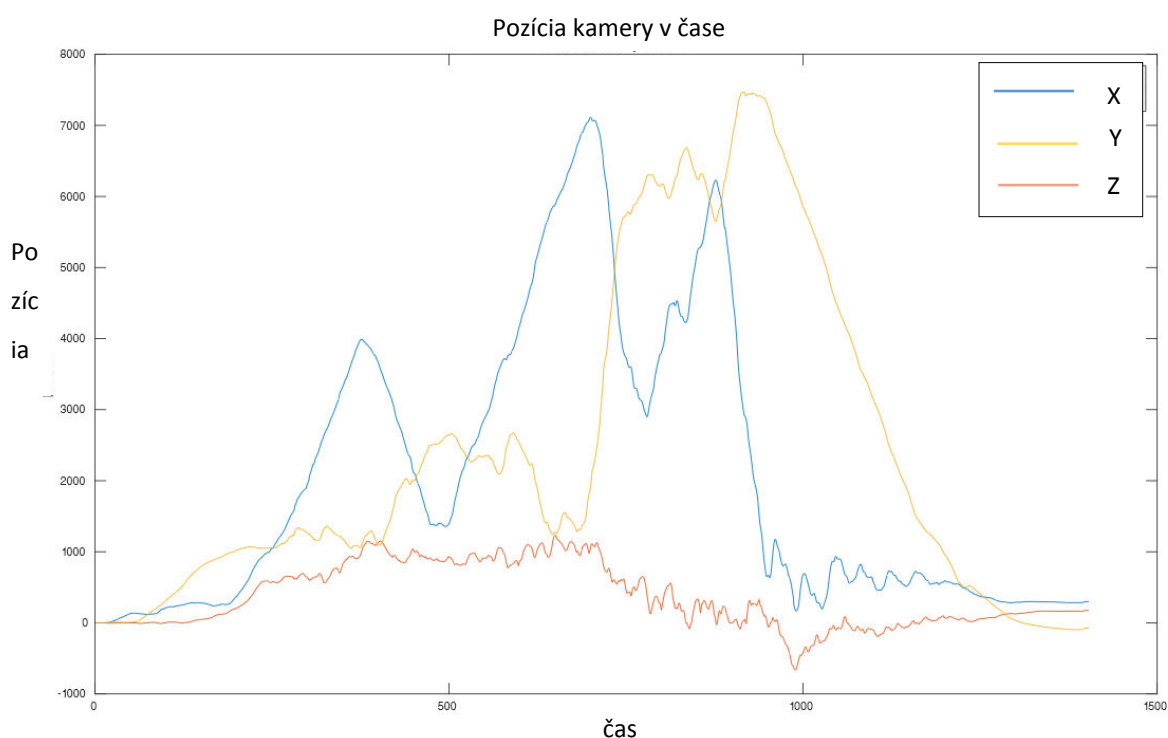
Označenie	Zmenený parameter	Graf	Screenshoty
Test 4_1_LSD	doSLAM vypnutý		
Test 4_2_LSD	doSLAM vypnutý + Sobel zapnutý		

Tab. 11 - Tabuľka k Test 4_LSD.

Test 4_1_LSD – Správne sledovanie. Nie úplne stabilné no správne s jednou väčšou chybou, a to, že konečná poloha kamery je v správnom mieste no pootočená o 90° okolo osi X. Táto chyba je zrejme spôsobená jedliatosťou textúry. Celková stabilita pri otvorenej scéne.

Test 4_2_LSD – Úplne správne sledovanie. Sobel naozaj veľmi pomohol. Skvelo vidieť správny začiatok aj koniec kamery na screenshotoch aj grafe. Koniec s jemnou odchýlkou.

Z testov je zjavné, že Sobel je naozaj veľmi použiteľný a pomocný filter pre LSD-SLAM. Takisto je možné pozorovať väčšiu stabilitu v otvorených priestoroch oproti Ethzasl-PTAM.






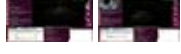















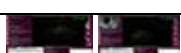

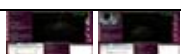



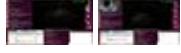


Obr. 35 – Výsledný najsprávnejší graf odhadovanej pozície kamery pomocou LSD-SLAM. Test 4_2_LSD.

Zobrazený graf je vhodné porovnať s grafom Test 4_2_PTAM.

4.7.1.2. Test 5

Nasleduje testovanie správania sa algoritmu v rôznych prípadoch na otvorenom priestranstve.

Označenie	Prípád	Zmenený parameter	Graf	Screenshotty
Test 5_1_1_LSD	Prechádzka po dvore	doSLAM vypnuté		
Test 5_1_2_LSD	Prechádzka po dvore	doSLAM vypnuté + Sobel zapnuté		
Test 5_1_3_LSD	Prechádzka po dvore	doSLAM vypnuté +		

		Gauss Kernel 6 + Sobel zapnuté		
Test 5_2_1_LSD	Detail k zemi	doSLAM vypnuté		
Test 5_2_2_LSD	Detail k zemi	doSLAM vypnuté + Sobel zapnuté		
Test 5_3_1_LSD	Záber na oblohu	doSLAM vypnuté		
Test 5_3_2_LSD	Záber na oblohu	doSLAM vypnuté + Sobel zapnuté		
Test 5_4_1_LSD	Odlesk auta	doSLAM vypnuté		
Test 5_4_2_LSD	Odlesk auta	doSLAM vypnuté + Sobel zapnuté		
Test 5_5_1_LSD	Jednoliata stena	doSLAM vypnuté		
Test 5_5_2_LSD	Jednoliata stena	doSLAM vypnuté + Sobel zapnuté		
Test 5_6_1_LSD	Čistá rotácia	doSLAM vypnuté		
Test 5_6_2_LSD	Čistá rotácia	doSLAM vypnuté + Sobel zapnuté		
Test 5_7_1_LSD	Rýchly posuvný pohyb	doSLAM vypnuté		
Test 5_7_2_LSD	Rýchly posuvný pohyb	doSLAM vypnuté + Sobel zapnuté		
Test 5_8_1_LSD	Realita	doSLAM vypnuté		
Test 5_8_2_LSD	Realita	doSLAM vypnuté + Sobel zapnuté		

Tab. 12 - Tabuľka k Test 5_LSD.

Test 5_1_1_LSD – Problémy pri začiatku, neskôr veľmi správne no v strede stratená mierka a zvyšok scény veľmi zmenšený. Sledovanie je celkom dobré no problém mierky je zjavný. Vid' screenshots.

Test 5_1_2_LSD – To isté ako 5_1_1_LSD.

Test 5_1_3_LSD - To isté ako 5_1_1_LSD. Je zrejme že Sobel nie je pri problémoch s mierkou nápomocný.

Test 5_2_1_LSD – Obe detaily správne no pri vracaní sa na základňu strata mierky.

Test 5_2_2_LSD – Rovnaké ako v Test 5_2_1_LSD.

Test 5_3_1_LSD – Správne sledovaný pohyb, no s veľkým natriasaním. Zmena expozície aj pohľad na oblohu ustálo vďaka hranám domov v okrajoch. Rozpad pri konci, mimo oblohu.

Test 5_3_2_LSD – Rovnaké ako Test 5_3_1_LSD.

Test 5_4_1_LSD – Zvládnuté správne aj napriek odleskom.

Test 5_4_2_LSD – Zvládnuté správne. Konečné plochy sú natočené no vzhľadom na fakt, že z optických SLAM algoritmov je použitý iba posun, je táto skutočnosť v poriadku keďže je orientácia načítaná priamo z Oculusu.

Test 5_5_1_LSD – Zo začiatku je všetko v poriadku. Pri zabraní celého obrazu jednoliatou textúrou virtuálna kamera stojí, čo je možné predpokladať. Strata mierky po obnovení.

Test 5_5_2_LSD – To isté ako Test 5_5_1_LSD.

Test 5_6_1_LSD – Celkom stabilné držanie no veľké výkyvy kvôli rotácii. Neskôr veľká strata mierky a rozpad.

Test 5_6_2_LSD – Stabilné a správne. Správne zvládnutá takmer čistá rotácia. Opäť dôkaz Sobela.

Test 5_7_1_LSD – Prvé dve posuny správne, ďalší spôsobil stratu mierky a miesta. Celkovo stabilnejšie ako Ethzasl-PTAM.

Test 5_7_2_LSD – To isté ako Test 5_7_1_LSD.




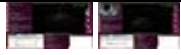
Test 5_8_1_LSD – Strata pri prvej väčšej veľmi rýchlej rotácii.




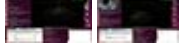

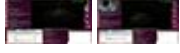



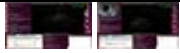


Test 5_8_2_LSD – Rovnaký výsledok ako v Test 5_8_1_LSD. LSD-SLAM nie je stále pripravený na reálne použitie.

Z vyhotovených testov vyplýva, že algoritmus je oveľa stabilnejší ako Ethzasl-PTAM. Správne zvládol prípad pohľadu do oblohy, detailov, a aj mnohých ďalších kde ale pri niektorých bola veľkou chybou mierka. Algoritmus sa mnohokrát správal správne no po stratení mierky čo je v realite menej použiteľné. Celkovo robustnejší, viac drží a zvláda aj napr. čistú rotáciu, ktorá sa pri bežnom používaní vyskytuje veľmi často.

4.7.1.3. Test 6

Ďalšie testovanie rôznych prípadov no vo vnútorných priestoroch.

Označenie	Prípad	Zmenený parameter	Graf	Screenshoty
Test 6_1_1_LSD	Prechádzka vnútri	doSLAM vypnuté		
Test 6_1_2_LSD	Prechádzka vnútri	doSLAM vypnuté + Sobel zapnuté		

Test 6_2_1_LSD	Jednoliata textúra	doSLAM vypnuté		
Test 6_2_2_LSD	Jednoliata textúra	doSLAM vypnuté + Sobel zapnuté		
Test 6_3_1_LSD	Rýchly pohyb	doSLAM vypnuté		
Test 6_3_2_LSD	Rýchly pohyb	doSLAM vypnuté + Sobel zapnuté		
Test 6_4_1_LSD	Prechádzka vnútri 2	doSLAM vypnuté		
Test 6_4_1_LSD	Prechádzka vnútri 2	doSLAM vypnuté + Sobel zapnuté		

Tab. 13 – Tabuľka k Test 6_LSD.

Test 6_1_1_LSD a Test 6_1_2_LSD – Všetko úplne správne no postupne strácaná mierka.

Test 6_2_1_LSD a Test 6_2_2_LSD – Zvládnuté správne. Táto jednoliata textúra nie je úplne jednoliata a algoritmus ju zvládol správne naproti Ethzasl-PTAM.


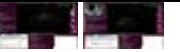





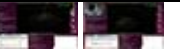

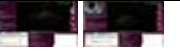






Test 6_3_1_LSD a Test 6_3_2_LSD – Nezládnuté. Prvých pár pohybov áno ale potom odlet kamery a strata mierky.

Test 6_4_1_LSD a Test 6_4_1_LSD – Správne no s odchýlkou.

Z testov je zrejme že algoritmus je opäť oveľa robustnejší ako Ethzasl-PTAM kde zvládol obe prechádzky, hoci jednu so stratou mierky, a takisto zvládol jednoliatu textúru.

4.7.1.4. Test 7

Posledné testovanie rôznych prípadov.

Označenie	Prípad	Zmenený parameter	Graf	Screenshoty
Test 7_1_1_LSD	Slnko oproti, tieň, expozičný prechod	doSLAM vypnuté		
Test 7_1_2_LSD	Slnko oproti, tieň, expozičný prechod	doSLAM vypnuté + Sobel zapnuté		
Test 7_2_1_LSD	Slnko oproti, tieň, expozičný prechod 2	doSLAM vypnuté		
Test 7_2_2_LSD	Slnko oproti, tieň, expozičný prechod 2	doSLAM vypnuté + Sobel zapnuté		
Test 7_3_1_LSD	Prechod von do dnu	doSLAM vypnuté		
Test 7_3_2_LSD	Prechod von do dnu	doSLAM vypnuté + Sobel zapnuté		
Test 7_4_1_LSD	Prechod osoby	doSLAM vypnuté		
Test 7_4_1_LSD	Prechod osoby	doSLAM vypnuté + Sobel zapnuté		

Tab. 14 – Tabuľka k Test 7_LSD.

Test 7_1_1_LSD – Začiatok v poriadku a stabilný. Pri zobrazení slnka oproti chyba algoritmu, oddialenie virtuálnej kamery a strata mierky. Príchod naspať v poriadku.

Test 7_1_2_LSD – Celý priebeh úplne v poriadku. So Sobelom slnko oproti nenaruša sledovanie. Všetko v poriadku a koniec s malou odchýlkou.

Test 7_2_1_LSD a Test 7_2_2_LSD – Stabilné a celkom správne. Menšia odchýlka so Sobelom.

Test 7_3_1_LSD – Priebeh v poriadku no pri konci strata mierky.

Test 7_3_2_LSD – Celý priebeh v poriadku, trochu väčšie natriasanie.

Test 7_4_1_LSD – Všetko v poriadku. Prechádzajúca osoba nenarušila výpočet.

Test 7_4_1_LSD – Všetko v poriadku. Prechádzajúca osoba nenarušila výpočet. Zakončenie so stratou mierky. Je zjavné, že hoci je Sobel veľkým zlepšením, pri niektorých rotačných pohyboch naopak prospieva negatívne.

4.7.2. Vybrané nastavenia LSD-SLAM a záver testovania

Z vyhotovených testov je zjavné, že LSD-SLAM je oveľa robustnejší a stabilnejší oproti Ethzasl-PTAM. Boli zvládnuté mnohé testy ktoré v prípade Ethzasl-PTAM nie. Nastavenia LSD-SLAM sú ponechané na východzie s rozdielom parametra doSLAM ktorý je vypnutý. Zvolené rozlíšenie je 640x352 a jediným filtrom, ktorý je naozaj veľmi nápomocný a je výhodné ho v mnohých prípadoch, napr. slnka oproti a iným, zapnúť pre lepšiu funkcionality algoritmu sú Sobelové derivácie.

4.8. Testovanie a nastavenie S-PTAM

Ďalším z algoritmov ktorý je vhodné otestovať a nastaviť je algoritmus S-PTAM.

4.8.1. Kalibrácia kamery

Keďže algoritmus pracuje s natívnymi obrazovými kanálmi prostredia ROS ako aj s natívnymi informačnými kanálmi, ktoré sú publikované, sú oba kamery stereokulárnej kamery kalibrované podľa postupu [30]. Takto je teda všetko pripravené pre správne spustenie a použitie algoritmu S-PTAM.

4.8.2. Zlyhanie používania

Aj po obrovskej snahe, algoritmus nebolo možné použiť. Algoritmus je pomerne nový a aj napriek správnej funkcionalite príkladov od autorov, nevykazuje pri zhotovených súboroch žiadne použiteľné výsledky. Po komunikácii s autormi, je možnosťou malá robustnosť tohto algoritmu ako aj pomerne vysoká zložitosť testovaných scén.

Z týchto dôvodov nie je ďalej tento algoritmus zahrnutý v práci aj napriek tomu, že sú zhotovené a pripravené všetky materiály a podklady pre jeho testovanie.

4.9. Komunikácia medzi Oculus a ROS

Uskutočnenie Informačného toku medzi modulmi Oculus a ROS je veľmi zložitá a to z dôvodu slabej alebo takmer žiadnej podpory Oculusu na platforme Linux. Z tohto dôvodu je po hľadaní a skúšaní rôznych ciest zvolený voľne dostupný modul [31] pre modul Rviz ktorý je potrebné upraviť.

4.9.1. Vykresľovanie

V module bolo potrebné vypnutie korekcie skreslenia pre funkčnosť. Pre vykresľovanie je použitá knižnica Ogre3D ktorá je natívne podporovaná v prostredí ROS. Tomuto vykresľovaniu je venovaný odsek pretože mu bolo potrebné venovať veľké množstvo času.

4.9.2. Získavanie surových dát z Oculusu

Pre potrebu získania surových dát zo zariadenia Oculus bolo po značnom úsilí a pochopení témy až na úroveň ovládačov a API vytvorená vlastná funkcia pre získavanie týchto dát.

V konečnom dôsledku je tento modul značne upravený pre správne získavanie dát z Oculusu a správne vykresľovanie virtuálnej scény prostredia Rviz späť do Oculusu. Modul je takisto naviazaný na už spomenuté parametre predfiltrácie pre ovládanie scény ako je mierka kamery či voľba algoritmu a je takisto naviazaný na Matlab a to formou odosielania akcelerácie a prijímania vypočítanej polohy. V schéme je daný modul zahrnutý v štvorci Fúzia dát.

Je treba povedať že korekcia skreslenia je vypnutá a takisto, že tento modul je podrobnejšie opísaný v užívateľskej a softvérovej príručke.

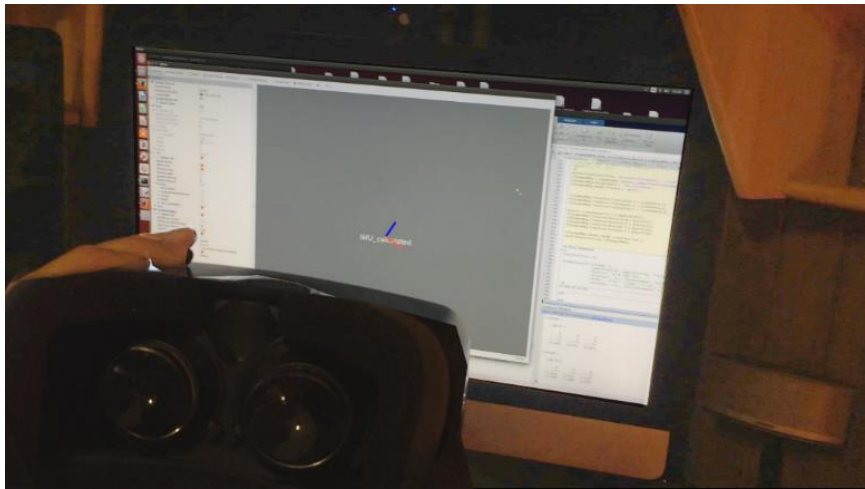
4.10. Komunikácia medzi ostatnými modulmi

V celom celku je veľké množstvo tokov dát ktoré neboli opísané. Tieto toky sú štandardnými tokmi a nevyznačujú sa špeciálnymi požiadavkami a preto im nie je venovaná väčšia

pozornosť. Tieto toky vytvorené sú pre funkčnosť celého celku a bola im venovaná značná pozornosť a značný čas.

4.11. Testovanie dvojitej integrácie akcelerácie

Algoritmus je implementovaný jednoduchou integráciou v prostredí Matlab. Hlavným a vlastne jediným atribútom ktorý je posudzovaný pre jeho testovaní je vizuálny posudok. Testovanie prebieha nasledovne a je zobrazené na obrázku nižšie.



Obr. 36 – Vizuálne testovanie inerčného SLAM.

Užívateľ drží Oculus a pohybom Oculusu sleduje ako správne či nesprávne sú tieto pohyby opisované v scéne virtuálnej. Pre zopakovanie testu je potrebné postupovať podľa návodu v užívateľskej a systémovej príručke.

Po vizuálnom otestovaní tohto algoritmu je treba povedať, že inerčný SLAM je veľmi nepresný. Správanie sa skutočného pohybu je veľmi málokedy správne odzrkadlené a takisto je treba povedať, že akumulovaná chyba je obrovská a akumulácia je veľmi rýchla.

Celkovo je teda možné povedať že inerčný SLAM je pre naše použitie nepoužiteľný v takejto forme a kvôli neurčitosti charakteru pohybu ani v podobe filtrovanej ktorá testovaná a ani implementovaná nie je.

Z týchto dôvodov nie je inerčný SLAM zahrnutý do fúzie dát a spĺňa takto skôr iba formu prezentačnú a náučnú.

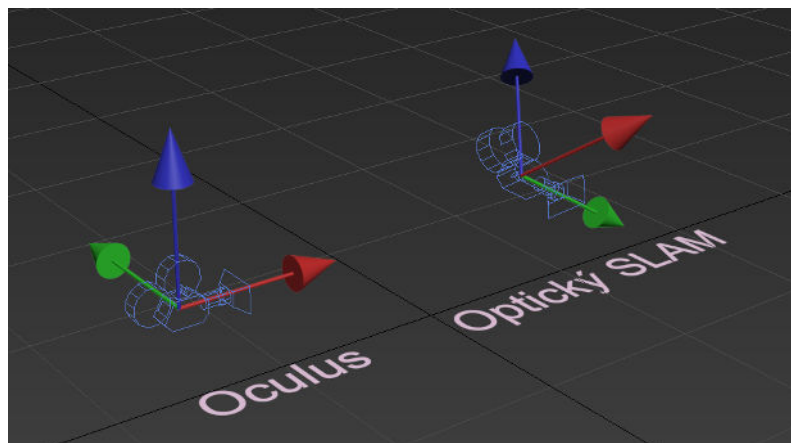
5. Návrh a implementácia fúzie dát a ladenie celku

Posledným modulom ktorý je potrebné pre celkovú funkčnosť opísať je fúzia dát. S celkovým dojmom a funkčnosťou je spojená takisto príkladova scéna a vyladenosť či rýchlosť spracovania.

5.1. Návrh a implementácia fúzie dát

Keďže vstupom do modulu fúzie dát, ako je možné vidieť na Obr. 21, sú dáta z Oculusu a výstupné dáta z optických SLAM, vzniká potreba dáta správne spojiť a využiť pre maximálne a správne využitie.

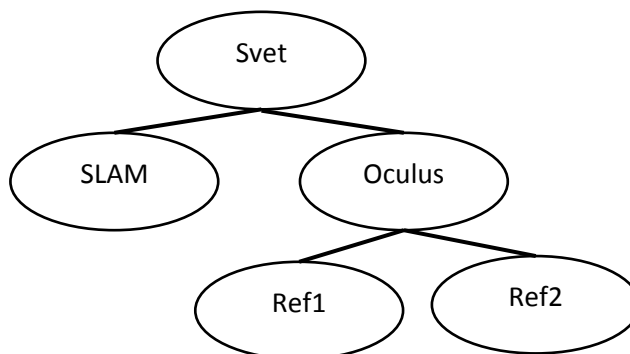
Je treba povedať, že výstupom optických SLAM je virtuálna kamera pohybujúca sa vo virtuálnom priestore. Táto póza kamery teda obsahuje ako pozíciu tak aj orientáciu virtuálnej kamery. Obe sú iba vypočítané odhadované hodnoty. Naproti tomu, Oculus ponúka veľmi presnú a správnu informáciu o orientácii v každom čase počítanú pomocou svojich inerčných senzorov. Práve preto, je veľmi vhodné a potrebné spojenie informácie o orientácii poskytnutej Oculusom s informáciou o zmene pozície kamery poskytnutej optickým SLAM. Problém ktorý je potrebné vyriešiť je znázornení na obrázku nižšie.



Obr. 37 – Problém fúzie dát.

Problém spočíva v zrkadlení pohybu optického SLAM do smeru orientácie Oculusu. Teda vo virtuálnom svete môže mať virtuálna kamera SLAM úplne inú pozíciu a natočenie ako virtuálna kamera Oculusu, ktorá sa nepohybuje, no je správne natočená pretože práma orientáciu z Oculusu. Ďalším problémom je rôzna vlastná súradnicová sústava pre obe kamery kde napríklad pohyb kamery SLAM po osi X zrkadlene znamená pohyb kamery Oculus v zápornom smere po osi

Y. Je teda potrebné spárovať vlastné súradnicové systémy a odzrkadliť pohyb kamery optického SLAM a to takým spôsobom, že pohyb optického SLAM v smere vlastnej už spárovanej osi X, teda dopredu, je odzrkadlený rovnakým pohybom virtuálnej kamery Oculus v smere svojej vlastnej osi X, teda takisto dopredu a obdobne pre všetky osy. Tento proces je pomerne zložitý a časovo veľmi náročný. Proces zahŕňa prácu s modelmi pohybu a orientácie v prostredí ROS kde je poloha vyjadrená súradnicami XYZ vo svete a orientácia je vyjadrená quaternionom. Pri opisovaní je vhodné zobrazíť štruktúru používaného stromu tf koordinačných rámcov:



Obr. 38 – Štruktúra stromu tf koordinačných rámcov fúzie dát.

Všetky bubliny stromu predstavujú tf koordinačné rámce. Svet je základný koordinačný systém v ktorom sa nachádzajú virtuálne kamery SLAM a Oculus vo forme tf koordinačných rámcov. Tf koordinačné rámce Ref1 a Ref2 sú opísané neskôr. Pre teóriu tf vid' kapitolu 2.4.2. Rviz a tf.

Prvým potrebným krokom je získanie vektora posunutia virtuálnej kamery SLAM v každom čase, teda:

$$\Delta p_{Svet}^{SLAM} = (p_{Svet}^{SLAM}(t)) - (p_{Svet}^{SLAM}(t-1))$$

kde Δp_{Svet}^{SLAM} predstavuje vektor posunutia v najmenšom časovom okamihu odpočítaním polohy kamery v predchádzajúcom časovom okamihu $t-1$ od terajšieho časového okamihu t . Ďalším potrebným krokom je získanie rozdielu natočenia tf Oculus od tf SLAM pre získanie rotačnej matice, teda:

$$Q_{Svet}^o = Q_{Svet}^{Oculus} * (Q_{Svet}^{SLAM})^{-1}$$

kde Q_{Svet}^o predstavuje výsledné rozdielové natočenie oboch kamier v rámci Svet čo predstavuje takisto rotačnú maticu. Pre premietnutie pohybu SLAM do Oculus je teda potrebné vynásobenie transponovaného (T) vektora posunutia Δp_{Svet}^{SLAM} a rotačnej matice Q_{Svet}^o :

$$\Delta p_{Svet}^{Temp} = Q_o * (\Delta p_{Svet}^{SLAM})^T$$

kde Δp_{Svet}^{Temp} predstavuje natočený vektor posunutia ktorý je ale ešte potrebné znova natočiť pretože koordinačné rámce Oculus a SLAM nie sú spárované, teda os X pre kameru Oculus, ktorá ukazuje vpred, je v prípade kamery SLAM os Y atď..

Vzniká potreba spárovania koordinačných systémov virtuálnych kamier. Je potrebné získať rozdielový quaternion tohto natočenia vzhľadom na koordinačný systém Oculusu a následne získať transformáciu tohto quaternionu vzhľadom na koordinačný systém Svet. Z tohto dôvodu existujú tri koordinačné rámce Ref1 a Ref2 s materským rámcom Oculus, kde Ref1 predstavuje koordinačný rámec s orientáciou koordinačného rámca Oculus, teda os X znamená vpred atď., a Ref2 predstavuje koordinačný rámec s orientáciou koordinačného rámca SLAM, teda os Y znamená vpred atď. pričom je poloha nepodstatná, zadaná ako nula. Potom pre získanie rozdielového quaternionu natočenia v koordinačnom systéme Oculus:

$$T_{Svet}^{Ref1} = T_{Svet}^{Oculus} * Q_{Oculus}^{Ref1}$$

$$T_{Svet}^{Ref2} = T_{Svet}^{Oculus} * Q_{Oculus}^{Ref2}$$

A keďže $T_a^b = [p_a^b, Q_a^b]$:

$$Q_{Svet}^S = Q_{Svet}^{Ref2} * (Q_{Svet}^{Ref1})^{-1}$$

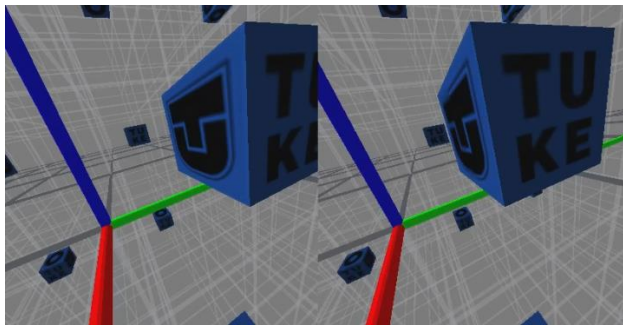
A teda pre získanie konečného vektora posunutia:

$$\Delta p_{Svet}^{Oculus} = (Q_{Svet}^S * \Delta p_{Svet}^{Temp})^T$$

Teda Δp_{Svet}^{Oculus} predstavuje vektor posunutia ktorý je pripočítaný k polohe p_{Svet}^{Oculus} , a teda kamera je posunutá správnym odzrkadleným smerom a fúzia dát je úspešná a správna.

5.2. Testovacia RVIZ TUKE scéna

Za účelom demonštrácie celého systému je vytvorená virtuálna TUKE scéna v prostredí Rviz. Táto scéna je optimalizovaná pre nízku výpočtovú náročnosť.



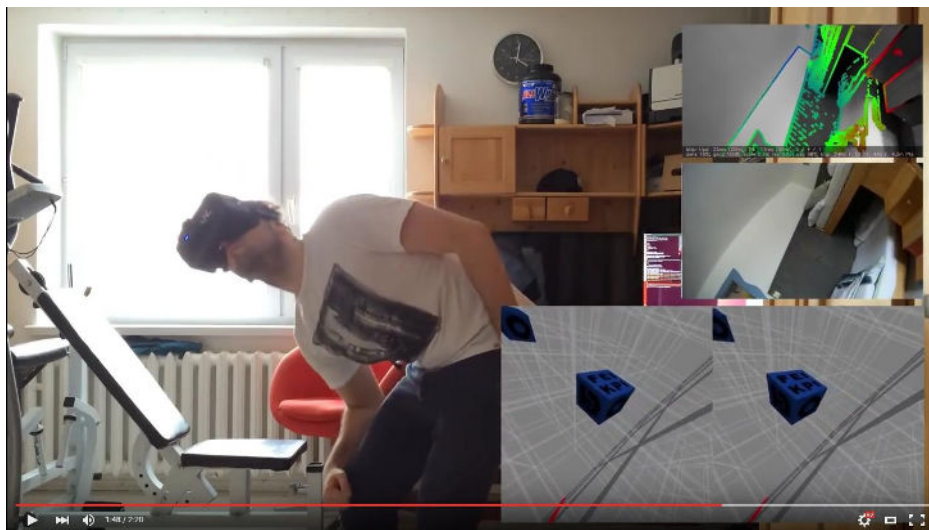
Obr. 39 – Testovacia TUKE Rviz scéna.

5.3. Vykresľovanie na X screen a ladenie rýchlosti

Pre správny a rýchly chod celku, ktorý je veľmi dôležitý a pre užívateľa ohromne dôležitý, je veľké množstvo prvkov optimalizovaných a vyladených. Od Rviz testovacej scény, cez zobrazované prvky Rviz, upravenú obnovovaciu frekvenciu a takisto zobrazovanie a vykresľovanie vstupu pre Oculus v separátnom X screene v linuxe podľa [33]. Takisto sú vyladené a optimalizované komunikačné kanály ROS, komunikácie a všetky prvky tak, aby bol systém rýchly a testovanie bolo naozaj zážitkom pre užívateľa. Takisto sú vyladené a vybrané správne ovládače grafickej karty a nastavenie celého systému a celý systém je plne funkčný, stabilný a vyladený.

5.4. Testovanie celku

Celok je plne funkčný a testovaný. Celý navrhnutý systém pracuje správne a podľa predpokladov. Pri reálnom testovaní sú potvrdené všetky závery z testovania samotných algoritmov. Pre zhladnutie funkčnosti a testovania je zhotovené video dostupné online [36] a takisto na prílohovom DVD.



Obr. 40 – Náhľad demonštračného videa.

6. Záver

Táto práca predstavuje prienik ohromného množstva technológií, postupov, zariadení, platforiem, modulov do funkčného celku ktorý je spustiteľný a dokazuje správnosť konceptu použitia optického SLAM ako pozičného sledovania s premietnutím tohto odhadovaného pohybu do virtuálnej scény dátovej helmy pre virtuálnu realitu.

Výstupom je funkčný systém ktorý dokáže kombinovať orientáciu získanú z Oculusu a polohu virtuálnej kamery vypočítanej optickými SLAM algoritmi pracujúcimi nad obrazovým tokom z kamery pripevnenej na Oculusu a výslednú transformáciu spätne premietajú ako vykresľovanie obrazu v Oculusu pohybujúcom sa vo virtuálnom prostredí.

Pre naštudovanie problematiky virtuálnej reality, zmiešanej reality a spracovania obrazu bola daná problematika detailne preštudovaná a zistená a vytýčená dôležitosť témy problému SLAM ktorá je veľmi detailne a podrobne opísaná v kapitolách 1. Teoretický rozbor SLAM a 2. Analýza SLAM pre naše použitie a opis vybraných algoritmov.

Pre navrhnutie systému spracovania obrazu pre technológiu zmiešanej reality s orientáciou na monokulárnu rekonštrukciu snímaného obrazu z pohľadu pozorovateľa bol navrhnutý systém pripevnenia monokulárnej kamery k Oculusu s následnou rekonštrukciou obrazu formou vykonávania algoritmov optického SLAM nad obrazovým tokom z kamery. Vybrané algoritmy optického SLAM boli Ethzasl-PTAM, LSD-SLAM a S-PTAM. Pre inerčný SLAM bola vybraná dvojitá integrácie akcelerácie na polohu. Daným návrhom sa zaoberajú podkapitoly v kapitole 4. Návrh, implementácia a testovanie podčastí systému.

Pre Implementovanie systému z predchádzajúceho odseku vrátane prípadnej možnosti zmeny algoritmu spracovania scény resp. filtrácie snímaného obrazu je systém implementovaný podľa navrhnutého postupu. Ako prostredie je použitý operačný systém Linux so spustením robotickým operačným systémom ROS a množstvom modulov, podmodulov a postupov. Takisto je implementovaná detailná predfiltrácia. Danou implementáciou a predfiltráciou sa zaoberajú podkapitoly v kapitole 4. Návrh, implementácia a testovanie podčastí systému a podkapitoly v kapitole 5. Návrh a implementácia fúzie dát a ladenie celku. Najdôležitejšími poznatkami z testovania predfiltrácie je zistenie prospešnosti Gaussovského filtra a Sobelových derivácií.

Pre vykonanie experimentálnych prác s implementovaným systémom vrátane spracovania alebo dotvorenia potrebných modelov sú rozsiahle experimenty vykonané v podkapitolách kapitoly 4. Návrh, implementácia a testovanie podčastí systému. Dotvorením potrebných

modulov a súčastí je sprevádzaná celá práca pričom najdôležitejšie moduly tvoria moduly pre získavanie surových dát z Oculus a fúzia dát z Oculusu s dátami z optických SLAM algoritmov. Poznatky z testovania takisto zahrňujú stanovisko, že takýto optický SLAM je v mnohých prípadoch nestabilný a náchylný k pádu.

Práca sa veľmi detailne zameriava a opisuje každú podstatnú časť potrebných tém. Hlavnou rozobranou témou je problém SLAM respektíve optický SLAM. Obetoval som veľké množstvo času pre takýto rešerš danej témy aj s ohľadom na súvisiace témy. Je pravdou že práca s takýmto obrovským množstvom možností bola veľmi obtiažna a bola potreba správneho uvažovania a množstva rozhodnutí.

Práca je rozsiahla aj čo sa týka použitých zariadení a bolo potrebné vyskúšanie a hlavne zohnanie v celku veľkého množstva zariadení. Rád by som vyzdvihol vlastnú iniciatívu a zaobstaranie si Oculus Rift DK2 ktorý je v súčasnosti veľmi nedostupný. Takisto bolo potreba všetky testované kamery požičať a vyskúšať čo je aj spolu s Oculusom pomerne finančne a hlavne časovo náročné.

Pri práci boli vyskúšané takisto rôzne iné technológie ktoré ale použité neboli no takisto zabrali väčšie množstvo času ako skúšanie rôznych platforiem, Unreal Engine, a pri riešení každého problému skúšanie rôznych ciest. Vytýčená cesta je výsledkom veľkého zapálenia a skúšania už spomínaných ciest. Viaceré problémy ako získavanie dát z Oculusu, vykresľovanie pre Oculus, fúzia dát a iné vyžadovali ohromné množstvo času. Takisto témy ktorým nie je v práci venovaná veľká pozornosť ako ladenie komunikácie, tokov, X screen atď. sú naozaj veľkými a dôležitými problémami ktoré boli vyriešené.

Pri práci som komunikoval takisto s mnohými odborníkmi z daných oblastí. Takisto boli pokusy o získanie profesionálnej kamery a množstvo ďalších komunikácií. Rád by som vyzdvihol fakt, že pri práci som čerpal takmer výhradne z odborných a akademických kruhov a publikácií.

Čo sa týka prínosu, tak veľkým prínosom okrem dokázania konceptu je takisto poukázanie správnosti predfiltrácie. Po veľkom počte uskutočnených testov je možné povedať že Gaussovský filter a Sobelové derivácie sú nápomocné a vo väčšine prípadov vylepšujú výkon optických SLAM algoritmov.

Práca načrtla ďalšie cesty akými je možné sa uberať a uvažovať a vytyčuje témy ktorými je možné sa v budúcnosti zaoberať ako predfiltrácia obrazu pred spracovaním či vytváranie vlastných optických SLAM algoritmov.

Zoznam použitej literatúry

- [1]. FUENTES-PACHECO, J. – RUIZ-ASCENCIO, J. – RENDÓN-MANCHA, J.: Visual simultaneous localization and mapping: a survey. [Online] Mexiko: Centro Nacional de Investigación y Desarrollo Tecnológico, 2012. Dostupné na internete: <<http://link.springer.com/article/10.1007%2Fs10462-012-9365-8>>.
- [2]. ZHANG, J. – SINGH, S.: LOAM: Lidar Odometry and Mapping in Real-time. [Online], Berkeley: Robotics: Science and Systems Conference, 2014. Dostupné na internete: <http://www.frc.ri.cmu.edu/~jizhang03/Publications/RSS_2014.pdf>.
- [3]. GÄRTNER, J.: Provedení rešeršní studie v oblasti metod simultánní lokalizace a mapování. [Online], Brno: Vysoké učení technické v Brně, 2008. Dostupné na internete: <https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=6471>.
- [4]. SOBOTA, B. – HROZEK, F.: Systémy virtuálnej reality. [Online], Košice: Technická univerzita v Košiciach, 2015. Dostupné na internete: <http://hornad.fei.tuke.sk/predmety/svr/doc/SVR_ucebnica_2015_v1.pdf>.
- [5]. ZHANG, J. – KAESS, M. - SINGH, S.: LOAM: Real-time Depth Enhanced Monocular Odometry. [Online], Chicago: IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), 2014. Dostupné na internete: <http://www.frc.ri.cmu.edu/~jizhang03/Publications/IROS_2014.pdf>.
- [6]. ZHANG, J. - SINGH, S.: LOAM: Visual-lidar Odometry and Mapping: Low-drift, Robust, and Fast. [Online], Seattle: IEEE Intl. Conf. on Robotics and Automation (ICRA), 2015. Dostupné na internete: <http://www.frc.ri.cmu.edu/~jizhang03/Publications/ICRA_2015.pdf>.
- [7]. ZHANG, J. - SINGH, S.: Visual-Inertial Combined Odometry System for Aerial Vehicles. [Online], Journal of Field Robotics. vol. 32, no. 8, pp. 1043–1055, 2015. Dostupné na internete: <http://www.frc.ri.cmu.edu/~jizhang03/Publications/JFR_2015.pdf>.
- [8]. AULINAS, J. - PETILLOT, Y. – SALVI, J. – LLADÓ, X.: The SLAM problem: a survey. [Online], Girona: University of Girona. Edinburgh: Heriot-Watt University, 2008. Dostupné na internete: <<http://eia.udg.es/~qsalvi/papers/2008-CCIAa.pdf>>.
- [9]. Internetová adresa Robotického Operačného Systému. [Online]. Dostupné na internete: <<http://www.ros.org/>>.
- [10]. GEIGER, A. - LENZ, P. – URTASUN, R.: Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. [Online], Karlsruhe Institute of Technology, Toyota Technological Institute at Chicago, 2012. Dostupné na internete: <<http://www.cvlibs.net/publications/Geiger2012CVPR.pdf>>.

- [11]. Internetová adresa The KITTI Vision Benchmark Tool. [Online]. Dostupné na internete: <http://www.cvlibs.net/datasets/kitti/>.
- [12]. ENGEL, J. – STUCKLER, J. – CREMERS, D.: Stereo Parallel Tracking and Mapping for robot localization. [Online], Munchen: Technische Universität München, 2015. Dostupné na internete: https://vision.in.tum.de/_media/spezial/bib/engel2015_stereo_lsdlam.pdf.
- [13]. PIRE, T. – FISCHER, T – et al.: Nazov. [Online], Zaragoza: Universidad de Zaragoza, 2015. Dostupné na internete: http://webdiis.unizar.es/~jccivera/papers/pire_etal_iros15.pdf.
- [14]. KLEIN, G. – MURRAY, D.: Parallel Tracking and Mapping for Small AR Workspaces. [Online], Oxford: Active Vision Laboratory, University of Oxford, 2007. Dostupné na internete: https://www.robots.ox.ac.uk/~vgg/rg/papers/klein_murray__2007__ptam.pdf.
- [15]. WEISS, S.: Vision Based Navigation for Micro Helicopters. [Online], Zurich: University of Zurich, 2012. Dostupné na internete: <http://e-collection.library.ethz.ch/eserv/eth:5889/eth-5889-02.pdf>.
- [16]. Internetová adresa Ethzasl-PTAM knižnice pre ROS. [Online]. Dostupné na internete: http://wiki.ros.org/ethzasl_ptam.
- [17]. ENGEL, J. – SCHOPS, T. – CREMERS, D.: LSD-SLAM: Large-Scale Direct Monocular SLAM. [Online], Munich: Technical University Munich, 2014. Dostupné na internete: http://vision.in.tum.de/_media/spezial/bib/engel14eccv.pdf.
- [18]. Internetová adresa Algoritmu pána Seb Madwick. [Online]. Dostupné na internete: <http://www.x-io.co.uk/gait-tracking-with-x-imu/>.
- [19]. STEWENIUS, H. - ENGELS, C. - NISTER, D.: Recent developments on direct relative orientation. ISPRS Journal of Photogrammetry and Remote Sensing, 60:284–294, June 2006.
- [20]. Internetová adresa internetovej stránky výskumníka Guoxuan Zhang. [Online]. Dostupné na internete: <https://scholar.google.com/citations?user=wJ7C09oAAAAJ&hl=en>.
- [21]. ENGEL, J. – STURM, J. – CREMERS, D.: Semi-Dense Visual Odometry for a Monocular Camera. [Online], Munchen: Technical University München, 2013. Dostupné na internete: http://vision.in.tum.de/_media/spezial/bib/engel2013iccv.pdf.
- [22]. STURM, J. - ENGELHARD, L. – et al.: A benchmark for the evaluation of RGB-D SLAM systems. In Intelligent Robot Systems (IROS), 2012. 2, 7.
- [23]. MATTHIES, L. – SZELISKI, R. – KANADE, T.: Incremental estimation of dense depth maps from image image sequences. In CVPR, 1988.
- [24]. Internetová adresa Wikipédia stránky o IMU. [Online]. Dostupné na internete: https://en.wikipedia.org/wiki/Inertial_measurement_unit.

- [25]. Internetová adresa podrobné rozobratie Oculus Rift DK2. [Online]. Dostupné na internete: <<https://www.ifixit.com/Teardown/Oculus+Rift+Development+Kit+2+Teardown/27613>>.
- [26]. Internetová adresa dostupného open source kódu LSD-SLAM. [Online]. Dostupné na internete: <https://github.com/tum-vision/lst_slam>.
- [27]. Internetová adresa druhého algoritmu pána Seb Madwick. [Online]. Dostupné na internete: <<http://www.x-io.co.uk/oscillatory-motion-tracking-with-x-imu/>>.
- [28]. Internetová adresa kalibrácie kamery pre PTAM. [Online]. Dostupné na internete: <http://wiki.ros.org/ethzasl_ptam/Tutorials/camera_calibration>.
- [29]. Internetová adresa odporúčaných nastavení PTAM od samotných autorov. [Online]. Dostupné na internete: <https://github.com/ethz-asl/ethzasl_ptam/issues/19>.
- [30]. Internetová adresa kalibrácie stereo kamery v prostredí ROS. [Online]. Dostupné na internete: <http://wiki.ros.org/camera_calibration/Tutorials/StereoCalibration>.
- [31]. Internetová adresa pluginu pre Rviz podporujúceho Oculus v prostredí ROS. [Online]. Dostupné na internete: <https://github.com/OSUrobotics/oculus_rviz_plugins>.
- [32]. FOOTE, T.: tf: The Transform Library. [Online], California: Open Source Robotics Foundation, 2013. Dostupné na internete: <http://wiki.ros.org/Papers/TePRA2013_Foote>.
- [33]. Internetová adresa postupu správneho používania X screen v Linuxe. [Online]. Dostupné na internete: <<https://codelab.wordpress.com/2015/04/02/proper-oculus-rift-dk2-setup-on-gnulinux/>>.
- [34]. MONTEMERLO, M. – THRUN, S. – KOLLER, D. – WEGBREIT, B.: FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. [Online], Pittsburgh: Carnegie Mellon University. Stanford: Stanford University, 2002. Dostupné na internete: <<http://ai.stanford.edu/~koller/Papers/Montemerlo+al:AAAI02.pdf>>.
- [35]. MONTEMERLO, M. – THRUN, S. – KOLLER, D. – WEGBREIT, B.: FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges. [Online], Pittsburgh: Carnegie Mellon University. Stanford: Stanford University, 2003. Dostupné na internete: <<http://www.cs.cmu.edu/~mmde/mmdeijcai2003.pdf>>.
- [36]. Internetová adresa demonštračného videa tejto práce. [Online]. Dostupné na internete: <<https://www.youtube.com/watch?v=lyMMbEEbLq4>>.

Prílohy

- Príloha A: DVD médium – diplomová práca v elektronickej podobe, prílohy v elektronickej podobe, zdrojové kódy, testy, videá, demonštračné videá, odborný článok. Použité sú 3 DVD média.
- Príloha B: Systémová príručka
- Príloha C: Používateľská príručka
- Príloha D: Systémová a Používateľská príručka S-PTAM