

БЕЛАРУСКІ ДЗЯРЖАЎНЫ УНІВЕРСІТЭТ

Факультэт прыкладной матэматыкі і інфарматыкі

Кафедра дыскрэтнай матэматыкі і алгарытмікі

**Справаздача
аб праходжанні пераддыпломнай практыкі**

студэнта 4 курса
Богдана Уладзіслава Уладзіміравіча
спецыяльнасць “інфарматыка”

Кіраўнік практыкі:
асістэнт Жылка Андрэй Ігаравіч

Мінск, 2018

ЗМЕСТ

УВОДЗІНЫ	3
ГЛАВА 1. ЗАДАЧА РЭКАНСТРУКЦЫІ ПАВЕРХНІ	4
1.1 Агульныя звесткі	4
1.2 Сферы прымянення	4
ГЛАВА 2. SLAM-АЛГАРЫТМЫ	5
2.1 Агляд	5
2.1.1 Лакалізацыя	6
2.1.2 Пошук на мапе	6
2.2 Падыходы да рэалізацыі	7
2.3 Агляд SLAM-рэалізацыяў	7
2.3.1 ORB-SLAM	7
2.3.3 LSD-SLAM	9
2.3.2 SVO	10
ГЛАВА 3. ПРАКТЫЧНАЯ РЭАЛІЗАЦЫЯ	12
3.1 ROS	12
3.2 Архітэктурa сістэмы	13
3.3 Тыпы ўваходных інтэрфейсаў	14
3.4 Выманне дадатковых дадзеных са SLAM-алгарытмаў . .	14
3.5 Апра і інтэрфейс для дадзеных са SLAM-а	15
3.6 Аптымізацыя камунікацыяў з дапамогай ROS-а	16
ВЫСНОВЫ	17
СПІС КРЫНІЦАЎ	18

УВОДЗІНЫ

Задача рэканструкцыі паверхні зямлі па дадзеных з беспілотных лятальных апаратаў (БПЛА) узнікае ўсё ў большай колькасці сфераў жыццядзейнасці: ад сельскай гаспадаркі да ацэнкі наступстваў прыродных катастрофаў. Патрэбнасць у эфектыўным рашэнні задачы таксама звязаная з шырокай даступнасцю БПЛА і з распаўсюдам танных камераў.

Задача рэканструкцыі цесна звязаная са шматлікімі сумежнымі задачамі: распазнаванне аб'ектаў, навігацыя ў прасторы, пабудова мапы мясцовасці. Асаблівую цікаўнасць прадстаўляюць рашэнні, якія выконваюцца ў рэальным часе; праца ў рэальным часе для алгарытмаў навігацыі і абхода перашкодаў можа быць крытычнай для аўтаномных БПЛА, у адрозненні ад наземных робатаў, якія могуць на нейкі час спыніцца і дачакацца пабудовы маршрута.

Падчас праходжання пераддыпломнай практыкі я працягнуў даследаваць тэорыю ў галіне трохмернай рэканструкцыі, шчыльна пазнаёміўся з канцэпцыяй SLAM (Simultaneous Localization and Mapping) алгарытмаў, а таксама часткова рэалізаваў сістэму, якая для эфектыўнасці рэканструкцыі спалучае ў сабе SLAM-частку і больш традыцыйныя метады рэканструкцыі паверхні.

ГЛАВА 1.

Задача рэканструкцыі паверхні

1.1 Агульныя звесткі

У агульным выпадку задача рэканструкцыі паверхні фармулюецца наступным чынам: неабходна рэканструяваць трохмерны аб'ект па мностве здымкаў зробленых з розных ракурсаў. Задача досыць натуральная і калі чалавеку дастаткова кінуць позірк на аб'ект, каб уявіць ягоную трохмерную структуру, алгарытмічна задача ўсё яшчэ застаецца не цалкам вырашанай, патрабуе досыць вялікіх вылічальных магутнасцяў і не працуе ўніверсальна добра для любых асяроддзяў і любых умоваў здымак.

Задача рэканструкцыі можа таксама фармулявацца і для іншых тыпаў уваходных дадзеных, для дадзеных з іншых датчыкаў, такіх як акселерометр, гіраскоп ці GPS-датчык; замест манакулярнай камеры можа прымяняцца RGB-D (вяртае дадатковы слой глыбіняў) ці стэрэакамера, якая ў пэўным сэнсе імітуе бінакулярны чалавечы зрок.

У залежнасці ад таго, якія выходныя дадзеныя нас цікавяць, алгарытм можа будаваць воблака кропак рознай шчыльнасці альбо будаваць шчыльную тэкстураваную мадэль, максімальна падобную на рэальны трохмерны аб'ект. Розныя патрабаванні да выходных дадзеных патрабуюць адрозных алгарытмічных рашэнняў і наяўнасці розных вылічальных магутнасцяў.

Варта дадаць, што даследаванні ў гэтай галіне кампутарнага зроку развіваюцца таксама праз удасканаленне апаратнага забеспячэння: падыходы, якія некалькі год таму былі практычна нерэалізаванымі, з развіццём тэхналогіяў даюць ім новае жыццё.

1.2 Сферы прымянення

Цікавасць задачы рэканструкцыі таксама ў запатрабаванні атрымання рашэння ў абсалютна розных сферах жыцця, у кожнай на сваімі асаблівасцямі. БПЛА выкарыстоўваюцца надзвычайнымі службаў для ацэнкі наступстваў прыродных катастроф, у сельскай гаспадарцы, дарожнымі службаў. Патрабаванні да хуткасці працы алгарытмаў рэканструкцыі натуральныя - хуткасць працы ў некаторых галінах жыцця крытычная і разбор вялікіх аб'ёмаў неапрацаваных дадзеных чалавекам можа быць немагчымы. Патрабаванні да працы алгарытмаў рэканструкцыі ў рэальным часе ў большасці выпадкаў з'яўляюцца пры навігацыі і аўтаномным руху, у такім выпадку шчыльная рэканструкцыя можа быць залішняй і разрэджаная мадэль у выглядзе воблака кропак цалкам задаволіць патрабаванні.

ГЛАВА 2.

SLAM-АЛГАРЫТМЫ

2.1 Агляд

Задача адначасовай лакалізацыі і пошуку на мапе (Simultaneous Localization and Mapping, SLAM) спалучае ў сабе выкарыстанне разнастайных датчыкаў (лазерныя сканеры, RGB альбо RGB-D камеры і іншае) з мэтай ацэнкі пазіцыі ў прасторы і адначасовай пабудовы мапы мясцовасці. Задача фармулюецца як для двухмерных, так і для трохмерных асяроддзяў: нас больш цікавіць трохмерная задача, тым больш што двухмерная версія задачы лічыцца вырашанай. З адначасовай пабудовы мапы і ацэнкі пазіцыі ў прасторы вынікае патрэбаваць да алгарытма працаваць у рэальным часе, што накладвае асаблівыя патрабаванні да алгарытмаў і абсталявання, на якім алгарытмы запускаюцца.

Сваё найбольшае развіццё рашэнне SLAM задачы атрымала ў апошнія дзесяцігоддзі. Даследванне задачы можна ўмоўна падзяліць на тры этапы (прапанаваныя ў [1], [2]). Першы перыяд, прыблізна 1986-2004, можна назваць “класічным”. У гэтыя часы асноўным накірункам працы па рашэнні задачы была імавернасная фармуліроўка і падыходы, заснаваныя на фільтрах. У другі перыяд, 2004-2015, былі даследаваныя асноўныя ўласцівасці і падыходы, такія як збежнасць і ўзгодненасць алгарытмаў. У гэты час былі праведзеныя цесныя аналогія з існуючымі падыходамі ў галіне кампутарнага зроку.

Мяркуюцца, што на трэці этап распрацоўка перайшла зусім нядаўна, у апошнія гады: гэта звязваюць з публікацыямі такіх алгарытмаў, як ORB-SLAM альбо LSD-SLAM, якія па сваёй функцыянальнасці моцна пераўзыходзяць любыя ранейшыя публікацыі. Прынцып працы гэтых алгарытмаў будзе разгледжаны далей.

Важна успрымаць SLAM не як адзіны суцэльны алгарытм, але як канцэпцыю. SLAM складаецца са шматлікіх этапаў і кожны з гэтых этапаў можа вырашацца рознымі алгарытмамі з той ці іншай эфектыўнасцю.

Для задачы рэканструкцыі паверхні SLAM-сістэма наўрад ці будзе аптымальным рашэннем: пабудова шчыльных мадэляў у рэальным часе пакуль яшчэ застаецца слаба вырашанай задачай. Тут і далей SLAM-сістэмы нас цікавяць сваімі выходнымі дадзенымі: пасля апрацоўкі дадзеных у рэальным часе захаваныя пазіцыі, павароты камераў у часе і пабудаваная глабальная мапа могуць быць эфектыўна выкарыстаныя ў якасці ўваходных дадзеных афлайн алгарытма, падвышаючы ягоную дасканаласць і хуткасць працы.

Як вынікае з назвы, алгарытмы рашаюць дзве асноўныя задачы: лакалізацыю і пошук на мапе. На першых этапах даследавання задачы разглядаліся асобна, але з цягам часу стала зразумела, што задачы цесна звязаныя і рашэнне адной з іх моцна дапамагае ў рашэнні іншай. Мапа патрэбная для эфектыўнага пошуку пазіцыі ў прасторы, у той жа час з ацэнкай становішча мапа сабірае больш дадзеных і становіцца больш дакладнай. Асобнай задачай заўжды стаіць магчымасць алгарытма замыкаць цыклы (*loop closure*).

Разгледзім асноўныя этапы падрабязней.

2.1.1 Лакалізацыя

Пры руху пазіцыя БПЛА ў дыскрэтным часе задаецца матрыцай $T \in \mathbb{R}^{4 \times 4}$. Паслядоўнасць такіх матрыцаў - траекторыя пралёту. Лакалізацыя - пошук такой матрыцы ў вызначаны момант часу. Для адкрытых прастораў задача была часткова вырашаная з запускам GPS - глабальнай сістэмы пазіцыянавання; у прасторах, дзе GPS недаступны, таксама можа выкарыстоўвацца вонкавае абсталяванне для лакалізацыі, што часта з'яўляецца непрактычным альбо дарагім рашэннем. Выкарыстанне датчыкаў, усталяваных на БПЛА, для лакалізацыі ў прасторы, робіць падыходы да рашэння задачы больш універсальнымі. Лакалізацыю з дапамогай толькі камераў часам таксама называюць візуальнай адыметрыяй (*Visual Odometry*). Надзейныя механізмы лакалізацыі аўтаномных БПЛА асабліва важныя праз знаходжанне БПЛА ў паветры і немагчымасць спыніцца, у адрозненні на наземных робатаў.

2.1.2 Пошук на мапе

Ёсць некалькі важных аспектаў пабудовы мапы. Па-першае, мапа выкарыстоўваецца для пракладкі маршрутаў і лакалізацыі перашкодаў - асноўныя элементы для аўтаномнай навігацыі. Па-другое - мапа на выхадзе можа быць цікавая сама па сабе. Яна можа пасля выкарыстоўвацца для візуалізацыі пралёта па мясцовасці альбо як уваходныя дадзеныя для алгарытмаў афлайн-рэканструкцыі, для распазнавання аб'ектаў і гэтак далей. Трэці і, магчыма, самы важны аспект у дачыненні да канцэпцыі SLAM - добра пабудаваная мапа дазваляе лакалізаваць БПЛА ў прасторы.

Такім чынам, лакалізацыя і пошук на мапе не выступаюць як дзве асобныя і незалежныя часткі алгарытма, але дапаўняюць і паляпшаюць адно аднаго. Пошук сябе на папярэдне пабудаванай мапе (напрыклад, пры паўторным пралёце той жа мясцовасці) дазваляе лакалізаваць БПЛА з якасцю, параўнальнай з якасцю пабудаванай мапы, у сваю чаргу

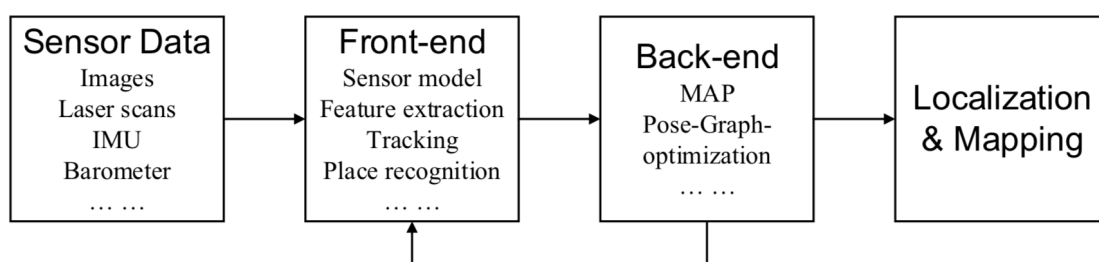
лакалізацыя пры дапамозе адаметрыі і старонніх датчыкаў паслядоўна ўдасканальвае мапу.

2.2 Падыходы да рэалізацыі

Як зазначаецца ў [2], архітэктара любой SLAM-сістэмы можа быць прадстаўленая схемай як на малюнку 1. Фронтэнд выконвае першасную апрацоўку дадзеных, якія паступаюць са знешніх датчыкаў. Для алгарытмаў, заснаваных на пошуку асаблівых кропак, фронтэнд займаецца іх пошукам і апісаннем; для алгарытмаў, якія працуюць непасрэдна са значэннямі пікселяў, выконваецца трэкінг паміж кадрамі. Усе вынікі першаснай апрацоўкі пасля перадаюцца бэкэнду.

Бэкэнд, у сваю чаргу, займаецца пабудовай графа залежнасцяў, аптымізацыяй на графе.

Важным крокам у рэалізацыі SLAM-алгарытмаў стала дамінаванне ідэі раздзялення лакалізацыі і пошуку на мапе ў асобныя плыні. Упершыню практычнае прымяненне ідэя знайшла ў сістэме PTAM (*Parallel Tracking and Mapping for Small AR Workspaces*, [3]). Лакалізацыйная плыня PTAM займаецца пошукам адпаведнасцяў у дадзеных і запуская аптымізацыю па рухах камеры. У гэты ж час плыня з мапай калекцыянуе вынікі лакалізацыі, трыангулюе выяўленыя асаблівасці ў трохмерныя кропкі і абнаўляе глабальную мапу. Са з'яўленнем такога падыходу сталі відавочнымі ягонныя перавагі і амаль кожная сучасная SLAM-сістэма пабудаваная па падобным прынцыпе.



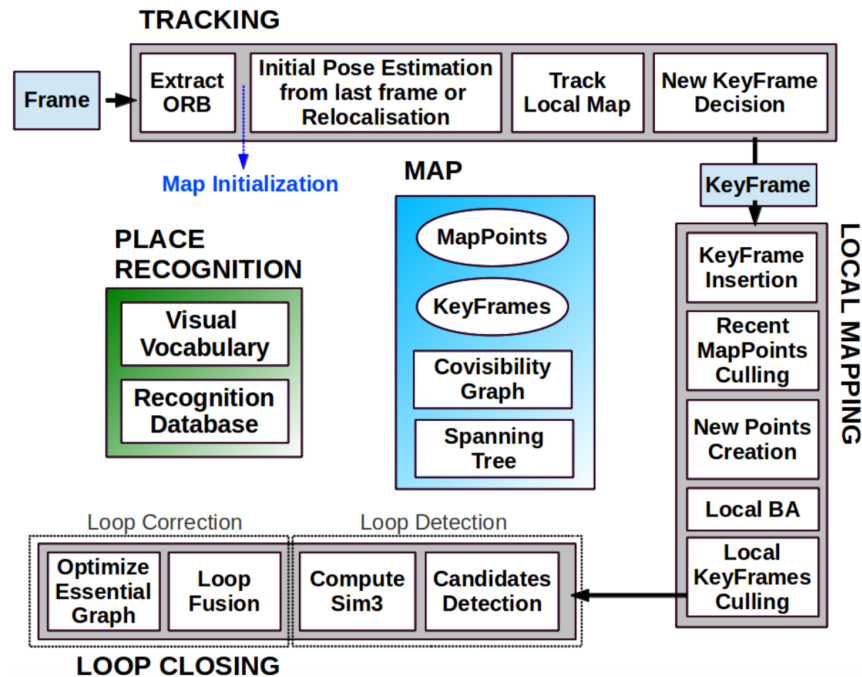
Малюнак 1: архітэктара SLAM-алгарытмаў

2.3 Агляд SLAM-рэалізацыяў

Далей ідзе апісанне асаблівасцяў некалькі SLAM-сістэм, якія паказваюць найлепшыя вынікі. Адна з іх заснаваная на пошуку асаблівых кропак (*feature-based SLAM*), другая - на простых метадах (*direct SLAM*), трэцяе - камбінуе ў сабе абодва падыхода (*semi-direct SLAM*).

2.3.1 ORB-SLAM

ORB-SLAM ([4], [5]) - манакулярная SLAM-систэма, заснаваная на пошуку асаблівасцяў на выявах (*feature-based*), спраектаваная для працы ў маленькіх і вялікіх, замкнёных і адкрытых прасторах.



Малюнак 2: ORB-SLAM

Асноўныя асаблівасці:

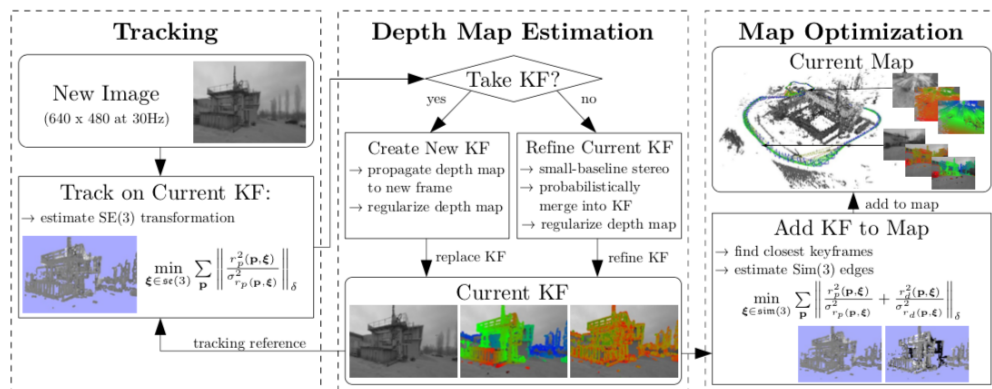
- выкарыстанне адных і тых жа ключавых кропак на ўсіх этапах працы алгарытма;
- чатыры асноўныя задачы, якія адначасова рашае алгарытм: трэкінг (*tracking*), нанясенне дадзеных на мапу (*mapping*), рэлакалізацыя (*relocalization*), замкненне цыклаў (*loop closing*);
- неабмежаваны рост памераў мапы *толькі* з ростам аглядаемай тэрыторыі;
- выкарыстанне ORB-дэскрыптараў як найлепшых па хуткасці (без ужывання GPU) і інварыянтнасці да зменаў у павароце і асвятленні;
- хуткі трэкінг і пошук на мапе на лакальных абласцях бачнасці, што дасягаецца праз выкарыстанне *графы бачнасці* (*covisibility graph*);
- новы аўтаматычны і надзейны спосаб ініцыялізацыі, які стварае пачатковыя мапы як для планарных, так і непланарных паверхняў;

- пазбаўленне ад лішніх ключавых кадраў па стратэгіі “выжывання наймацнейшага” (*survival of the fittest*);
- агульная схема працы алгарытма прадстаўленая на малюнку 2.

Арыгінальная публікацыя [4] прыводзіла апісанне манакулярнай сістэмы, у другой рэдакцыі алгарытма (ORB-SLAM2, [5]) дадалася падтрымка стэрэа і RGB-D камер. Пра рэалізацыю варта зазначыць, што першая версія была даступная толькі пад ROS, у сваю чаргу ў другой версіі з’явіўся ўласны візуалізатар, здольны працаваць незалежна.

2.3.3 LSD-SLAM

У адрозненні ад разгледжанай вышэй сістэмы ORB-SLAM, заснаванай на пошуку асаблівых кропак, LSD-SLAM ([6]) выкарыстоўвае *простыя* (*direct*) падыходы да працы з выявамі, то бок працуе непасрэдна са значэннямі ў пікселях.



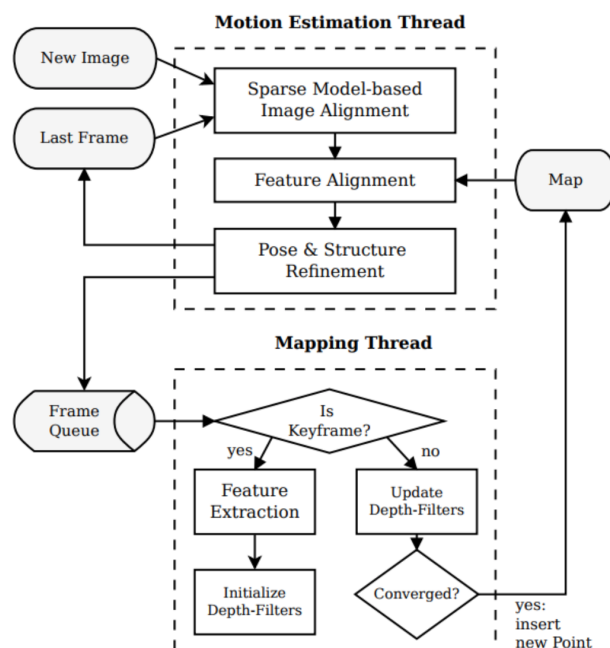
Малюнак 3: LSD-SLAM

- працуе ў рэальным часе на CPU;
- складаецца з трох асноўных кампанентаў: трэкінг, падлік набліжэння для мапы глыбіняў і аптымізацыя мапы;
- трэкінг кампанента паслядоўна апрацоўвае новыя здымкі, прымаючы да ўвагі папярэдні і бягучы ключавы здымкі;
- другая кампанента выкарыстоўвае здымкі каб альбо палепшыць, альбо замяніць бягучы ключавы кадр і ацэньвае глыбіню сцэны простымі (*direct*) і імавернаснымі падыходамі;
- здымкі, якія канчаткова пазначаныя як ключавыя (мапа глыбіняў больш змяняцца не будзе) заносзяцца на глабальную мапу;

- мапа - граф узаемаразмяшчэнне ключавых кадраў. Паколькі алгарытм таксама ацэньвае мапы глыбіняў, то мапа, па сутнасці, ўяўляе сабой напauшчыльную трохмерную рэканструкцыю паверхні;
- аўтары ініцыялізуюць першы ключавы кадр *выпадковай* мапай глыбіняў з вялікімі перападамі. Сцвярджаецца, што пры значных рухах камеры ў прасторы ў першую секунду, першасная мапа глыбіняў будзе хутка збягацца. Аўтары прызнаюцца ў неабходнасці дапрацаваць гэтую частку сістэмы; пры тэставых запусках мапа глыбіняў для ініцыялізацыі бралася непасрэдна з датасэтаў (то бок была дакладна вядомая), на практыцы падобны падыход можа запатрабаваць пэўнага чалавечага ўдзелу;
- архітэктара сістэмы прадстаўленая на малюнку 3.

2.3.2 SVO

У адрозненні ад папярэдніх двух прыкладаў, SVO (*Semi-Direct Monocular Visual Odometry*, [7]) спалучае ў сабе як пошук і апісанне ключавых кропак, так і простыя метады.



Малюнак 4: SVO

- спалучае ў сабе лепшае ад метадаў, заснаваных на ключавых кропках (выбар ключавых кадраў, сачэнне за мноствам кропак, паралельны трэкінг і пошук на мапе), і ад простых метадаў (хуткасць і дакладнасць);

- пошук адпаведнасцяў паміж ключавымі кропкамі здзяйснення няўна як вынік прымянення простых метадаў (а не, напрыклад, matching-a);
- выманне ключавых кропак здзяйснення толькі на ключавых кадрах;
- трохмерная кропка дадаецца на мапу толькі ў выпадку збежнасці адпаведнага фільтра глыбіняў, што азначае патрэбнасць у шматлікіх вымярэннях для кожнай асобнай кропкі;
- свядзенне да мінімуму выкідаў у вымярэннях;
- архітэктара сістэмы прадстаўленая на малюнку 4.

ГЛАВА 3.

ПРАКТЫЧНАЯ РЭАЛІЗАЦЫЯ

Рэалізацыі алгарытмаў, разгледжаных у папярэдняй главе, даступныя ў вольным доступе як праекты з адкрытым зыходным кодам. Аўтары каментуюць публікацыю як адначасова унёсак у SLAM супольнасць і сродак для даследчыкаў з памежных галінаў [5].

У супольнасці робататэхнікаў папулярнасцю карыстаецца ROS - аперацыйная сістэма для распрацоўкі прыкладанняў для робатаў. Праз тое, што SLAM-алгарытмы ў шматлікіх выпадках знаходзяць прымяненне ў галіне робататэхнікі, большасць рэалізацыяў SLAM-алгарытмаў мае падтрымку інтэрфейсаў ROS-а, а ў некаторых выпадках гэта з’яўляецца наогул адзіным рэжымам працы.

Канечная мэта маёй працы - пабудова эфектыўнай сістэмы аўтаматычнай рэканструкцыі трохмерных паверхняў па дадзеных з беспілотных лятальных апаратаў; у сваіх даследаваннях я рабіў упор на спалучэнне SLAM-рэалізацыяў з “класічнымі” падыходамі. Пры рэалізацыі я сутыкнуся з наступнымі аспектамі, пра якія мне хацелася б распавесці ў асобных пунктах ніжэй.

3.1 ROS

ROS (The Robot Operating System) - фрэймворк для напісання праграмага забеспячэння для робатаў, з’яўляецца наборам інструментаў, бібліятэкаў і канвенцыяў для зручнай і надзейнай працы з робатамі ([8]). Патрэбнасць у фрэймворку ўзнікла праз той факт, што стварэнне праграмага забеспячэння для робатаў - вельмі складаны і шматэтапны працэс; уніфікацыя спосабу камунікацыяў паміж часткамі вялікай сістэмы дазваляе камандам працаваць над навігацыяй, рухам і зрокам робата асобна.

ROS пабудаваны на ідэі кліент-сервернага ўзаемадзеяння, што спрашчае працу ў размеркаваных сістэмах, хаця і можа часам падавацца залішнім пры запуску на адной лакальнай машыне.

У аснове ROS ляжаць:

- нізкаўзроўневы інтэрфейс перадачы паведамленняў,
- ананімная і асінхронная сістэма публікацыі і падпіскі на паведамленні,
- магчымасць рэалізацыі RPC для сінхроннага выкліку метадаў іншых пакетаў,
- глабальнае сховішча ключэй і значэнняў,

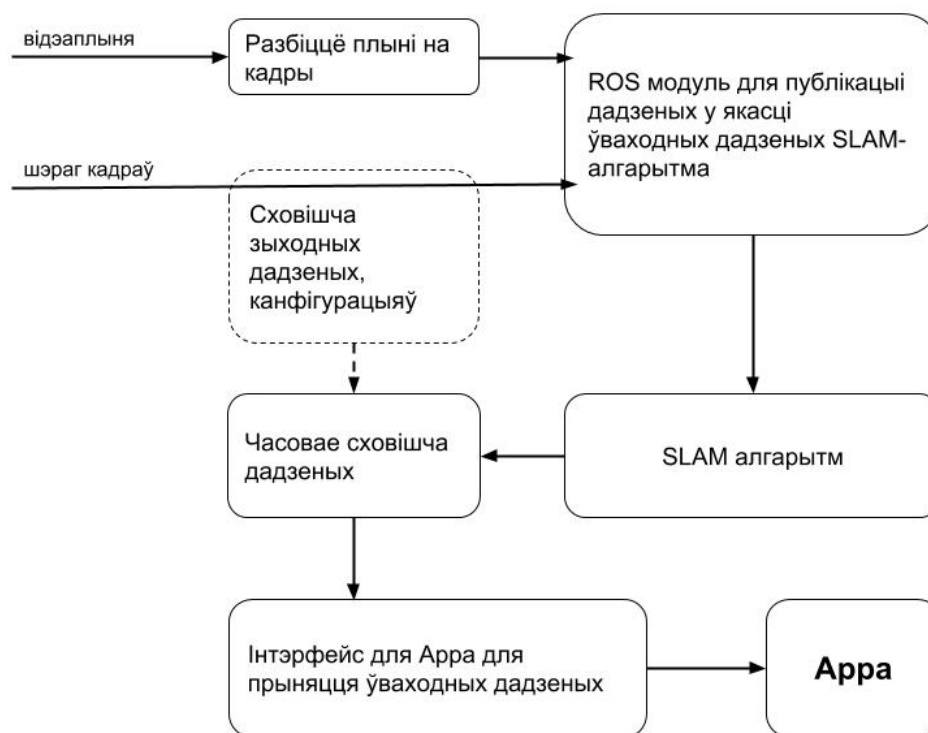
- зручныя сродкі дыягностыкі,
- інструменты для працы з ROS-ам праз графічны інтэрфейс: *rviz*, *rqt*.

Усе гэтыя ўласцівасці ROS-а робяць распрацоўку зручнай і лёгка ўбудовуемай ў іншыя сістэмы. Вялікая колькасць SLAM-алгарытмаў рэалізаваных менавіта пад ROS і могуць не мець уласных сродкаў візуалізацыі альбо ўводу дадзеных.

Падчас рэалізацыі ўласнага праграмнага забеспячэння мне давялося шчыльна пазнаёміцца з ROS-ам і прынцыпамі арганізацыі працы ў гэтай аперацыйнай сістэме.

3.2 Архітэктара сістэмы

Архітэктара сістэмы, рэалізацыяй якой я займаўся, і ўсе яе ключавыя вузлы схематычна прадстаўлены на малюнку 5.



Малюнак 5: архітэктара спалучанай сістэмы

Тут і далей, *Арпа* ([9]) - праграмнае забеспячэнне для рэканструкцыі паверхні па зыходных дадзеных у выглядзе набору малюнкаў. Распрацоўка праграмы пачалася ў межах курсавой працы і працягваецца

зараз. Асноўныя магчымасці праграмы: выманне ключавых кропак, ажыццяўленне пошуку адпаведнасцяў, рэканструкцыя паверхні, нанясенне колераў, праца з праектамі, падтрымка шматлікіх мадэляў у межах аднаго праекта. *Арра* добра спраўляецца з утрыманнем вялікай колькасці параметраў і опцыяў; праекты *Арра* лёгка пераносяцца паміж сістэмамі.

3.3 Тыпы ўваходных інтэрфейсаў

У сваёй сістэме, як бачна на малюнку 5, я дадаў падтрымку двух інтэрфейсаў для ўводу дадзеных.

Першы - перанакіраванне відэаплыні наўпрост з вэб-камеры ці любой іншай сістэмы захопу відэа. У такім рэжыме мы найбольш набліжаемся да ўмоваў, якія можа сустрэць алгарытм пры запуску на лятальным апарце пры выкананні рэальнага палёту.

Другі - больш “лабараторны” - публікацыя паслядоўнасці кадраў загадзя падрыхтаванага набору дадзеных. Такі рэжым значна лепш падыходзіць для тэставання і адладкі алгарытма і дае магчымасць паўтараць эксперыменты на аднолькавых дадзеных.

Падтрымка абодвух рэжымаў стала магчымай праз напісанне ўласных утылітаў для ROS альбо мадыфікацыю існуючых. Трэба дадаць, што на наступных этапах мы працуем з дадзенымі як з мноствам кадраў альбо выяваў - такім чынам, пры наяўнасці відэаплыні на ўваходзе ўзнікае патрэба ў разбіцці пlynі на кадры з вызначанай частатой, захаванне кадраў у форме асобнага датасэту, а таксама сінхранізацыя пазіцыяў камер з абранымі кадрамі з пачатковай відэаплыні.

3.4 Выманне дадатковых дадзеных са SLAM-алгарытмаў

Адным з найбольш крытычных месцаў працы сістэмы з’яўляецца выманне ўсіх патрэбных нам дадзеных са SLAM-алгарытмаў і узгадненне гэтага фармату дадзеных, для паспяховага пераходу да наступных крокаў.

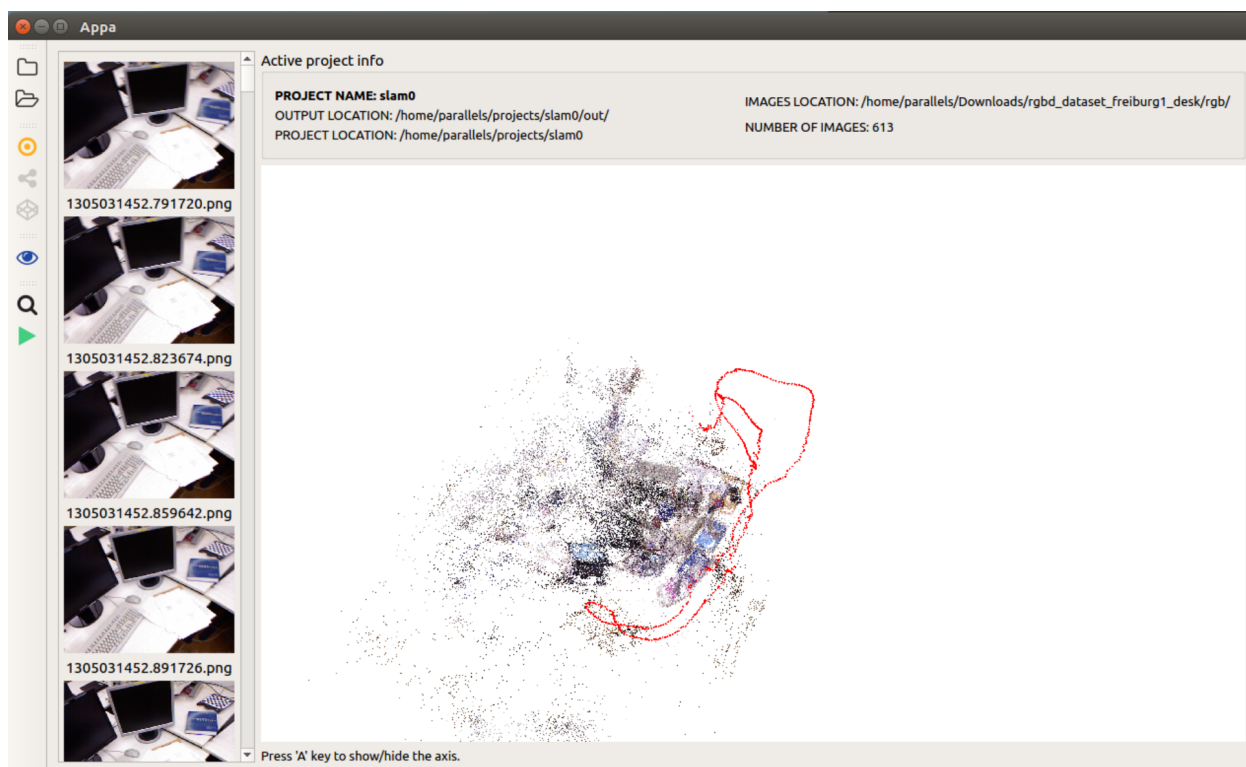
Першае, што варта тут згадаць - патрэбныя нам дадзеныя не выдаюцца алгарытмамі ў гатовым выглядзе. У якасці выходных дадзеных нас найбольш цікавіць граф узаемабачнасці кропак з камераў (*covisibility graph*) - граф, вяршынямі якога з’яўляюцца камеры (альбо ключавыя кадры), рэбры праводзяцца паміж тымі вяршынямі, адпаведныя камеры якіх маюць дастаткова вялікую агульную аглядную прастору (вымяраецца ў колькасці знойдзеных агульных трохмерных кропак). Таксама цікаўнасць прадстаўляюць ацэнкі пазіцыяў камер, якія

можна альбо прыняць за ісціну, альбо скарыстаць у якасці апрыорнай ацэнкі. Патрэба ва ўсіх гэтых дадзеных прымушае нас разбіраць той ці іншы SLAM-алгарытм на кавалкі, шукаць патрэбныя дадзеныя у кодзе і знаходзіць магчымасці выгрузкі іх у вонкавы сусвет.

Па-другое, працуючы з рознымі SLAM-алгарытмамі, мы павінны падыходзіць да кожнага з іх індывідуальна, шукаючы магчымасць выгрузіць патрэбныя нам дадзеныя, але, ў выніку, прадставіць дадзеныя ва ўніфікаванай форме, незалежнай ад выкарыстанага SLAM-алгарытма. Індывідуальная праца з кожным асобна ўзятым алгарытмам досыць моцна ўскладняе нашу задачу.

3.5 Appa і інтэрфейс для дадзеных са SLAM-а

У межах пераддыпломнай практыкі я працягнуў працу над праграмным забеспячэннем для рэканструкцыі паверхні па здымках з беспілотных лятальных апаратаў *Appa* [9], працу над якім я пачаў раней у межах курсавой працы.



Малюнак 6: праграмнае забеспячэнне для трохмернай рэкаструкцыі Appa

Першасныя ўдасканалеванні, якія патрабуе *Appa* - распрацоўка інтэрфейса, які б даваў магчымасць ўводзіць дадзеныя, сабраныя на папярэдніх этапах працы сістэмы. З ростам аб'ёмаў дадзеных да інтэрфейсу ўводу прад'яўляюцца ўсё больш жорсткія патрабаванні да хуткасці працы, што прымушае шукаць больш дасканалыя структуры

дадзеных для часовага захавання графа ўзаемабачнасці з усімі патрэбнымі дадатковымі параметрамі.

3.6 Аптымізацыя камунікацыяў з дапамогай ROS-a

Яшчэ адным накірункам маёй працы ёсць аптымізацыя камунікацыяў паміж часткамі сістэмы. У папярэдніх пунктах у асноўным апісваліся камунікацыі праз файлавую сістэму - камунікацыі могуць быць аптымізаваныя з дапамогай убудаваных сродкаў камунікацыяў у ROS: публікацыя ў ROS-аўскі “topic” з аднаго боку і праслухоўванне з іншага з’яўляецца, у агульным выпадку, значна больш дасканалым спосабам злучэння асобных элементаў сістэмы з малюнку 5 у адно суцэльнае. Гэты спосаб таксама дазваляе паскорыць камунікацыі, бо абмен дадзенымі ідзе праз RAM-памяць замест файлавай сістэмы.

ВЫСНОВЫ

У справаздачы былі апісаныя асноўныя паняцці, датычныя да задачы трохмернай рэканструкцыі паверхні па дадзеных з БПЛА, сфармуляваная канцэпцыя SLAM, апісаныя некаторыя найбольш вядомыя і паспяхова рэалізаваныя SLAM сістэмы.

Негледзячы на тое, што SLAM, рэалізаваны для працы ў рэальным часе, і часцей за ўсё выкарыстоўваемы для навігацыі ў прасторы, кепска спраўляецца з задачай пабудовы шчыльнай трохмернай мадэлі, вынікі адпрацоўкі SLAM могуць быць вельмі карыснымі ў якасці апрыорных дадзеных для запуску афлайн алгарытмаў.

У якасці практычнай рэалізацыі я распрацаваў дызайн і часткова рэалізаваў сістэму, якая б займалася эфектыўнай рэканструкцыяй паверхні па дадзеных з беспілотных лятальных апаратаў, у аснове якой ляжыць два канкуруючых падыхода: SLAM-алгарытм спалучаны разам з класічным падыходам да рэканструкцыі. Надалей я працягну ўдасканальваць кожны асобны элемент гэтай сістэмы, а таксама буду паляпшаць камунікацыі паміж рознымі часткамі сістэмы, каб сістэма ўспрымалася адзіным суцэльным механізмам.

СПИС КРЫНИЦАЎ

- [1] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian D. Reid, and John J. Leonard. Simultaneous localization and mapping: Present, future, and the robust-perception age. *CoRR*, abs/1606.05830, 2016.
- [2] Jiaxin Li, Yingcai Bi, Menglu Lan, Hailong Qin, Mo Shan, Feng Lin, and Ben M. Chen. Real-time simultaneous localization and mapping for uav: A survey. 2016.
- [3] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR '07*, pages 1–10, Washington, DC, USA, 2007. IEEE Computer Society.
- [4] Montiel J. M. M. Mur-Artal, Raúl and Juan D. Tardós. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.
- [5] Raúl Mur-Artal and Juan D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.
- [6] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *European Conference on Computer Vision (ECCV)*, September 2014.
- [7] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
- [8] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.
- [9] Appa: Gui and cli for theiasfm multiview geometry and structure from motion library. <http://git.io/appa>.