

МИНІСТЭРСТВА АДУКАЦЫІ РЭСПУБЛІКІ БЕЛАРУСЬ

БЕЛАРУСКІ ДЗЯРЖАЎНЫ УНІВЕРСІТЭТ

ФАКУЛЬТЭТ ПРЫКЛАДНОЙ МАТЭМАТЫКІ і  
ІНФАРМАТЫКІ

Кафедра дыскрэтнай матэматыкі і алгарытмікі

БОГДАН

Уладзіслаў Уладзіміравіч

РЭКАНСТРУКЦЫЯ ПАВЕРХНІ ПА ДАДЗЕНЫХ З  
БЕСПЛОТНЫХ ЛЯТАЛЬНЫХ АПАРАТАЎ

Дыпломная праца

Навуковы кіраўнік:  
доктар фіз.-мат. навук,  
прафесар Тузікаў А. В.

Дапушчаны да абароны:

23 траўня 2018 г.

Загадчык кафедры дыскрэтнай матэматыкі і алгарытмікі  
доктар фіз.-мат. навук, прафесар Котаў У. М.

Мінск, 2018

# РЭФЕРАТ

Дыпломная праца, 41 с., 16 выяў, 5 формул, 18 крыніц

БПЛА, ТРОХМЕРНАЯ РЭКАНСТРУКЦЫЯ, СТЭРЭАБАЧАННЕ,  
SLAM-АЛГАРЫТМЫ, ЭПІПАЛЯРНАЯ ГЕАМЕТРЫЯ,  
МАНАКУЛЯРНАЯ КАМЕРА, КЛЮЧАВЫЯ КРОПКІ,  
ДЭСКРЫПТАРЫ КЛЮЧАВЫХ КРОПАК, ПОШУК  
АДПАВЕДНАСЦЯЎ

Аб'ектам даследвання з'яўляюцца алгарытмы трохмернай рэканструкцыі паверхні па дадзеных з беспілотных лятальных апаратуў.

Мэта працы — даследаваць будову алгарытмаў рэканструкцыі паверхні па наборы здымкаў, правесці аналіз існуючых SLAM-алгарытмаў, разгледзець магчымасці інтэграцыі двух падыходаў да рэканструкцыі паверхні, расправаць адпаведнае праграмнае забеспячэнне.

Метады даследвання: аналіз публікацыяў і вынікаў эксперыменту, даследванне ўнутранай будовы алгарытмаў, правядзенне уласных эксперыменту, распрацоўка праграмнага забеспячэння.

У ходзе працы атрыманыя наступныя вынікі:

1. Праведзены аналіз і прыведзеная апісанне сучасных SLAM-алгарытмаў.
2. Пропанаваныя падыходы, пры якіх SLAM-алгарытмы могуць быць інтэграваныя з класічнымі алгарытмамі для рэканструкцыі паверхні па дадзеных з беспілотных лятальных апаратуў.
3. Распрацаванае праграмнае забеспячэнне для маніпуляцыі з наборамі дадзеных, пабудовы трохмерных мадэляў па наборы здымкаў, візуалізацыі. Праграмнае забеспячэнне, апрач рэалізацыі традыцыйных падыходаў да рэканструкцыі, інтэграванае з SLAM-сістэмамі, адкуль яно вымае дадатковыя дадзенія і выкарыстоўвае для ўдасканалення сваёй працы.

Галіны прымянення: камп'ютарны зрок, трохмерная рэканструкцыя, іншыя прыкладанні.

# РЕФЕРАТ

Дипломная работа, 40 с., 16 изображений, 5 формул, 18 источников

БПЛА, ТРЁХМЕРНАЯ РЕКОНСТРУКЦИЯ, СТЕРЕОМЕТРИЯ,  
SLAM-АЛГОРИТМЫ, ЭПИПОЛЯРНАЯ ГЕОМЕТРИЯ,  
МОНОКУЛЯРНАЯ КАМЕРА, КЛЮЧЕВЫЕ ТОЧКИ, ДЕСКРИПТОРЫ  
КЛЮЧЕВЫХ ТОЧЕК, ПОИСК СООТВЕТСТВИЙ

Объектом исследования являются алгоритмы трёхмерной реконструкции поверхности по данным с беспилотных летательных аппаратов.

Цель работы — исследовать строение алгоритмов реконструкции поверхности по набору снимков, провести анализ существующих SLAM-алгоритмов, рассмотреть возможности интеграции двух подходов к реконструкции поверхности, разработать соответствующее программное обеспечение.

Методы исследования: анализ публикаций и результатов экспериментов, исследование внутреннего строения алгоритмов, проведение собственных экспериментов, разработка программного обеспечения.

В ходе работы получены следующие результаты:

1. Проведён анализ и приведено описание современных SLAM-алгоритмов.
2. Предложены подходы, при которых SLAM-алгоритмы могут быть интегрированы с классическими алгоритмами для реконструкции поверхности по данным с беспилотных летательных аппаратов.
3. Разработано программное обеспечение для манипуляции с наборами данных, построения трёхмерных моделей на набору снимков, визуализации. Программное обеспечение, кроме реализации традиционных подходов к реконструкции, интегрировано с SLAM-системами, откуда оно достаёт дополнительные данные и использует для своей работы.

Область применения: компьютерное зрение, трёхмерная реконструкция, приложения.

# ABSTRACT

Graduate work, 41 p., 16 figures, 5 formulas, 18 sources

UAV, THREE-DIMENSIONAL RECONSTRUCTION, STEREOVISION, SLAM ALGORITHMS, EPIPOLAR GEOMETRY, MONOCULAR CAMERA, KEYPOINTS, FEATURES, FEATURES' DESCRIPTORS, FEATURE MATCHING

The object of the research are algorithms of three-dimensional reconstruction of the surface based on the data from unmanned aerial vehicles.

The purpose — to research the structure of the algorithms of surface reconstruction from a set of images, to analyze existing SLAM algorithms, to search for the possibilities to integrate two reconstruction approaches, to develop the software.

Methods of the research are: to analyze publications and results of the experiments, to research internal structure of the algorithms, to perform own experiments, to develop the software.

During the research the following results were obtained:

1. Analysis of state-of-art SLAM algorithms was performed and the description was provided.
2. The ways to integrate SLAM algorithms with traditional algorithms for surface reconstruction based on the data from UAV were suggested.
3. The software for data manipulation, three-dimensional models' construction and visualization was developed. Furthermore, the software was integrated with SLAM systems in order to obtain data to improve its work.

The scopes are: computer vision, three-dimensional reconstruction, other applications.

# ЗМЕСТ

УВОДЗІНЫ . . . . .	4
ГЛАВА 1. АКТУАЛЬНАСЦЬ ЗАДАЧЫ РЭКАНСТРУКЦЫИ ПАВЕРХНІ . . . . .	6
1.1 Агульныя звесткі . . . . .	6
1.2 Сферы прымянення . . . . .	6
1.3 Асаблівасці задачы пры выкарыстанні БПЛА . . . . .	7
ГЛАВА 2. ТЭАРЭТЫЧНЫЯ АСПЕКТЫ ЗАДАЧЫ РЭКАНСТРУКЦЫИ . . . . .	9
2.1 Этапы рашэння задачы рэканструкцыі . . . . .	9
2.2 Тыпы камер . . . . .	10
2.2.1 Стэрэа RGB камера . . . . .	10
2.2.2 RGB-D камера . . . . .	10
2.2.3 Манакулярная RGB камера . . . . .	11
2.3 Ключавыя кропкі, дэскрыптары, спосабы іх апісання . . . . .	14
2.3.1 Апісанне некаторых распаўсюджаных дэтэктараў і дэскрыптараў . . . . .	14
2.4 Спалучэнне простых метадаў і метадаў, заснаваных на ключавых кропках . . . . .	17
2.5 Пошук адпаведнасцяў паміж ключавымі кропкамі . . . . .	17
2.6 Фармальная пастановка задачы рэканструкцыі . . . . .	19
ГЛАВА 3. SLAM-АЛГАРЫТМЫ . . . . .	21
3.1 Асноўныя звесткі . . . . .	21
3.2 Ключавыя паняцці і падыходы . . . . .	22
3.2.1 Лакалізацыя . . . . .	22
3.2.2 Пошук на мапе . . . . .	22
3.3 Падыходы да рэалізацыі . . . . .	23
3.4 Агляд SLAM-сістэм . . . . .	24
3.4.1 ORB-SLAM . . . . .	24
3.4.2 LSD-SLAM . . . . .	26
3.4.3 SVO . . . . .	27
3.4.4 CNN-SLAM . . . . .	28
ГЛАВА 4. ПРАКТЫЧНАЯ РЭАЛІЗАЦЫЯ . . . . .	31
3.1 Распрацаванае праграмнае забеспечэнне . . . . .	31
3.2 Інтэграцыя з SLAM-сістэмамі . . . . .	34
3.2.1 ROS . . . . .	35
3.2.2 Тыпы ўваходных інтэрфейсаў . . . . .	36
3.2.3 Выманне дадатковых дадзеных са SLAM-алгарытмаў . . . . .	36
3.2.4 Аптымізацыя камунікацыяў з дапамогай ROS . . . . .	37
3.2.5 Статус распрацаванасці . . . . .	38

ВЫСНОВЫ . . . . .	39
СПІС КРЫНІЦ . . . . .	40

# УВОДЗІНЫ

Задача рэканструкцыі паверхні зямлі па дадзеных з беспілотных лятальных апаратоў (БПЛА) узнікае ў вялікай колькасці сфераў жыщчядзейнасці: ад сельскай гаспадаркі да ацэнкі наступстваў прыродных катастрофаў. З кожным годам усё больш шырокое развіццё ў свеце атрымліваюць аўтаномныя аўтамабілі, сродкі сачэння ў месцах вялікага скаплення людзей. Патрэбнасць у вырашэнні тых ці іншых задач, якія патрабуюць прымянеñня беспілотнага лятальнага апарату (дрона) можа ўзнікнуць у арганізацый ѿ розных памераў.

Сам па сабе беспілотны лятальны апарат звычайна выступае толькі у якасці пачатковага “зборшчыка” інфармацыі, тады як задача апрацоўкі, удасканалення і ўзгаднення інфармацыі стаіць асобна і патрабуе распрацоўкі эфектыўных, хуткіх і надзейных алгарытмаў. Згаданыя вышэй задачы штогод атрымліваюць усё большы распаўсюд, у тым ліку ў сувязі з шырокай даступнасцю БПЛА і з распаўсюдам танных камераў, якімі ўкамплектаваецца борт БПЛА.

Задача рэканструкцыі цесна звязаная са шматлікімі сумежнымі задачамі: распазнаванне аб'ектаў, навігацыя ў просторы, пабудова мапы мясцовасці. Асаблівую цікаўнасць прадстаўляюць раšэнні, якія выконваюцца ў рэальнym часе; праца ў рэальнym часе крытычная для алгарытмаў навігацыі і абыходу перашкодаў пры запуску на аўтаномных БПЛА, бо аўтаномнасць вымагае хуткага і беспамылковага реагавання на змены ў навакольным асяроддзі.

Алгарытмы, якія прымяняюцца пры здзяйсненні працэса рэканструкцыі, могуць быць умоўна падзеленыя на тыя, што выконваюцца пры наяўнасці поўнага набора дадзеных (“афлайн”-алгарытмы), і тыя, якія выконваюцца ў рэальнym часе і аперуюць з дадзенымі, якія паступаюць у “онлайн” рэжыме. Да апошніх, у прыватнасці, адносяцца алгарытмы, пабудаваныя па канцэпцыі SLAM (англ. *Simultaneous Localization and Mapping*) - алгарытмы адначасовай лакалізацыі і пабудовы мапы. SLAM-алгарытмы атрымалі моцны штуршок у развіцці ў апошнія гады ў сувязі з презентацыяй шэрага сістэм, якія моцна пераўзыходзяць усе папярэднія напрацоўкі. У наступных главах, сярод іншага, падрабязна распавядаецца пра некаторыя сучасныя SLAM-сістэмы.

Мэтамі распрацоўкі дадзенай тэмы ставіліся даследаванне тэарэтычных аспектаў задачы рэканструкцыі, а таксама практычная рэалізацыя шэрага праграмных модуляў, якія бы здзяйснялі рэканструкцыю паверхні. Пры гэтым дадзеныя, згенераваныя SLAM алгарытмамі, прапаноўваецца выкарыстоўваць для аптымізацыі гэтага працэса.

У гэтай працы быў зроблены агляд разнастайных падыходаў да рэканструкцыі паверхні, прыведзенае паразнанне алгарытмаў, якія працуюць у рэальнym часе, з тымі, што працуюць з папярэдне сабранымі БПЛА дадзенымі, разгледжаная магчымасць аб'яднання розных падыходаў у адну сістэму. У практичнай частцы прыводзіцца справаздача па распрацаваным праграмным забеспечэнні для рэканструкцыі паверхні па наборы здымкаў і па дадатковых дадзеных (пры наяўнасці), а таксама прапаноўваецца і прыводзіцца статус распрацаванасці архітэктуры сістэмы, якая спалучае ў сабе SLAM-падыходы з “афлайн” падыходамі для здзяйснення эффектыўнай рэканструкцыі паверхні па дадзеных з беспілотных лятальных апаратуў.

# ГЛАВА 1. АКТУАЛЬНАСЦЬ ЗАДАЧЫ РЭКАНСТРУКЦЫІ ПАВЕРХНІ

## 1.1 Агульныя звесткі

У агульным выпадку задача рэканструкцыі паверхні фармулюеца наступным чынам: неабходна рэканструяваць трохмерны аб'ект па мнстве зробленых з розных ракурсаў здымкаў. Задача фармулюеца дастаткова натуральна чынам, і калі чалавеку дастаткова кінуць позірк на аб'ект, каб уявіць ягоную трохмерную структуру, алгарытмічна задача ўсё яшчэ застаецца не да канца вырашанай, звычайна патрабуе вялікіх вылічальных магутнасцяў і не працуе ўніверсальна добра для любых асяроддзяў і любых умоваў здымак, такіх як, напрыклад, адрозныя па асвятленні сцэны. Пад трохмерным аб'ектам у залежнасці ад кантэксту могуць мецца на ўвазе адрозныя рэчы: калі ў некаторых сітуацыях раздрэджанае воблака кропак будзе лічыцца добрым прыкладам рэканструяванай структуры, то ў іншых пастаўленая задача можа запатрабаваць пабудову шчыльнай і гладкай мадэлі з нанесенымі тэкстурамі і колерамі. Падбор алгарытмаў і ацэнка вылічальных магутнасцяў здзяйсняеца ў залежнасці ад пастаўленых патрабаванняў.

Задача рэканструкцыі можа таксама фармулявацца для іншых тыпаў уваходных дадзеных: апроч той ці іншай камеры ўваходнымі дадзенымі для задачы могуць быць дадзеныя з іншых датчыкаў, такіх як акселерометр, гіраскоп і GPS-датчык; замест манакулярнай камеры можа прымяняцца RGB-D камера (вяртае дадатковы слой глыбіняў) стэрэа-камера (уяўляе сабой дзве RGB камеры на фіксаванай паміж сабой адлегласці, якія ў пэўным сэнсе імітуюць бінакулярны чалавечы зрок), альбо, напрыклад, лазерная камера.

Варта дадаць, што даследаванні ў гэтай галіне камп'ютарнага зроку развіваюцца таксама праз удасканаленне апаратнага забеспячэння: падыходы, якія некалькі год таму былі практычна нерэалізуемымі і былі магчымыя толькі ў тэорыі, з развіццём тэхналогіяў атрымліваюць новае жыццё.

Разам з tym, мнства проблемаў застаюцца нявырашанымі. Поспех ўсёй галіны даследаванняў залежыць ад того, наколькі адначасова добра будуць удасканальвацца вылічальныя магутнасці камп'ютарных сістэм, алгарытмы, а таксама сродкі захопу дадзеных - усё яшчэ мнства проблем узімае акурат праз недасканаласць, недакладнасць альбо нестабільнасць камер і іншых датчыкаў.

## 1.2 Сферы прымянењня

Цікавасць задачы рэканструкцыі таксама ў запатрабаванні атрымання ражэння ў абсалютна розных сферах жыцця, кожная з якіх дыктуе свае асаблівасці і прымушае развіваць даследаванні ў тым ці іншым кірунку.

БПЛА выкарыстоўваюцца надзвычайнімі службамі для ацэнкі наступстваў прыродных катастроф, у сельскай гаспадарцы, дарожнымі службамі для маніторынгу і аналізу трафіка. Патрабаванні да хуткасці працы алгарытмаў рэканструкцыі натуральныя - хуткасць працы ў некаторых галінах жыцця крытычная і разбор вялікіх аб'ёмаў неапрацаўваных дадзеных можа стацца непераадольна вялікай працай для чалавека. Патрабаванне да алгарытмаў рэканструкцыі працаўца ў рэальнym часе ў большасці выпадкаў з'яўляецца пры навігацыі і аўтаномным руху, у такім выпадку шчыльная рэканструкцыя можа быць залішнай і разрэджаная мадэль у выглядзе воблака кропак цалкам задаволіць патрабаванні. У такіх выпадках мы часта кажам пра пабудову мапы - аб'екта, які ўяўляе сабой своеасаблівую мадэль навакольнага свету, з пазначанымі на ім кропкамі, якія прадстаўляюць для нас цікаўнасць. Мапы часта выкарыстоўваюцца для лакалізацыі і навігацыі і могуць быць схаваныя ад вачей назіральніка. Падрабязней да тэмы пабудовы мапаў мы вернемся падчас абмеркавання SLAM-алгарытмаў.

## 1.3 Асаблівасці задачы пры выкарыстанні БПЛА

Той факт, што рэканструкцыя адбываецца не на выпадковым наборы дадзеных, пра які адсутнічае дадатковая інфармацыя, дае нам простору для аптымізацыі працэса рэканструкцыі: скарачэнне часу і паляпшэнне якасці пабудаванай мадэлі. Пералічым асаблівасці задачы рэканструкцыі па дадзеных з БПЛА ў пароўненні з агульнай задачай:

- Усе здымкі зробленыя адной фізічнай камерай, такім чынам унутраныя параметры ўсіх камераў застаюцца нязменнымі. Больш за тое, унутраныя параметры камеры застаюцца нязменнымі не толькі ў межах аднаго набора, але і для ўсіх здымкаў зробленым адным БПЛА з зафіксаванай на ім камерай. Апошні факт не ўносіць зменаў у працэс рэканструкцыі, але спрашчае працэс выкарыстання БПЛА на практицы (адсутнасць патрэбы ў шторазовым калібраванні для высвятлення ўнутраных параметраў камеры).
- Апроч камеры ў нашым распараджэнні, у залежнасці ад канфігурацыі БПЛА, могуць мецца і іншыя датчыкі: акселерометр, гіраскоп, GPS-датчык і інш. Дадзеныя сабраныя імі, могуць

выкарыстоўвацца як для вызначэння вонкавых параметраў камеры, так і для папярэдняга размяшчэння кропак у прасторы.

## Патэнцыйныя змены ў набор дадзеных і алгарытмы з улікам вышэйсказанага

Прычыны, па якіх дадатковыя дадзенія станоўча паўплываюць на якасць і хуткасць рэканструкцыі, абсалютна натуральныя: памяншэнне колькасці параметраў у глабальнай аптымізацыйнай задачы і больш якасная пачатковая апраксімацыя. Такім чынам мы маем два падыходы да выкарыстання дадзеных, якімі суправаджаецца набор выявав:

- Аб'яўленне параметраў канстантнымі і непасрэднай іх падстаноўка ва ўраўненні - такім чынам, маем яўнае памяншэнне колькасці параметраў што не можа не паўплываць на хуткасць адпрацоўкі алгарытма і на выніковае значэнне памылкі праекцыі станоўча. Відавочны і недахоп: немагчымасць дакладнага папярэдняга задання параметраў і наступнае аб'яўленне іх канстантнымі прывядзе да пісання канчатковых значэнняў трохмерных кропак і, адпаведна, пагаршэння якасці пабудаванай мадэлі. Дакладнасць вызначэння параметраў знаходзіцца ў простай залежнасці ад карэктнасці пабудаванай мадэлі. На практыцы высвятляеца, што такі падыход вядзе да значнага пагаршэння канчатковых вынікаў і кепска прымяняльны на рэальных дадзеных.
- Выкарыстанне дадатковых дадзеных для пачатковай апраксімацыі значэнняў параметраў камеры і палажэнняў кропак у прасторы. Пры такім падыходзе, мы не рызыкуем стаць ахвярамі кепской дакладнасці альбо памылак у апісанні дадзеных - разам з тым, колькасць ітэрацый у працэсе пучковай аптымізацыі можа паменшыцца ў разы. Гэты падыход можа паспяхова быць прыменены на практыцы: хуткасць працэса пабудовы праста залежыць ад якасці ўваходных дадзеных - яўных жа памылак у выніковай мадэлі (такіх, як у першым пункце) назірацца не будзе, бо алгарытм выправіць яўна хібныя ўваходныя дадзенія.

Альтэрнатывай можа стаць змяшаны падыход: напрыклад, аб'яўвіць фокусную адлегласць пастаяннай і недаступнай да зменаў (калі мы дастаткова ўпэўненыя ў лічбах, якія прадастаўляе вытворца камера, альбо атрыманых у выніку каліброўцы), але дазволіць алгарытму удакладняць параметры павароту камеры - пры дастаткова дакладных пачатковых значэннях яны будуць змененія нязначна.

Апісаныя вышэй падыходы я выкарыстоўваю ў сваім распрацаваным праграмным забеспечэнні, падрабязнае апісанне якога прыводзіцца ў апошнюю главе.

# ГЛАВА 2. ТЭАРЭТЫЧНЫЯ АСПЕКТЫ ЗАДАЧЫ РЭКАНСТРУКЦЫ

## 2.1 Этапы решэння задачы рэканструкцыі

Калі мы вядзем гаворку пра пабудову трохмернай мадэлі па здымках з адзінай манакулярнай камеры, мы заўсёды решаем некалькі важных задач. Да гэтых задач адносяцца:

- для двух ці некалькі кадраў – вымianne інфармацыі аб агульнай аглядаемасці сцэны;
- для здымка (пры наяўнасці іншых дадзеных) - пабудова мапы глыбіні сцэны;
- решэнне алтымізацыйнай задачы па аб'яднанні ўсёй інфармацыі ў адзіную сцэну (ці прыняцце решэння аб наяўнасці некалькіх незалежных сцэн у межах аднаго набора дадзеных), мінімізацыя геаметрычных і алтычных памылак.

Гэтые задачы ў той ці іншай форме і ў tym ці іншым выглядзе з'яўляюцца пры любой спробе рэканструяваць трохмерную паверхню сродкамі камп'ютарнага зроку. Спынімся на першай задачы і паглядзім падрабязней, якія наборы інструментаў даюць магчымасць ацаніць узаемаразмяшчэнне для двух ці некалькіх здымкаў і, такім чынам, даць уяўлянне пра арганізацыю камераў вакол сцэны.

Першы з такіх набораў інструментаў – метады, заснаваныя на *пошуку асаблівасцяў* (англ. *feature-based methods*). Ідэя у выманні з кожнай асобнай выявы пэўнай колькасці асаблівасцяў (імі могуць быць, напрыклад, кропкі, вуглы альбо лініі) і пошук адпаведнасцяў паміж асаблівымі кропкамі на розных выявах (англ. *feature matching*), аднаўленне пазіцыяў камер і структуры сцэны з дапамогай эпіполярнай геаметрыі (геаметрыя стэрэабачання) і, у завяршэнне, удакладненне параметраў праз мінімізацыю памылкі праекцыі. Падобную працэдуру выконвае вялікая колькасць алгарытмаў: даступнасць эффектыўных метадаў вымання асаблівасцяў і пошуку іх адпаведнасцяў дазваляе рабіць гэта хутка і надзейна; разам з tym, такі падыход можа даваць недастатковую дакладнасць вынікаў, быць прымяняльным толькі на вызначаным класе зыходных дадзеных (напрыклад, не дапускаць аднастайныя тэкстуры), патрабуе надзейных алгарытмаў ацэнкі пазіцыі.

Другі клас метадаў заснаваны на гэтак званых *простых метадах* (англ. *direct methods*): у такіх метадах непасрэдна выкарыстоўваюцца

значэнні запісаныя ў пікселях, не адываеца спробы выяўлення на выяве вызначанага кшталту асаблівасцяў. Для апрацоўкі выкарыстоўваеца накірунак і велічыні лакальнага градыента інтэнсіўнасці. Простыя метады, якія выкарыстоўваюць усю інфармацыю на выяве, нават у зонах з маленькім градыентам, апынуліся значна эфектыўнейшымі за метады, заснаваныя на пошуку асаблівасцяў у сцэнах з нізкай тэкстураванасцю альбо ў выпадку размытасці ці дэфакусіроўкі [17]. Дастаткова праца затратнай працэдурай становішча падлік фотаметрычнай памылкі (у параўнанні з падлікам памылкі праекцыі). Разам з тым, паколькі праца вядзеца непасрэдна са значэннямі пікселяў, час эканоміцца на пошуку асаблівасцяў і падліку дэскрыптараў.

## 2.2 Тыпы камер

Неглядзячы на цікавыя магчымасці такіх датчыкаў, як акселерометр, гіраскоп альбо лазер вымярэння адлегласці, яны не заўсёды доступны і не даюць універсальна добрых вынікаў у любых умовах. Далей мы будзем весці гутарку пра камеры розных тыпаў і больш за ёсё - пра манакулярную, як найбольш універсальную, распаўсюджаную і простую ў эксплуатацыі.

### 2.2.1 Стэрэа RGB камера

Для стэрэакамеры ўласцівая наяўнасць двух ці больш аб'ектываў, кожны з якіх стварае кадры незалежна ад іншага. Гэта дазваляе сімуляваць бінакулярны зрок чалавека і атрымліваць трохмерныя аб'екты са значна меншымі вылічальнымі выдаткамі. Кожны з аб'ектываў стэрэакамеры працуе як асобная манакулярная камера, плыні кадраў з розных аб'ектываў даюць магчымасць праз пошук асаблівых крапак і геаметрычныя падлікі размісціць крапкі ў трохмернай прасторы.

Трэба дадаць, што стэрэакамера заўжды наладжаная на працу з пэўным спектрам адлегласцяў: камера, якая працуе добра ў маленьких закрытых прасторах не дасць такіх жа вынікаў, калі прастора будзе змасштабаваная ў дзясяткі разоў. Такім чынам, стэрэакамерам не ўласцівая вялікая універсальнасць.

### 2.2.2 RGB-D камера

RGB-D камера, як і стэрэа, з'яўляеца шматканальнай: разам з RGB каналам, які вяртае звыклыя нашаму воку каляровыя выявы, існуе дадатковы канал, які для кожнага здымка вяртае гэтак званую мапу глыбіні. Прыклад вяртаемых дадзеных - на малюнку 2.1.

На практыцы RGB-D камеры не заўсёды працуе так добра, як нам хацелася б: справа ў амежаванасці спектра, у якім фармуюцца значэнні

глыбіняў. Мапа глыбіняў, атрыманая апаратным спосабам, часцей за ўсё выкарыстоўваецца для папярэдняй ініцыялізацыі і для паслядоўнай аптымізацыі алгарытмічнымі спосабамі. Да таго ж, RGB-D камеры даюць слабыя вынікі ў складаных умовах, такіх як уздзеянне простых сонечных прамянёў.

Прыкладамі RGB-D камер з'яўляюцца Kinect для Xbox 360 альбо Intel RealSense камеры.



Малюнак 2.1: здымкі з RGB-D камеры

### 2.2.3 Манакулярная RGB камера

Самымі распаўсюджанымі з'яўляюцца добра знаёмыя нам манакулярныя камеры - камеры, якія прысутнічаюць у смартфонах і ўжываюцца ў быце. Манакулярныя камеры, ў сваю чаргу, таксама бываюць розных тыпаў, але на практыцы мы часцей за ўсё працуем з праектыўнымі (англ. “projective”, “pinhole”) камерамі.

Да таго ж гэта адзіны від камераў, з якімі ў рамках дадзенай работы вялася шчыльная праца, для якіх рэалізаваныя алгарытмы і на якіх праводзіліся эксперыменты.

Праецыраванне трохмерных крапак у выпадку з манакулярнай камерай можа быць апісанае наступнай формулай:

$$x = PX \quad (1)$$

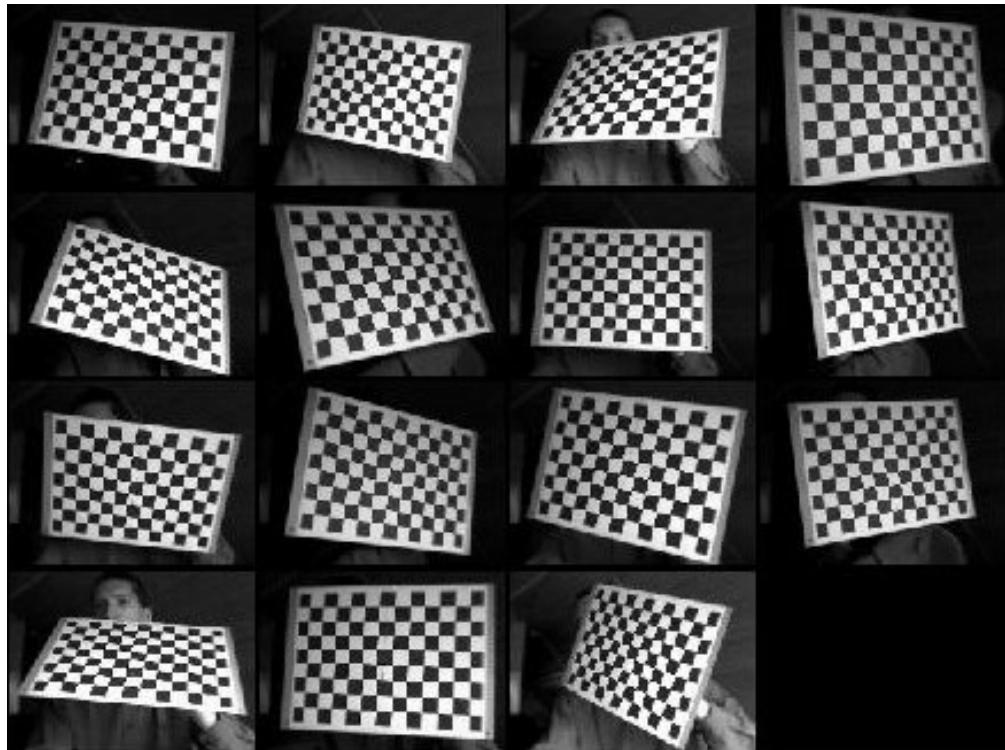
дзе  $P$  - матрыца, якая здзяйсняе пераўтварэнне,  $X, x$  - вектары, якія адпавядаюць трохмернай і двухмернай крапцы адпаведна, запісаныя ў аднародных каардынатах. У сваю чаргу матрыцу  $P$  можна прадставіць як:

$$P = K[R|t] \quad (2)$$

дзе  $K$  - вектар унутраных параметраў камеры,  $R$  -  $3 \times 3$  матрыца павароту,  $t$  -  $3 \times 1$  вектар зрушэння камеры ў прасторы ў сусветных коардынатах. Гэтыя параметры задаюць праектыўную камеру, якая здзяйсніе праекцыю кропкі прасторы на матрыцу камеры. Унутраныя параметры могуць задавацца разнастайным чынам, распаўсюджанымі прыкладамі выступаюць матрыцы:

$$K = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}, \quad K = \begin{bmatrix} \alpha_x & 0 & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

дзе  $f$  - фокусная адлегласць,  $\alpha_x, \alpha_y$  - фокусная адлегласць уздоўж восяў (для выпадкаў, калі не супадае),  $p_x, p_y$  - зрушэнні цэнтру праекцыі адносна цэнтра матрыцы. Іншымі параметрамі камеры могуць быць радыяльныя скажэнні  $k_1, k_2$ , якія ўзнікаюць з-за неідэальнасці лінзаў у камерах.



Малюнак 2.2: працэс каліброўкі манакулярнай камеры

Унутраныя параметры камеры заўжды вядомыя загадзя альбо праз документацыю да прылады ад вытворца, альбо праз каліброўку: каліброўка можа выяўляць хібы, дапушчаныя вытворцам. Каліброўка дае добрыя значэнні ўнутраных параметраў камеры, таму яны рэдка выкарыстоўваюцца ў якасці аптымізуемых параметраў. Працэс каліброўкі, у выпадку патрэбы, не складана здзейсніць самастойна - існуе вялікая колькасць утылітаў (напрыклад, у складзе OpenCV), якія дазваляюць гэта зрабіць, маючи ў наяўнасці толькі раздукаваную дошку,

якая нагадвае шахматную, памерамі  $9 \times 7$  клетак (гэты параметр вар'юеца). Працэс каліброўкі паказаны на малюнку 2.2.

У сваю чаргу, матрыца  $R$  і вектар  $t$  ва ўраўненні 2 з'яўляюца *знешнімі параметрамі камеры* і вызначаюца пазіцыяй камеры ў прасторы.

Варта зазначыць, што адной з найбольшых перавагаў і, адначасова, адной з найбольшых перашкодаў у працы ў манакулярнымі камерамі з'яўляеца неадназначнасць працы з масштабам. Проблема ў тым, што масштаб не можа быць высветлены праз перасоўванні ў прасторы і змяненні ў вуглах агляду, што з'яўляеца крыніцай мноства памылак. Разам з тым, перавага неадназначнасці масштабу ва ўніверсальнасці: алгарытм будзе працаваць з аднолькавымі вынікамі як у маленьких закрытых памяшканнях, так і на вялікіх адкрытых просторах. З гэтым аспектам (немачыгмасць высвятлення рэальных памераў аб'ектаў) мы яшчэ сутыкнемся пры абмеркаванні SLAM-алгарытмаў - некаторыя з іх, такія як CNN-SLAM ([18]), прапаноўваюць шляхі вырашэння гэтай проблемы.

## Заданне матрыцы павароту праз кватэрніоны

Альтэрнатывай апісанай вышэй матрыцы павароту памера  $3 \times 3$  можа быць апісанне павароту з дапамогай кватэрніонаў.

**Кватэрніоны** - сістэма гіперкомплексных лікаў, якія ўтвараюць вектарную прастору размернасцю 4 над полем рэчаісных лікаў.

Кватэрніон можа быць прадстаўлены як фармальная сума  $q(a, b, c, d) = a + bi + cj + dk$ , дзе  $a, b, c, d \in \mathbb{R}$ ,  $i, j, k$  - уяўныя адзінкі, такія, што  $i^2 = j^2 = k^2 = ijk = -1$ .

Кватэрніон часта запісываецца як пара  $(a, \vec{u})$ , дзе  $a \in \mathbb{R}$ ,  $\vec{u} = (b, c, d) \in \mathbb{R}^3$  - вектар у трохмернай прасторы, што дае нагоду задумацца аб прымененні кватэрніонаў для заданняў паваротаў у трохмернай прасторы.

Негледзячы на тое, што агулам кватэрніоны не знайшлі шырокага прыменення, іх выкарыстанне часта апраўданае ў некаторых галінах матэматыкі ды інфарматыкі, такіх як камп'ютарная графіка, навігацыя альбо праграмаванне гульняў. Кватэрніоны мінімальнай колькасцю скалярных параметраў задаюць паварот, пры гэтым яны пазбаўленыя выраджанасці, якая сустракаецца пра заданні паварота пры дапамозе толькі трох параметраў (напрыклад - вугламі Эйлера).

Калі маём кватэрніон  $q = (w, x, y, z)$ , тады матрыца павароту можа быць запісаная праз кватэрніон як:

$$R = \begin{pmatrix} 1 - 2y^2 - 2z^2 & 2xy - 2zw & 2xz + 2yw \\ 2xy + 2zw & 1 - 2x^2 - 2z^2 & 2yz - 2xw \\ 2xz - 2yw & 2yz + 2xw & 1 - 2x^2 - 2y^2 \end{pmatrix} \quad (4)$$

У практычнай рэалізацыі з главы 4 кватэрніоны выкарыстоўваюцца пры перадачы зневідных параметраў камер паміж кампанентамі сістэмы, схематычна прадстаўленай на малюнку 4.6.

## 2.3 Ключавыя кропкі, дэскрыптары, спосабы іх апісання

Тут і далей мы будзем казаць пра “ключавыя кропкі” (англ. *keypoints*) альбо “асаблівасці” (англ. *features*), хаця ў тэрмінах таго ці іншага дэтэктара ключавой кропкай можа называцца любая адзінка інфармацыі: кропка, акружнасць, лінія, вектар і інш.

Дэтэктарам (англ. *detector*) будзем называць алгарытм, які знаходзіць на выяве ключавыя кропкі.

Знойдзеная ключавая кропка пасля звычайна апісваецца ў нейкай вызначанай кароткай форме. Такое апісанне ключавой кропкі называецца дэскрыптарам (англ. *descriptor*). Часцей за ўсё сустракаюцца дэскрыптары ў выглядзе рэчаісных вектароў альбо бінарных радкоў. Дэскрыптары дапамагаюць хутка знаходзіць адпаведныя адна адной ключавыя кропкі на розных выявах. Знайсці адлегласць паміж двумя дэскрыптарамі і, такім чынам, вызначыць іхняе падабенства, з'яўляеца асноўнай аперацыяй пры пошуку адпаведнасцяў паміж дзвюма выявамі, таму да надзейнасці і хуткасці падліку функцыі адлегласці прад'яўляюцца асаблівыя патрабаванні. На практыцы адлегласць часцей за ўсё падлічваецца праз Эўклідаву адлегласць L2 (для рэчаісных вектароў) альбо праз адлегласць Хэмінга (для бінарных радкоў).

Ніжэй прыведзенае кароткае апісанне найбольш распаўсюджаных дэтэктараў ключавых кропак і спосабаў іх апісання.

### 2.3.1 Апісанне некаторых распаўсюджаных дэтэктараў і дэскрыптараў

Дэтэктар **FAST** (*Features from Accelerated Segment Test*) [12]. Як вынікае з назвы, асноўнай перавагай дэтэктара з'яўляеца ягоная хуткасць, што асабліва важна ў выпадку, калі дадзеныя паступаюць і патрабуюць апрацоўкі ў рэжыме рэальнага часу і абмежаваных рэсурсаў. Найлепшым прыкладам будуць SLAM-прыкладанні (*Simultaneous Localization and Mapping*), у якіх у рэжыме рэальнага часу адбываеца ацэнка месцазнаходжання і пошук яго на мапе; SLAM-прыкладанні часта знаходзяць прымяненне на мабільных прыладах, у тым ліку на БПЛА.

FAST з'яўляеца алгарытмам “пошуку вуглоў” (англ. *corner detector*):



Малюнак 2.3: зыходная выява

кропка з'яўляеца ключавой, калі ў маленькіх ваколіцах пікселя алгарытму атрымліваеца распазнаць адпаведны шаблон, іншымі словамі “вугал”. Дадатковая аптымізацыя адбываеца з дапамогай метадаў машыннага навучання.

Даследаванні паказваюць, што ён у некалькі разоў хутчэйшы за іншыя існуючыя алгарытмы пошука ключавых кропак на аснове вылучэння вуглоў, але, у сваю чаргу, ён недастаткова ўстойлівы да моцна зашумленых выяваў.

Дэскрыптар і дэтэктар **SIFT** [10] быў прадстаўлены ў 2004 годзе і па стане на сёння з'яўляеца адным з найбольыш распаўсюджаных. Важнымі харектарыстыкамі алгарытмамі з'яўляеца ўстойлівасць да масштабавання і змянення арыентацыі выявы, яе афінных пераўтварэнняў і зашумленых варыянтаў. Яскрава выраджаная харектэрнасць ключавых кропак дазваляе паспяхова шукаць адпаведнасці паміж рознымі выявамі. У [10], апроч апісання спосаба вымання ключавых кропак і фармата дэскрыптара, апісаюцца тэхнікі для эфектыўнага пошука адпаведнасцяў паміж выявамі з SIFT дэскрыптарамі, а таксама алгарытмы эфектыўнага распазнавання патэрнаў. Такая завершанасць даследавання алгарытма дае яму вялікую перавагу перад іншымі і моцна паспрыяла ягонаму распаўсюду. SIFT дэскрыптарам з'яўляеца рэчаісны вектар размернасці 128.

На малюнку 2.4 можна бачыць візуалізацыю знайдзеных алгарытмам SIFT ключавых кропак па пачатковым малюнку 2.3.



Малюнак 2.4: выява з нанесенымі ключавымі кропкамі (SIFT)

Дэскрыптар і дэтэктар **SURF** [11] быў упершыню апублікованы ў 2006 годзе і пачаткова ідэя стварэння заключалася ў стварэнне “больш хуткага” SIFT-а. Ён таксама з’яўляецца ўстойлівым да змены масштаба і паварота, можа быць паспяхова прыменены для размытых выявав; разам з тым, выкарыстанне SURF-а абмежаванае, калі мы маем справу са зменамі ў вугле назірання. Дэскрыптар SURF можа быць як размернасці 64, так і 128 (як SIFT).

Звернем увагу на тое, што і SIFT і SURF з’яўляюцца запатэнтаванымі алгарытмамі і, негледзячы на вялікую колькасць рэалізацыяў з адкрытым зыходным кодам, іх выкарыстанне ў камерцыйных прадуктах абмежаванае. У сваю чаргу, алгарытм ORB прапрыетарным не з’яўляецца.

Дэскрыптар і дэтэктар **ORB** (*Oriented FAST and Rotated BRIEF*) [13] з’яўляецца своеасаблівой камбінацыяй ключавых кропак FAST і дэскрыптара BRIEF. Мэтай стварэння ORB было пераўзысці SIFT у хуткасці (нават ахвяруючы надзеянасцю і дакладнасцю) у сувязі з распаўсюдам мабільных маламагутных прыладаў. Дэскрыптары маюць выгляд бінарных радкоў. Адпаведнасці паміж ORB дэскрыптарамі эфектыўна могуць быць знайдзеныя з дапамогай LSH (Locality-sensitive hashing).

Дэскрыптар **BRIEF** (*Binary Robust Independent Elementary Features*) [14] стаў адным з першых дэскрыптараў, якія апісываюць ключавыя

кропкі з дапамогай бінарных радкоў (SIFT альбо SURF, напрыклад, выкарыстоўваюць вектары з рэчаіснымі лікамі якія, у большасці выпадкаў, пасля ўсё адно ў мэтах эфектыўнасці фарматуюцца ў бінарныя радкі, што дазваляе выкарыстоўваць эфектыўную адлегласць Хэмінга).

Дэскрыптар і дэтэктар **BRISK** (*Binary Robust Invariant Scalable Keypoints*) [15] ствараўся як альтэрнатыва вышэйзгаданым SIFT і SURF, хуткасць працы якога дасягаецца праз выкарыстанне новага, заснаванага на FAST, дэтэктара ў камбінацыі з бінарным дэскрыптарам.

Звернем увагу на тое, што некаторыя алгарытмы спалучаюць у сабе адразу як пошук ключавых крапак, так і падлік іх дэскрыптараў. У агульным выпадку гэты працэс можа быць яўна падзелены: на мностве ключавых крапак атрыманых любым з дэтэктараў можа быць запушчаны алгарытм, які для кожнай кропцы паставіць у адпаведнасць дэскрыптар вызначанага фармату. На практыцы, некаторыя пары дэтэктара і дэскрыптара спалучаюцца добра і заўжды выкарыстоўваюцца разам, некаторыя сумяшчаюцца вельмі кепска. Большаясць алгарытмаў, якія знайшлі шырокое прымяленне, апісываюць адразу спосаб дэтэкцыі і апісання і ўтвараюць такім чынам закрытую сістэму, раздзяленне якой не прыводзіць да добрых вынікаў.

Даследванне спалучальнасці разам разнастайных дэтэктараў і дэскрыптараў праводзіліся мной у рамках курсавой працы.

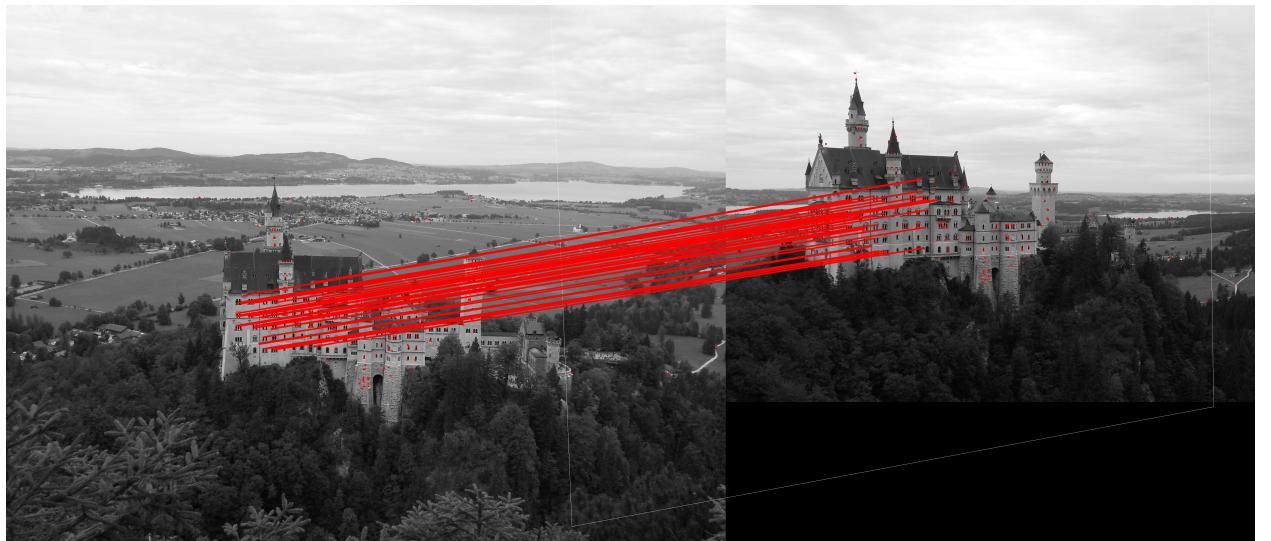
## 2.4 Спалучэнне простых метадаў і метадаў, заснаваных на ключавых крапках

Вядомыя эфектыўныя алгарытмы, якія спалучаюць у сабе простыя метады і метады, заснаваныя на выманні асаблівасцяў. Найбольшы распаўсюд такія алгарытмы знайшлі ў задачах апрацоўкі плыні дадзеных (відэаплыня, апрацоўка дадзеных у рэальнym часе з БПЛА), калі алгарытм будзе мапу мясцовасці і шукае сваё месцазнаходжанне ў рэальнym часе.

Напрыклад, у [3] яўны пошук асаблівасцяў выклікаецца толькі калі новы кадр запускае ініцыялізацыю новых трохмерных крапак у прасторы, іначай выкарыстоўваюцца простыя метады якія, у сваю чаргу, няяўна даюць адпаведнасці паміж крапкамі, пазначанымі раней у якасці асаблівых.

## 2.5 Пошук адпаведнасцяў паміж ключавымі кропкамі

Пошук асаблівых кропак і падлік адпаведных дэскрыптараў з'яўляецца, вядома ж, толькі падрыхтоўчым этапам да этапа параўнання дэскрыптараў з дзвюх выявав і пошуку найлепшых параў (працэс матчынга, англ. *feature matching*). Не існуе ўніверсальнаага спосаба, які даваў бы найлепшыя вынікі на любым тыпе асаблівых кропак і іх дэскрыптараў: для кожнага набора дэскрыптараў існуе найбольш эфектыўны спосаб пошуку адпаведнасцяў. Таксама, выбар алгарытма залежыць ад паставленых намі задачаў і расстаўленых прыарытэтаў: ці нам найважнейшая хуткасць ці дакладнасць, ці дастаткова нам толькі  $n$  найлепшых супадзенняў ці мы хочам атрымаць усе знайдзеныя. Важным пытаннем з'яўляецца вызначэнне парога, калі знайдзеная пара дэскрыптараў з'яўляецца заведама хібнай.



Малюнак 2.5: вынік пошуку адпаведнасцяў паміж ключавымі кропкамі

Самым простым спосабам знайсці адпаведнасці паміж дэскрыптарамі з дзвюх розных выявав з'яўляецца прсты пералік усіх магчымых параў, падлік нормы ( $L_2$ , Хэмінга, альбо любой іншай) і сартыроўка ўсіх параў па ўзрастанні значэнняў адлегласцяў (гэтак званы Brute-Force Matcher). Такі спосаб з'яўляецца прстым у рэалізацыі, беспамылковым, але вельмі марудным. Практычнае прымяне ва ўмовах апрацоўкі плыняў дадзеных альбо пры пост-апрацоўцы вялікіх набораў дадзеных часта немагчымае.

Для выпадкаў, калі дакладнасцю можна ахвяраваць на карысць хуткасці (напрыклад, у працы з БПЛА хуткасць апрацоўкі грае першасную ролю), былі вынайдзены і іншыя алгарытмы пошуку адпаведнасцяў.

Метады, пра якія ідзе гаворка, маюць агульную назvu *метадай набліжсанага пошуку бліжэйшых суседзяў* (англ. *Approximate Nearest*

*Neighbors*). Падлічаныя намі дэскрыптары з'яўляюцца звычайным мноствам вектароў, на якім зададзеныя суадносіны адлегласці. У практычнай частцы такія метады будуть выкарыстоўвацца. Для дэскрыптараў з рэчаіснымі вектарамі (SIFT, SURF) гэта будзе метад заснаваны на *k*-мерных дрэвах (англ. *k-dimensional trees*), для бінарных дэскрыптараў - Locality-sensitive Hashing (LSH) - імавернасны метад паніжэння размернасці дадзеных.

На малюнку 2.5 можна бачыць нанесеная прастыя лініі, якія злучаюць крапкі, адпаведныя адной і той жа кропцы прасторы. Дэскрыптары былі падлічаныя алгарытмам SIFT.

## 2.6 Фармальная пастановка задачи рэканструкцыі

Задача аднаўлення трохмерных каардынатаў кропак па наборы выяваў, зробленых з розных ракурсаў і пазіцыяў, атрымала агульную назу задачы пучковай аптымізацыі (англ. **bundle adjustment**). Задача заключаецца ў мінімізацыі памылкі функцыянала, які задае суадносіны паміж мноствам кропак прасторы і праекцыямі гэтых кропак. Строга гэта можа быць запісаны як:

$$\min_{a_j, b_i} \sum_{i=1}^n \sum_{j=1}^m v_{ij} d(Q(a_j, b_i), x_{ij})^2 \quad (5)$$

дзе:

$n$  - колькасць кропак у трохмернай прасторы, якія бачныя хаця б на адным фотаздымку,

$m$  - колькасць выяваў (колькасць камер),

$x_{ij}$  - праекцыя крапкі  $i$  на выяву  $j$ ,

$v_{ij}$  - дваічная зменная, якая вызначае, ці бачная кропка  $i$  на выяве  $j$ ,

$a_j$  - вектар параметраў камеры  $j$ ,

$b_i$  - набліжэнне для крапкі  $i$  прасторы,

$Q(a_j, b_i)$  - функцыя пошуку праекцыі кропкі  $i$  на выяву  $j$ ,

$d(x, y)$  - Эўклідава адлегласць паміж кропкамі  $x$  і  $y$ .

Мінімізаваўшы памылку функцыянала, мы знайдзем найлепшае магчымае набліжэнне для параметраў камераў  $a_j$  і кропак трохмернай прасторы  $b_i$ , што дасць нам магчымасць зрабіць візуалізацыю мадэлі, што і з'яўляецца нашай мэтай.

Дадзеная задача можа рашацца як агульнымі падыходамі да мінімізацыі функцыяналу, так і спецыяльнай распрацаванымі алгарытмамі, якія ўлічваюць разрэджаную структуру матрыцы, якая апісвае функцыянал, такім чынам эфектыўна рашаючы задачу пучковай

аптымізацыі. Алгарытмам, які атрымаў найбольшы распаўсюд пры практычнай рэалізацыі, з'яўляеца алгарытм Левенберга-Марквардта (англ. *Levenberg–Marquardt algorithm, LMA*). Гэта ітэратыўны алгарытм рашэння задачы мінімізацыі нелінейнага функцыянала спосабам найменшых квадратаў (англ. *non-linear least squares problem*).

# ГЛАВА 3. SLAM-АЛГАРЫТМЫ

## 3.1 Асноўныя звесткі

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

Сваё найбольшое развіццё рашэнне SLAM задачы атрымала ў апошняе 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, такія як назіральнасць, збежнасць і ўзгодненасць алгарытмаў. У гэты час былі праведзеныя цесныя аналогіі з існуючымі падыходамі ў галіне камп'ютарнага зроку, была ствароная вялікая колькасць рэалізацыяў з адкрытым зыходным кодам, з'яўляюцца першыя спробы прымянення SLAM у практычных прыкладаннях.

Мяркуеца, што на трэці этап распрацоўка перайшла зусім нядаўна, у апошняе гады: гэта звязана з публікацыямі такіх алгарытмаў, як ORB-SLAM і LSD-SLAM, якія па сваёй функцыянальнасці моцна пераўзыходзяць любыя ранейшыя публікацыі. Прынцыпы працы гэтых алгарытмаў, як і некаторых іншых, будуць разгледжаныя далей.

Важна успрымаць SLAM не як адзіны прапанаваны алгарытм, але як канцепцыю. SLAM-сістэмы складаюцца са шматлікіх частак і кожная з гэтих частак можа ўдзельнічаць у рашэнні агульнай задачы з той ці іншай эфектыўнасцю.

Для задачы рэканструкцыі паверхні SLAM-сістэма наўрад ці будзе аптымальным рашэннем: пабудова шчыльных мадэляў у рэальнym часе пакуль яшчэ застаецца слаба вырашанай задачай. Тут і далей SLAM-сістэмы нас цікавяць сваімі выхаднымі дадзенымі: пасля апрацоўкі дадзеных у рэальнym 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 паміж сабой 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 становіща больш дакладнай.

Большасці сучасных SLAM-алгарытмаў прысутная магчымасць замыкаць цыклы (*loop closure*) - знаходзіць аб'екты, якія сустракаліся ў прасторы раней у межах таго ж пралёту/праходу асяроддзя, распознаваць іх у якасці *знаёмых* і глабальна ўдасканальваць мапу на падставе факта паўторна сустрэнутага аб'екта. Падыход добра паўплываў на якасць працы мноства алгарытмаў і зарэкамендаваў сябе адпаведным чынам – прычына поспеху ў надзеінасці вонкавых сістэм, здольных успрымаць відэаплыню і знаходзіць аб'екты незалежна ад асноўных плыняў алгарытма.

Разгледзім асноўныя этапы падрабязней.

## 3.2 Ключавыя паняцці і падыходы

### 3.2.1 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 абсталявання для лакалізацыі, што часта з'яўляецца непрактичным 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 адаметрыяй (англ. *Visual Odometry*). Надзеіныя механізмы лакалізацыі аўтаномных БПЛА асабліва важныя праз знаходжанне БПЛА ў паветры і немагчымасць спыніцца, у адрозненні на наземных робатаў.

### 3.2.2 Пошук на мапе

Ёсць некалькі важных аспектаў пабудовы мапы. Па-першае, мапа выкарыстоўваецца для пракладкі маршрутаў і лакалізацыі перашкодаў –

асноўныя элементы для аўтаномнай навігациі. 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 распознавання аб'ектаў і гэтак далей. Трэці і, магчыма, самы важны аспект у дачыненні да канцэпцыі SLAM – добра пабудаваная мапа дазваляе лакалізаваць БПЛА ў просторы.

Адзін з важных аспектаў пошука на мапе – магчымасць пошуку элементаў, месцаў, якія былі наведаныя і пабачаныя раней. Гэты працэс называецца *замкненнем цыклаў* (англ. *loop closure*).

Такім чынам, лакалізацыя і пошук на мапе не выступаюць як дзве асобныя і незалежныя часткі алгарытма, але дапаўняюць і паляпшаюць адна адну. Пошук сябе на папярэдне пабудаванай мапе (напрыклад, пры паўторным пралёце той жа мясцовасці) дазваляе лакалізаваць БПЛА з якасцю, параўнальнай з якасцю пабудованай мапы, у сваю чаргу лакалізацыя пры дапамозе адаметрыі і старонніх датчыкаў паслядоўна ўдасканальвае мапу. Сфармуляваўшы гэтую залежнасць іншымі словамі, пры наяўнасці дакладна вырашанай задачы альбо лакалізацыі, альбо пошуку на мапе, іншая задача таксама можа лічыцца вырашанай.

### 3.3 Падыходы да рэалізацыі

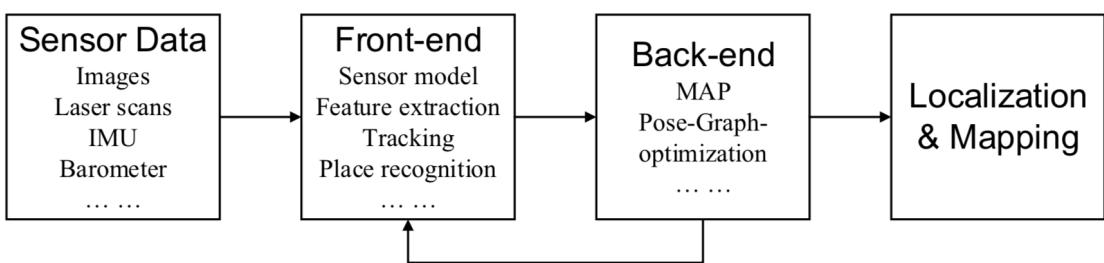
Як зазначаецца ў [2], архітэктура любой SLAM-сістэмы можа быць прадстаўленая схемай з малюнку 3.1. Фронтэнд выконвае першасную апрацоўку дадзеных, якія паступаюць са зневніх датчыкаў. Для алгарытмаў, заснаваных на пошуку асаблівых крапак, фронтэнд займаецца іх пошукам і апісаннем; для алгарытмаў, якія працуяць 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. Можна сказаць, што сёння асноўныя намаганні даследчыкаў прыкладаюцца да фронтэнду; бэкэнд жа можна ўмоўна лічыць стандартаваным, яго задача – аптымізацыя графа, які змяшчае інфармацыю пра пазіцыі ключавых кадраў і залежнасці паміж імі.

Задача SLAM з выкарыстаннем адзінай камеры ў якасці сэнсара на сённяшні момант лічыцца нявырашанай па прычыне нястачы вылічальных магутнасцяў, недахопе дадзеных пра глыбіню выявы і складанасці пошуку і асацыяцыі паміж сабой ключавых крапак. У некаторых ситуациях некаторыя з вышэйзгаданых проблемаў могуць быць вырашаныя (альбо часткова вырашаныя). Напрыклад, задача высвятлення глыбіні выявы не стаіць так востра пры выкарыстанні

стэрэакамеры, альбо калі пра манакулярную камеру вядома, што яна глядзіць уніз (англ. *downward looking camera*).

Важным крокам у рэалізацыі SLAM-алгарытмаў стала дамінаванне ідэі раздзялення лакалізацыі і пошуку на мале ў асобныя плыні. Упершыню практичнае прымяенне ідэя знайшла ў сістэме PTAM (*Parallel Tracking and Mapping for Small AR Workspaces*, [9]). Лакалізацыйная плыня PTAM займаецца пошукам адпаведнасцяў у дадзеных і запускае алтымізацыю па рухах камеры. У гэты ж час плыня з мапай калекцыяне вынікі лакалізацыі, трывангулюе выяўленыя асаблівасці ў трохмерныя кропкі і абнаўляе глабальную мапу. Са з'яўленнем такога падыходу сталі відавочнымі ягоныя перавагі і амаль кожная сучасная SLAM-сістэма пабудаваная па падобным прынцыпе.



Малюнак 3.1: высокаўзрознавая архітэктура SLAM-алгарытмаў

### 3.4 Агляд SLAM-сістэм

Далей ідзе апісанне асаблівасцяў некаторых SLAM-сістэм, якія на сённяшні дзень паказваюць найлепшыя вынікі. Некаторыя з іх заснаваныя на пошуку асаблівых крапак (*feature-based SLAM*), другія – на простых метадах (*direct SLAM*), трэція – камбінуюць у сабе абедва падыходы (*semi-direct SLAM*). Да прыкладу, ORB-SLAM ([4], [5]) адносіцца да першай катэгорыі SLAM-сістэм, LSD-SLAM ([6]) – да другой, SVO([3]) – да трэцяй.

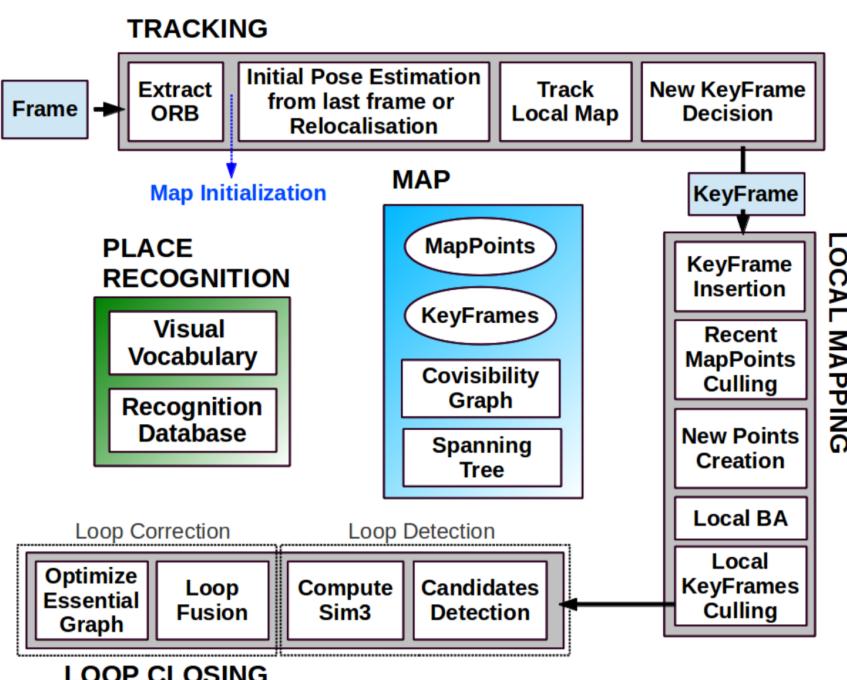
#### 3.4.1 ORB-SLAM

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

Асноўныя асаблівасці:

- выкарыстанне адных і тых жа ключавых крапак на ўсіх этапах працы алгарытма;

- четыры асноўныя задачы, якія адначасова реаце алгарытм: трэкінг (*tracking*), наяненне дадзеных на мапу (*mapping*), рэлакалізацыя (*relocalization*), замкненне цыклаў (*loop closing*);
- неабмежаваны рост памераў мапы *толъкі* 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 *графа узаемабачнасці* (*covisibility graph*);
- новы аўтаматычны і надзеіны спосаб ініцыялізацыі, які стварае пачатковыя мапы як для планарных, так і для непланарных паверхняў;
- пазбаўленне ад лішніх ключавых кадраў па стратэгіі “выжывання наймацнейшага” (*survival of the fittest*);



Малюнак 3.2: архітэктура ORB-SLAM

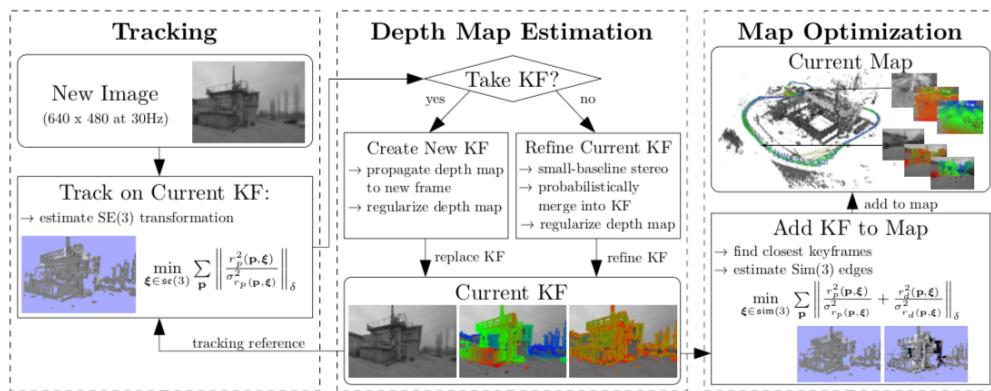
- агульная схема працы алгарытма прадстаўленая на малюнку 3.2.

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 (аперацыйная сістэма, шырокая распаўсюджаная сярод робататэхнікаў), у сваю чаргу ў другой версіі з'явіліся ўласныя сродкі візуалізацыі, здольныя працаўца незалежна.

### 3.4.2 LSD-SLAM

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



Малюнак 3.3: архітэктура LSD-SLAM

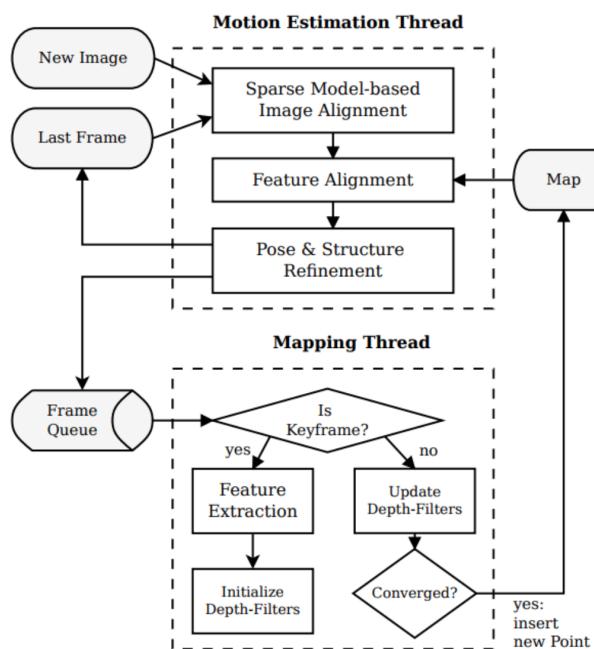
- працуе ў рэальнym часе на CPU;
- складаецца з трох асноўных кампанент: трэкінг, падлік набліжэння для мапы глыбіняў і аптымізацыя мапы;
- трэкінг кампанента паслядоўна апрацоўвае новыя здымкі, прымаючы да ўвагі папярэдні і бягучы ключавыя здымкі;
- другая кампанента выкарыстоўвае здымкі каб альбо палепшыць, альбо замяніць бягучы ключавы кадр і ацэнывае глыбіню сцэны простымі (*direct*) і імавернаснымі падыходамі;
- здымкі, якія канчаткова пазначаныя як ключавыя (мапа глыбіняў больш змяняцца не будзе) заносяцца на глабальную мапу;
- мапа - граф узаемаразмешчэння ключавых кадраў. Паколькі алгарытм таксама ацэнывае мапы глыбіняў, то мапа, па сутнасці, ўяўляе сабой напаўшчыльную 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

глыбінняў будзе хутка збягацца. Аўтары прызнаюцца ў неабходнасці дапрацаваць гэту частку сістэмы; пры тэставых запусках мапа глыбінняў для ініцыялізацыі бралася непасрэдна з датасэтаў (то бок была дакладна вядомая), на практыцы падобны падыход можа запатрабаваць пэўнага чалавечага ўдзелу;

- архітэктура сістэмы прадстаўленая на малюнку 3.3.

### 3.4.3 SVO

У адрозненні ад папярэдніх двух прыкладаў, SVO (*Semi-Direct Monocular Visual Odometry*, [3]) спалучае ў сабе як пошук і апісанне ключавых крапак, так і простыя метады, праз што і атрымаў сваю назvu (*Semi-Direct, паўпросты*).



Малюнак 3.4: архітэктура SVO

- спалучае ў сабе лепшае ад метадаў, заснаваных на ключавых крапках (выбар ключавых кадраў, сачэнне за мноствам крапак, паралельны трэкінг і пошук на мапе), і ад простых метадаў (хуткасць і дакладнасць);
- пошук адпаведнасцяў паміж ключавымі крапкамі здзяйсняецца няяўна як вынік прымянення простых метадаў (а не, напрыклад, matching-a);
- выманне ключавых крапак здзяйсняецца толькі на ключавых кадрах;

- пасля таго, як адпаведнасці паміж ключавымі кропкамі, а таксама пачатковая пазіцыя камеры ўсталяваныя, алгарытм толькі працуе з непасрэднымі значэннямі пікселяў у кропках. Пазіцыя камеры адносна папярэдняга кадра падлічваецца праз мінімізацыю фотаметрычнай памылкі;
- трохмерная кропка дадаецца на мапу толькі ў выпадку збежнасці адпаведнага фільтра глыбіні, што азначае патрэбнасць у шматлікіх вымярэннях для кожнай асобнай кропкі;
- глыбінныя фільтры для ключавых кадраў ініцыялізуюцца сярэднім значэннем сярод усіх глыбінных фільтраў сцэны;
- свядзенне да мінімума выкідаў у вымярэннях;
- архітэктура сістэмы прадстаўленая на малюнку 3.4.

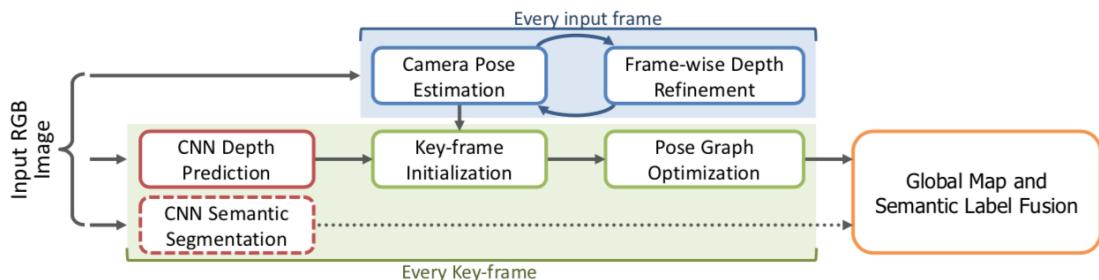
### 3.4.4 CNN-SLAM

Аўтары алгарытма CNN-SLAM [18] прарапаноўваюць альтэрнатывы падыход да ацэнкі глыбіні сцэны пры наяўнасці адзінай камеры. Найболыш класічным ў такім выпадку лічыцца імітацыя стэрэакамеры (ацэнка глыбіні сцэны пра невялікіх рухах камеры ў просторы), аднак такі падыход мае пэўныя проблемы, якія не дазваляюць ягонае надзеянае выкарыстанне. Напрыклад, падыход не прымняльны пры паваротах у просторы. Аўтары прарапаноўваюць будаваць глыбіню, базуючыся на адзіным кадры з дапамогай канвалюцыйных нейронных сетак (*Convolutional Neural Networks, CNN*, адсюль паходзіць назва алгарытма). Сярод іншых асаблівасцяў алгарытма:

- дазваляе абысці адно з найвялікшых абмежаванняў усіх SLAM-алгарытмаў – немагчымасць ацаніць рэальныя масштабы сцэны, якая аглядаецца. SLAM-алгарытмы заўжды даюць вынікі з дакладнасцю да масштабу, калі толькі не прымняюцца спецыяльныя падыходы, якія дапамагаюць зразумець рэальныя памеры аб'ектаў. Прыкладам такога падыхода можа быць распознаванне аб'ектаў і наступнае парашнанне з базай аб'ектаў, для якіх вядомыя іх фізічныя характеристыкі. Зразумела, што такі падыход не дасць аніякіх вынікаў пры адсутнасці на сцэне хаця б аднаго аб'екта з базы;
- нейронная сетка навучаецца не на прыкладах з аднаго канкрэтнага асяроддзя, але па наборы дадзеных, які змяшчае ў сабе вельмі разнастайныя выявы. Гэта дазваляе, аднойчы навучыўшы

SLAM-сістэму, запускаць яе ў разнастайных асяроддзях (у тым ліку – у незнаёмых) з адолькавай паспяховасцю;

- мапы глыбіняў, згенераваныя нейроннай сеткай, хаця і глабальна карэктныя, даюць размытасці на вуглах, граніцах. Гэта не дазваляе выкарыстоўваць згенераваныя мапы глыбіняў у сваім пачатковым выглядзе, але дазваляе паспяхова выкарыстоўваць іх у якасці апрыорных ацэнак;
- мапы глыбіняў, хоць і з размытасцямі, але даюць уяўлянне пра сапраўдную глыбіню сцэны, яе масштаб;
- мапы глыбіняў не трапляюць пад уплыў праблемаў з паваротамі камераў у просторы (без адпаведнага перасоўвання ў просторы), праз што трэкінг становіцца значна больш надзейным;
- наяўнасць працэса нармалізацыі для ўзгаднення памераў і каліяровых характеристыстыкаў выяву паміж наборамі, на якіх алгарытм вучыўся, і выявамі, якія паступаюць падчас працы алгарытма;
- для падтрымкі працы ў рэальнym часе глыбіня ацэньваецца толькі для ключавых кадраў, а не для ўсіх, якія паступаюць з відэаплыні;



Малюнак 3.5: архітэктура CNN-SLAM

- дадатковая нейронная сетка для семантычнай сегментацыі выяву – асобнае даследаванне, праведзенае ў межах працы і рэалізаванае ў тым жа фрэймворку;
- вынікі эксперымантаў паказваюць, што алгарытм на большасці набораў дадзеных працуе хутчэй за LSD-SLAM і ORB-SLAM, сённяшніх лідэраў сярод SLAM-алгарытмаў, з якімі праводзілася параўнанне, а таксама дае непараўнальная лепшую ацэнку для глыбіні сцэны;
- апроч таго, былі праведзеныя асобныя эксперыменты па прымяняльнасці алгарытма да дадзеных, якія атрымліваліся толькі праз павароты камеры. Паколькі гэтая папулярная сярод

SLAM-сістэмаў праблема ў выпадку з CNN-SLAM абыходзіцца праз выкарыстанне нейронных сетак для ацэнкі глыбіні, то, як вынік, алгарытм паказвае дастойныя вынікі на такога кшталту дадзеных і моцна апярэджвае любая іншыя сучасныя алгарытмы;

- архітэктура сістэмы прадстаўленая на малюнку 3.5.

# ГЛАВА 4. ПРАКТЫЧНАЯ РЭАЛІЗАЦЫЯ

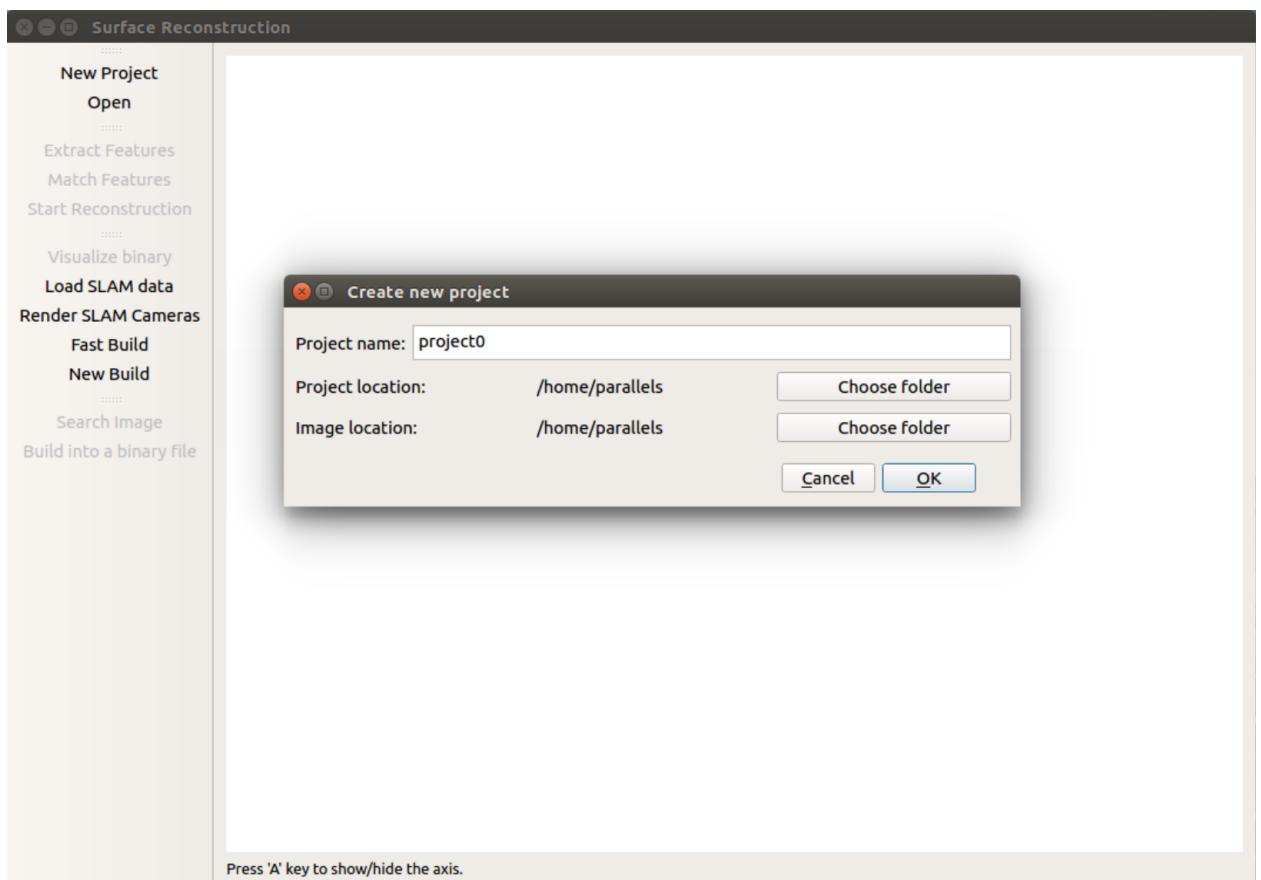
## 3.1 Распрацаванае праграмнае забеспячэнне

У якасці практычнай часткі, сярод іншага, было распрацаванае праграмнае забеспячэнне для правядзення рэканструкцыі паверхні па наборы здымкаў і, пры наяўнасці, па дадатковых дадзеных, такіх як значэнні зневідных альбо ўнутраных параметраў камераў.

Праграма распрацоўвалася на C++, з выкарыстаннем бібліятэкаў Qt, OpenGL (для візуалізацыі трохмерных мадэляў), TheiaSfm ([16], змяшчае некаторыя алгарытмы камп'ютарнага зроку). Праграмнае забеспячэнне даступнае для запуску на Ubuntu і macOS.

Базавы функцыонал уключае ў сябе:

- працу з праектамі: стварэнне (прыклад на малюнку 4.1), кіраванне, паўторнае адкрыццё, кіраванне пабудаванымі мадэлямі.

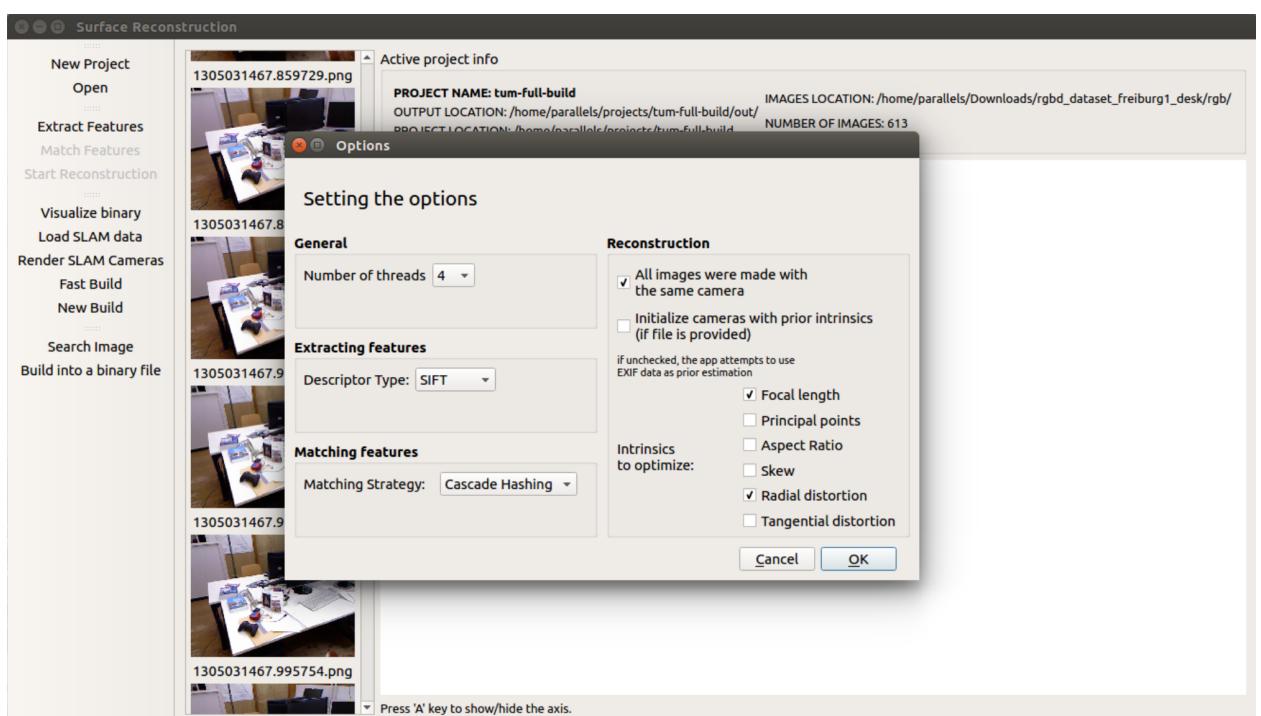


Малюнак 4.1: стварэнне новага праекта

- выманне і апісанне ключавых кропак, захаванне іх у файлавую сістэму. Захаванне прамежкавых этапаў рэканструкцыі, такіх як

апісаныя ключавыя кропкі, вельмі станоўча адбіаеца на часе запускаў на тых жа наборах дадзеных: замест паўторнага вымання кропак яны чытаюцца з файлавай сістэмы, тады як наступныя этапы могуць запускацца ўжо з іншымі параметрамі.

- падобны падыход і з пошукам адпаведнасцяў: знайдзеныя тым ці іншым спосабам адпаведнасці захоўваюцца ў файлавую сістэму для паўторнага выкарыстання.
- усе этапы рэканструкцыі кантралююцца вялікай колькасцю параметраў, асабліва зручна кіраваць якімі праз графічны інтэрфейс (як на малюнку 4.2).



Малюнак 4.2: налады, даступныя да зменаў пры запуску працэса рэканструкцыі

- наяўнасць рэжыма запуска праз камандны радок (малюнак 4.3), вельмі зручнага для частых запускаў і для напісання скрыптоў для правядзення эксперыменту. Магчымасць праглядзець трохмерную мадэль прысутнічае пасля ў графічным інтэрфейсе.

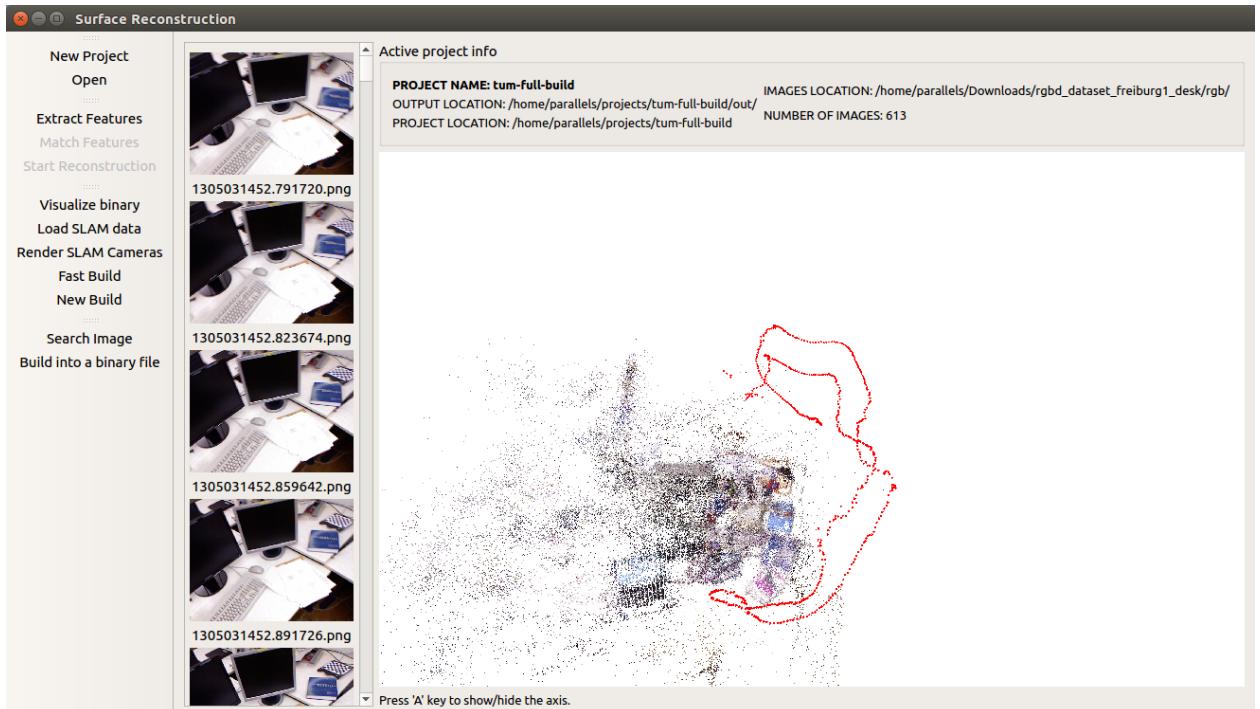
```

parallels@ubuntu:~/reconstruction
parallels@ubuntu:~/reconstruction$ ./app --mode=cli \
> --cli_mode=open \
> --project_path=/home/parallels/projects/swiss-church/ \
> --action=reconstruction \
> --shared_calibration=true \
> --use_camera_intrinsics_prior=false \
> --intrinsics_to_optimize="focal,raddist"
I0527 19:17:00.656311 21881 main.cpp:17] Logging is enabled.
I0527 19:17:00.656687 21881 cli.cpp:82] Running in CLI mode.
I0527 19:17:00.659006 21881 storage.cpp:54] Force initialization: success 37 read
I0527 19:17:00.660213 21881 options.cpp:54] Options flags parsed successfully.
I0527 19:17:00.660301 21881 cli.cpp:113] Starting reconstruction...
I0527 19:17:00.772471 21881 reconstructor.cpp:133] Matches read from filesystem successfully
I0527 19:17:00.772588 21881 reconstructor.cpp:56] Starting reconstruction process...

```

Малюнак 4.3: інтэрфейс каманднага радка

- адна з асноўных функцыяў для якой, уласна, спатрэбілася распрацоўка графічнага інтэрфейса - акно візуалізацыі пабудаваных мадэляў. Прыклад праграмы з загружанай у памяць мадэллю можна пабачыць на малюнку 4.4.

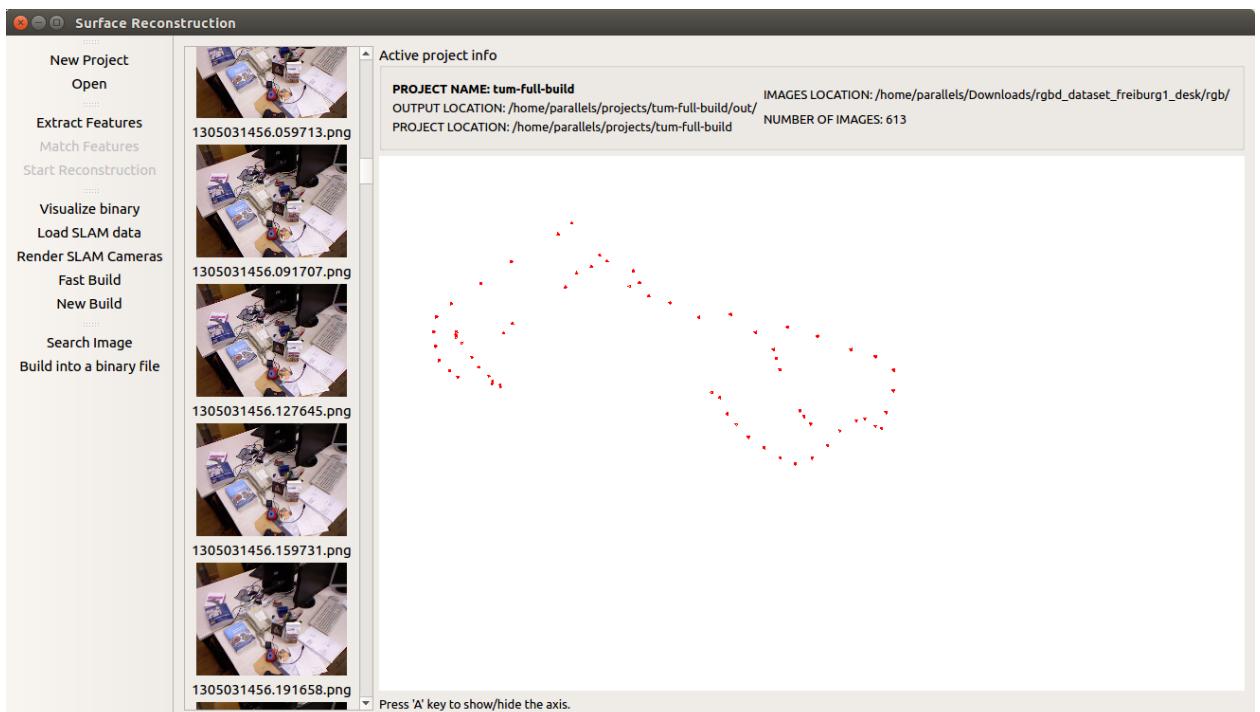


Малюнак 4.4: візуалізацыя воблака кропак ў рэжыме графічнага інтэрфейса

- магчымасць націскаць выявы ў акенцы злева і падсвечваць адпаведныя ім камеры ў акне візуалізацыі.
- ініцыялізацыя унутраных параметраў камераў папярэднімі значэннямі.
- магчымасць працы ў рэжыме, калі кожная камера (з пункту гледжання алгарытма) з'яўляецца адной і той жа фізічнай камерай (для нашых задачаў мы амаль заўсёды працуем у гэтым рэжыме). Дадзены рэжым абагульніяецца на выпадкі, калі, напрыклад, сярод

усяго мноства здымкаў палова зроблена адной фізічнай камерай (з адным наборам унутраных параметраў), іншая палова - іншай камерай.

Важней частка праграмы з'яўляецца інтэграцыя з SLAM-алгарытмамі: дадзеныя, якія падчас пралёту генеруюць SLAM-сістэмы, захоўваюцца і пасля выкарыстоўваюцца ў праграме. Напрыклад, на малюнку 4.5 паказаныя адмаліваныя камеры, пазіцыі якіх падлічаныя SLAM-алгарытмам. На жаль, некаторыя з гэтых магчымасцяў толькі даступныя ў выглядзе прататыпаў і патрабуюць наступнай распрацоўкі. Падрабязней пра распрацоўку сістэмы, якая б спалучала SLAM-алгарытм з традыцыйным алгарытмам пабудовы мадэлі, распавядаецца ў наступным пункце.



Малюнак 4.5: адмалёўка камераў, згенераваных SLAM-алгарытмам

### 3.2 Інтэграцыя з SLAM-сістэмамі

Архітэктура сістэмы, рэалізацыяй якой я займаўся, і ўсе яе ключавыя вузлы схематычна прадстаўленыя на малюнку 4.6.

Падрабязней будова сістэмы і яе кампанент, а таксама ўзаемадзеянне паміж яе часткамі раскрываюцца ў наступных пунктах.



Малюнак 4.6: архітэктура спалучанай сістэмы

### 3.2.1 ROS

ROS (The Robot Operating System) – фрэймворк для напісання праграмнага забеспячэння для робатаў, з'яўляеца наборам інструментаў, бібліятэкаў і пагадненняў для зручнай і надзейнай працы з робатамі ([8]). Патрэбнасць у фрэймворку ўзнікла праз той факт, што стварэнне праграмнага забеспячэння для робатаў – вельмі складаны і шматэтапны працэс; уніфікацыя способу камунікацыяў паміж часткамі вялікай сістэмы дазваляе камандам працеваць 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,
- аナンімная і асінхронная сістэма публікацыі і падпіскі на паведамленні,
- магчымасць реалізацыі 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 ўбудоўваемай ў іншыя сістэмы. Вялікая колькасць SLAM-алгарытмаў рэалізаваныя менавіта пад ROS і могуць не мець уласных сродкаў візуалізацыі альбо ўводу дадзеных.

Казаць пра ROS можна не толькі у дачыненні да робатаў, але і, як у нашым выпадку, да беспілотных лятальных апаратоў, бо разважанні не губляюць сваёй агульнасці. ROS зручны для працы з любымі сістэмамі, неаўтаномнымі, паўаўтаномнымі альбо цалкам аўтаномнымі, якія ўключаюць у сябе пэўны набор датчыкаў (такіх як камеры), маюць альбо не маюць уласных вылічальных магутнасцяў, могуць кантроліраваць сваё становішча ў просторы. Пад такое азначэнне любы БПЛА, безумоўна, падыходзіць.

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

### 3.2.2 Тыпы ўваходных інтэрфейсаў

У сістэме з малюнку 4.6 вялася праца з двумя тыпамі ўваходных інтэрфейсаў для прыняцця 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 апарате пры выкананні рэальнага палёту. Прыблізна ў такім жа рэжыме працуе сапраўдны БПЛА пры пралёце над мясцовасцю.

Другі – публікацыя паслядоўнасці кадраў загадзя падрыхтаванага набора дадзеных. Такі рэжым значна лепш падыходзіць для тэставання і адладкі алгарытмаў і дае магчымасць паўтараць эксперыменты на аднолькавых дадзеных.

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

### 3.2.3 Выманне дадатковых дадзеных са SLAM-алгарытмаў

Рэалізацыі большасці SLAM-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].

Распрацоўка ўласных SLAM-алгарытмаў не ставілася ў якасці мэты – гэта вялікая праца, якая патрабуе годы даследаванняў і, да таго ж, звычайна робіцца вялікай камандай распрацоўшчыкаў. У гэтай працы выкарыстоўваюцца існуючыя SLAM рэалізацыі, удасканаленыя для нашых патрэбаў.

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 кроکаў.

Першае, што варта тут згадаць – патрэбныя нам дадзенія не выдаюцца алгарытмамі ў гатовым выглядзе. Цікавыя нам значэнні параметраў схаваныя ўнутры рэалізацыяў, але праз наяўнасць адкрытага зыходнага кода з'яўляецца шанец працеваць з патрэбнымі нам дадзенімі. У якасці выходных дадзеных нас найбольш цікавіць граф узаемабачнасці крапак з камераў (*covisibility graph*) – граф, вяршынямі якога з'ўляюцца камеры (альбо ключавыя кадры), рэбрь праводзяцца паміж тымі вяршынямі, адпаведныя камеры якіх маюць дастаткова вялікую агульную аглядаемую прастору (выміраецца ў колькасці знойдзеных агульных трохмерных крапак). 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 дадзенія у кодзе і знаходзіць магчымасці выгрузкі іх у вонкавую прастору.

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

У выніку, большасць часу была прысвеченая ORB-SLAM ([4], [5]) як сістэме, якая давала найбольш надзейныя вынікі для ацэнак пазіцыяў камер, а таксама якая досыць зручна інтэгравалася у архітэктуру з малюнку 4.6.

### **3.2.4 Аптымізацыя камунікацыяў з дапамогай ROS**

Яшчэ адным накірункам працы з'яўлялася аптымізацыя камунікацыяў паміж часткамі сістэмы. У некаторых месцах дадзеныя ў кодзе перадаюцца праз файлавую сістэму, але ў большасці сітуацыяў камунікацыі аптымізаваныя з дапамогай убудаваных сродкаў ROS: публікацыя ў ROS-аўскі “topic” з аднаго боку і праслушоўванне з іншага з'яўляеца, у агульным выпадку, значна больш дасканалым спосабам злучэння асобных элементаў сістэмы з малюнку 4.6 у адно суцэльнае. Гэты способ таксама дазваляе павялічыць хуткасць камунікацыяў, бо абмен дадзенымі ідзе праз RAM-памяць замест файлавай сістэмы.

### **3.2.5 Статус распрацаванасці**

Была рэалізаваная большасць кампанентаў сістэмы, пазначаных на схеме 4.6, але праз той факт, што для ўзгодненай працы ўсёй сістэмы патрэбная яе поўная рэалізацыя, патрэбнае правядзенні серыі інтэграцыйных тэстаў і серыі дадатковых эксперыменталаў, сістэма не была сапраўды даведзеная да крапкі, на якой з'явілася б магчымасць падзяліцца вынікамі, якія б сцвярджалі, наколькі жыццяздольнай з'яўляеца падобная архітэктура. У любым выпадку, ёсць упэўненасць, што паколькі запуск SLAM-алгарытма з'яўляеца “бясплатным” – у тым сэнсе, што адбываеца ў рэальнym часе і затрачвае толькі вылічальныя магутнасці сістэмы, але не часавыя, то інтэграцыя SLAM-сістэмаў з традыцыйнымі будзе мець станоўчы эфект і пазітыўна адлюструеца на колькасных і якасных характеристыках рашэння задачы рэканструкцыі паверхні.

# ВЫСНОВЫ

У дадзенай справа задачы былі апісаныя асноўныя паняцці і канцэпцыі, датычныя да задачы трохмернай рэканструкцыі паверхні па дадзеных з БПЛА, сформуляваная канцэпцыя SLAM, апісаныя некаторыя найбольш вядомыя і паспяхова рэалізаваныя SLAM-сістэмы.

Негледзячы на тое, што SLAM, рэалізаваны для працы ў рэальнym часе, і які часцей за ўсё выкарыстоўваецца для навігациі ў прасторы, кепска спраўляеца з задачай пабудовы шчыльнай трохмернай мадэлі, вынікі адпрацоўкі SLAM могуць быць вельмі карыснымі ў якасці пачатковых дадзеных для запуску традыцыйных алгарытмаў рэканструкцыі.

У якасці практычнай рэалізацыі было расправацаванае праграмнае забеспячэнне, якое здзяйсняе поўны цыкл рэканструкцыі па наборы здымкаў з мноствам дадатковага функцыяналу, такога як: падтрымка праектаў і захаванне іх у файлавую сістэму, візуалізацыя трохмерных мадэляў і захаванне мноства мадэляў у межах аднаго праекта, выманне і апісанне ключавых кропак, пошук адпаведнасцяў паміж асаблівымі кропкамі, правядзенне непасрэдна працэсу рэканструкцыі. Другой практычнай задачай з'яўлялася распрацоўка сістэмы, якая б інтэгравала ў сябе SLAM-алгарытм для падліку папярэдніх значэнняў некаторых параметраў, такіх як пазіцыі камераў у прасторы, якія пасля бы выкарыстоўваліся ўсё tym жа праграмным забеспячэннем для правядзення рэканструкцыі з большай хуткасцю і якасцю. Была распрацаваная архітэктура такой сістэмы і рэалізаваная большасць яе кампанентаў, аднак не была дасягнутая суцэльнасць сістэмы, пры якой можна было б казаць пра паспяховую яе рэалізацыю і дасягненне канчатковай мэты. Тым не менш, ёсць перакананасць, што SLAM-алгарытмы сапраўды могуць заняць моцныя пазіцыі ў задачах пабудовы шчыльнай трохмернай рэканструкцыі, нават калі частка абавязкаў па ўшчыльненні мадэлі будзе ляжаць на алгарытмах, няздольных працаваць у рэальнym часе.

## СПІС КРЫНІЦ

- [1] Simultaneous localization and mapping: Present, future, and the robust-perception age. / Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian D. Reid, and John J. Leonard. // CoRR, abs/1606.05830, 2016.
- [2] Real-time simultaneous localization and mapping for uav: A survey. / Jiaxin Li, Yingcai Bi, Menglu Lan, Hailong Qin, Mo Shan, Feng Lin, and Ben M. Chen. // 2016. - Mode of access: [http://www.imavs.org/papers/2016/237\\_IMAV2016\\_Proceedings.pdf](http://www.imavs.org/papers/2016/237_IMAV2016_Proceedings.pdf). - Date of access: February 2018.
- [3] SVO: Fast semi- direct monocular visual odometry. / Christian Forster, Matia Pizzoli, and Davide Scaramuzza. // In IEEE International Conference on Robotics and Automation (ICRA), 33(2):249-265, 2014.
- [4] ORB-SLAM: a versatile and accurate monocular SLAM system. / Montiel J. M. M. Mur-Artal, Raúl and Juan D. Tardós. // IEEE Transactions on Robotics, 31(5):1147–1163, 2015.
- [5] ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. / Raúl Mur-Artal and Juan D. Tardós. // IEEE Transactions on Robotics, 33(5):1255–1262, 2017.
- [6] LSD-SLAM: Large-scale direct monocular SLAM. / J. Engel, T. Schöps, and D. Cremers. // In European Conference on Computer Vision (ECCV), September 2014.
- [7] A benchmark for the evaluation of rgb-d slam systems. / J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. // In Proc. of the International Conference on Intelligent Robot Systems (IROS), Oct. 2012.
- [8] Ros: an open-source robot operating system. / Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. // In ICRA Workshop on Open Source Software, 2009.
- [9] Parallel tracking and mapping for small ar workspaces. / Georg Klein and David Murray. // 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.
- [10] Distinctive image features from scale-invariant keypoints. / David G. Lowe. // Int. J. Comput. Vision, 60(2):91–110, November 2004.

- [11] Speeded-up robust features (surf) / Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. // Comput. Vis. Image Underst., 110(3):346–359, June 2008.
- [12] Machine learning for high-speed corner detection. / Edward Rosten and Tom Drummond. // In Proceedings of the 9th European Conference on Computer Vision - Volume Part I, ECCV’06, pages 430–443, Berlin, Heidelberg, 2006. Springer-Verlag.
- [13] Orb: An efficient alternative to sift or surf. / Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. // In Proceedings of the 2011 International Conference on Computer Vision, ICCV ’11, pages 2564–2571, Washington, DC, USA, 2011. IEEE Computer Society.
- [14] Brief: Binary robust independent elementary features. / Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. // In Proceedings of the 11th European Conference on Computer Vision: Part IV, ECCV’10, pages 778–792, Berlin, Heidelberg, 2010. Springer-Verlag.
- [15] Brisk: Binary robust invariant scalable keypoints. / Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart. // In Proceedings of the 2011 International Conference on Computer Vision, ICCV ’11, pages 2548–2555, Washington, DC, USA, 2011. IEEE Computer Society.
- [16] Theia multiview geometry library: Tutorial & reference. [Electronic resource] / Chris Sweeney. - Mode of access: <http://theia-sfm.org>. - Date of access: March 2018.
- [17] About direct methods. / Michal Irani and P. Anandan. // In Proceedings of the International Workshop on Vision Algorithms: Theory and Practice, ICCV ’99, pages 267–277, London, UK, UK, 2000. Springer-Verlag.
- [18] CNN-SLAM: real-time dense monocular SLAM with learned depth prediction. / Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. // CoRR, abs/1704.03489, 2017.