

**МІНІСТЭРСТВА АДУКАЦЫІ РЭСПУБЛІКІ БЕЛАРУСЬ**

**БЕЛАРУСКІ ДЗЯРЖАЎНЫ УНІВЕРСІТЭТ**

**Факультэт прыкладной матэматыкі і інфарматыкі**

**Кафедра дыскрэтнай матэматыкі і алгарытмікі**

**РЭКАНСТРУКЦЫЯ ПАВЕРХНІ ПА ДАДЗЕННЫХ З  
БЕСПЛОТНЫХ ЛЯТАЛЬНЫХ АПАРАТАЎ**

Курсавы праект

Богдана Уладзіслава Уладзіміравіча  
студэнта 4 курса  
спецыяльнасць "інфарматыка"

Навуковы кіраўнік:  
асістэнт Жылка Андрэй Ігаравіч

Мінск, 2017

## **Анотацыя**

У курсавым праекце апісваецца задача рэканструкцыі паверхні па дадзеных з беспілотных лятальных апаратаў (БПЛА), прыводзіцца апісанне канцэпта SLAM-сістэм (адначасовай лакалізацыі і пошуку на мапе), разглядаюцца некаторыя найбольш паспяховыя SLAM-алгарытмы. У працы прадстаўленыя вынікі эксперыментаў па даследаванні збежнасці SLAM-сістэм і апісваюцца магчымасці інтэграцыі з класічным падыходам да рэканструкцыі.

## **Аннотация**

В курсовом проекте описывается задача реконструкции поверхности по данным с беспилотных летательных аппаратов (БПЛА), приводится описание концепта SLAM-систем (одновременной локализации и поиска на карте), рассматриваются некоторые наиболее успешные SLAM-алгоритмы. В работе представлены результаты экспериментов по исследованию сходимости SLAM-систем и описываются возможности интеграции с классическим подходом к реконструкции.

## **Annotation**

The surface reconstruction based on data from unmanned aerial vehicles (UAV's) problem is described as well as the concept of SLAM (simultaneous localization and mapping) and some most successful SLAM implementations. The work contains results of the experiments determining convergence abilities of SLAM systems. Ways to integrate SLAM with classical approach are considered.

# ЗМЕСТ

УВОДЗІНЫ . . . . .	4
ГЛАВА 1. ЗАДАЧА РЭКАНСТРУКЦЫІ ПАВЕРХНІ . . . . .	5
1.1 Агульныя звесткі . . . . .	5
1.2 Сферы прымянення . . . . .	5
1.3 Адрозненні ў тыпах камер . . . . .	6
1.4 Рэканструкцыя паверхні афлайн . . . . .	8
ГЛАВА 2. SLAM-АЛГАРЫТМЫ . . . . .	10
2.1 Агляд . . . . .	10
2.1.1 Лакалізацыя . . . . .	11
2.1.2 Пошук на мапе . . . . .	11
2.2 Падыходы да рэалізацыі . . . . .	11
2.3 Агляд SLAM-рэалізацыяў . . . . .	12
2.3.1 ORB-SLAM . . . . .	12
2.3.3 LSD-SLAM . . . . .	14
2.3.2 SVO . . . . .	15
ГЛАВА 3. SLAM НА ПРАКТЫЦЫ . . . . .	17
3.1 ROS . . . . .	17
3.2 Даследванне збежнасці ORB-SLAM . . . . .	17
ВЫСНОВЫ . . . . .	20
СПІС КРЫНІЦАЎ . . . . .	21

# УВОДЗІНЫ

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

Непасрэдна задача рэканструкцыі цесна звязаная са шматлікімі сумежнымі задачамі: распазнаванне аб'ектаў, навігацыя ў прасторы, пабудова мапы мясцовасці. Асабліваю 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.

У гэтай працы я падсумую найважнейшыя аспекты, якія ўзнікаюць пры рашэнні задачы рэканструкцыі паверхні, а таксама разгледжу шэраг алгарытмаў з агульнай назвай SLAM (Simultaneous Localization and Mapping), якія могуць быць эфектыўна прымененыя да задачаў, якія патрабуюць найбольшай хуткасці выканання, такіх як навігацыя.

# 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 зробленых з розных ракурсаў здымкаў. 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 для іншых тыпаў уваходных дадзеных: апроч камеры ўваходнымі дадзенымі для задачы могуць быць дадзеныя з іншых датчыкаў, такіх як акселерометр, гіраскоп ці GPS-датчык; замест манакулярнай камеры можа прымяняцца RGB-D (вяртае дадатковы слой глыбіняў) ці стэрэа камера, якая ўяўляе сабой дзве RGB камеры і 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 службамі для ацэнкі наступстваў прыродных катастроф, у сельскай гаспадарцы, дарожнымі службамі. 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.

### 1.3 Адрозненні ў тыпах камер

Негледзячы на цікавыя магчымасці такіх датчыкаў, як акселерометр, гіраскоп альбо лазер вымярэння адлегласці, яны не заўсёды даступныя і не даюць універсальна добрых вынікаў у любых умовах. Далей мы будзем весці гутарку пра розныя тыпы камер і больш за ўсё - пра манакулярную, як найбольш універсальную, распаўсюджаную і простую ў эксплуатацыі.

#### Манакулярная RGB камера

Самымі распаўсюджанымі з'яўляюцца манакулярныя камеры - камеры, якія прысутнічаюць у смартфонах, ужываюцца ў быце і з якімі мы ўсе выдатна знаёмыя. Манакулярныя камеры ў сваю чаргу таксама бываюць розных тыпаў, але на практыцы мы часцей за ўсё працуем з праектыўнымі (projective, pinhole) камерамі.

Праецыраванне трохмерных кропак у гэтым выпадку можа быць апісанае наступнай формулай:

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

дзе  $P$  - матрыца, якая здзяйсняе пераўтварэнне,  $X, x$  - вектары, якія адпавядаюць трохмернай і двухмернай кропцы 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,  $R$  - матрыца павароту,  $t$  - вектар зрушэння камеры ў прасторы ў сусветных каардынатах. Гэтыя параметры задаюць праектыўную камеру, якая здзяйсняе праекцыю кропкі прасторы на матрыцу камеры. Унутраныя параметры могуць задавацца разнастайным чынам, распаўсюджанымі прыкладамі могуць быць 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 цэнтру матрыцы. Іншымі параметрамі камеры могуць быць радыяльныя скажэнні  $k_1, k_2$ , якія ўзнікаюць з-за неідэальнасці лінзаў у камерах.

Унутраныя параметры камеры заўжды вядомыя загадзя альбо праз дакументацыю 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.

Варта зазначыць, што адной з найбольшых перавагаў і, адначасова, адной з найбольшых перашкодаў у працы ў манакулярнымі камерамі з'яўляецца неадназначнасць працы з маштабам. Праблема ў тым, што маштаб не можа быць высветлены праз перасоўванні ў прасторы і змяненні ў вуглах агляду, што з'яўляецца крыніцай мноства памылак. Разам з тым, перавага неадназначнасці маштабу ва ўніверсальнасці: алгарытм будзе працаваць з аднолькавымі вынікамі як у маленькіх закрытых памяшканнях, так і на вялікіх адкрытых прасторах.

### **Стэрэа RGB камера**

Для стэрэа камеры ўласцівая наяўнасць двух ці больш аб'ектываў, кожны з якіх стварае кадры незалежна ад іншых. 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 кропак і геаметрыю размясціць кропкі ў трохмернай прасторы.

Трэба дадаць, што стэрэа камера заўжды наладжаная на працу з пэўным спектрам адлегласцяў: камера, якая працуе добра ў маленькіх закрытых прасторах не дасць такіх жа вынікаў, калі прастора будзе змаштабаваная ў дзясяткі разоў. Такім чынам, стэрэа камеры не ўласцівая такая ж універсальнасць у сферах прымянення, як для манакулярнай.

### **RGB-D камера**

RGB-D камера, як і стэрэа, з'яўляецца шматканальнай: разам з RGB каналам, які вяртае звыклыя нашаму воку каляровыя выявы, існуе дадатковы канал, які для кожнага здымка вяртае гэтак званую мапу глыбіняў. Прыклад вяртаемых дадзеных - на малюнку 1.

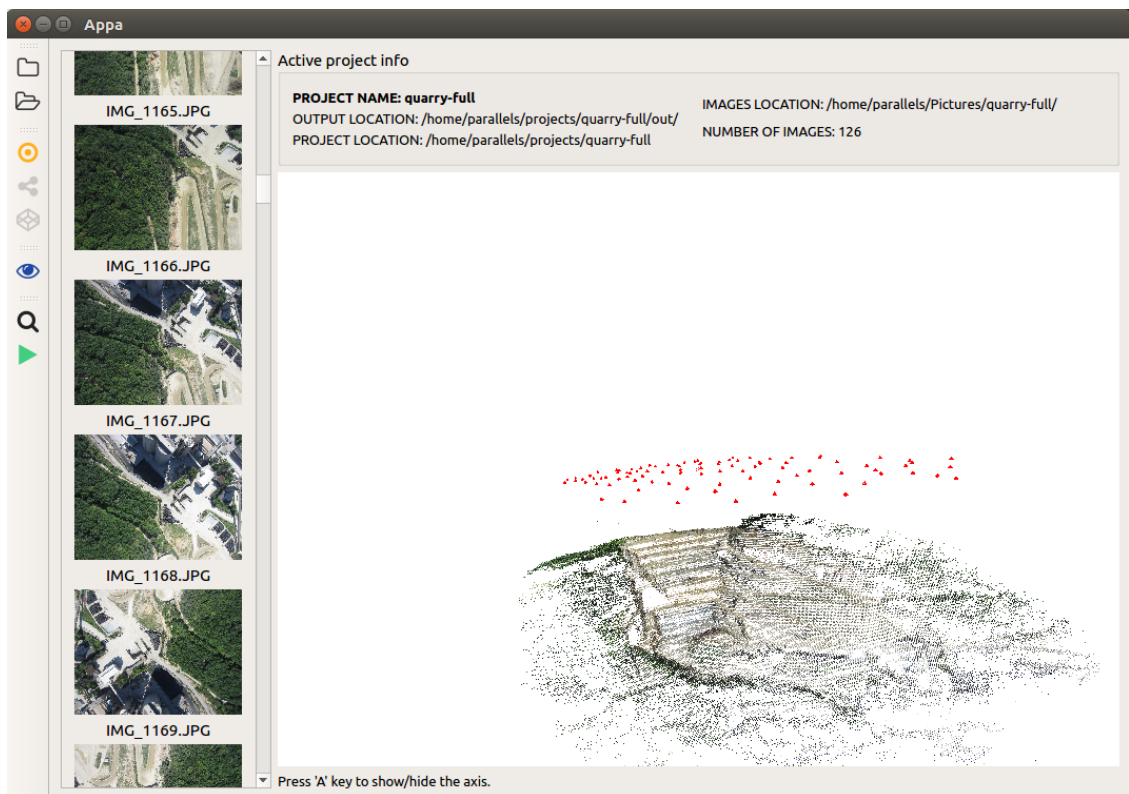
На практыцы RGB-D камеры не заўсёды працуюць так добра, як нам хацелася б: справа ў абмежаванасці спектра, у якім фармуюцца значэнні глыбіняў. Мапа глыбіняў, атрыманая апаратным спосабам, часцей за ўсё выкарыстоўваецца для папярэдняй ініцыялізацыі і для паслядоўнай аптымізацыі алгарытмічнымі спосабамі.

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



Малюнак 1: здымкі з RGB-D камеры

## 1.4 Рэканструкцыя паверхні афлайн



Малюнак 2: праграма рэканструкцыі паверхняў Арпа

Праект афлайн пабудовы трохмернай паверхні па здымках з манакулярнай камеры можа ўключаць ў сябе наступныя этапы:

1. Пошук і апісанне асаблівых кропак на выяве. Дасягаецца праз прымяненне такіх алгарытмаў і тыпаў дэскрыптарнаў, як SIFT [1], SURF [2], ORB [3] і іншых.



2. Пошук адпаведнасцяў паміж двухмернымі кропкамі з мэтай знайсці кропкі, якія адпавядаюць адным і тым жа кропкам трохмернай прасторы (працэс *matching*).
3. Прымяненне эпіпалярнай геаметрыі для пошуку пазіцыяў трохмерных кропак і пазіцыяў камераў у прасторы. Унутраныя параметры камеры могуць як быць вядомымі і замацаванымі, так і ўдзельнічаць у наступнай аптымізацыі.
4. Запуск алгарытмаў пучковай аптымізацыі (*Bundle Adjustment*) для ўдасканалення пазіцый кропак, пазіцый і паваротаў камер у прасторы, унутраных параметраў камер (калі не вядомыя).
5. Ушчыльненне, нанясення тэкстураў ды іншыя пераўтварэнні для перахода ад разрэджанага воблака кропак да шчыльных трохмерных мадэляў.

Прыкладам бібліятэкі для пабудовы трохмернай мадэлі па такой схеме можа выступіць TheiaSfm ([4]), праграмным сродкам для візуалізацыі працэса - распрацаваная мной раней у рамках курсавой працы праграма *Arpa*, працэс працы якой можна назіраць на здымку 2.

# 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 гэтак далей) 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 патрэбнасць 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. Даследаванне задачы можна ўмоўна падзяліць на тры этапы (прапанаваныя ў [5], [6]). 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 аналогіі 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, напрыклад, 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 адзін суцэльны алгарытм, але як канцэпт. 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 і павароты камер 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 лакалізацыю 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.

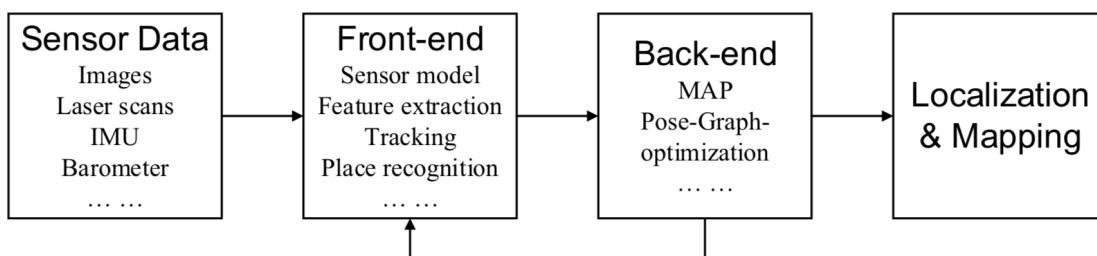
Такім чынам, лакалізацыя і пошук на мапе не выступаюць як дзве асобныя і незалежныя часткі алгарытма, але дапаўняюць і паляпшаюць адно аднаго. Пошук сябе на папярэдне пабудаванай мапе (напрыклад, пры паўторным пралёце той жа мясцовасці) дазваляе лакалізаваць БПЛА з якасцю параўнальнай з якасцю пабудаванай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

## 2.2 Падыходы да рэалізацыі

Як зазначаецца ў [6], архітэктара любой SLAM-сістэмы можа быць прадстаўленая схемай як на малюнку 3. 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*, [7]). Лакалізацыйная плынь PTAM займаецца пошукам адпаведнасцяў у дадзеных і запуская аптымізацыю па рухах камеры. У гэты ж час плынь 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 3: архітэктара SLAM-алгарытмаў

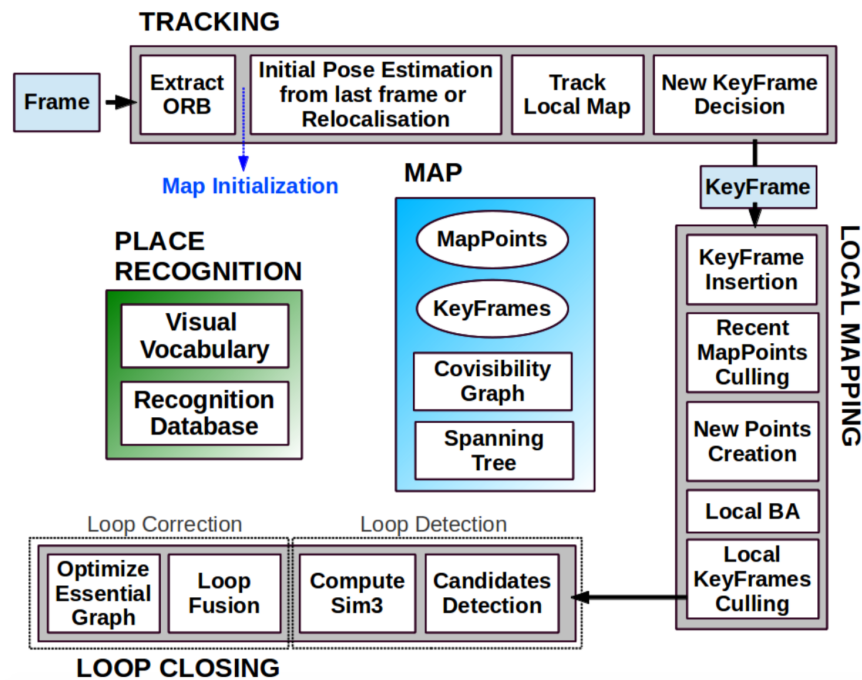
## 2.3 Агляд SLAM-рэалізацыяў

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 ([8], [9]) - манакулярная SLAM-сістэма, заснаваная на пошуку асаблівасцяў на выявах (*feature-based*), спраектаваная для працы

ў маленькіх і вялікіх, замкнёных і адкрытых прасторах.



Малюнак 4: ORB-SLAM

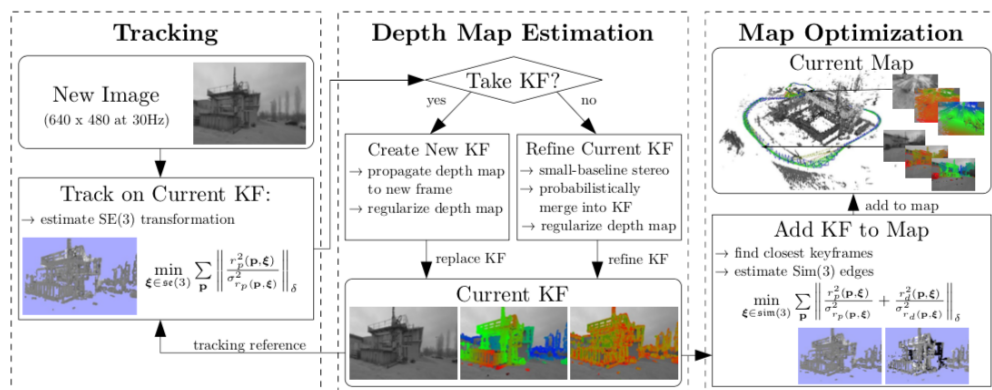
Асноўныя асаблівасці:

- 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 *графу бачнасці* (*covisibility graph*);
- новы аўтаматычны і надзейны спосаб ініцыялізацыі, які стварае пачатковы мапу 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 4.

Арыгінальная публікацыя [8] прыводзіла апісанне манакулярнай сістэмы, у другой рэдакцыі алгарытма (ORB-SLAM2, [9]) 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 ([10]) выкарыстоўвае *простыя* (*direct*) падыходы да працы з выявамі, то бок працуе непасрэдна са значэннямі ў пікселях.



Малюнак 5: LSD-SLAM

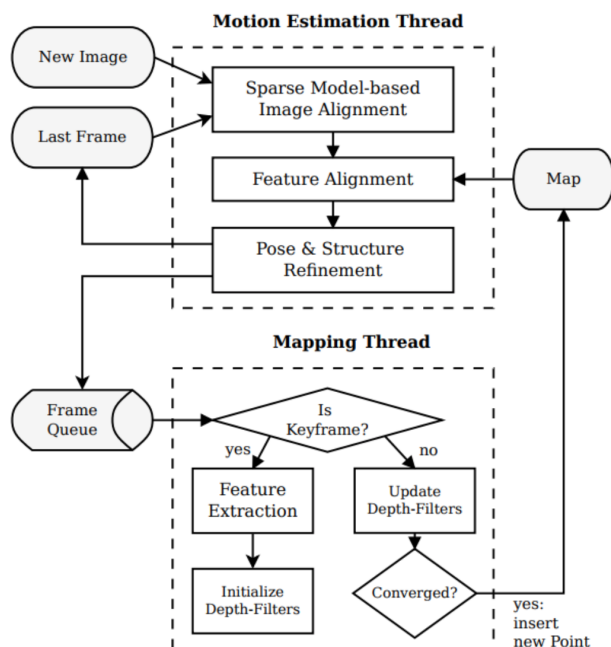
- працуе ў рэальным часе на CPU;
- складаецца з трох асноўных кампанентаў: трэкінг, падлік набліжэння для мапы глыбіняў і аптымізацыя мапы;
- трэкінг кампанента паслядоўна апрацоўвае новыя здымкі, прымаючы да ўвагі папярэдні і бягучы ключавы здымкі;
- другая кампанента выкарыстоўвае здымкі каб альбо 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 напаўшчыльную трохмерную рэканструкцыю паверхні;
- аўтары ініцыялізуюць першы ключавы кадр *выпадковай* мапай глыбіняў з вялікімі перападамі. Сцвярджаецца, што пры значных

рухах камери 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 5.

### 2.3.2 SVO

У адрозненні ад папярэдніх двух прыкладаў, SVO (*Semi-Direct Monocular Visual Odometry*, [11]) спалучае ў сабе як пошук і апісанне ключавых кропак, так і простыя метады.



Малюнак 6: 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 здзяйсняецца няўна як вынік прымянення простых метадаў (а не matching-а, напрыклад);
- выманне ключавых кропак здзяйсняецца 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 мінімума выбрасаў у вымярэннях;
- архітэктурная сістэма прадстаўленая на малюнку 6.



## ГЛАВА 3.

# SLAM НА ПРАКТЫЦЫ

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 [9].

У супольнасці робататэхнікаў папулярнасцю карыстаецца ROS - аперацыйная сістэма для распрацоўкі прыкладанняў для робатаў.

### 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 ([12]). Патрэбнасць у фрэймворку ўзнікла праз той факт, што стварэнне праграмага забеспячэння для робатаў - вельмі складаны і шматэтапны працэс; уніфікацыя спосабу камунікацый 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 рэалізаваныя менавіта пад ROS і могуць не мець уласных сродкаў візуалізацыі альбо ўводу дадзеных.

## 3.2 Даследванне збежнасці ORB-SLAM

У працэсе знаёмства з разнастайнымі SLAM-рэалізацыямі я правёў даследванне збежнасці пазіцый камер у прасторы на прыкладзе ORB-SLAM. Сутнасць эксперыментаў заключалася ў наступным:

- Запусціць алгарытм (у нашым выпадку - ORB-SLAM) на наборы дадзеных. У прыведзеных ніжэй выніках я выкарыстоўваў датасэт *freiburg1\_desk* апублікаваны ў [13].
- Пасля кожнага наступнага дададзенага здымка праверыць, ці была дададзеная новая камера. У выпадку, калі была - выгрузіць вонкавыя параметры ўсіх камераў, захаваць.
- Пасля таго, як усе дадзеныя прайшлі праз алгарытм, мы атрымалі мноства пазіцыяў камер, зарэгістраваных у розныя моманты часу. Пачынаем працу з гэтым дадзенымі.
- Для кожнай камеры знаходзім мноства яе пазіцый і сартыруем іх па часе. Пазіцыю, атрыманую ў канцы выканання алгарытма, прымаем за эталонную.
- Для кожнай камеры вылічваем яе “блізкасць” да эталоннай пазіцыі асобна па пазіцыі ў прасторы і па павароце. Адлегласць паміж пазіцыямі ў прасторы вылічваецца пры дапамозе Эўклідавай адлегласці (4), адлегласць паміж паваротамі, кожны з якіх задаецца кватэрніонам, вылічваецца па формуле (5). Адлегласць паміж паваротамі - значэнне памеру вугла, паварот на які прывядзе адзін кватэрніон у іншы.

$$d(a, b) = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2 + (a_z - b_z)^2} \quad (4)$$

$$d(q_1, q_2) = \arccos(2(q_1, q_2)^2 - 1) \quad (5)$$

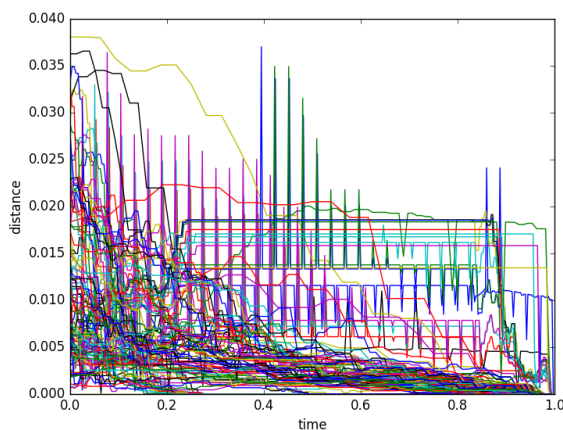
$$\text{дзе } a_{q_i}^2 + b_{q_i}^2 + c_{q_i}^2 + d_{q_i}^2 = 1$$

- Далей праводзіцца аналіз дадзеных, атрыманых на папярэдніх кроках. Графічна дадзеныя прадстаўленыя на малюнках 7 - 10. На кожным з графікаў 7 - 10 па восі абсцысаў адкладзены час: 0.0 - момант, калі камера ўпершыню была размешчаная, 1.0 - момант, калі камера была размешчаная на сцэне канчаткова і больш не ўдасканалася. Кожны графік (адзін колер) на малюнках 7, 9 адпавядае адной камеры. Малюнкi 8, 10 - гістаграмы, пабудаваныя па дадзеных для пазіцыяў і паваротаў камераў у прасторы

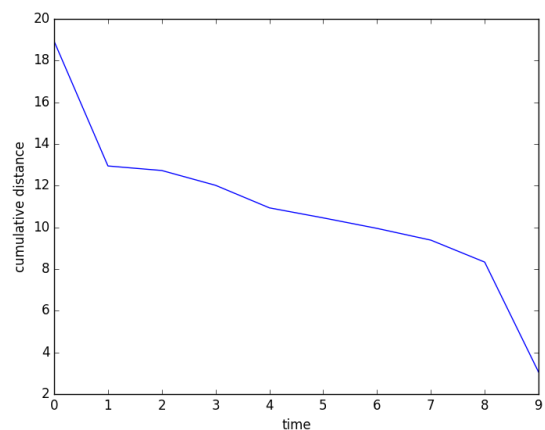
адпаведна і з'яўляюцца абагульненнем малюнкаў 7 і 9. Натуральна, што ўсе графікі сустракаюцца ў нулі ў момант часу 1.0, бо ў гэты момант былі сфармаваныя эталонныя пазіцыі і павароты.

- Негледзячы на тое, што некаторыя камеры паводзяць сябе дастаткова дзіўным чынам і напрацягу доўгага часу пераходзяць паміж двума станамі, агульная тэндэнцыя да збежнасці пазіцыяў і паваротаў камер да сваіх фінальных становішчаў добра праглядаецца.

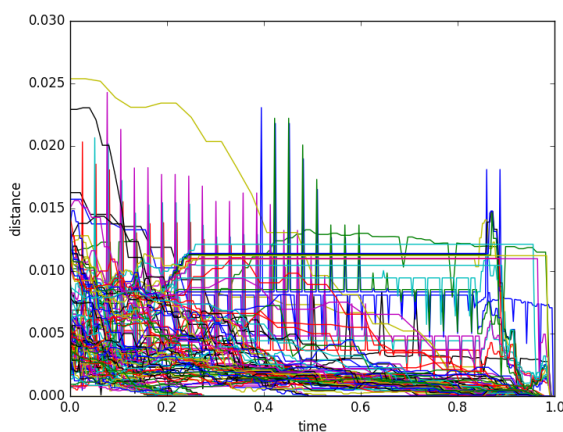
Правядзенне эксперыменту запатрабала рэалізацыю дадатковых метадаў у ORB-SLAM, а таксама распрацоўку дапаможных утылітаў пад ROS.



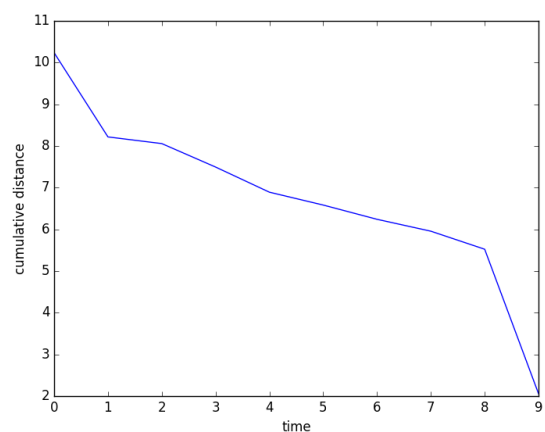
Мал. 7: збежнасць пазіцый камер



Мал. 8: адпаведная гістаграма



Мал. 9: збежнасць паваротаў камер



Мал. 10: адпаведная гістаграма

Агулам, у межах працы над курсавым праектаў я напісаў некалькі ўтылітаў для ROS, якія ажыццяўлялі камунікацыі паміж ROS-кампанентамі, такія як перадача выяваў, часавых характарыстык, геаметрычныя параметры камераў, стрымінг дадзеных у анлайн-рэжыме.

# ВЫСНОВЫ

У працы былі разгледжаныя 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 працы таксама прыводзіцца апісанне фрэймворка ROS, які выкарыстоўваецца для запуску большасці SLAM рэалізацыяў, а таксама прыведзеныя вынікі правядзення эксперыментаў па даследванні збежнасці ORB-SLAM.

# СПИС КРЫНИЦАЎ

- [1] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.
- [2] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008.
- [3] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Proceedings of the 2011 International Conference on Computer Vision, ICCV '11*, pages 2564–2571, Washington, DC, USA, 2011. IEEE Computer Society.
- [4] Chris Sweeney. Theia multiview geometry library: Tutorial & reference. <http://theia-sfm.org>.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *European Conference on Computer Vision (ECCV)*, September 2014.

- [11] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
- [12] 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.
- [13] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.