

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій  
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)  
Розроблення 3D моделювання робочого середовища мобільного робота  
засобами VSLAM  
(тема)

Виконав:

здобувач 2 року навчання,  
групи КТРСм-23-2

Поддубняк І.А.

(прізвище, ініціали)

Спеціальність 174 Автоматизація,  
комп'ютерно-інтегровані технології та  
робототехніка

(код і повна назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютеризовані та  
робототехнічні системи

(повна назва освітньої програми)

Керівник д.т.н. Цимбал О.М.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

\_\_\_\_\_

(підпис)

Невлюдов І.Ш.

(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та  
робототехніки

Рівень вищої освіти другий (магістерський)

Спеціальність 174 Автоматизація, комп'ютерно-інтегровані технології та  
робототехніка

(код і повна назва)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютеризовані та робототехнічні системи

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри

\_\_\_\_\_ (підпис)

«\_\_\_\_\_» \_\_\_\_\_ 2024 р.

### ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Поддубняку Іллі Андрійовичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення 3D моделювання робочого середовища  
мобільного робота засобами VSLAM

затверджена наказом університету від 03 листопада 2024 р. № 1287 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 23  
січня 2025 р.

3. Вихідні дані до роботи виконання SLAM через візуальні дані; інтеграція з  
симульованими та реальними програмними та апаратними середовищами  
робототехніки; споживання до 8 ГБ ОЗП; можливість діагностування та  
модифікації системи; ОС GNU/Linux дистрибутиву Ubuntu 22.04.4; версія  
ROS2 Humble; бібліотека VSLAM ORB-SLAM3

4. Перелік питань, що потрібно опрацювати в роботі Вступ; Аналіз  
предметної області; Аналіз технічного завдання; Розробка системи проекту;  
Аналіз функціоналу проекту; Розрахункові дослідження системи; Охорона  
здоров'я

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Демонстраційний матеріал представлений у форматі презентації PowerPoint (\*.ppt) – 14 с. формату А4


6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1 )

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Складання технічного завдання	01.09.24 – 05.09.24	виконано
2	Аналіз предметної області	05.09.24 – 15.09.24	виконано
3	Аналіз експериментів	15.09.24 – 28.09.24	виконано
4	Підготовка програмного середовища	28.09.24 – 28.10.24	виконано
5	Розробка поєднаної системи	28.10.24 – 28.11.24	виконано
6	Налаштування системи в симуляторі	28.11.24 – 01.12.24	виконано
7	Функціональні доповнення системи	01.12.24 – 25.12.24	виконано
8	Складання розділу розрахунків	25.12.24 – 05.01.25	виконано
9	Подання роботи на перевірку Інтернет-сервісом Unichек	21.01.25	виконано
10	Оформлення пояснювальної записки	22.01.25	виконано
11	Подання роботи на рецензію	22.01.2025	виконано
12	Подання роботи на підпис зав. кафедри	22.01.2025	виконано
13	Подання кваліфікаційної роботи в ЕК	23.01.2025	виконано

Дата видачі завдання 1 вересня 2024 р.

Здобувач  \_\_\_\_\_  
(підпис)

Поддубняк І.А. \_\_\_\_\_

Керівник роботи \_\_\_\_\_  
(підпис)

д.т.н. Цимбал О.М. \_\_\_\_\_  
(посада, прізвище, ініціали)

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

23.01.2025



Поддубняк Ілля Андрійович

## РЕФЕРАТ

Пояснювальна записка: 82 с., 5 табл., 67 рис., 6 дод., 40 джерел.

РОБОТОТЕХНІКА, КОМП'ЮТЕРНИЙ ЗІР, КАРТОГРАФУВАННЯ,  
ЛОКАЛІЗАЦІЯ, SLAM, VSLAM.

Мета роботи – підвищення ефективності 3D моделювання робочого середовища мобільного робота за рахунок використання VSLAM.

Об'єкт розробки – технології комп'ютерного зору та SLAM.

Предмет розробки – інтегрована VSLAM система робота.

Для досягнення мети було проведено аналіз предметної області та аналогів імплементацій VSLAM систем в роботах. Проведений аналіз вибору операційної системи, бібліотек та програмних компонентів проекту. Обрано системно-програмну платформу, програмні компоненти, та розроблено VSLAM систему на їх основі. Розроблено схему взаємодій компонентів програми. Розроблено 3D моделювання робота та середовища операцій робота у симуляторі. Проведено дослідження щодо функціоналу VSLAM системи. Проведено розрахунки ефективності та математичної моделі VSLAM системи.

Результати можна віднести до Цілі сталого розвитку 9, п. 9.3 “Забезпечити доступність дорожньотранспортної інфраструктури, яка базується на використанні інноваційних технологій, зокрема через розширення форм участі держави у різних інфраструктурних проектах” та 9.4 “Сприяти прискореному розвитку високо- та середньовисокотехнологічних секторів переробної промисловості, які формуються на основі використання ланцюгів «освіта – наука – виробництво» та кластерного підходу за напрямками: розвиток інноваційної екосистеми”.

## **ABSTRACT**

Explanatory note: 82 p., 5 tabl., 67 pic., 6 applications, 40 sources.

**ROBOTICS, COMPUTER VISION, MAPPING, LOCALIZATION, SLAM, VSLAM.**

The purpose of the work is to increase the efficiency of 3D modeling of the mobile robot's working environment through the use of VSLAM.

The object of development is computer vision and SLAM technologies.

The subject of development is an integrated VSLAM robot system.

In order to achieve the goal, an analysis of the subject area and analogues of VSLAM system implementations in robots was conducted. An analysis of the choice of the operating system, libraries and software components of the project was made. A system-software platform and software components were selected, and a VSLAM system was developed based on them. Schematics of program component interactions were made. A 3D modeling of the robot and the robot's operation environment in the simulator was developed. A study was conducted on the functionality of the resulting VSLAM system. Calculations of the efficiency and mathematical model of the VSLAM system were performed.

The results can be attributed to Sustainable Development Goal 9, paragraph 9.3 “Ensure the accessibility of road transport infrastructure based on the use of innovative technologies, in particular through the expansion of forms of state participation in various infrastructure projects” and 9.4 “Promote the accelerated development of high- and medium-high-tech sectors of the processing industry, which are formed on the basis of the use of the “education - science - production” chains and a cluster approach in the areas: development of an innovative ecosystem”.

## ЗМІСТ

Перелік скорочень .....	7
Вступ.....	8
1 Аналіз предметної області.....	10
1.1 Аналіз технічного завдання.....	10
1.2 Аналіз сучасних універсальних 3D-симуляторів роботів.....	10
1.2.1 Аналіз експериментів з симуляціями.....	14
1.3 Аналіз сучасності використання комп'ютерного зору у симуляторах робототехнічних систем .....	16
1.4 Аналіз сучасних візуальних SLAM систем в робототехніці .....	18
1.5 Висновки до першого розділу.....	33
2 Розробка системної реалізації проекту .....	34
2.1 Обґрунтування прийнятих рішень системної реалізації розробки .....	34
2.2 Засоби та результати системної реалізації розробки.....	36
2.3 Висновки до другого розділу .....	47
3 Аналіз функціоналу проекту.....	48
3.1 Висновки до третього розділу.....	61
4 Розрахункові дослідження системи.....	62
4.1 Базові показники ефективності.....	62
4.2 Комп'ютерне моделювання системи автоматичного управління .....	67
4.3 Огляд та розрахунки необхідних умов для приміщення для підготовки та використання системи .....	73
4.4 Висновки до четвертого розділу.....	75
Висновки .....	76
Перелік джерел посилання .....	78
Додаток А Код URDF файлу робота .....	83
Додаток Б Код файлу YAML для уточнення параметрів монокулярної камери для VSLAM.....	91

	6
Додаток В Апробація результатів кваліфікаційної роботи.....	94
Додаток Г Висвітлення результатів кваліфікаційної роботи.....	109
Додаток Д Висвітлення результатів кваліфікаційної роботи .....	116
Додаток Е Презентація .....	130



**ПЕРЕЛІК СКОРОЧЕНЬ**

GPS – global positioning system;

IMU – inertial measurement unit;

LTS – long-term support;

OS – operating system;

RGB-D – red, green, blue and depth;

ROS – robot operating system;

SLAM – simultaneous localization and mapping;

VO – visual odometry;

VSLAM – visual simultaneous localization and mapping.

## ВСТУП

SLAM є критично важливою технологією для розвитку галузі мобільних роботів до високого рівня – вона дозволяє їм та їх операторам ефективно орієнтуватися по мапі невідомих середовищ, що побудована по даних з сенсорів, датчиків, камер. Одним з підходів до SLAM є VSLAM, де використовується виключно візуальна інформація.

VSLAM використовує дані тільки з камер, що встановлені на роботі, для одночасного визначення свого місцезнаходження та побудови карти навколишнього середовища. Ця технологія має ряд переваг, таких як низька вартість, відсутність необхідності в додаткових сенсорах, як GPS або лідар, можливість сенсорів у одну спільну SLAM-систему та можливість роботи в приміщеннях. Однак VSLAM також має й технічні виклики, пов'язані з обробкою великих обсягів візуальних даних, врахуванням динамічних змін у середовищі та забезпеченням стабільної роботи в реальних умовах.

Розробка та ретельне тестування ефективних VSLAM систем є важливим завданням створення якісних роботів, здатних функціонувати в реальному світі. Моделювання та симуляція мають ключову роль у цьому процесі, дозволяючи дослідникам та розробникам оцінювати та вдосконалювати алгоритми VSLAM в безпечному віртуальному середовищі, де відсутні ризики пошкодити реальну апаратну частину систем.

Ця кваліфікаційна робота спрямована на актуальні розробку та вдосконалення сучасних методів та інструментів VSLAM для проектів створення модельованих та реальних робототехнічних систем, що використовують візуальні картографію та локалізацію. Розроблена система інтеграбельна з симульованими та реальними програмними та апаратними середовищами та має здатність виконання SLAM через візуальні дані та надання даних до діагностики у реальному часі.

Можливі сфери застосування: будь-яка галузь роботехніки, що

пов'язана з невідомими чи динамічними середовищами застосування роботів.

Мета роботи – підвищення ефективності 3D моделювання робочого середовища мобільного робота за рахунок використання VSLAM.

Об'єкт розробки – технології комп'ютерного зору та SLAM.

Предмет розробки – інтегрована VSLAM система робота.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз сучасних універсальних 3D-симуляторів роботів
- провести аналіз сучасності використання комп'ютерного зору у симуляторах робототехнічних систем;
- провести аналіз сучасних візуальних SLAM систем в робототехніці та аналогів їх імплементацій;
- провести вибір та розробку програмних компонентів системи;
- розробити команди та схеми підключень та взаємозв'язків системи;
- розробити моделювання тестового робота та його середовища для симуляцій;
- розробити імплементацію керування роботом та втілення VSLAM системи;
- дослідити функціонал системи;
- розрахувати ефективність та комп'ютерне моделювання системи.

Робота виконана згідно [1–6] . Результати дослідження надані та опубліковані у [7-9].

Апробація та висвітлення результатів кваліфікаційної роботи представлено у додатках В, Г, Д.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Аналіз технічного завдання

Мета роботи – підвищення ефективності 3D моделювання робочого середовища мобільного робота за рахунок використання VSLAM.

Об’єкт розробки – технології комп’ютерного зору та SLAM.

Предмет розробки – інтегрована VSLAM система робота.

Актуальність розробки зумовлена тим, що VSLAM є критично важливою технологією для мобільних роботів, бо вона дозволяє їм та їх операторам ефективно орієнтуватися по мапі невідомих середовищ, що побудована по зображень камер.

Ця технологія має ряд переваг, таких як низька вартість, відсутність необхідності в додаткових сенсорах по типу GPS або лідар, можливість інтеграції камер та інших сенсорів у одну спільну SLAM-систему, та можливість роботи в приміщеннях.

Розробка спрямована на прототипування та вдосконалення такої системи VSLAM, що можна впровадити у робототехнічні проекти.

Цільова система повинна мати достатні функціонал та простоту використання та модифікацій для таких робототехнічних розробок, що використовують технологію VSLAM.

## 1.2 Аналіз сучасних універсальних 3D-симуляторів роботів

З точки зору симуляції апаратної частини роботів, основними підходами є геометричне, фізичне та кінематичне моделювання. Ці методи дозволяють створити точну віртуальну модель фізичної структури робота, яку можна налаштовувати та змінювати. Використовуючи 3D-модель та відповідні програмні інструменти, можна створити інтерактивну модель, яка точно

відображає цільовий пристрій і його можливості у певному середовищі, уникаючи ризиків і проблем, пов'язаних із тестуванням реальної апаратури. За допомогою геометричної моделі також можна кінематично описати рухи робота (за допомогою прямої чи зворотної кінематики) [10].

Щодо симуляції програмної частини, під час моделювання управління, поведінки та рухів використовуються ті самі алгоритми, які застосовуються для реального робота-аналога. Деякі симуляційні програми підтримують програмування за допомогою блоків коду, однак у високоякісних програмах головна увага приділяється традиційному кодуванню та машинному навчанню. Важливо зазначити, що управління симуляцією або її об'єктами може виконуватися в режимі користувача, автономно, або в гібридному режимі [11].

До додаткових параметрів симуляції можна віднести різні режими, такі як масова симуляція, паралельна симуляція для прискорення машинного навчання, або симуляція в безмоніторному режимі ("headless").

Також можна синхронізувати симуляцію з реальним роботом через технологію "цифрового близнюка" та моделювати впливи навколишнього середовища, такі як вітер, погода, тиск, сила тяжіння, інтерактивні властивості поверхонь або місцеві механічні елементи, які не є частиною роботів, що розробляються [12].

На базовому рівні можливості для симуляції апаратної та програмної частини є однаковими у всіх симуляційних середовищах, як видно з досліджень канадських учених [13]. Проте для розширених параметрів симуляцій та вдосконалених розробок існують значні відмінності між різними симуляційними програмами [14].

Через ці відмінності та широкий спектр потреб у робототехніці, розробникам слід заздалегідь ознайомитися з сучасними симуляційними середовищами, щоб полегшити та покращити цикл розробки проектів, над якими вони працюють.

У таблицях 1.1 та 1.2 представлені назви та характеристики сучасних

програмних середовищ для робототехнічних симуляцій, які використовуються для розробки та рекомендовані дослідниками з Шотландії та Китаю [15-16].

Таблиця 1.1 – Базові дані про властивості сучасних симуляційних середовищ

<b>Назва</b>	<b>Мова програмування</b>	<b>Операційна система</b>	<b>Підтримка безмоніторного (headless) функціонування</b>
Gazebo	C, C++, Python	Linux, macOS, Windows	Вбудована
Gazebo Ignition	C++, Python	Linux, macOS, Windows	Вбудована
WeBots	C, C++, Python, Java, MATLAB	Linux, macOS, Windows	Зовнішня
Isaac Sim	C++	Linux	Вбудована
Project Chrono	C++, Python	Linux, macOS, Windows	Вбудована
PyBullet	Python	Linux, macOS, Windows	Вбудована
CoppeliaSim (V-REP)	C, C++, Python, Java, Lua, MATLAB, Octave	Linux, macOS, Windows	Вбудована
MuJoCo	C	Linux, macOS, Windows	Вбудована
ARGoS	Lua, C++	Linux, macOS, Windows	Вбудована

Таблиця 1.2 Додаткові дані про властивості сучасних симуляційних середовищ

Назва	Фізичний рушій	Відкритий доступ	Підтримка ROS2	Підтримка машинного навчання
Gazebo	Bullet, DART, ODE, Simbody	Так	Так	Зовнішня
Ignition	DART	Так	Так	Зовнішня
WeBots	ODE	Так	Так	Зовнішня
Isaac Sim	PhysX	Ні	Так	Вбудована
Project Chrono	Chrono::Engine	Так	Так	Зовнішня
PyBullet	Bullet	Так	Так	Зовнішня
CoppeliaSim (V-REP)	Bullet, Newton, ODE, Vortex Dynamics	Ні	Так	Зовнішня
MuJoCo	MuJoCo	Так	Ні	Зовнішня
ARGoS	ODE, 3D particle engine, 2D-dynamics open-source physics engine library Chipmunk, 2D-kinematics engine	Так	Ні	Вбудована

Варто зазначити, що підтримка програмного пакету Robot Operating System 2 (ROS2) є особливо важливою для виконання симуляцій, які працюють паралельно з роботом в апаратній формі.

Найкращу сумісність із ROS2 та більшістю якісних симуляційних програм (зокрема Isaac Sim) забезпечують стабільні версії операційних систем Linux, такі як Ubuntu. Тому на етапі планування розробки симуляції слід заздалегідь врахувати вибір ОС і сумісність симуляційного ПЗ з нею.

При порівнянні WeBots, Gazebo, CoppeliaSim та ARGoS, під час вибору симуляційного ПЗ також враховують [17] такі фактори, як:

- доступність симуляції різних датчиків та сенсорів (з цим проблеми тільки в ARGoS);
- простота моделювання (особливо гарна в WeBots та CoppeliaSim);
- величина бібліотек готових моделей (гарна в WeBots та CoppeliaSim та невеличка тільки в ARGoS);
- якість технічної підтримки від розробників та дослідників (нижче гарної тільки в CoppeliaSim – через обмеження кількості користувачів комерційною моделлю дистрибуції).

### 1.2.1 Аналіз експериментів з симуляціями

В симуляторах коефіцієнт реального часу (real-time factor, RTF) в результатах дослідження розглядається як максимально можлива швидкість симуляції відносно реального часу.

Навантаження на сервер (CPU 2×AMD EPYC 7542, GPU 4×NVIDIA Quadro RTX 8000, RAM 512 GiB DDR4, накопичувач Samsung PM1733), мобільну робочу станцію (CPU Intel Core i7-8700, GPU NVIDIA GeForce RTX 2080 Mobile, RAM 32 GiB DDR4, накопичувач Samsung 960 EVO) та ноутбук (CPU Intel Core i7-7500U, GPU Intel HD Graphics 620, RAM 8 GiB DDR4, накопичувач Samsung 950 Pro) з тих симуляторів, які було проаналізовано дослідниками з Німеччини, вказані в таблиці 1.3 [18].



Таблиця 1.3 – Середні використання ресурсів симуляційного ПЗ та коефіцієнти реального часу з кроком часу 1 мс

ПЗ	Тип	CPU (С)	RTF (С)	CPU (МРБ)	RTF (МРБ)	CPU (Н)	RTF (Н)
Gazebo	робот	273,0%	4,3	264,4%	4,4	11,9%	2,3
	кулі	223,1%	1,1	208,7%	1,4	213,2%	0,8
MuJoCo	робот	116,3%	2,4	100,8%	2,8	103,1%	2,1
	кулі	113,4%	0,6	99,8%	0,8	102,8%	0,6
PyBullet	робот	119,3%	0,7	102,0%	0,8	104,0%	0,6
	кулі	118,0%	1,1	100,7%	1,3	103,8%	1,0
WeBots	робот	124,3%	1,8	105,6%	1,7	109,1%	1,3
	кулі	120,7%	1,2	103,8%	1,2	103,3%	0,5

Щодо якості роботи симуляцій та використання комп'ютерних ресурсів у порівнянні між Ignition, Isaac Sim, PyBullet, CoppeliaSim та WeBots, дослідники з Шотландії вивели такі критерії для оцінювання симуляційного ПЗ [19]:

- підтримка виконання довгострокових операцій з успішним виконанням завдань симуляції (К1);
- повторюваність моделювання симуляції за тієї самої сцени та обмежень завдання (К2);
- придатність для досліджень з машинного навчання з точки зору використання ресурсів і часу простою (К3).

WeBots підходить для довготривалих операцій і успішно виконує поставлені завдання (К1), а також здатен відтворювати однакові умови моделювання під час повторних спроб (К2). Однак WeBots придатний для машинного навчання лише тоді, коли час виконання та використання ресурсів не є критичними вимогами під час навчання моделей (К3).

Ignition, хоча його порівнюють з WeBots, краще відповідає вимогам К1

та K3. Вимога K2 задовольняється за умов повільного й стабільного виконання завдань.

CoppeliaSim і PyBullet мають менший вплив на ресурси, тому вони є найбільш придатними для K3.

Isaac Sim відповідає лише K1, оскільки моделювання в ньому важко повторити при повторних спробах.

Важливо зазначити, що для виконання складних робототехнічних проєктів іноді необхідно використовувати кілька симуляційних середовищ. Наприклад, нідерландські дослідники під час розробки машинного керування для роботизованої руки використовують одночасно Gazebo (завдяки гарній сумісності з ROS) та PyBullet (через найкращу інтеграцію зі стачами машинного навчання) [20].

### 1.3 Аналіз сучасності використання комп'ютерного зору у симуляторах робототехнічних систем

Згідно з результатами аналізів публікацій з галузі [21], дослідники та розробники мають успіх у використанні симуляцій для впровадження таких дій роботів, що відповідають візуальним даним про середовища навколо роботів. До таких дій відносяться: відстеження цілей, виявлення цілей, пошук шляхів, прогнозування майбутніх станів за необробленими сенсорними показаннями, маніпуляції об'єктами, автономні приземлення, контроль групами роботів у ієрархії лідер-послідовник, оцінка погляду, оцінка руху, точна взаємодія з руками людини тощо [22].

Зазвичай, такі моделювання мають спрощений тип рендерінгу об'єктів, але, завдяки особливостям ходу обробки зображень в програмній частині комп'ютерного зору, результати співпадали з наявними аналогічними експериментами у реальних умовах; як зазначено публікаціями з галузі [23-24], це пов'язано з тим, що для більшості цілей розробок потрібний об'єктивно оцінений реалізм даних (проекції тривимірних об'єктів у двовимірному

просторі, контури об'єктів, ознаки об'єктів тощо), що не обов'язково повністю перетинається з фотореалізмом, який оцінюється більш суб'єктивно. Це особливо варто пам'ятати через те, що фотореалістичний рендерінг вимагає більш ресурсозатратної комп'ютерної апаратури. Для подальшого аналізу необхідно окреслити різновиди камер, що використовуються у робототехніці – вони приведені в таблиці 1.4.

Таблиця 1.4 – Типові види камер, що використовуються в робототехніці для комп'ютерного зору

<b>Пристрій</b>	<b>Тип знімання</b>	<b>Приклади застосування</b>
Камера RGB	Зображення з широким діапазоном кольорів за низької вартості	Виявлення об'єктів
Камера дальнісного зображення	Поєднання RGB зображень з даними відстані з далекоміру	Система керування з точною взаємодією з людиною
Стереокамера	Імітація двоокого зору людини кількома лінзами	Навігація роботів зі спрощеним розпізнаванням об'єктів
Ендоскопічні, мікроскопічні камери	Роздільні зображення у важкодоступних місцях	Керування мікророботами та роботами з м'якими компонентами
Монокулярна камера	Цілісні зображення з широким полем зору за низької вартості та маленької ваги	Відстеження цілі, швидкий аналіз місцевості «сферичними зображеннями»
Інфрачервона (IR) камера	Знімання зображень за IR випромінюванням	Перевірка на перегрівання, нічний зір, відстеження очей по IR відблиску

#### 1.4 Аналіз сучасних візуальних SLAM систем в робототехніці

Щоб зрозуміти VSLAM, важливо спочатку окреслити його характеристики по відношенню до загальних концепцій SLAM [25].

Щоб мати локалізацію та картографування достатньо практичної точності, процеси SLAM мають зберігати старі стани та одночасно адаптуватися до їх нових змін.

Для оптимізації використання комп'ютерних ресурсів, географічний розмір мапи має залежати лише від конкретних цілей щодо дослідження території. Технологія SLAM підходить до більшості мехатронних галузей, зокрема повітряні та підводні мобільні роботи, автономні транспортні засоби, дрони, та може бути визначений як така, що складається з двох основних частин [26]:

- побудова карти невідомого зовнішнього чи внутрішнього середовища;
- одночасне відстеження положення або переміщення датчиків і камери (що встановлені на мобільній системі) через різні позиції та різний час у цьому середовищі.

Для отримання даних, у SLAM використовуються датчики оцінки власного переміщення з колісних, візуальних, інерційних, лазерних або комбінацій різних джерел (одометрії), датчики зору, лазери (як 2D, так й 3D), радар, сонар, гіроскопи, акселерометри, та GPS.

Поняття датчиків зору описує будь-які приймачі першоджерельних візуальних даних: монокулярні камери, стереокамери, камери на основі подій, інфрачервоні камери, всенаправлені камери, триколіорові камери глибини (RGB-D) тощо; робот оцінює свої положення й орієнтацію щодо власного оточення завдяки візуальним даним, що надаються з датчиків зору.

У сучасності, у SLAM входить декілька груп напрямків розпізнавання рис середовища [27]:

- візуальні ознаки: ознаки, отримані із зображень камери, наприклад ознаки згорткової нейронної мережі (CNN) із глибоких нейронних мереж, ознаки, що створені «вручну» (програмно, без нейронних мереж) ознаки, як от SIFT (масштабно-інваріантне перетворення ознак), SURF (прискорені надійні ознаки), HOG (гістограма орієнтованих градієнтів);
- ознаки LiDAR/лазера: ознаки, що отримані з 3D-хмари точок або 2D-лазерного сканування, такі як дескриптори точок і гістограми;
- семантичні ознаки: ознаки, що представляють семантичні об'єкти або класи, сегментовані або виявлені на зображеннях чи хмарах точок, наприклад стовпи, дерева, будівлі;
- об'єднання кількох датчиків: поєднання ознак з різних датчиків, таких як камери, LiDAR, радар, для використання сильних сторін кожного з них (наприклад, поєднання камер та датчиків дальності для виявлення візуальних і глибинних ознак місцевості);
- стабільність ознак: означає, наскільки добре отримані візуальні або інші ознаки місцевості можна надійно виявляти та зіставляти протягом тривалого періоду часу в мінливих середовищах.

Загальний алгоритм послідовності техніки для будь-якого SLAM – це збір даних з датчиків, вилучення ознак із зображень, зіставлення ознак, оцінка пози, замикання циклів та створення мапи – детальніше це зображено на рисунку 1.1 [9].

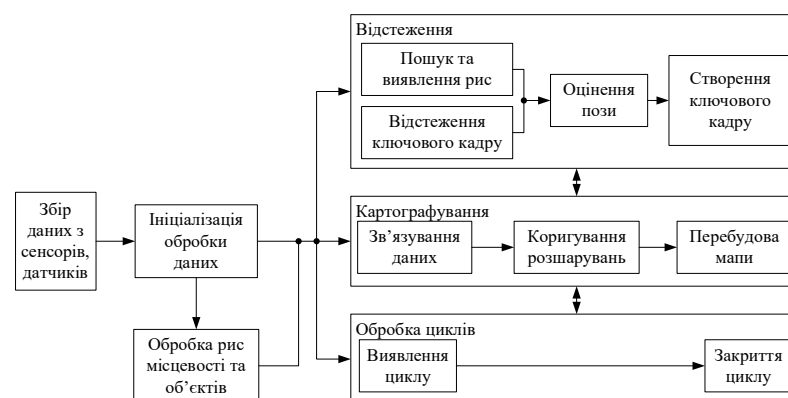


Рисунок 1.1 – Принципова схема алгоритму SLAM

Через те, що здатність виявляти реальні навколишні простір та середовище є одним з ключових елементів у системах SLAM, автономні роботи зі SLAM повинні мати відповідні сенсори для отримання та обробки найкращої інформації та сигналів з навколишнього середовища, щоб створити карту простору та виконувати інші споріднені зі SLAM функції, як зображено на рисунку 1.2.

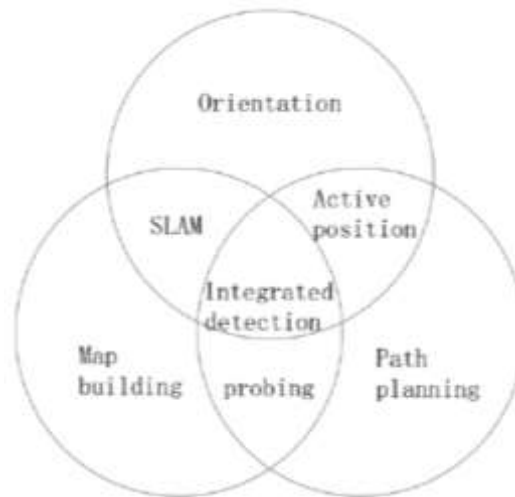


Рисунок 1.2 – Діаграма Венна для кібернетичних методів та функцій, що сумісні зі SLAM

VSLAM також має меншу похибку локалізації позиції робота у порівнянні з одометрією. Графік порівняння зображено на рисунку 1.3.

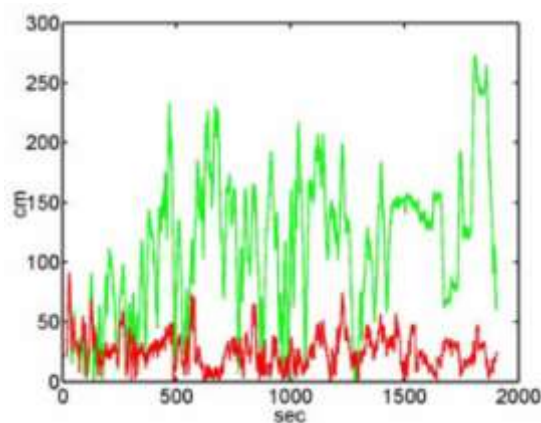


Рисунок 1.3 – VSLAM (червоним) та одометрія (зеленим)

Сенсори, що типово використовуються у SLAM та можуть бути використані у мультисенсорній SLAM системі, наведено у таблиці 1.5.

Таблиця 1.5 – Сенсори SLAM

Сенсор	Застосування	Діапазон	Тип SLAM	Споживання (Вт)
Сонар	Підводний	1 кГц та 500 кГц	Bat SLAM	0,01–5
Лазер	Внутрішній, зовнішній та дрони	1–100 Гц	Laser SLAM, TinySLAM	1–100
LiDAR	Внутрішній, зовнішній та дрони	1–50 Гц	LiDAR SLAM, CT-SLAM	5–200
RGB-D	Внутрішній	10–400 Гц	RGB-D VSLAM	0,3–5
Моно-камера	Внутрішній та зовнішній	20–200 Гц	Mono VSLAM	0,01–10
Стерео	Внутрішній, зовнішній та БПЛА	1–400 Гц	Stereo LSD-VSLAM	2–15

Датчики Sound Navigation and Ranging (SONAR) виявляють об'єкти за допомогою еха ультразвукового сигналу, що відбивається від них, і широко використовуються в мобільних роботах. Робота датчиків SONAR базується на звукових сигналах, що дозволяє їм функціонувати в темряві, що робить їх популярними для підводних роботів, де використання світлових зображень стикається з труднощами.

Системи SONAR може охоплювати великий простір для картографування та навігації, але вони не підходять для цілей виявлення об'єктів, оскільки вони не можуть точно визначити кути об'єктів. Базовий модуль сонару зображено на

рисунку 1.4.

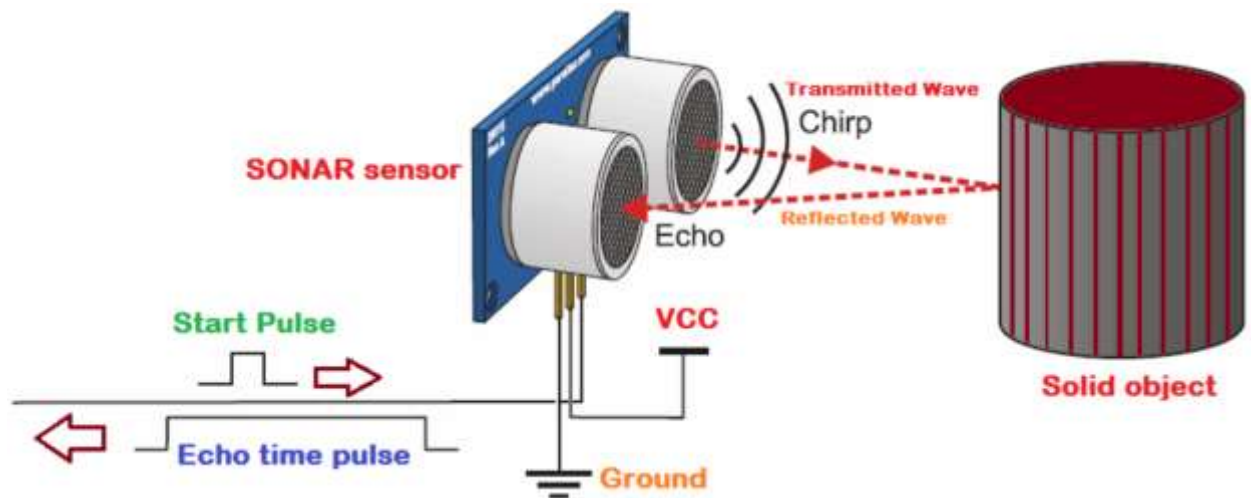


Рисунок 1.4 – Принцип роботи сонару на прикладі простого модулю ультразвукового датчика, зпареного з відповідним випромінювачем

Для сонару, SLAM зручніше виконувати скупченням кількох датчиків у двох площинах (інформація глибини від затримки відбиття дає дані для побудовання хмар точок у трьох площинах).

SLAM, засноване на базі 2D лазерного сканеру є поширено використаною моделлю. Цей підходи зазвичай застосовуються в ситуаціях, коли рух робота відносно обмежений по двом осям, наприклад, для колісного робота, що рухається по площині, або літаючого робота, що летить на постійній висоті над землею. Завдяки лазерним сенсорам роботи можуть створювати карти як внутрішніх, так і зовнішніх просторів. Лазерні сенсори з частотою оновлення 20 Гц часто використовуються для внутрішніх SLAM робіт.

Хоча лазерні сенсори не є оптимальним вибором для технік виявлення об'єктів, вони можуть бути використані для 2D та 3D картографування з високошвидкісною обробкою, відомою як *tiny SLAM*. Це рішення легше застосовувати в сценаріях, де висота або альтитуда є постійними. Принцип роботи двовимірною лазерного сканеру зображено на рисунку 1.5.



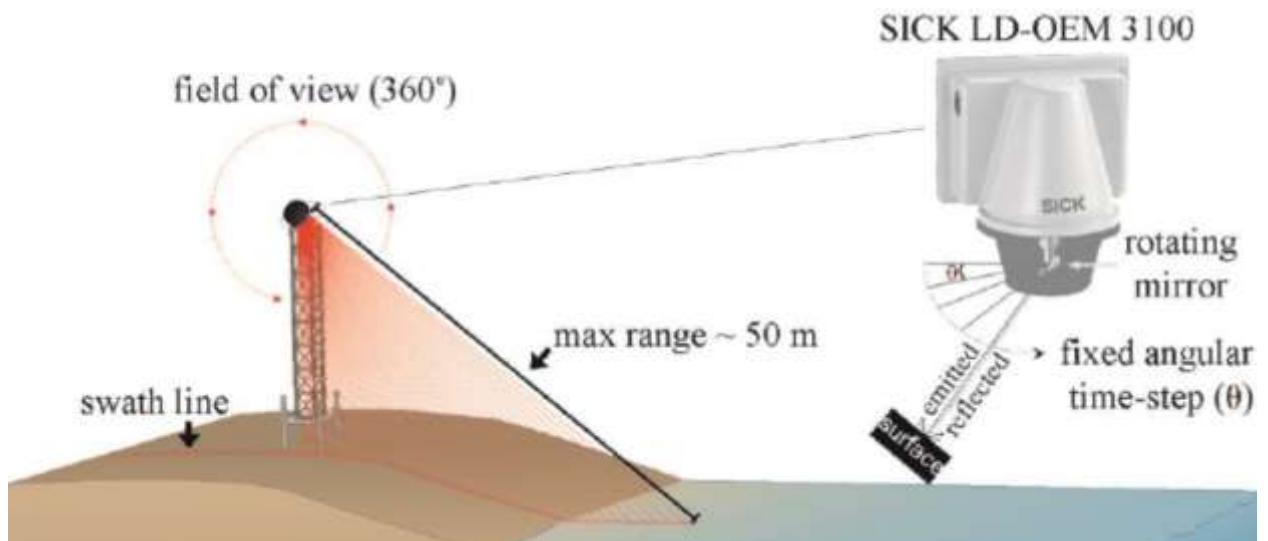


Рисунок 1.5 – Принцип роботи 2D лазерного сканеру

Якщо висота змінюється, це додає рівень складності при використанні 2D лазера. У таких випадках необхідно ідентифікувати зміни за допомогою іншого сенсора та об'єднати цю інформацію для компенсації проблем картографування.

SLAM, засноване на LiDAR, є однією з сучасних систем картографування, яка широко застосовується в робототехніці. Хоча попередні версії LiDAR зазвичай були громіздкими та важкими сенсорами, нещодавно з'явилися компактні та легкі LiDAR, які можна використовувати для швидких і точних процесів зображення. LiDAR працює аналогічно RADAR, але використовує світло від лазера замість радіохвиль для виявлення та зображення території, об'єктів і глибини до 300 метрів. Ця технологія дозволяє створювати точні карти як внутрішніх, так і зовнішніх просторів. LiDAR-системи забезпечують високу точність картографування та реального часу локалізації в різноманітних застосуваннях, таких як автономні транспортні засоби, мобільна робототехніка та внутрішнє картографування для навігації і відстеження активів.

LiDAR забезпечує детальні 3D точкові хмари, що представляють навколишнє середовище, та, завдяки своїй здатності адаптуватися до різних умов, LiDAR широко використовується у складних середовищах, таких як складські приміщення або зони надзвичайних ситуацій. Принцип роботи

LiDAR зображено на рисунку 1.6.

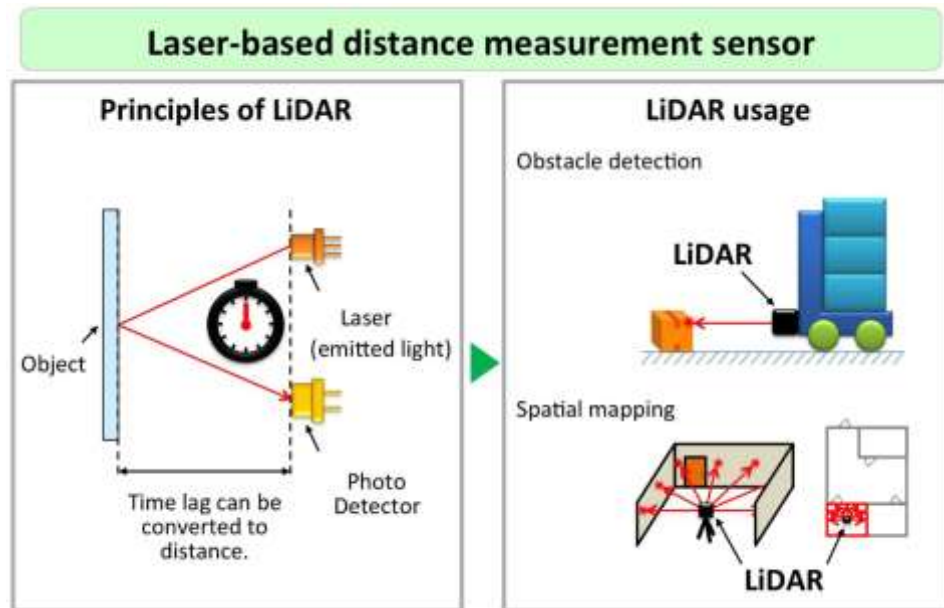


Рисунок 1.6 – Принцип та застосування LiDAR

Камери RGB-D є поширеним вибором для виявлень об'єктів, що додаються у комплексну карту території з можливістю розрізняти впізнавані об'єкти в навколишньому середовищі, оскільки камери RGB-D допомагають знаходити об'єкти та виявляти кути і контури для скупчення ознак.

Камери RGB-D включають інфрачервоні передавачі, приймачі та монофокальну камеру, які формують RGB-зображення з структурованими світловими патернами в інфрачервоному діапазоні. Ці патерни передаються на територію і приймаються, що дозволяє визначити глибину (або відстань) кожного пікселя зображення. Камери забезпечують детальне зображення об'єктів та їх контурів, дозволяють ефективно кластеризувати дані для подальшої обробки, та, як компоненти SLAM систем, вони використовуються в різних сферах, включаючи автономні роботи, системи безпеки та інтерактивні додатки. Принцип роботи таких камер зображено на рисунку 1.7.

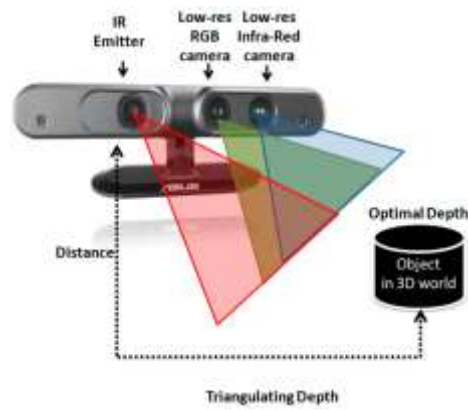


Рисунок 1.7 – Принцип роботи RGB-D камер

Монофокальні (монокулярні) камери є поширеними, дешевими та зазвичай доступними сенсорами, які включені до складу багатьох роботів і можуть використовуватися для технік SLAM. Існує тисячі алгоритмів, вихідного коду, книг і технічних статей, присвячених обробці зображень, машинному зору та глибокому навчанню на основі виходу камер, що забезпечують RGB-зображення. Для створення карти середовища та виявлення об'єктів існує кілька алгоритмів, які використовують об'єднання даних і аналіз зображень.

Принцип роботи монокулярних камер, як загальних цифрових камер, представлено на рисунках 1.8-1.9 для CCD та CMOS типів відповідно.

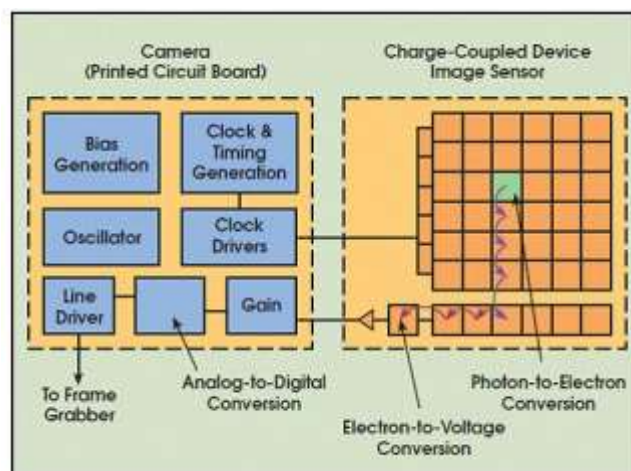


Рисунок 1.8 – Принцип CCD камер

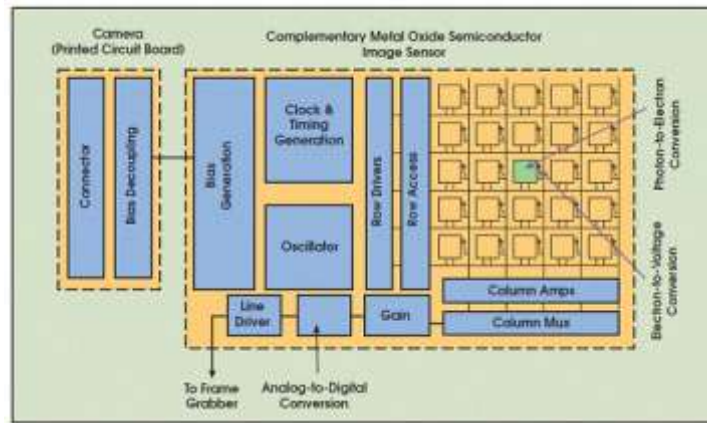


Рисунок 1.9 – Принцип CMOS камер

Монофокальні камери є сенсорами з низьким споживанням енергії і підходять для проектів SLAM з обмеженим у бюджеті. Їх недоліки це, в базових апаратних та програмних конфігурації, відсутність інформації про глибину (що призводить до неоднозначності масштабу мапи та дрейфу програмної позиції робота та ознак з часом), й тому вони найкраще підходять для малих середовищ і бюджетних застосувань через їх переваги низької вартості, компактного розміру, та легкості у використанні.

Стерео камери є ще однією системою для комп'ютерного зору у SLAM, яка може оцінювати глибину в зображеннях, використовуючи два захоплені зображення одного об'єкта з різних кутів. Переваги цих систем полягають у низькій вартості та високій мобільності. На рисунку 1.10 зображено модель роботи стереокамери за геометричними принципами.

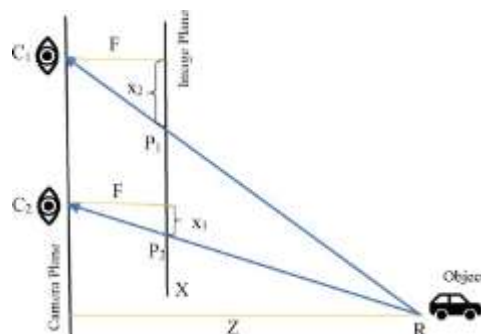


Рисунок 1.10 – Геометрія стереозору

Геометрія стереозору є основою для оцінки глибини в зображеннях, що отримуються з двох камер. У цій системі важливими параметрами є фокусна відстань ( $F$ ) камер, позиції камер ( $C_1$  і  $C_2$ ), а також координати в зображенні ( $x_1$  і  $x_2$ ). На основі цієї геометрії, глибина отримується формулою:

$$\left\{ \begin{array}{l} \frac{X}{Z} = \frac{x_2}{F} \\ \frac{X - C_1 - C_2}{Z} = \frac{x_1}{F} \\ Z = \frac{F(C_1 - C_2)}{x_2 - x_1} \end{array} \right. \quad (1.1)$$

Дисперсія, визначена як різниця між координатами в зображенні ( $x_2 - x_1$ ), є відстанню між точками на площині зображення, що відповідають центру камери та 3D-точці сцени. Знаючи глибину кожної точки в стереозображенні, можна отримати 3D-огляд області. Глибина ( $Z$ ) обчислюється через співвідношення між фокусною відстанню, дисперсією та відстанню між камерами. Це дозволяє визначити просторове розташування об'єктів у сцені, що є ключовим для застосувань SLAM.

Спільний недолік RGB-D камер – вони можуть використовуватися на обмеженій відстані та мають обмежене поле зору (FoV). Це може призвести до втрати трекінгу через нестачу просторової структури, необхідної для обмеження вирівнювання ICP (ітеративна найближча точка).

Типова система VSLAM включає програмне забезпечення з інтегрованою візуальною одометрією (VO) для оцінки траєкторії камери та компонент SLAM для оптимізації створеної карти. Існує багато варіацій реалізації VSLAM на основі цього підходу. Важливо зазначити різницю між VSLAM і VO: VO не використовує глобальну карту і не працює з алгоритмами прогнозування або траєкторії, а лише забезпечує попередню оцінку місця розташування та поз робота, використовуючи локальні стабільні ознаки, які надсилаються для оптимізації.

Проблема оцінки 6D пози, як категорія проблеми реєстрації зображень,

пов'язує два набори вимірювань  $\mathbf{M}$  і  $\mathbf{M}^*$  (в цьому підрозділі  $*$  позначає еталонні дані), отриманих з однієї сцени з різних точок зору. У випадку візуального SLAM, вимірювання отримуються з високою частотою (зазвичай понад 30 Гц), і передбачається, що точки зору не змінюються суттєво між послідовними вимірюваннями – в такому випадку проблему можна вирішити за допомогою локальної рамки оцінки пози.

На рисунку 1.11 представлено принцип проблеми відстеження зображень.

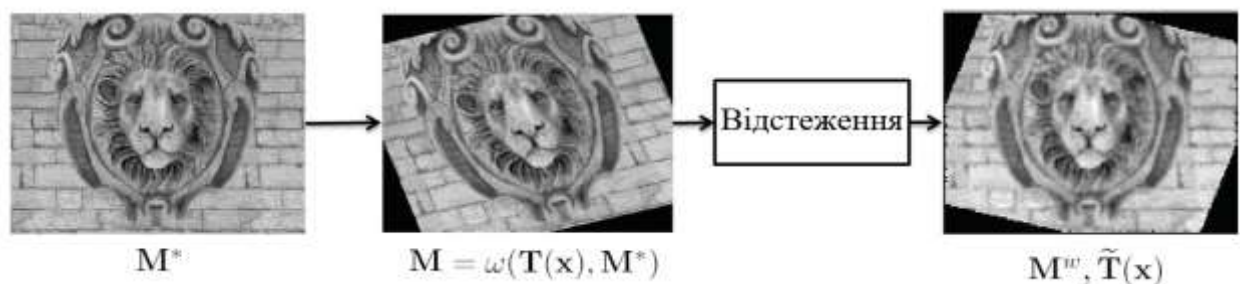


Рисунок 1.11 – Проблема відстеження зображень

Коли точки зору знаходяться далеко одна від одної, локальний метод не може бути надійно використаний, і виникає потреба в методах глобальної оцінки пози.

Два вимірювання  $\mathbf{M}$  і  $\mathbf{M}^*$  отримані з однієї сцени. Вимірювання  $\mathbf{M}$  отримується шляхом переміщення сенсора на  $\mathbf{T}(\mathbf{x})$ . Якщо модель придбання відома, її можна повторно проєкціювати на точку зору  $\mathbf{M}^*$  через функцію спотворення  $\omega$ . Процес реєстрації зображень має на меті оцінити значення  $\mathbf{T}(\mathbf{x})$  з двох вимірювань ( $\mathbf{M}, \mathbf{M}^*$ ).

Основна ідея вирішення проблеми локального оцінювання пози полягає в тому, щоб знайти позу  $\mathbf{T}(\mathbf{x})$ , яка мінімізує помилку репроекції між обома наборами вимірювань. Це означає, що модель проєкції сенсора, яка перетворює точку  $\mathbf{P}_i$  в середовищі на вимірювання  $\mathbf{M}_i$ , є відомою. У випадку камер зазвичай використовується модель камери-обскури. У своїй найбільш

загальній формі це часто виражається як така нелінійна задача мінімізації:

$$\mathbf{T}(\mathbf{x}) = \underset{\mathbf{T}(\mathbf{x})}{\operatorname{argmin}} \sum_{i=1}^N \|\mathbf{M}_i^* - \omega(\mathbf{T}(\mathbf{x}), \mathbf{M}_i)\|^2 \in \mathbb{SE}(3) \quad (1.2)$$

де  $\omega(\mathbf{T}(\mathbf{x}), \mathbf{M}_i)$  – функція деформації, яка перетворює набір вимірювань  $\mathbf{M}_i$  на набір  $\mathbf{M}_w$  за допомогою перетворення  $\mathbf{T}(\mathbf{x})$ ;

$N$  – кількість вимірювань, які присутні як в  $\mathbf{M}$ , так і в  $\mathbf{M}^*$ ;

$\mathbf{T}(\mathbf{x})$  – однорідна матриця перетворення, яку можна розкласти на компоненти обертання та переміщення:

$$\mathbf{T}(\mathbf{x}) = (\mathbf{R}(\mathbf{x}), \mathbf{t}(\mathbf{x})) \in \mathbb{SE}(3) \quad (1.3)$$

де  $\mathbf{x} = [vx, vy, vz, \omega x, \omega y, \omega z] \in \mathbb{SE}(3)$  – параметризація пози в алгебрі Лі.

Зв'язок між ними задається експоненціальним відображенням як:

$$\mathbf{T}(\mathbf{x}) = e^{[\mathbf{x}]^\wedge} \in \mathbb{SE}(3) \quad (1.4)$$

де  $[\cdot]^\wedge$  – оператор кручення матриці.

Для оцінки пози з RGB-D зображень, коли виконується реєстрація RGB-D зображень з кадру в кадр, частота отримання кадрів дозволяє підтримувати достатній перекриття між послідовними зображеннями для формулювання прямої функції помилки, яка порівнює фотометричну інформацію з RGB зображень і геометричну інформацію з глибинних зображень. Переваг у порівнянні з ранніми методами, які враховували лише ключові точки в RGB зображеннях, є багато. По-перше, потреба в розробці надійних дескрипторів ключових точок, які можуть бути використані для надійного знаходження відповідностей між зображеннями, отриманими з різних точок зору, усувається.

По-друге, жодна з отриманих даних не відкидається, що призводить до більш надійної формулювання. Нарешті, спільне врахування фотометричної та геометричної інформації забезпечує значні переваги в плані надійності.

Коли сцена позбавлена текстур, але має багато геометрії, геометрична вартість обмежує оцінку пози. Навпаки, коли присутня невелика геометрія, але багато текстури, позиція також добре обмежена. У ситуаціях, коли сцена багата як текстурою, так і геометрією, умови стають ідеальними, і позиція буде обмежена обома факторами. Завдяки цьому підходу у більшості випадків проблема оцінки пози буде майже завжди добре обмежена. Однак існують специфічні випадки невдач, такі як коли не доступні ні геометрія, ні текстура або коли спостерігається геометрія здалеку з малою текстурою. На практиці ці ситуації досить рідкісні, що робить таку формулювання трекінгу добре підходящою для застосувань у робототехніці.

Щодо категоризації саме алгоритмів VSLAM, базова класифікація алгоритмів полягає в тому, як дослідники використовують різні методи та стратегії для кожного потоку. Методи поділяють на непрямі та прямі:

- непрямі методи виділяють ознаки (ключові точки), що отримані з текстур шляхом обробки сцени та відстеження їх шляхом зіставлення їхніх дескрипторів у послідовних кадрах. Етапи виділення ознак і зіставлень є ресурсозатратними, але вони надають точність та стійкість до фотометричних змін інтенсивності кадру;

- прямі методи оцінюють рух камери саме з даних на рівні пікселів та мають задачу оптимізації для зменшення фотометричних похибок. Використовуючи принципи фотограмметрії, такі методи використовують усі вихідні пікселі камери та слідкують за їх замінами в наступних кадрах щодо певних обмежених аспектів (колір, яскравість тощо). У порівнянні з непрямими методами, такі характеристики дозволяють використовувати прямі підходи для моделювань збільшеної кількості інформації з зображень, й вони надають більш точну 3D-реконструкцію.

Прямі методи працюють краще в середовищах без текстур і не вимагають



більше обчислень для виділення ознак, але вони мають проблеми у великомасштабній оптимізації; а різні умови нестандартного освітлення негативно впливають на їх точність (що є проблемою, коли не можна впровадити використання інфрачервоних камер чи джерел освітлення у системі робота). Переваги та недоліки кожного підходу спонукали дослідників до розробки гібридних рішень, що об'єднують обидва методи. Змішані підходи зазвичай поєднують етапи непрямого та прямого виявлення, де один метод ініціює та коригує інший.

Деякі з найсучасніших реалізацій VSLAM включають два додаткові модулі: замикання циклу та картографування, що дозволяють розпізнавати раніше відвідані місця для точнішого відстеження і вдосконалення карти на основі пози камери.

За останні роки різні дослідники та розробники представили свої підходи і вдосконалення методів та налаштувань VSLAM, деякі з яких спільні з іншими методами SLAM [28-30].

Підходи до мультисенсорної обробки:

– використання кількох камер. Оскільки відтворити тривимірні траєкторії рухомих об'єктів за допомогою однієї камери складно, деякі дослідники пропонують використовувати кілька камер. Завдяки перекриттю оглядів камери стає легше відтворювати динамічні точки в 3D, об'єднуючи оцінки позицій всередині і між камерами, а також відображення. Крім того, система VSLAM стає більш ефективною в складних умовах, оскільки зменшується залежність від однієї камери, що усуває проблеми з оклюзією та розрідженістю текстури. Деякі методи використовують зображення з камер із ширококутними об'єктивами типу «риб'яче око», перетворюючи їх на ключові кадри для подальшої швидкої, але ресурсомісткої обробки;

– мультимодальні датчики. Деякі підходи об'єднують різні типи датчиків, поєднуючи дані з візуальних та інерційних сенсорів для покращення продуктивності. Зокрема, інтегруються карти глибини LiDAR з RGB-

зображеннями камер. Також можливе розширення типів вхідних даних шляхом використання радарів, гіроскопів, акселерометрів тощо.

Підходи до оцінювання пози:

- оцінювання за даними ліній/точок. Це метод побудови структурних ліній як корисних ознак для визначення позиції камери. Структурні лінії зазвичай відповідають домінуючим напрямкам і містять інформацію про глобальну орієнтацію, що покращує прогнозовані траєкторії. Один із прикладів точково-лінійного SLAM – це система, яка одночасно використовує лінійні та точкові ознаки для покращення оцінки позиції, особливо в ситуаціях з обмеженою кількістю точок ознак. Недолік – висока обчислювальна вартість та залежність від складних геометричних примітивів;

- оцінювання за комплексними ознаками. Деякі методи пропонують оцінку положення з шістьма ступенями свободи через складні обчислення хмар точок зі стереокамер та оптичних потоків. Вони також дозволяють відтворювати великомасштабні карти інтер'єрів, використовуючи штучні маркери або орієнтири для систем комп'ютерного зору.

Підходи до забезпечення точності відносно реального середовища:

- відокремлення ознак місцевості і використання різних джерел даних із датчиків (наприклад, поєднання камер з LiDAR та радарями) допомагає підвищити стійкість до змін у зовнішньому вигляді середовища;

- методи динамічних середовищ дозволяють створювати карти в умовах змінної обстановки. Найбільш поширений підхід – це відокремлення статичних, динамічних і напівстатичних змін в середовищі, використовуючи тільки статичні елементи для локалізації, що підвищує надійність оцінки позиції. Деякі методи використовують п'ятипоточковий процес, що включає відстеження, семантичну сегментацію, локальне картографування, замикання циклу та побудову щільної семантичної карти. Інші методи залучають сортування значень оптичного потоку для розпізнавання ознак і реконструкції карти середовища, використовуючи семантичну сегментацію, узгодженість руху та геометричні обмеження.

Підходи з обмеженням використання ресурсів:

- обмежені обчислювальні можливості. Деякі системи VSLAM, розроблені для мобільних пристроїв або пристроїв з обмеженими ресурсами, використовують стереокамери та модулі для виділення ознак середовища або об'єднують візуальну одометрію з методами виявлення об'єктів;
- розвантаження обчислень. Ця практика передбачає передачу складних обчислень SLAM у хмару або віддалені комп'ютери, що робить роботизовані системи більш компактними, енергоефективними та легкими.

Ці методи спільно сприяють поліпшенню довготривалої локалізації та картографічних можливостей мобільних роботів, усуваючи дисперсію візуальних даних, виключаючи динамічні елементи з аналізу, підвищуючи точність представлення карт і синхронізуючи паралельні операції. Результатом впровадження будь-якого з цих підходів є здатність роботів адаптуватися до змін у середовищі, підтримувати точні карти та ефективно орієнтуватися протягом тривалого часу.

Також варто відзначити, що більшість із цих підходів можна поєднувати з машинним навчанням та нейронними мережами, які забезпечують універсальні й адаптивні обчислювальні процеси, необхідні для швидкого та точного виявлення ознак і картографування. Наприклад, створення ознак-описів на основі згорткових нейронних мереж (CNN) є більш детальним і точним порівняно з традиційними методами програмного опису середовища.

### 1.5 Висновки до першого розділу

В даному розділі було ретельно розглянуто сучасні універсальні 3D-симулятори роботів, експерименти з симуляціями, сучасні використання комп'ютерного зору у симуляторах, та сучасні SLAM та VSLAM технології та принципи їх роботи, що є необхідним для практичної реалізації VSLAM системи, яку, для виконання цілей роботи, можна використовувати, досліджувати, та модифікувати з залученням симулятора.

## 2 РОЗРОБКА СИСТЕМНОЇ РЕАЛІЗАЦІЇ ПРОЕКТУ

### 2.1 Обґрунтування прийнятих рішень системної реалізації розробки

Список програмних компонентів, що були обрані для розробки:

- OS середовища: Ubuntu 22.04.4 LTS (Jammy Jellyfish) 64-bit;
- версія ROS2: Humble;
- пакет VSLAM: ORB-SLAM3;
- симуляційне ПЗ: Gazebo.

Обрані ті OS, симуляційне ПЗ, та версія ROS2, що відомо сумісні з ORB-SLAM3.

Було вирішено обрати VSLAM бібліотеку ORB-SLAM3 через те, що ця бібліотека:

- працює в реальному часі;
- може виконувати візуальний, візуально-інерціальний і багатокартковий SLAM з монокулярними, стереокамерами та камерами RGB-D, використовуючи моделі об'єктів із отворами та «риб'яче око»;
- у всіх конфігураціях датчиків ORB-SLAM3 настільки ж надійна, як і найкращі відомі й доступні системи;
- має тісно інтегровану візуально-інерціальну систему SLAM на основі функцій, яка повністю покладається на оцінку Maximum-a-Posteriori (MAP), навіть під час фази ініціалізації IMU. Це призводить до створення надійності системи під час її функціонування в реальному часі;
- має систему кількох карт, яка базується на новому методі розпізнавання місць із покращеним запам'ятовуванням. Це дозволяє ORB-SLAM3 переживати тривалі періоди поганої візуальної інформації, створюючи нові карти, які плавно об'єднуються під час повторного перегляду раніше нанесених на карту територій;

– здатна повторно використовувати всю попередню інформацію на всіх етапах алгоритму. Це дозволяє включати видимі разом ключові кадри, які забезпечують спостереження високого паралакса, підвищуючи точність, навіть якщо вони значно рознесені в часі або від попередніх сеансів картографування;

– доступна як програмне забезпечення з відкритим кодом за ліцензією GPL-3.0, що дозволяє спільноті користуватися перевагами цієї найсучаснішої системи SLAM.

Для зручності налаштування робочого процесу, для цієї роботи Ubuntu використаний через VNC (Virtual Network Computing) з контейнеру Docker образу Ubuntu.

Під час побудування з відкритого вихідного коду по рекомендованих інструкціях оригінальних розробників ORB-SLAM3 [29] було враховано й ті відомості про помилки та способи їх виправлення у встановленні та використанні ORB-SLAM3, що доступні на розробниковому вебсервісі GitHub [31-35].

Для інтеграції ROS2 з ORB-SLAM3 була використана спеціальна програмна обгортка від розробника zang09 [36].

Спираючись на аналіз симуляційного ПЗ як в рамках цієї роботи, так й з публікацій дослідників з Австралії [37], для задоволення потреби технічного завдання проєкту – можливості його впровадження у реальні робототехнічні системи – та для тестування та оцінки системи розробки використано середовище моделювання Gazebo. Це середовище моделювання є кращим в тих випадках, коли планується розробка не тільки для симуляції, але й для реальних систем – завдяки однорідному підключенню ROS2 (що впроваджено інтеграцією в структуру ROS2). Інтеграція Gazebo з ROS2 дозволяє використовувати модельованих роботів у імітаційних середовищах з текстурованими будівельними об'єктами, перешкодами та різними умовами освітлення; для розробки можна використовувати спільний інтерфейс керування для моделювання та для справжньої системи (підхід типу

«цифрового близнюка»).

Gazebo чудово підходить для розробки та тестування VSLAM-систем, особливо для додатків з великою кількістю фізики та сенсорів. Gazebo робить акцент на точному моделюванні фізики та сенсорів. Причини, з яких це середовище особливо цінне для VSLAM:

а) кілька фізичних рушіїв, таких як ODE, Bullet і DART, які є критично важливими для моделювання реалістичних взаємодій робота з навколишнім середовищем. Це забезпечує надійне тестування системи VSLAM в динамічних середовищах;

б) вбудована підтримка різних датчиків, таких як RGB-камери, глибинні камери та LiDAR. Вони мають важливе значення для VSLAM, оскільки дозволяють точно емітувати сприйняття реального світу;

в) підтримка плагінів для додаткових типів датчиків і функцій, що робить його універсальним для індивідуальної розробки VSLAM;

г) глибока інтеграція з ROS полегшує тестування та розгортання алгоритмів VSLAM в робототехнічних системах за допомогою додаткових програмних інструментів.

В рамках даної роботи, було обрано проведення тестування саме монокулярного SLAM через розповсюдженість та дешевизну проектів VSLAM на основі звичайної монокулярної камери.

## 2.2 Засоби та результати системної реалізації розробки

Команди запуску RViz для загального дебагінгу, візуалізації та взаємодії різних аспектів самого робота – на рисунку 2.1.

```
ubuntu@e9b5cf60895a:~$ cd ~/ros2_demows
source install/local_setup.bash
ros2 launch my_first_pkg display.launch.py
```

Рисунок 2.1 – Команди запуску RViz

Команди запуску Gazebo з параметрами проекту для тестування системи приведено на рисунку 2.2.

```
ubuntu@e9b5cf60895a:~$ cd ~/ros2_demos
source install/local_setup.bash
ros2 launch my_first_pkg gazebo.launch.py
```

Рисунок 2.2 – Команди запуску Gazebo

Команди запуску базового скрипту керування роботом приведено на рисунку 2.3.

```
ubuntu@e9b5cf60895a:/$ cd ~/ros2_demos
source install/local_setup.bash
ros2 run teleop_twist_keyboard teleop_twist_keyboard
```

Рисунок 2.3 – Команди запуску скрипту керування

Команда запуску ORB-SLAM3 ROS2 – на рисунку 2.4.

```
ubuntu@e9b5cf60895a:/$ cd ~/orbslam3_ws
source install/local_setup.bash
ros2 run orbslam3 mono ~/orbslam3_ws/src/orbslam3_ros2/vocabulary/ORBvoc.txt ~/orbslam3_ws/src/orbslam3_ros2/config/monocular/TUM1.yaml
```

Рисунок 2.4 – Команди запуску ORB-SLAM3 ROS2

Команди запуску `rqt graph` (для візуалізації потоків повідомлень і обчислювального графіка вузлів системи ROS) – на рисунку 2.5.

```
ubuntu@e9b5cf60895a:/$ ros2 run rqt_graph rqt_graph
```

Рисунок 2.5 – Команди запуску rqt graph

Для створення сцени у симуляторі, можна або завантажити існуючу з файлу, або скористатися внутрішніми інструментами додавання об'єктів. Вкладка бібліотек-каталогів з файлами об'єктів – зображена на рисунку 2.6.



Рисунок 2.6 – Вкладка бібліотек - каталогів з файлами об'єктів

Розроблену модель робота (кодом стандарту URDF, додаток А) для тестування системи зображено на рисунку 2.7.



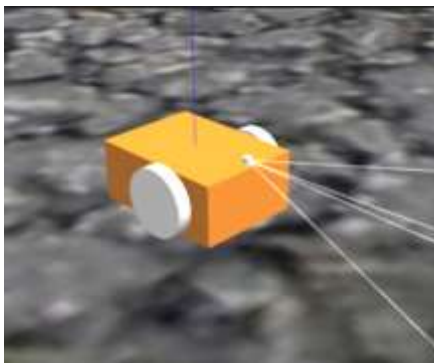


Рисунок 2.7 – Модель робота для тестування VSLAM функціоналу системи

Вкладка поточного складу сцени вікна – включаючи загальні параметри гравітації та освітлення зображено на рисунку 2.8. Вікно перегляду внутрішніх тем симулятора зображено на рисунку 2.9.

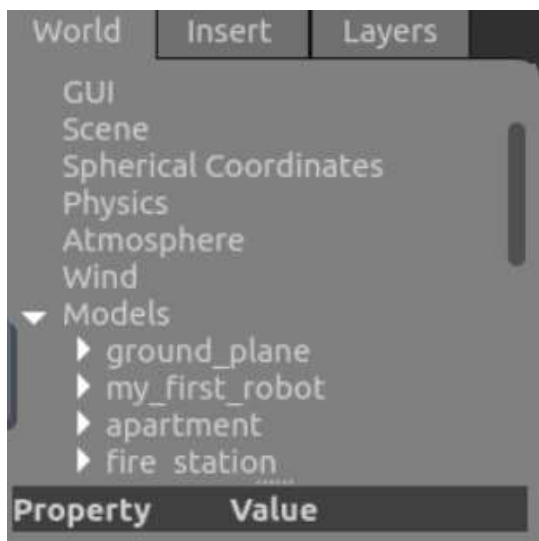


Рисунок 2.8 – Вкладка поточного складу сцени вікна



Рисунок 2.9 – Вікно перегляду внутрішніх тем симулятора

Змодельоване у Gazebo тривимірне середовище для тестування системи зображено на рисунку 2.10.

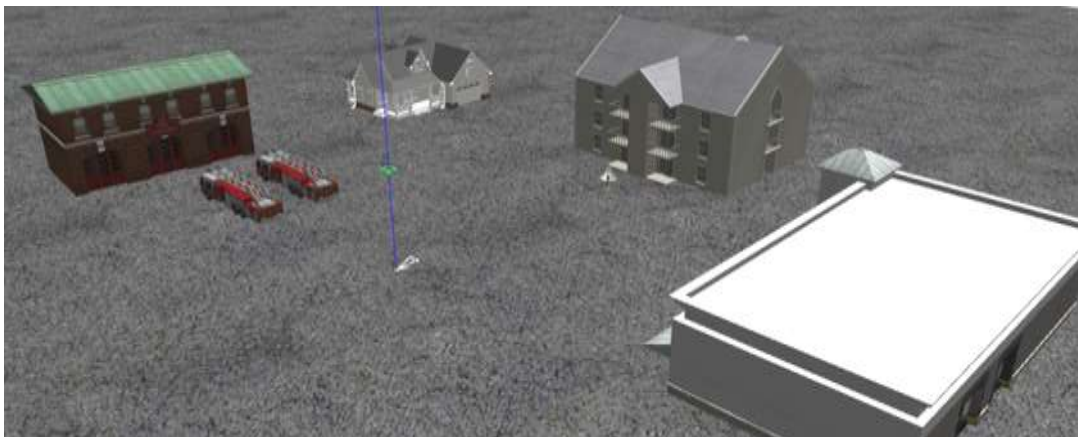


Рисунок 2.10 – Модель середовища для перевірки працездатності VSLAM функціоналу системи



Повідомлення з інструкціями при активації скрипту керування роботом з терміналу – зображено на рисунку 2.12.

```
Moving around:
  u   i   o
  j   k   l
  m   ,   .

For Holonomic mode (strafing), hold down the shift key:
-----
  U   I   O
  J   K   L
  M   <   >

t : up (+z)
b : down (-z)

anything else : stop

q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
e/c : increase/decrease only angular speed by 10%

CTRL-C to quit

currently:      speed 0.5      turn 1.0
```

Рисунок 2.12 – Термінал з повідомленням активного скрипту керування роботом

Щоб оцінити працездатність системи, були відстежені показники інтерфейсу ORB-SLAM3.

Початкова ініціалізація комп'ютерного зору ORB-SLAM3 та початкова ініціалізація вікна відображень хмари точок картографування зображені на рисунках 2.13-2.14. Відображення чорного екрану з «WAITING FOR IMAGES» та порожньої мапи продовжується до підвантаження зображення кадрів з камери та помічення алгоритмами візуальних рис з кадрів. Фід з вікна системи може прпинитися при таймауті через те, що не було знайдено ознак поверхонь та об'єктів у поточних кадрах з камери. Параметри камери вказано для системи кодом, у додатку Б.



Рисунок 2.13 – Початкова ініціалізація комп'ютерного зору ORB-SLAM3



Рисунок 2.14 – Початкова ініціалізація вікна відображень хмари точок картографування

З точки зору користувача, для повного ініціалізування ORB-SLAM3 необхідно, щоб активна камера, яка подає зображення на ORB-SLAM3, трохи порухалась та покрутилась. Для модельованих середовищ, простір винний бути достатньо «реалістичним», тобто об'єкти середовища мусять мати текстури, з яких алгоритми можуть знаходити необхідні риси об'єктів, як це підмічено на сторінці розробки ORB-SLAM3 [38].

Перший приклад дисплею ORB-SLAM3 для відображення активних процесів комп'ютерного зору знаходження рис з поточного кадру з камери – на рисунку 2.15.

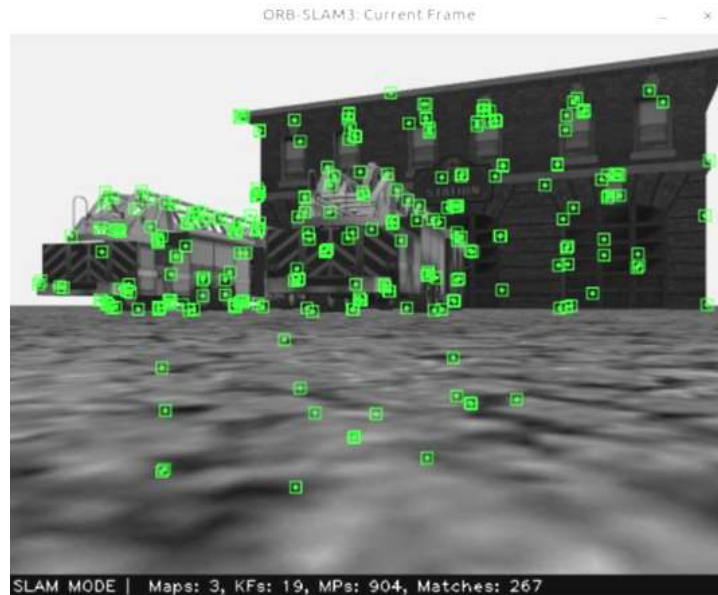


Рисунок 2.15 – Перший приклад роботи комп’ютерного зору ORB-SLAM3

Перший приклад дисплею ORB-SLAM3 для відображення поточної побудованої мапи з хмари точок – на рисунку 2.16. Зелений прямокутник – поточне положення камери. Інші прямокутники – минулі положення камери. Червоні точки – риси, як їх поточно бачить камери. Чорні точки – точки, що зафіксовані як частини побудованої мапи.

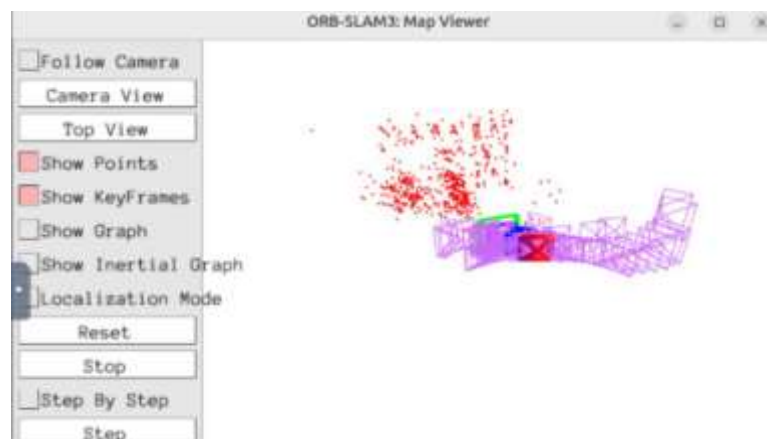


Рисунок 2.16 – Відображення хмари точок картографування, перший приклад

Другий приклад дисплею ORB-SLAM3 для відображення поточних рис, помічених комп'ютерним зором системи – на рисунку 2.17. Система здатна одночасно знаходити риси кількох різних об'єктів для побудови спільної мапи.

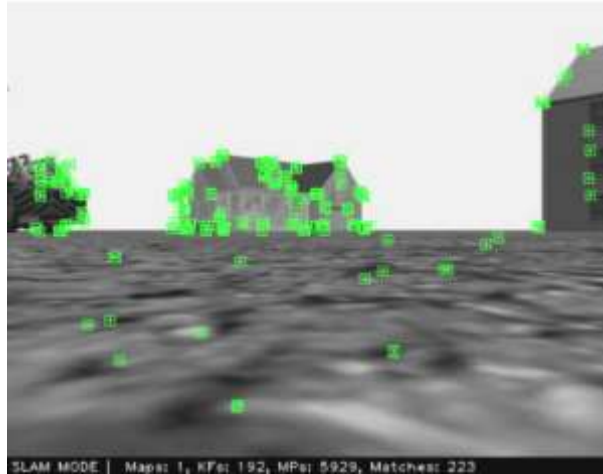


Рисунок 2.17 – Другий приклад роботи комп'ютерного зору ORB-SLAM3

Перший приклад дисплею ORB-SLAM3 для відображення поточної побудованої мапи з хмари точок – на рисунку 2.18. Варто відмітити, що алгоритми локалізацій здатні самостійно виправляти помилки у визначенні положення камери – на скріншоті це помітно по нещільній купці синіх прямокутників, які позначають помилкові відхилення у історії визначення положення камери.

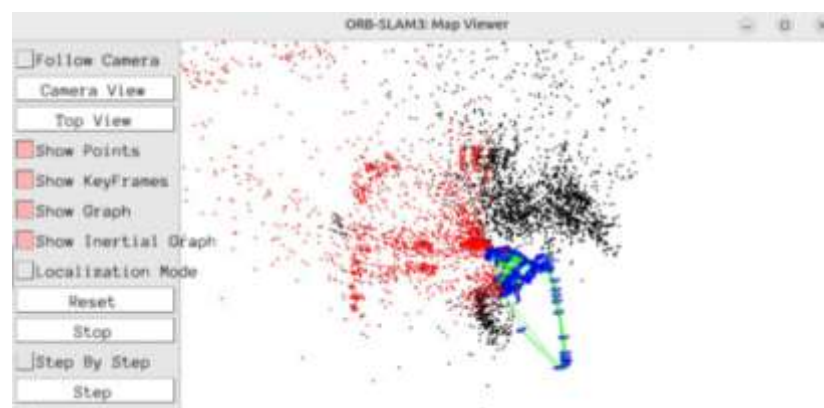


Рисунок 2.18 – Відображення хмари точок картографування, другий приклад

Joint State Publisher — це пакет у екосистемі ROS, призначений для керування та публікації стану суглобів робота. Має функцію відстеження суглобів – кути у радіанах або зсуви в метрах, та швидкість кожного шарніра. Він публікує ці значення як повідомлення `sensor_msgs/JointState`, які є фундаментальними для ефективного функціонування інших компонентів архітектури ROS.

Joint State Publisher працює разом із Robot State Publisher, що вимагає двох основних вхідних даних:

- повідомлення `sensor_msgs/JointState` від видавця Joint State;
- модель робота, визначена в URDF (Unified Robot Description Format).

Robot State Publisher використовує ці дані для виведення положення та орієнтації кожної системи координат робота, публікуючи їх у пакеті `tf2`, який відстежує всі системи координат протягом певного часу. У сценаріях симуляції, наприклад із використанням Gazebo, Joint State Publisher можна інтегрувати через плагін Gazebo для ефективної публікації спільних станів.

На рисунку 2.19 у вікні Joint State Publisher можна побачити можливість перевірки та взаємодії з моторами колес робота проєкту. Так як у даному випадку колеса виконано ідеально плоскими та обертальний рух від моторів іде на великих швидкостях постійним обертанням, та Joint State Publisher стандартно розглядає суглоби як серводвигуни з обмеженим кутом руху, то зміна значень не має помітних результатів у симуляції.



Рисунок 2.19 – Вікно Joint State Publisher проєкту



### 2.3 Висновки до другого розділу

Отже, в даному розділі були розглянуті результати проведеної розробки системної реалізації проєкту, включаючи: раціональні обґрунтування щодо обраних системних та програмних компонентів проєкту, використані засоби та результати системної реалізації розробки, огляд, підготовка інструментів та активного середовища симуляції, огляд базового функціоналу системи.

### 3 АНАЛІЗ ФУНКЦІОНАЛУ ПРОЕКТУ

На рисунках 3.1-3.4 представлено rqt граfi тем даних в різних режимах відображення: повний, без сумаризації tf, tf2 та зображень, з сумаризацією tf та tf2, з сумаризацією зображень.

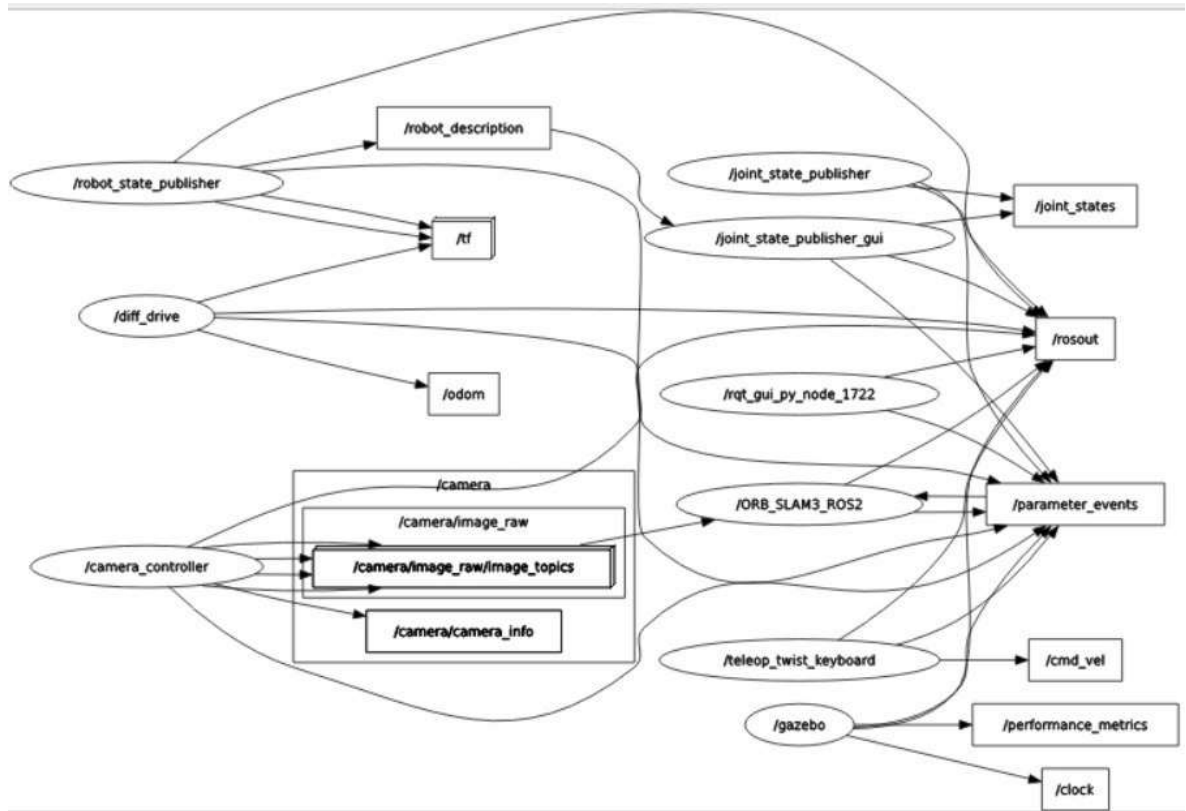


Рисунок 3.1 – Повний rqt граф



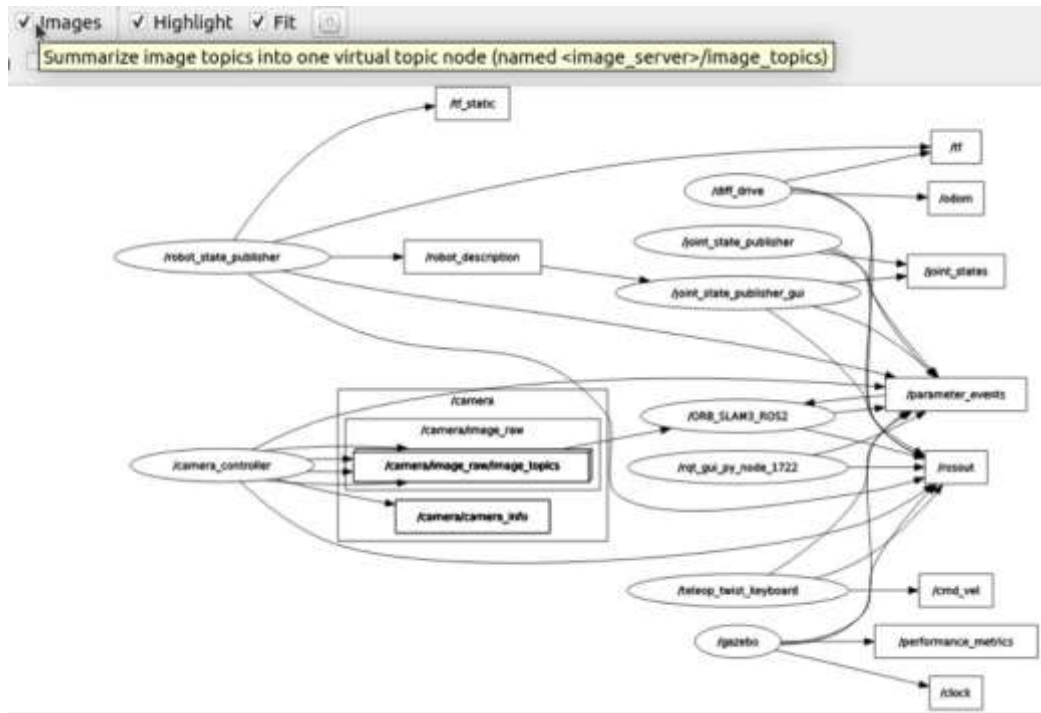


Рисунок 3.4 – rqt граф з сумаризацією зображень

На рисунках 3.5-3.6 представлено середовище для досліджень системи. Середовище побудоване методами вставки об'єктів з готов бібліотек.



Рисунок 3.5 – Вигляд сцени для аналізу ефективності VSLAM системи,  
ракурс 1

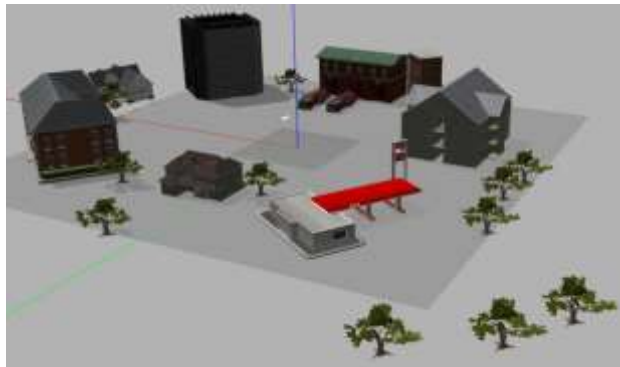


Рисунок 3.6 – Вигляд сцени для аналізу ефективності VSLAM системи,  
ракурс 2

Під час дослідження функціонала системи, емпірично було виявлено, що найшвидший метод VSLAM збору даних середовища з певної зони на місцевості – рух носія камери по колу. Таким чином, VSLAM робота можна реалізувати зверху конструкції монокулярною камерою, що обертається зі зсувом (для імітації саме руху по колу а не обертання), майже як радари старих зразків. Але рух не повинний бути занадто швидким, інакше алгоритми VSLAM можуть не встигнути нормально опрацьовувати зображення. Конкретну швидкість можна виявити вже практичною перевіркою на готовій системі, бо швидкодія та ефективність VSLAM реальній системі може залежити, серед іншого, як від висоти, на якій знаходиться камера (чим більше висота, тим більше видно ознак, які потрібно опрацьовувати) так й від завантаженості апаратних можливостей системи.

Також під час процесу VSLAM, при обробці частин місцевості, де незадовільний реалізм даних, підсистема мапи може відображати помилки типу значних похибок локалізації – буде показувати, що робот різко та випадково змінює свою позицію по всій території зкартографованої мапи, але це виправляється до норми з подільшим збором даних навіть у «неповній» віртуальній місцевості.

Скріншоти тестування системи з даними для розрахунків представлено на рисунках 3.7-3.18.

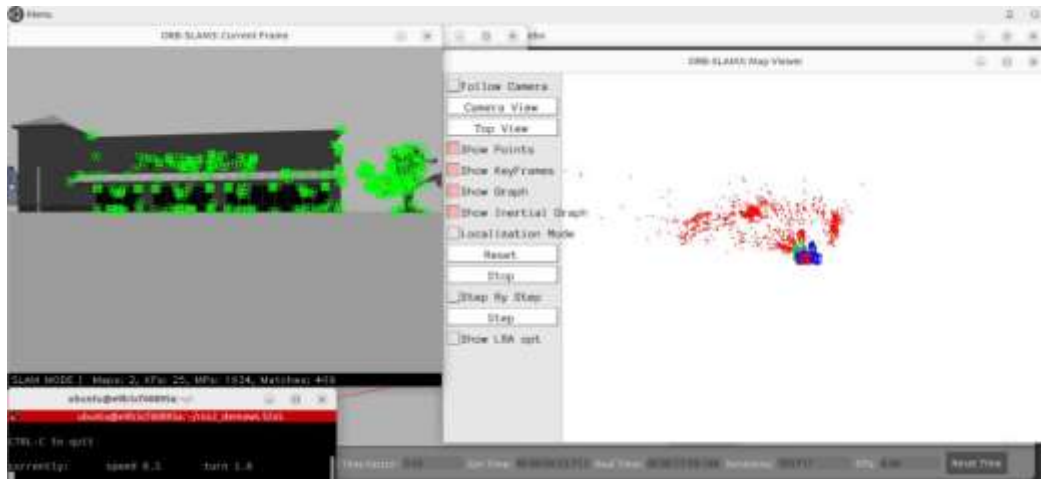


Рисунок 3.7 – Робота системи у підготовленому середовищі, кадр-зразок 1

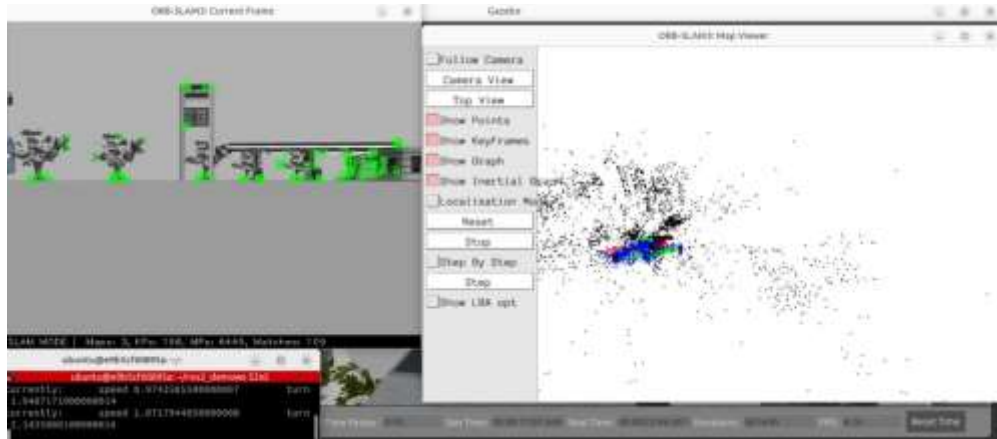


Рисунок 3.8 – Робота системи у підготовленому середовищі, кадр-зразок 2

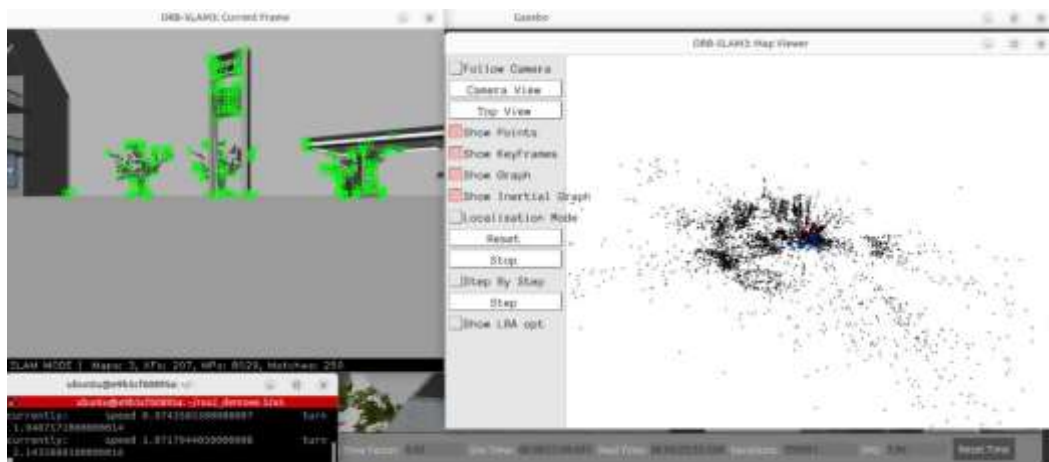


Рисунок 3.9 – Робота системи у підготовленому середовищі, кадр-зразок 3

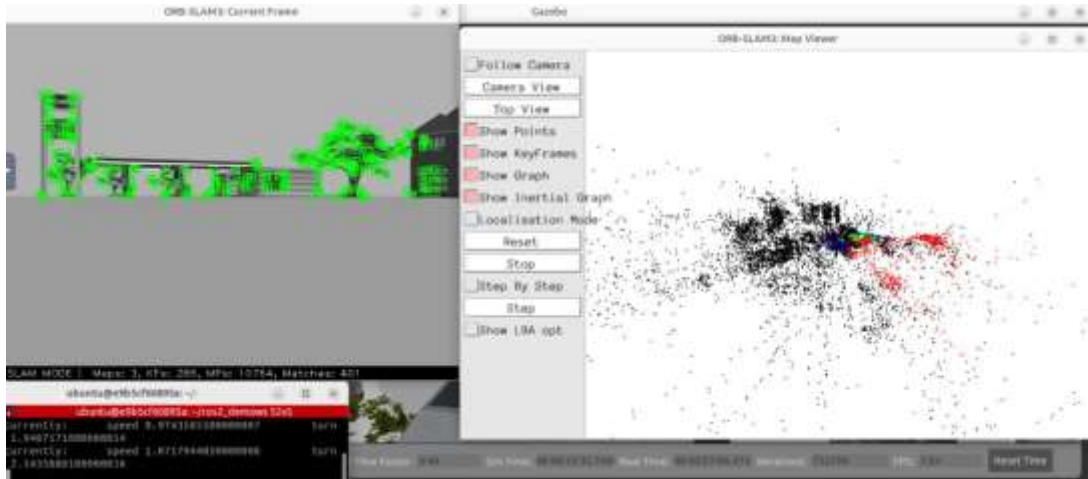


Рисунок 3.10 – Робота системи у підготовленому середовищі, кадр-зразок 4

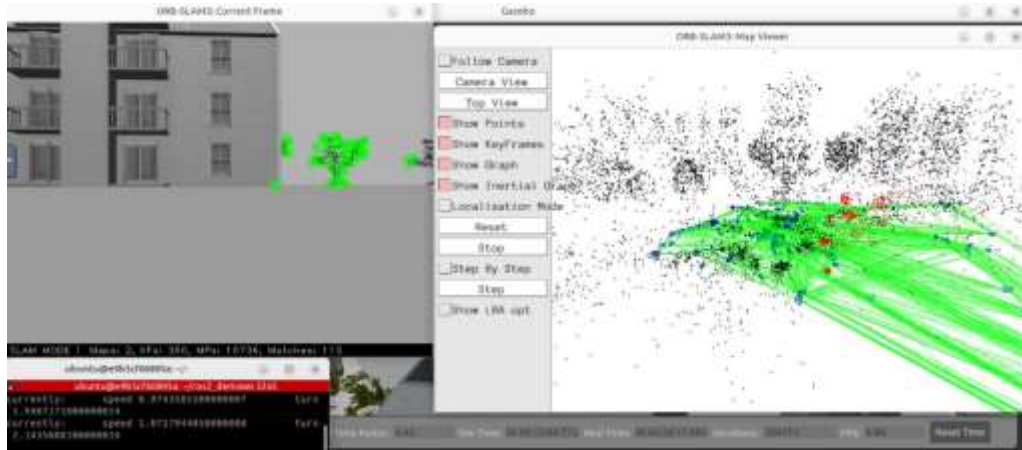


Рисунок 3.11 – Робота системи у підготовленому середовищі, кадр-зразок 5

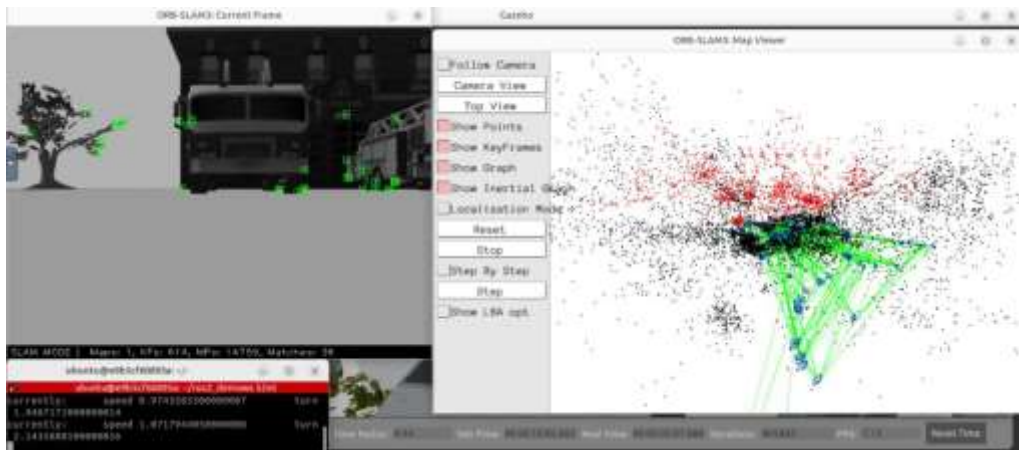


Рисунок 3.12 – Робота системи у підготовленому середовищі, кадр-зразок 6

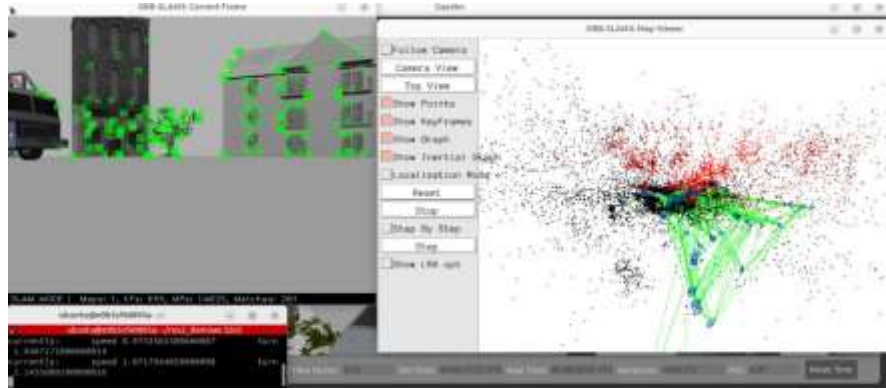


Рисунок 3.13 – Робота системи у підготовленому середовищі, кадр-зразок 7

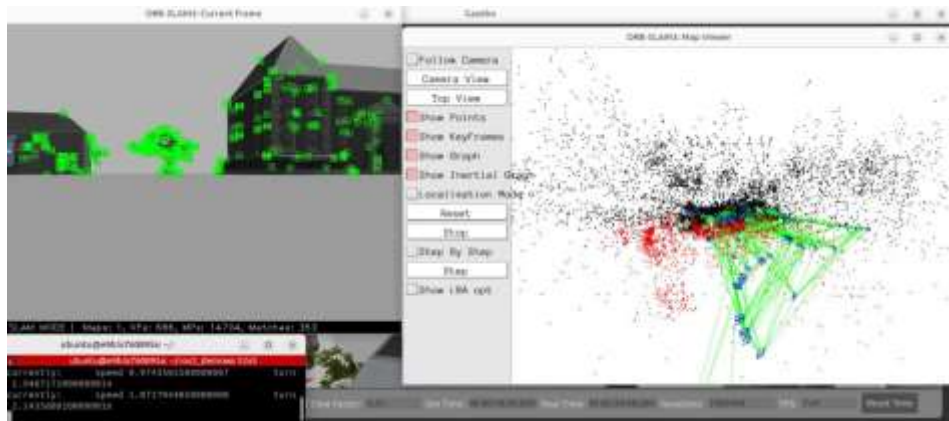


Рисунок 3.14 – Робота системи у підготовленому середовищі, кадр-зразок 8

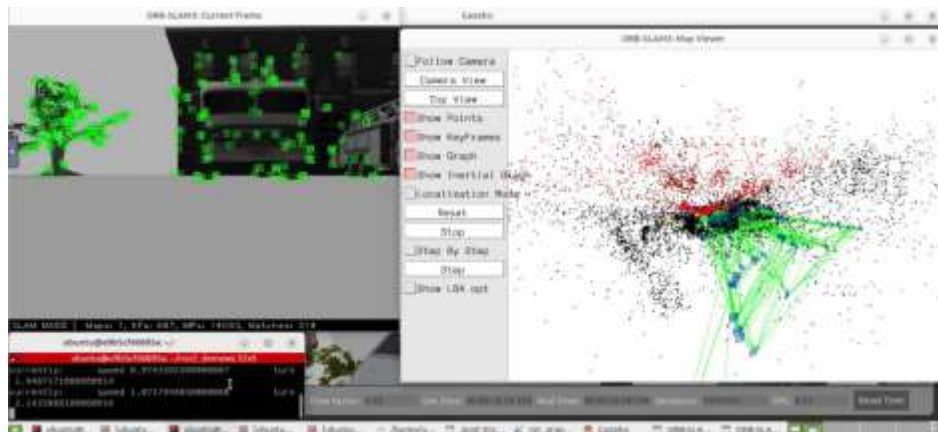


Рисунок 3.15 – Робота системи у підготовленому середовищі, кадр-зразок 9



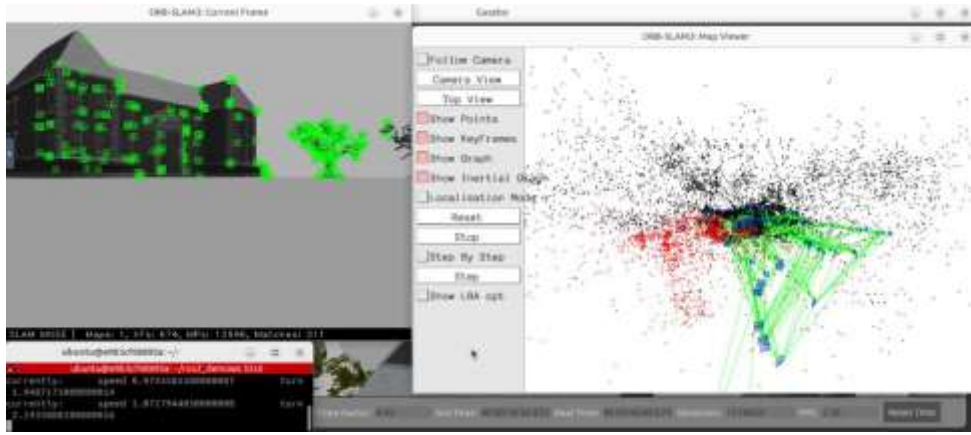


Рисунок 3.16 – Робота системи у підготовленому середовищі, кадр-зразок 10

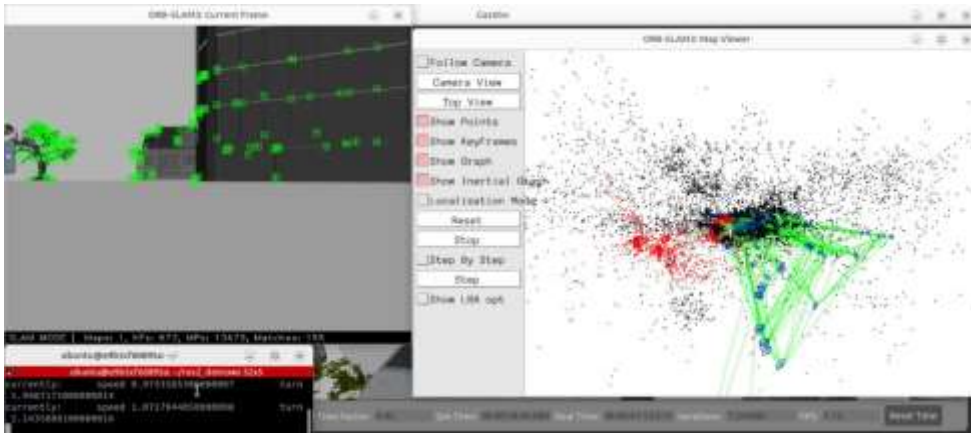


Рисунок 3.17 – Робота системи у підготовленому середовищі, кадр-зразок 11

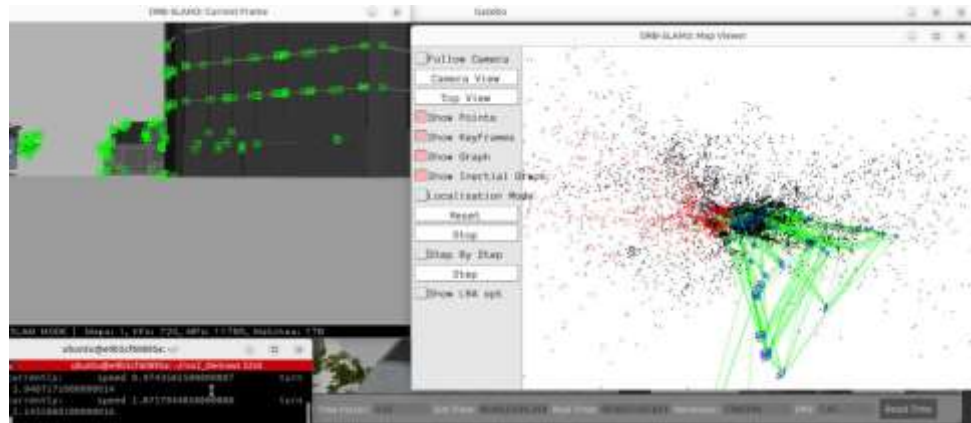


Рисунок 3.18 – Робота системи у підготовленому середовищі, кадр-зразок 12

Варто відмітити прогресію значення кількості мап з точки зору системи. На початку роботи та “ознайомлення” з середовщем, система тільки набирає ознаки та розглядає їх як мапи-скупчення ознак, але при подальшій інтерпретації та зв’язуванню даних, система виявляє середовище як суцільне.

На рисунку 3.19 зображено під’єднання RViz до топіку зображень з камери, де видно й кольоровий фід кадрів камери, й системні 3D дані RViz щодо робота.

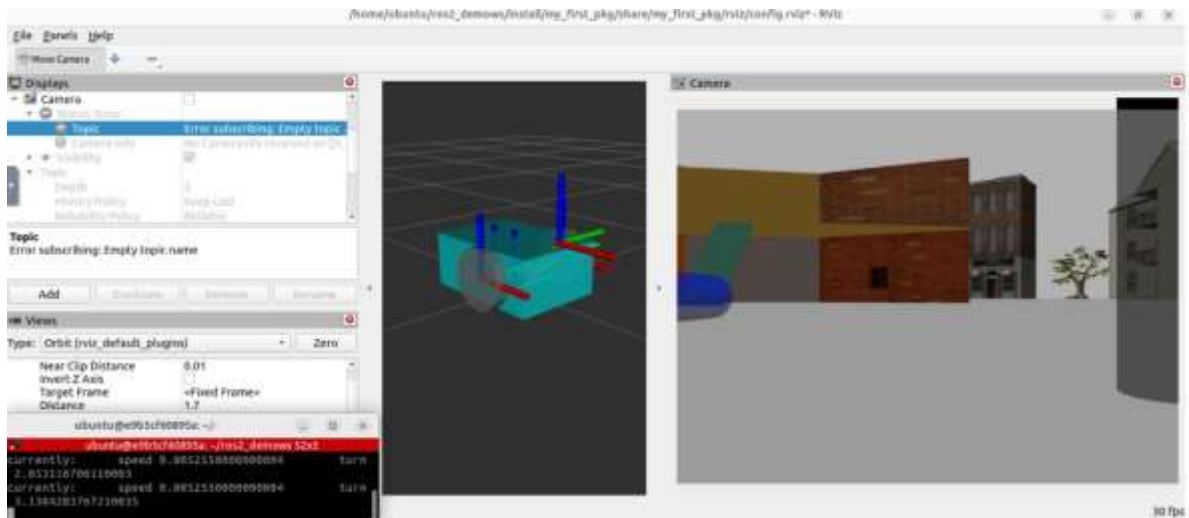


Рисунок 3.19 – Підключення повного кольорового фіду з камери у RViz перегляді статусу робота

Для перегляду, вибору, та додавання саме простих об’єктів типу форм, Gazebo має режим редагування форм.

Також цей режим зручний для швидкого та наглядного оцінювання кількості полігонів об’єктів, що потрібно як для приблизної оцінки споживання комп’ютерних ресурсів сценою, так й для розуміння впливів форм на процеси розпізнавання VSLAM.

Режим редагування форм зображено на рисунку 3.20.



Рисунок 3.20 – Режим форм

Для остаточного дослідження роботи системи, було використано режим будування, що є інструментом Gazebo. Початкові будування зображено на рисунках 3.21-3.26.

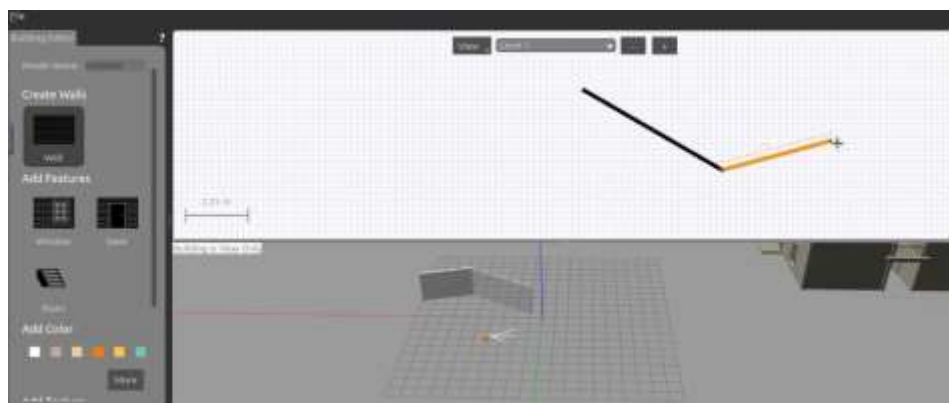


Рисунок 3.21 – Режим будівництва, побудова стін

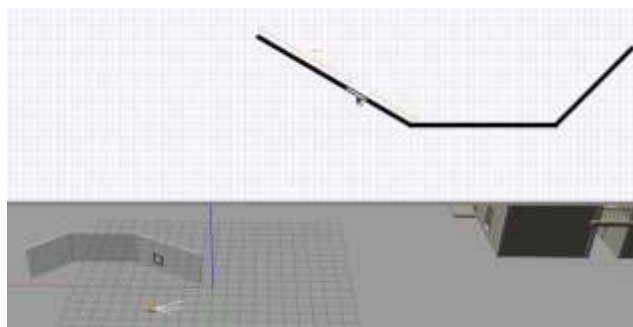


Рисунок 3.22 – Виставлення вікна

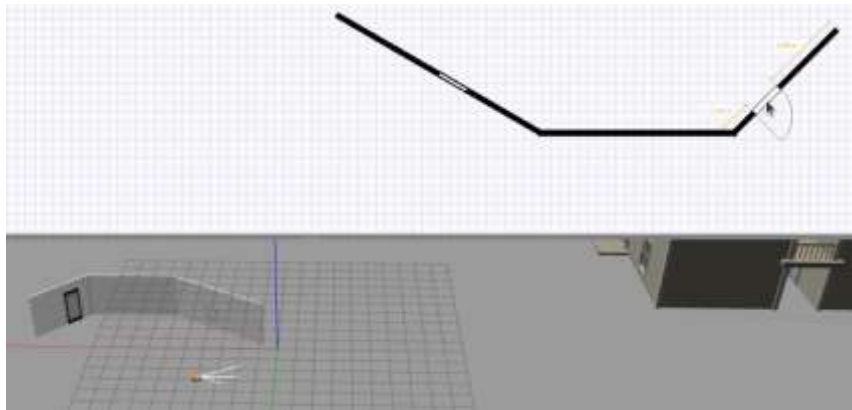


Рисунок 3.23 – Виставлення дверей

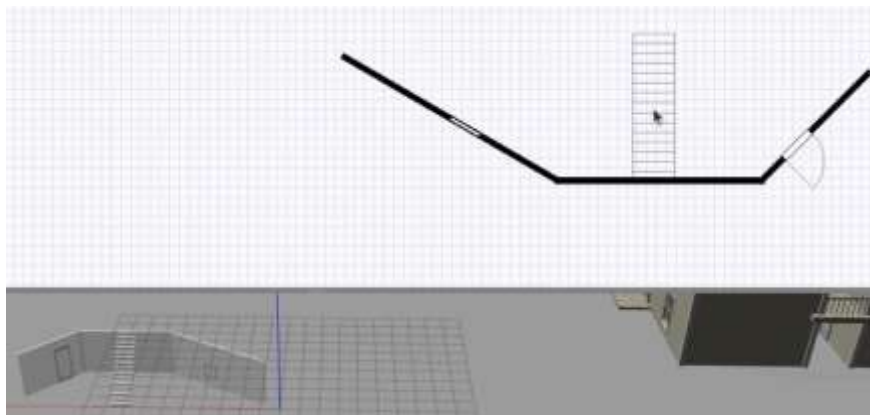


Рисунок 3.24 – Встановлення сходів



Рисунок 3.25 – Фарбування об'єктів

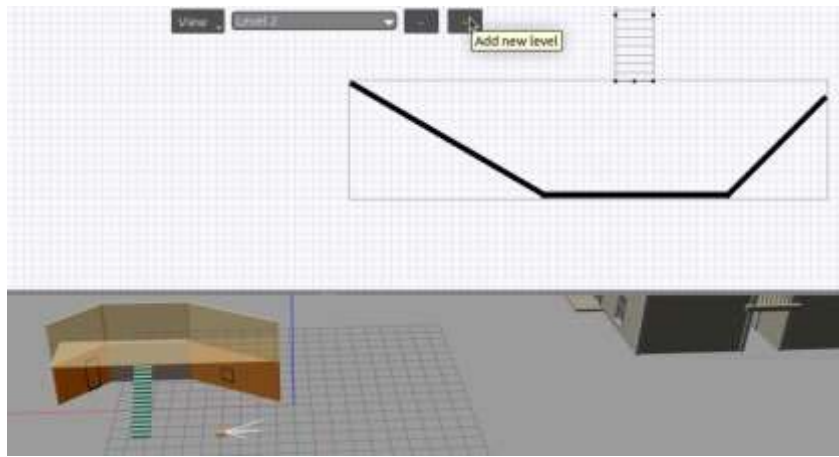


Рисунок 3.26 – Створення нового поверху

Як вже було зазначено, об'єкти середовища мусять мати текстури для того, щоб VSLAM розпізнав об'єкти.

Функціонал створення та «фарбування» будівель симулятора надає можливість надати наглядну демонстрацію залежності виявлення об'єктів від їх реалізму даних.

На рисунку 3.27 зображено часткову нанесення текстур, а на рисунку 3.28 зображено порівняння розпізнавання текстурованих та нетекстурованих об'єктів.



Рисунок 3.27 – Накладання текстур на елементи будівлі

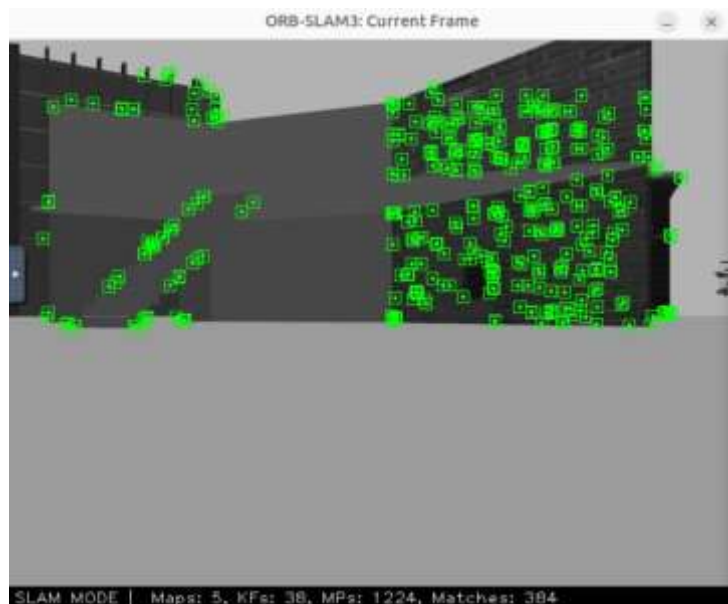


Рисунок 3.28 – Демонстрація незадовільного функціоналу системи у випадку недостатньої реалістичності візуальних даних

На рисунку 3.29 відображено межі полігонів будівлі (wireframe), де текстуровані поверхні так само плоскі, як нетекстуровані.

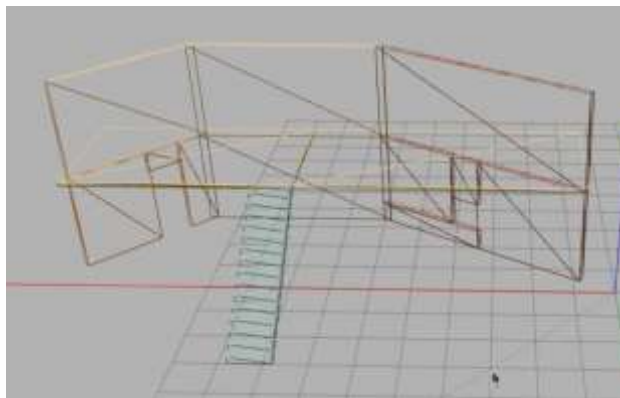


Рисунок 3.29 – Wireframe вигляд будівлі

Ідеально плоскі стіни не є фотореалістичними, та дана перевірка підтверджує, що для картографування та локалізації з візуальних даних з місцевості, місцевості потрібна лише реалістичність даних, яку можна надати простим текстуруванням.

### 3.1 Висновки до третього розділу

В даному розділі було розглянуто повну структуру графу топиків активної системи в різних рівнях представлення графу, було розроблено та оглянуто тривимірне середовище сцени симуляції та її складову, було проаналізовано особливості та обмеження процесів системи та додаткових інструментів створення, модифікації, та відображення об'єктів сцени, та було отримано чисельні дані станів параметрів системи протягом кількох етапів роботи VSLAM, що можна використати для розрахунків.

## 4 РОЗРАХУНКОВІ ДОСЛІДЖЕННЯ СИСТЕМИ

З розділу дослідження функціоналу системи у комплексному середовищі, зі зразкових кадрів доступні наступні масиви вихідних даних: кількість ключових кадрів (Total Keyframes (TK)); кількість карт (Total Maps (TM)); кількість збігів (Total Matches (TMA)); кількість зіставлених точок (Total Matched Points (TMP)).

### 4.1 Базові показники ефективності

Щільність ключових кадрів – показник щільності ключових кадрів допомагає оцінити, як часто система створює ключові кадри відносно створених карт, що свідчить про ефективність картографування:

$$KD = \frac{TK}{TM}. \quad (4.1)$$

Ефективність зіставлення – метрика ефективності зіставлення вказує на ефективність процесу підбору, показуючи, скільки з усіх спроб зіставлень були успішними:

$$ME = \left( \frac{TMA}{TMP} \right) \cdot 100. \quad (4.2)$$

Середня щільність мапи – параметр щільності мапи дає уявлення про те, скільки деталей містить кожна мапа, що може відобразити повноту представлення оточення:

$$MD = \frac{TMP}{TM}. \quad (4.3)$$

Код для проведення розрахунків та створення графіків:



```
efficiency.m

% Initial Data (from the screenshots)
Maps = [2; 3; 3; 3; 2; 1; 1; 1; 1; 1; 1; 1];
KFs = [25; 158; 207; 285; 350; 614; 655; 686; 687; 674; 673; 720];
MPs = [1934; 6445; 8029; 10764; 10736; 14799; 14825; 14704; 14093;
12996; 13479; 11785];
Matches = [409; 109; 250; 401; 113; 36; 281; 353; 314; 311; 155; 178];

% X-axis values (indices)
i = (1:length(Maps))';

% Plotting Initial Data
figure;
hold on;

% Plotting each variable
plot(i, Maps, '-o', 'DisplayName', 'Maps', 'LineWidth', 2);
plot(i, KFs, '-o', 'DisplayName', 'KFs', 'LineWidth', 2);
plot(i, MPs, '-o', 'DisplayName', 'MPs', 'LineWidth', 2);
plot(i, Matches, '-o', 'DisplayName', 'Matches', 'LineWidth', 2);

% Formatting the plot
title('Initial Data Comparison');
xlabel('Index (i)');
ylabel('Values');
legend('show');
grid on;
hold off;

% Derived Data Calculations
```

```
KD = KFs ./ Maps;
ME = (Matches ./ MPs) * 100;
MD = MPs ./ Maps;
% Plotting Derived Data
figure;

% KD Plot
subplot(3,1,1);
plot(i, KD, '-o', 'DisplayName', 'KD (KFs/Maps)', 'LineWidth', 2);
title('KD Calculation');
xlabel('Index (i)');
ylabel('KD Value');
grid on;

% ME Plot
subplot(3,1,2);
plot(i, ME, '-o', 'DisplayName', 'ME (Matches/MPs *100)', 'LineWidth', 2);
title('ME Calculation');
xlabel('Index (i)');
ylabel('ME Value (%)');
grid on;

% MD Plot
subplot(3,1,3);
plot(i, MD, '-o', 'DisplayName', 'MD (MPs/Maps)', 'LineWidth', 2);
title('MD Calculation');
xlabel('Index (i)');
ylabel('MD Value');
grid on;
```

% Adjust layout

sgtitle('Derived Data Calculations');

На рисунках 4.1-4.2 відображено графіки величини початкових даних в залежності від прогресу виконання VSLAM функцій на новій для робота місцевості (прогрес умовно відображений дискретизацією у кадри).



Рисунок 4.1 – Графік карт, ключових кадрів, збігів

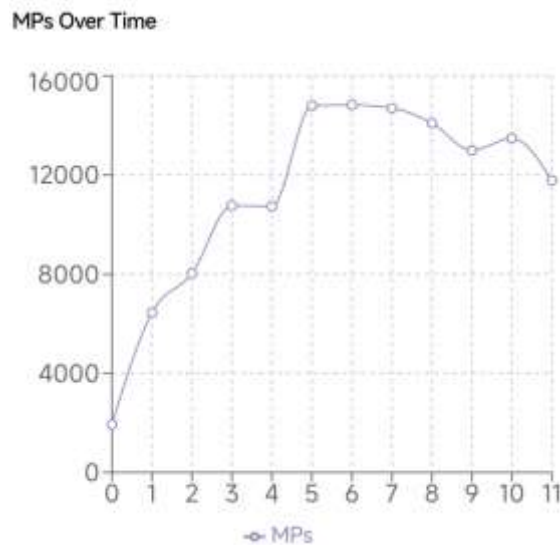


Рисунок 4.2 – Графік точок збіжностей

На рисунках 4.3-4.4 відображено графіки розрахованих величин в залежності від прогресу виконання VSLAM функцій на новій для робота місцевості.

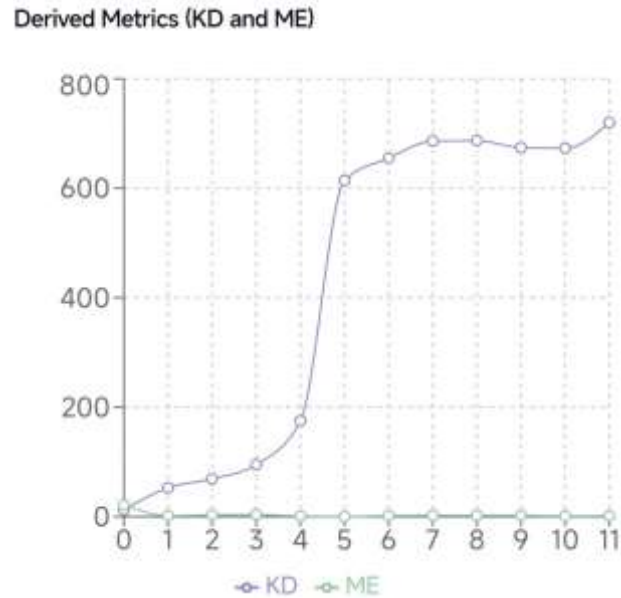


Рисунок 4.3 – Графік щільності ключових кадрів та ефективності знаходження збігів

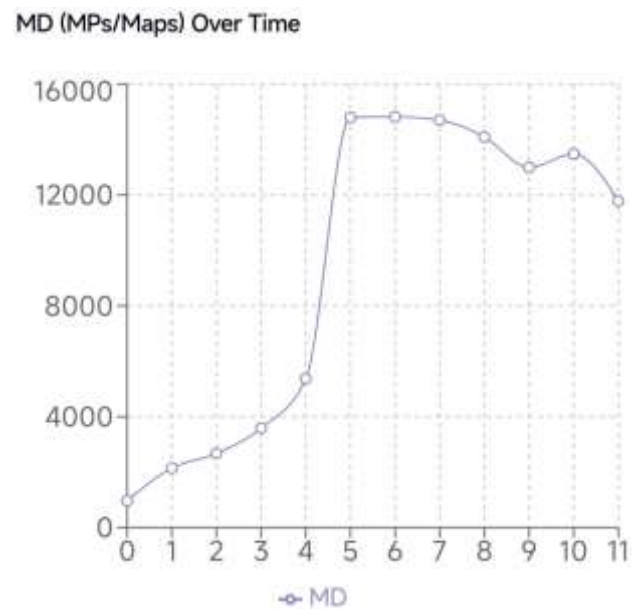


Рисунок 4.4 – Графік щільності мапи

## 4.2 Комп'ютерне моделювання системи автоматичного управління

Для того, щоб розглянути VSLAM систему цієї роботи як систему автоматичного управління, було змодельовано цю систему, як таку, що:

- є інтегрованою у мобільного робота;
- керує системою рушія робота;
- використовується для задачі позиціонування робота на відстані 1 метр від такої розпізненої ознаки, що відповідає особливостям референтної ознаки (тобто співпадає з такою ознакою, яку потрібно розпізнати).

Проведемо аналіз стабільності та точності функцій відстеження ознак такої моделі.

Спочатку виявимо ті формули та рівняння, що будуть для цього використані.

ПД-регулятор для візуального відстеження функцій:

$$C(s) = K_p + \frac{K_i}{s} + K_d \cdot s, \quad (4.4)$$

де  $K_p$  – посилення помилки позиції (пікселі для контрольного сигналу);

$K_i$  – накопичений приріст корекції дрейфу;

$K_d$  – посилення передбачення руху функції.

$$K_p = \frac{1}{2 \cdot \text{focal length} \cdot \text{baseline}}, \quad (4.5)$$

де  $\text{focal length}$  – фокусна відстань камери в пікселях;

$\text{baseline}$  – відстань між камерою та центром робота;

$$K_i = \frac{K_p}{8 \cdot \text{frame rate}}, \quad (4.6)$$

де  $\text{frame\_rate}$  – частота кадрів камери в Гц;

$$K_d = K_p \cdot \text{predict horizon}, \quad (4.7)$$

де `predict_horizon`: час передбачення функції (зазвичай 2-3 кадри).

У моделі, з урахуванням вхідних даних  $e(t)$ , формула (4.4) виглядає так:

$$C(s) = \left( K_p \cdot e(t) \right) + \left( K_i \cdot \int e(t) dt \right) + \left( K_d \cdot \frac{de(t)}{dt} \right). \quad (4.8)$$

Передаточна функція виявлення ознак в зображеннях Лапласа:

$$H(s) = \frac{1}{s^3 + 3s^2 + 2s + 3}. \quad (4.9)$$

Передаточна функція умовного руху робота в одному напрямку в зображеннях Лапласа:

$$G(s) = \frac{K_v}{s+1}. \quad (4.10)$$

В моделі SIMULINK:

$$G(s) = \frac{K_v \cdot u(t)}{s+1}. \quad (4.11)$$

Параметри системи, що симулюється у MATLAB:

- частота оновлення камери – 30 Гц;
- фокусна відстань камери – 500 пікселів;
- відстань між камерою та центром робота – 0,5 метрів.

Деякі з інших параметрів у коді скрипту MATLAB для задання параметрів ланок було змінено підбором після тестувань для виявлень максимальної стабільності системи.

Код .m MATLAB файлу, що задає параметри симуляції, початкові параметри, та рівняння для ланок моделі SIMULINK:

```
acs.m
```

```
% Opening the Simulink model in the same folder
```

```
mdl = 'VisualFeatureTrackingv2';
```

```
open_system(mdl);
```

```
% Set up model parameters
```

```
ts = 1/30; % Sample time (30 Hz camera)
```

```
stop_time = 60; % Simulation stop time
```

```
% System parameters
```

```
Kv = 0.01; % Visual scale factor (pixels to meters)
```

```
focal_length = 500; % Camera focal length in pixels
```

```
baseline = 0.5; % Distance between camera and robot center
```

```
frame_rate = 30; % Camera frame rate in Hz
```

```
% PID controller gains
```

```
Kp = 100/(2*focal_length*baseline);
```

```
Ki = Kp/(100*frame_rate);
```

```
Kd = Kp*100; % 2-frame prediction horizon
```

```
% Feature detector parameters
```

```
sigma_min = 10;
```

```
sigma_max = 100;
```

```
d_min = 5;
```

```
d_max = 50;
```

```
% Set blocks
```

```
set_param([mdl '/Reference'], 'Time', '0', 'After', '1');
```

```

% PID Controller
set_param([mdl '/PID'], ...
    'P', num2str(Kp), ...
    'I', num2str(Ki), ...
    'D', num2str(Kd));

% Coefficients for the feature detection transfer function
a0 = 1;
a1 = 3;
a2 = 2;
a3 = 3;
set_param([mdl '/FeatureDetector'], ...
    'Numerator', '[1]', ...
    'Denominator', ['1' num2str(a0) ' ' num2str(a1) ' ' num2str(a2) ' ' num2str(a3)
    ']);

% Robot Motion Response
set_param([mdl '/RobotResponse'], ...
    'Numerator', '[Kv]', ...
    'Denominator', '[1 1]'); % Simple first-order response

% Visual Processing Delay
set_param([mdl '/ProcessingDelay'], 'DelayTime', '0.04'); % 40ms delay

% Feature Matching Error
set_param([mdl '/ErrorCalc'], 'Inputs', '++-');

% Measurement Noise
set_param([mdl '/MeasurementNoise'], ...
    'Cov', '0.1', ...
    'Ts', num2str(ts));

% Configure simulation parameters

```



```

set_param mdl, ...
'Solver', 'ode45', ...
'StopTime', num2str(stop_time), ...
'MaxStep', num2str(ts / 10));

```

Модель системи функцій відстеження ознак у Simulink та графіки ключових параметрів зображено на рисунках 4.5-4.8.

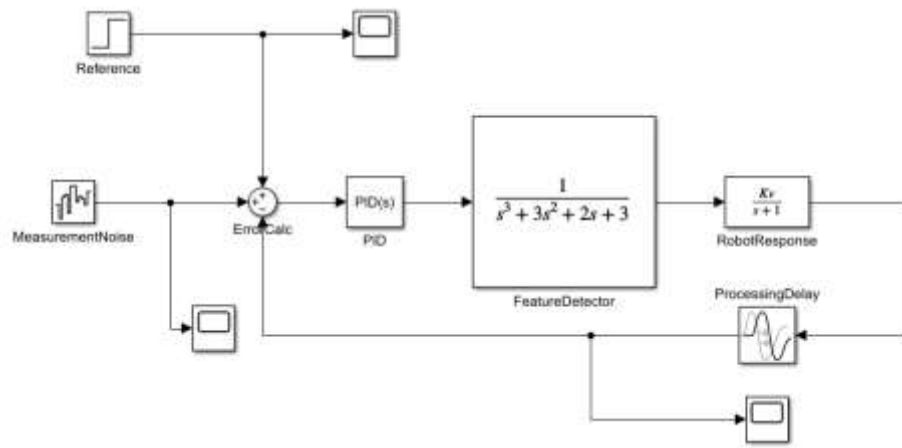


Рисунок 4.5 – Модель системи функцій відстеження ознак у Simulink



Рисунок 4.6 – Референтне значення

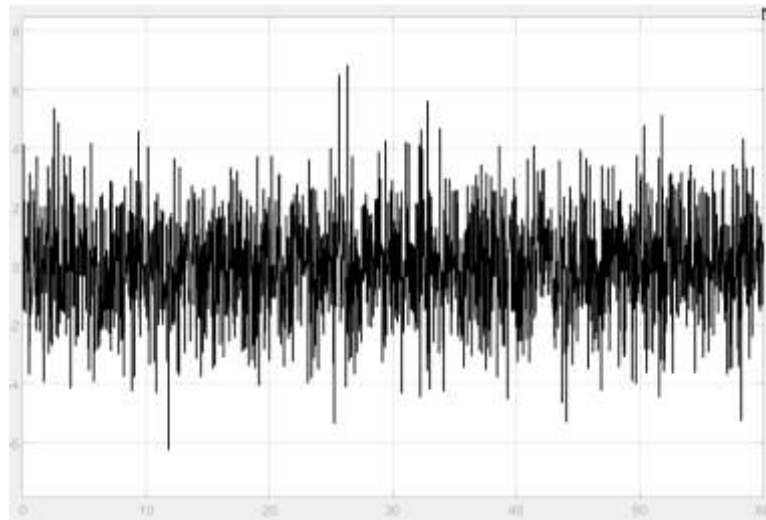


Рисунок 4.7 – Шум як симуляція похибок

Похибка між опорним положенням елемента та фактично виявленим положенням елемента походить від блоку «ErrorCalc», який обчислює різницю між: еталонним входом під шумом й фактичним положенням після проходження ланцюжка виявлення функцій і відповіді робота.

Отже, кожне коливання показує, наскільки далеко система знаходиться від бажаного положення функції.

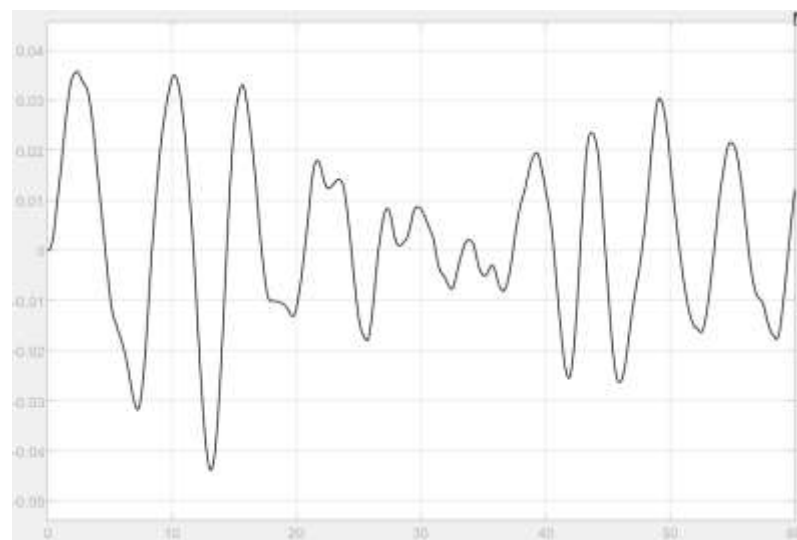


Рисунок 4.8 – Виміряна похибка між опорним положенням елемента та фактично виявленим положенням елемента

Коливання існує в межах приблизно від 0 до  $40 \cdot 10^{-5}$  метрів (0,4 міліметрів) свідчить про дуже точні позиційні налаштування.

#### 4.3 Огляд та розрахунки необхідних умов для приміщення для підготовки та використання системи

Повноцінної підготовки та експлуатація системи VSLAM, як у реальних, так й у симуляційних середовищах, потребує щонайменше наявності робочого місця з комп'ютером. Таке робоче місце має відповідати здоровим умовам праці, які встановлені конкретними правилами держави [39-40].

Мікроклімат робочого місця винний задовольняти такі параметри:

- відносна вологість повітря: допустиме значення не більше 75 % у холодну пору року та не більше 55% (при 28 °C) у теплу пору року. Оптимальним значенням є 40-60% в будь-яку пору року;
- температура повітря робочої зони: допустимі значення: 21-25 °C холодну пору року, та від 22-28 °C у теплу пору року. Оптимальні значення це: від 22-24 °C у холодну пору року та 23-25 °C у теплу пору року;
- швидкість руху повітря: допустиме значення є не більше ніж 0,1 м/с для холодної пори року, та 0,1-0,2 м/с у теплу пору року. Оптимальне значення це 0,1 м/с незалежно від пори року;
- освітлення приміщення або робочої зони, що спроектоване згідно Державною будівельною нормою України ДБН В.2.5-28:2018.

За нормами щодо освітлення робочого приміщення, в практичному результаті облаштування робочого місця, освітленість від штучного освітлення має бути не менше 300 Лк. Для створення задовільних умов освітленості має розраховуватися світловий потік, що падає на робочу поверхню, за формулою:

$$F = \frac{E * K * S * Z}{\eta}, \quad (4.12)$$

де  $F$  – шуканий світловий потік, Лм;

$E$  – нормована мінімальна освітленість, Лк ( $E = 300$  Лк);

$K$  – коефіцієнт запасу, для врахування зменшення світлового потоку пристрою освітлення через поступові забруднення освітлювальних приладів у процесі експлуатації ( $K = 1,5$ );

$S$  – площа освітлюваного приміщення;

$Z$  – коефіцієнт відносності середньої освітленості до мінімальної (приймається рівним в діапазоні 1,1-1,2, в даному випадку  $Z=1,1$ );

$\eta$  – коефіцієнт використання світлового потоку (табличне значення в залежності від індексу приміщення  $I$ ).

$I$  обчислюється формулою:

$$I = \frac{S}{h(A + B)}, \quad (4.13)$$

де  $S$  – площа приміщення, м<sup>2</sup>;

$h$  – висота встановлення освітлювальних приладів, м;

$A$  та  $B$  – ширина та довжина приміщення відповідно площі, м.

З врахуванням того, що налаштування та експлуатація VSLAM може бути виконана з роботом у апаратному виконанні, маємо виробниче приміщення площею 50 м<sup>2</sup>, вимірами 10 м та 6 м, висота встановлення освітлення дорівнює 3 м. При підставленні значення до формули (4.13) отримується:

$$I = \frac{50}{3 * (10 + 5)} = 1,11. \quad (4.14)$$

За значенням  $I$  виявляється значення  $\eta$  згідно довідкової таблиці у ДБН В.2.5-28:2018 [38]. Значення  $\eta$  дорівнює 0,48.

Йде підставлення всіх значення у формулу (4.12) й отримується:

$$F = \frac{300 * 1,5 * 50 * 1,1}{0,48} = 51562,5 \text{ Лм.} \quad (4.15)$$

Потрібно отримати світловий потік у 51562,5 Лм.

Світловий потік однієї світлодіодної лампи типу Т8 це 1620 Лм, тому розрахунок:

$$n = \frac{F}{F_{л}} = \frac{51562,5}{1620} = 31,8 = 32 \text{ шт.} \quad (4.16)$$

Отже, для освітлення приміщення знадобиться 32 лампи типу Т8, що дорівнює потрібній потужності освітлення у 15 Вт/м як відповідно до стандарту [39].

#### 4.4 Висновки до четвертого розділу

В даному розділі, у MATLAB було проведено розрахунки базових показників ефективності та виведення графіків для них, та у Simulink було проведено математичне моделювання системи у відображенні ланок САУ, з результатом в якості графіків. Також було проведено огляд нормативів охорони праці та розрахунки для того, щоб визначити оптимальні параметри для робочого середовища, в якому буде відбуватися експлуатація розробленої VSLAM системи.

Система дає більш ефективні результати з часом активного обслідування середовища.

Змодельована САУ має достатні підналаштування для забезпечення стабільності функціоналу завдяки роботі ПІД (пропорційно-інтегрально-диференціального) регулятора.

## ВИСНОВКИ

За результатами виконання роботи була розглянута предметна область візуальної одночасної локалізації та картографування. Було проаналізовано принципи та технології VSLAM у робототехніці. Було досліджено наявні принципи імплементації VSLAM у робототехнічні проєкти.

Були виконані такі завдання:

- аналіз сучасних універсальних 3D-симуляторів роботів;
- аналіз сучасності використання комп'ютерного зору у симуляторах робототехнічних систем;
- аналіз сучасних візуальних SLAM систем в робототехніці та аналогів їх імплементацій;
- вибір та розробка програмних компонентів системи;
- розробка команд та схем підключень та взаємозв'язків системи;
- розробка моделювання тестового робота та його середовища для симуляцій;
- розробка імплементації керування роботом та втілення VSLAM системи;
- дослідження функціоналу системи;
- розрахунки ефективності та комп'ютерного моделювання системи;
- виявлення оптимальних умов робочого місця експлуатації системи.

У результаті було досягнуто основну мету роботи: розроблено систему-проєкт для виробництва та подальших досліджень роботів, що використовують симуляційні середовища та методи VSLAM.

Результати можна віднести до Цілі сталого розвитку 9, п. 9.3 “Забезпечити доступність дорожньотранспортної інфраструктури, яка базується на використанні інноваційних технологій, зокрема через розширення форм участі держави у різних інфраструктурних проєктах” та 9.4

“Сприяти прискореному розвитку високо- та середньовисокотехнологічних секторів переробної промисловості, які формуються на основі використання ланцюгів «освіта – наука – виробництво» та кластерного підходу за напрямками: розвиток інноваційної екосистеми”. А саме це стосується індикаторів:

– №9.3.2 "Частка доріг загального користування державного значення з твердим покриттям, що відповідають нормативним вимогам, %", бо візуальні одночасні картографування та локалізація допоможуть аналізувати форми поверхнів доріг для знаходження невідповідностей інфраструктурним стандартам;

– №9.4.1 "Частка доданої вартості за витратами виробництва підприємств, які належать до високотехнологічного сектору переробної промисловості (зокрема, з виробництва фармацевтичних продуктів і препаратів; комп'ютерів, електронної та оптичної продукції; повітряних і космічних літальних апаратів, супутнього устаткування відповідно до КВЕД) у загальній доданій вартості за витратами виробництва, %" та №9.4.2 "Частка доданої вартості за витратами виробництва підприємств, які належать до середньовисокотехнологічного сектору переробної промисловості (зокрема, виробництва хімічної продукції; електричного устаткування; машин та устаткування; автотранспортних засобів, причепів і напівпричепів; інших транспортних засобів відповідно до КВЕД), у загальній доданій вартості за витратами виробництва, %", бо розвитку технології VSLAM необхідні для поліпшень та інновацій процесів у таких промислових секторах, де суцільні швидкі та дешеві дослідження об'ємних характеристик середовищ, що використовуються для промислових цілей – тобто діагностика важкодоступних місць на несправності чи дефекти, картографування місцевості для аналізу перед роботи з нею, автономні саморегулювання трафіку та здолення перешкод між автономними промисловими роботами на семістатичній місцевості тощо.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. Введ. 2015-06-22. К.: Держстандарт України, 2017. 29 с.
2. Положення про організацію проведення практики здобувачів вищої освіти Харківського національного університету радіоелектроніки [Електронний ресурс]: Наказ ХНУРЕ від 03 травня 2019 р. № 222. – URL: <https://nure.ua/wp-content/uploads/222-vid-03.05.2019-pro-vvedennja-vdiju-rishennja-vchenoi-radi-universite-tu.pdf> (дата звернення: 8.10.2024).
3. Положення про кваліфікаційну роботу здобувача вищої освіти на другому (магістерському) рівні [Електронний ресурс]: Наказ ХНУРЕ від 06 травня 2021 р. № 143. – URL: [https://nure.ua/wp-content/uploads/Main\\_Docs\\_NURE/143-vid-06.05.2021-pro-vvedennja-v-diju-rishennja-vchenoiradi-universitetu.pdf](https://nure.ua/wp-content/uploads/Main_Docs_NURE/143-vid-06.05.2021-pro-vvedennja-v-diju-rishennja-vchenoiradi-universitetu.pdf) (дата звернення: 8.10.2024).
4. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології, освітньо-професійних програм: «Автоматизоване управління технологічними процесами», «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи» / Упоряд. І. Ш. Невлюдов, Р. В. Артюх, В. В. Безкоровайний, Н. П. Демська, В. В. Євсєєв, О. І. Филипенко, О. М. Цимбал. – Харків: ХНУРЕ, 2021. – 55 с.
5. Положення про академічну доброчесність. Наказ ХНУРЕ від 02 лютого 2021 р. № 50. URL: [https://nure.ua/wp-content/uploads/Main\\_Docs\\_NURE/polozhennja-pro-akademichnu-dobrochesnist.pdf](https://nure.ua/wp-content/uploads/Main_Docs_NURE/polozhennja-pro-akademichnu-dobrochesnist.pdf) (дата звернення: 8.10.2024).
6. Методичні вказівки до виконання розділу «Охорона праці та безпека в надзвичайних ситуаціях» в атестаційних роботах окр «Магістр» /



упоряд. : Б. В. Дзюндзюк, В. А. Айвазов, Т. Є. Стиценко ; МОН України, ХНУРЕ. – Харків : ХНУРЕ, 2013. – 52 с.

7. Поддубняк ІА Аналіз сучасних універсальних 3D-симуляторів роботів/ІА Поддубняк//Комп'ютерно-інтегровані технології автоматизації технологічних процесів на транспорті та у виробництві: матеріали всеукр. наук.-практ. конф. здобувачів вищої освіти і молодих учених, 22 листоп. 2023 р./Харків. нац. автомоб.-дор. ун-т.—Харків, 2023.—С. 68-72.

8. Поддубняк І. А. Аналіз комп'ютерного зору в сучасних симуляторах роботів / І. А. Поддубняк, О. М. Цимбал // Автоматизація, електроніка та робототехніка. Стратегії розвитку та інноваційні технології (АЕРТ'2023): матеріали V форуму, 29–30 листопада 2023 р. – Харків: ХНУРЕ, 2023. – С. 36-39.

9. Поддубняк І. А. Аналіз сучасних візуальних SLAM систем в робототехніці / І. А. Поддубняк // Автоматизація та Приладобудування = Automation and Development of Electronic Devices (ADED'2023): збірник студентських наукових статей, 2023. – Харків: ХНУРЕ, 2023. – Вип. 2. – С. 286-291.

10. Liu J., Borja P., Della Santina C., “Physics-informed Neural Networks to Model and Control Robots: a Theoretical and Experimental Investigation,” arXiv, 2023.

11. Blanco-Claraco J.-L., Tymchenko B., Mañas-Alvarez F. J., Cañadas-Aránega F., López-Gázquez Á., Moreno J. C., “MultiVehicle Simulator (MVSIM): Lightweight dynamics simulator for multiagents and mobile robotics research,” SoftwareX, vol. 23, pp. 101443–101443, 2023.

12. Farley A., Wang J., Marshall J. A., “How to pick a mobile robot simulator: A quantitative comparison of CoppeliaSim, Gazebo, MORSE and Webots with a focus on accuracy of motion,” Simulation Modelling Practice and Theory, vol. 120, pp. 102629–102629, 2022.

13. Phadke A., Medrano F. A., Sekharan C. N., Chu T., “Designing UAV Swarm Experiments: A Simulator Selection and Experiment Design Process,” *Sensors*, vol. 23, no. 17, p. 7359, 2023.
14. Audonnet F. P., Hamilton A., Aragon-Camarasa G., “A Systematic Comparison of Simulation Software for Robotic Arm Manipulation using ROS2,” 2022 22nd International Conference on Control, Automation and Systems (ICCAS). IEEE Xplore, 2022.
15. Elmquist A. et al., “A software toolkit and hardware platform for investigating and comparing robot autonomy algorithms in simulation and reality,” arXiv (Cornell University), 2022.
16. Chen Z., Yan J., Ma B., Shi K., Yu Q., Yuan W., “A Survey on Open-Source Simulation Platforms for Multi-Copter UAV Swarms,” *Robotics*, vol. 12, no. 2, pp. 53–53, 2023.
17. Körber M., Lange J., Rediske S., Steinmann S., Glück R., “Comparing Popular Simulation Environments in the Scope of Robotics and Reinforcement Learning,” arXiv:2103.04616 [cs], 2021.
18. Audonnet F. P., Hamilton A., Aragon-Camarasa G., “A Systematic Comparison of Simulation Software for Robotic Arm Manipulation using ROS2,” 2022 22nd International Conference on Control, Automation and Systems (ICCAS). IEEE Xplore, 2022.
19. Wang Y. and Kasaei H., “IPPO: Obstacle Avoidance for Robotic Manipulators in Joint Space via Improved Proximal Policy Optimization,” arXiv (Cornell University), 2022.
20. M. T. Shahria, M. S. H. Sunny, M. I. I. Zarif, J. Ghommam, S. I. Ahamed, and M. H. Rahman, “A Comprehensive Review of Vision-Based Robotic Applications: Current State, Components, Approaches, Barriers, and Potential Solutions,” *Robotics*, vol. 11, no. 6, p. 139, Dec. 2022, doi: <https://doi.org/10.3390/robotics11060139>.

21. A. Elmquist, R. Serban, and D. Negrut, "Camera simulation for robot simulation: how important are various camera model components?," arXiv (Cornell University), Nov. 2022, doi: <https://doi.org/10.48550/arxiv.2211.08599>.
22. A. Elmquist, R. Serban, and D. Negrut, "A Sensor Simulation Framework for Training and Testing Robots and Autonomous Vehicles," *Journal of Autonomous Vehicles and Systems*, vol. 1, no. 2, Feb. 2021, doi: <https://doi.org/10.1115/1.4050080>.
23. R. B. Sousa, H. M. Sobreira, and A. Moreira, "A Systematic Literature Review on Long-Term Localization and Mapping for Mobile Robots," Authorea (Authorea), Nov. 2022, doi: <https://doi.org/10.22541/au.166739295.55264285/v1>.
24. M. F. Ahmed, K. Masood, V. Fremont, and I. Fantoni, "Active SLAM: A Review on Last Decade," *Sensors*, vol. 23, no. 19, p. 8097, Jan. 2023, doi: <https://doi.org/10.3390/s23198097>.
25. A. Tourani, H. Bavle, J. L. Sanchez-Lopez, and H. Voos, "Visual SLAM: What Are the Current Trends and What to Expect?," *Sensors*, vol. 22, no. 23, p. 9297, Nov. 2022, doi: <https://doi.org/10.3390/s22239297>.
26. J. A. Placed et al., "A Survey on Active Simultaneous Localization and Mapping: State of the Art and New Frontiers," arXiv (Cornell University), Jul. 2022, doi: <https://doi.org/10.48550/arxiv.2207.00254>.
27. W. Chen et al., "Overview of Multi-Robot Collaborative SLAM from the Perspective of Data Fusion," *Machines*, vol. 11, no. 6, pp. 653–653, Jun. 2023, doi: <https://doi.org/10.3390/machines11060653>.
28. C. Campos, R. Elvira, J. J. G. Rodriguez, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1–17, 2021, doi: <https://doi.org/10.1109/tro.2021.3075644>.
29. Q. Zang, K. Zhang, L. Wang, and L. Wu, "An Adaptive ORB-SLAM3 System for Outdoor Dynamic Environments," *Sensors*, vol. 23, no. 3, p. 1359, Jan. 2023, doi: <https://doi.org/10.3390/s23031359>.

30. Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel and Juan D. Tardós, ORB-SLAM3: Anurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM, *IEEE Transactions on Robotics* 37(6):1874-1890, Dec. 2021
31. URL: [https://github.com/alsora/ros2-ORB\\_SLAM2/issues/8#issuecomment-1461570970](https://github.com/alsora/ros2-ORB_SLAM2/issues/8#issuecomment-1461570970) (дата звернення: 10.01.2025).
32. URL: [https://github.com/Soldann/MORB\\_SLAM/issues/3](https://github.com/Soldann/MORB_SLAM/issues/3) (дата звернення: 10.01.2025).
33. URL: [https://github.com/UZ-SLAMLab/ORB\\_SLAM3/issues/387](https://github.com/UZ-SLAMLab/ORB_SLAM3/issues/387) (дата звернення: 10.01.2025).
34. URL: [https://github.com/UZ-SLAMLab/ORB\\_SLAM3/issues/566](https://github.com/UZ-SLAMLab/ORB_SLAM3/issues/566) (дата звернення: 10.01.2025).
35. URL: [https://github.com/zang09/ORB\\_SLAM3\\_ROS/issues/2#issuecomment-1251850857](https://github.com/zang09/ORB_SLAM3_ROS/issues/2#issuecomment-1251850857) (дата звернення: 10.01.2025).
36. URL: [https://github.com/zang09/ORB\\_SLAM3\\_ROS2](https://github.com/zang09/ORB_SLAM3_ROS2) (дата звернення: 10.01.2025).
37. Collins J., Chand S., Vanderkop A., Howard D., “A Review of Physics Simulators for Robotic Applications,” *IEEE Access*, vol. 9, pp. 51416–51431, 2021.
38. URL: [https://github.com/UZ-SLAMLab/ORB\\_SLAM3/issues/277#issuecomment-1916007795](https://github.com/UZ-SLAMLab/ORB_SLAM3/issues/277#issuecomment-1916007795) (дата звернення: 10.01.2025).
39. Закон України «Про охорону праці» від 14 жовтня 1992 р. № 2694 – XII (зі змінами та доповненнями) / [Електронний ресурс]. – Режим доступу: <http://zakon4.rada.gov.ua/laws/show/2694-12>.
40. ДБН В.2.5-28:2018. Природне і штучне освітлення. – Вид. офіц. – [Чинний від 2019-03-01]. Київ : Мінрегіон України, 2018. – 133 с.