Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп’ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

Програмно-апаратний засіб візуальної одометрії на базі камери та обчислювального модуля
Назва теми

Рівень вищої освіти *перший (бакалаврський)*

Галузь знань *12 «Інформаційні технології»*
Шифр, назва

Спеціальність *123 «Комп’ютерна інженерія»*
Шифр, назва

Освітня програма *«Комп’ютерна інженерія та програмування»*
Назва

Шифр КвРКІ 022060.22.01.88 ПЗ

Виконав здобувач IV курсу, група КІ2-22-1 ______ Дмитро МИШКО
Підпис Ініціали, прізвище

Керівник канд. техн. наук, доцент ______ Андрій НІЧЕПОРУК
Науковий ступінь, учене звання Підпис Ініціали, прізвище

Нормоконтролер канд. фіз.-мат. наук, доц. ______ Тетяна КИСІЛЬ
Науковий ступінь, учене звання Підпис Ініціали, прізвище

До захисту допускаю:
завідувач кафедри КІІС ______ Ольга ПАВЛОВА
підпис Ініціали, прізвище
«01» _червня_ 2026 р.
дата

Хмельницький 2026

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Комп'ютерної інженерії та інформаційних систем

Рівень вищої освіти перший (бакалаврський)

Галузь знань 12 Інформаційні технології

Спеціальність 123 Комп'ютерна інженерія

Освітня програма «Комп'ютерна інженерія та програмування»

ЗАТВЕРДЖУЮ
Завідувачка кафедри КІІС
________ Ольга ПАВЛОВА

" 10 " 01 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Мишку Дмитру Вадимовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмно-апаратний засіб візуальної одометрії на базі камери та обчислювального модуля

Керівник проекту (роботи) Нічепорук Андрій Олександрович, к.т.н., доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 20.01.2026 р. № 7

2. Термін подання здобувачем роботи на кафедру 01.06.2026 р.

3. Вихідні дані до роботи Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Аналіз предметної області візуально-інерціальної одометрії

Проєктування програмно-апаратного засобу візуальної одометрії

Реалізація та тестування програмно-апаратного засобу

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Структурна схема програмно-апаратного засобу

Алгоритми функціонування програмно-апаратного засобу

Архітектура ПЗ програмно-апаратного засобу

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «_10_»________01__________2026 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2026	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2026	виконано
3	Робота над розділом 1 – аналіз предметної області візуально-інерціальної одометрії	01.03.2026	виконано
4	Робота над розділом 2 – проєктування програмно-апаратного засобу візуальної одометрії	01.04.2026	виконано
5	Робота над розділом 3 – реалізація та тестування програмно-апаратного засобу	29.04.2026	виконано
6	Оформлення пояснювальної записки згідно вимог	24.05.2026	виконано
7	Попередній захист ВКР	25.05.2026	виконано
8	Захист ВКР на засіданні ЕК	Червень 2026 року	

Здобувач____________ Дмитро МИШКО
Підпис Імя, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи____________ Андрій НІЧЕПОРУК
Підпис Імя, ПРІЗВИЩЕ

№ рядка	формат	Позначення	Найменування	Кіл. листів	№ екз	Примітка
			<u>Текстові документи</u>			
1		КвРКІ 022060.22.01.88 ПЗ	Пояснювальна записка	60		
			<u>Графічні матеріали</u>			
2		КвРКІ 022060.22.01.88Е8	Структурна схема	1		
			програмно-апаратного			
			засобу			
3		КвРКІ 022060.22.01.88 Е8	Алгоритми функціонування	1		
			програмно-апаратного			
			засобу			
4		КвРКІ 022060.22.01.88 Е8	Архітектура ПЗ	1		
			програмно-апаратного			
			засобу			

Зм	Арк	№ докум	Підпис	Дата	КвРКІ 022060.22.01.88 ВП			
Розробив		Мишко		28.05.26	Відомість проекту	Літера	Аркуш	Аркушів
Перевір.		Нічепорук		28.05.26		У	1	1
						ХНУ, КІ2-22-1		
Н. контр.		Кисіль		28.05				
Затв.		Павлова		28.05				

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмно-апаратний засіб візуальної одометрії на базі камери та обчислювального модуля».

Автор роботи: Дмитро МИШКО.

Керівник роботи: Андрій НІЧЕПОРУК.

Пояснювальна записка: 62 с., 16 рис., 10 табл., 3 дод., 60 джерел.

Графічна частина: 3 креслення.

IMU, IMX219, VIO, KALIBR, OPENVINS, ORANGE PI 5 PRO, ROS2, ВІЗУАЛЬНА ОДОМЕТРІЯ.

Кваліфікаційна робота бакалавра присвячена розробленню та дослідженню програмно-апаратного засобу візуальної одометрії на базі камери, інерціального модуля та обчислювальної платформи Orange Pi 5 Pro. У роботі розглянуто теоретичні основи візуально-інерціальної одометрії, проєктування апаратної та програмної частини, а також практичне калібрування сенсорів засобами Kalibr.

Метою роботи є визначення умов створення програмно-апаратного засобу візуальної одометрії, налаштування середовища ROS2 Humble та OpenVINS, отримання потоків камери й IMU, а також перевірка можливості практичного калібрування системи за допомогою Kalibr.

________________ 30.05.2026

Підпис здобувача Дата

ЗМІСТ

Вступ 4

1 Аналіз предметної області візуально-інерціальної одометрії 6

1.1 Одометрія як задача оцінювання руху автономної системи 6

1.2 Геометричні основи візуальної одометрії 7

1.3 Монокулярна візуальна одометрія та проблема масштабу 9

1.4 Інерціальні вимірювання та модель похибок IMU 10

1.5 Принципи візуально-інерціального об'єднання даних 11

1.6 Фільтраційні та оптимізаційні підходи у VIO 13

1.7 Огляд програмних засобів для візуально-інерціальної одометрії 15

1.8 Висновки до першого розділу 16

2 Проєктування програмно-апаратного засобу візуальної одометрії 17

2.1 Загальна постановка проєктної задачі 17

2.2 Функціональні, часові, обчислювальні та калібрувальні вимоги 18

2.3 Обчислювальна платформа Orange Pi 5 Pro 27

2.4 Камерний модуль IMX219: параметри, обмеження та вибір одного каналу 30

2.5 Інерціальний модуль ICM20948 і вимоги до IMU-потоку 32

2.6 Програмна архітектура ROS2 і передача сенсорних даних 34

2.7 Конфігурація OpenVINS і логіка роботи фільтра 36

2.8 Калібрування камери та IMU засобами Kalibr 38

2.9 Висновки до розділу 2 40

3 Реалізація та тестування програмно-апаратного засобу 42

3.1 Загальна логіка практичної реалізації 42

3.2 Підготовка Ubuntu 22.04 та базового середовища 43

3.3 Створення робочого простору ROS2 та підготовка OpenVINS 44

3.4 Організація ROS2-топіків і повідомлень 46

3.5 Реалізація C++-вузла захоплення кадрів 47

3.6 Отримання та публікація IMU-даних ICM20948 49

3.7 Практична методика калібрування через Kalibr 50
3.8 Налаштування конфігурації OpenVINS 56
3.9 Live-тестування системи 57
3.10 Оцінка працездатності реалізованої системи 58
3.11 Методика відтворюваного експерименту 61
3.12 Висновки до розділу 3 63
Висновки 64
Перелік джерел посилань 66
Додаток А Копія креслення «Структурна схема програмно-апаратного засобу» 73
Додаток Б Копія креслення «Алгоритми функціонування програмно-апаратного засобу» 74
Додаток В Копія креслення «Архітектура ПЗ програмно-апаратного засобу» 75

ВСТУП

Актуальність дослідження. Системи технічного зору поступово стають важливою складовою мобільних роботів, автономного транспорту, промислових платформ і переносних навігаційних засобів. У таких системах часто виникає потреба оцінювати власний рух без постійної опори на GPS або зовнішні маркери. Саме тому задача візуальної та візуально-інерціальної одометрії є актуальною для сучасної комп'ютерної інженерії.

У даній роботі розглядається побудова програмно-апаратного засобу візуальної одометрії на базі камери, інерціального модуля та обчислювальної платформи. Камера дає змогу оцінювати переміщення за зміною зображення сцени, а IMU доповнює її вимірюваннями кутової швидкості та прискорення. Поєднання цих джерел даних підвищує стійкість оцінювання руху в умовах, коли один лише відеопотік не дає достатньої інформації.

Практична актуальність теми полягає в тому, що сучасні одноплатні комп'ютери, зокрема Orange Pi 5 Pro, мають достатні ресурси для первинної обробки відеопотоку та запуску алгоритмів VIO. Використання камери Waveshare IMX219-83 Stereo Camera в монокулярному режимі разом з IMU ICM20948 дає змогу побудувати компактний прототип, придатний для навчальних і дослідницьких експериментів.

Метою дипломної роботи є визначення умов та особливостей створення програмно-апаратного засобу візуальної одометрії на базі камери, інерціального модуля та обчислювального модуля Orange Pi 5 Pro, а також оцінка можливості використання OpenVINS у середовищі ROS2 Humble для обробки сенсорних даних.

Об'єктом дослідження є процес визначення переміщення мобільного пристрою за допомогою відеопотоку з камери та інерціальних вимірювань.

Предметом дослідження є програмно-апаратна реалізація засобу візуально-інерціальної одометрії на базі Orange Pi 5 Pro, камери IMX219 та IMU ICM20948 із використанням ROS2, OpenVINS і засобів калібрування Kalibr.

Для досягнення поставленої мети в роботі розглянуто теоретичні основи візуальної та візуально-інерціальної одометрії, проаналізовано апаратні компоненти системи, описано роль камери, інерціального модуля та обчислювальної платформи, а також досліджено особливості підготовки програмного середовища ROS2, OpenVINS і Kalibr.

Практичне значення роботи полягає у формуванні структури програмно-апаратного засобу, який може бути використаний як основа для подальших експериментів із візуально-інерціальною навігацією. Отримані результати, зокрема параметри калібрування камери та IMU, можуть бути використані для повторюваного запуску OpenVINS і подальшого аналізу точності траєкторії.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ВІЗУАЛЬНО-ІНЕРЦІАЛЬНОЇ ОДОМЕТРІЇ

1.1 Одометрія як задача оцінювання руху автономної системи

Одометрія у загальному розумінні є задачею визначення зміни положення та орієнтації рухомої системи за даними сенсорів, установлених на цій системі. На відміну від абсолютної навігації, де координати визначаються відносно зовнішньої системи відліку, одометрія здебільшого працює як відносний метод. Вона оцінює, як платформа перемістилася від початкового стану або від попереднього моменту часу, а потім інтегрує ці локальні переміщення у траєкторію.

У класичних наземних роботах одометрія часто пов'язана з енкодерами коліс. Оберти коліс перетворюються на пройдений шлях, після чого з урахуванням кінематики платформи визначається зміна координат. Такий метод простий і добре працює на рівній поверхні, однак він чутливий до пробуксовування, нерівностей, деформації шин, люфтів і неточності геометричних параметрів платформи. Для літальних апаратів, переносних пристроїв або мобільних роботів, які рухаються у складному середовищі, колісна одометрія або непридатна, або недостатня.

Розвиток комп'ютерного зору зробив можливим використання камери як основного джерела інформації про рух. Камера не вимірює координати безпосередньо, але фіксує зміну проєкції навколишньої сцени. Якщо між двома кадрами камера перемістилася, одні й ті самі точки сцени змінюють положення на зображенні. Ця зміна містить інформацію про відносний рух камери. Саме на такому принципі ґрунтується візуальна одометрія [1–4]. На відміну від класичних методів навігації, візуальна одометрія не потребує зовнішньої інфраструктури, такої як GPS або маяки позиціонування. Оцінка руху виконується лише на основі послідовності зображень, отриманих із камери.

Рисунок 1.1 – Загальна логіка одометричного оцінювання

Для одометрії характерне накопичення похибки. Навіть якщо кожне окреме переміщення оцінено з невеликою помилкою, після багаторазового інтегрування ці помилки додаються одна до одної, і траєкторія поступово відхиляється від реальної. У задачах SLAM це частково компенсується розпізнаванням уже відвіданих місць і оптимізацією карти, але в задачі одометрії без замикання циклів дрейф залишається природним обмеженням. Якість одометричної системи визначається не лише точністю одного вимірювання, а й здатністю довго зберігати стабільну оцінку руху.

1.2 Геометричні основи візуальної одометрії

Візуальна одометрія базується на моделі перспективної проєкції. Точка тривимірного простору, яка має координати у системі камери, проєктується на площину зображення відповідно до внутрішніх параметрів камери. До таких параметрів належать фокусні відстані по осях, координати головної точки та параметри дисторсії об'єктива. Якщо камера переміщується, координати тієї самої просторової точки у системі камери змінюються, а разом з ними змінюється і її положення на зображенні.

Типовий алгоритм візуальної одометрії починається з виділення характерних точок або візуальних ознак. Це можуть бути кути, локальні текстурні області або інші елементи зображення, які можна повторно знайти у

наступних кадрах. Далі алгоритм встановлює відповідності між ознаками у сусідніх кадрах. Якщо відповідності знайдені правильно, вони задають геометричні обмеження на рух камери. За цими обмеженнями обчислюється відносне переміщення, а потім окремі переміщення об'єднуються у траєкторію [26–28].

Рисунок 1.2 – Узагальнена послідовність обробки даних у візуальній одометрії

У практичних системах значну роль відіграє відкидання хибних відповідностей. Рухомі об'єкти у кадрі, повторювані текстури, розмиття або засвічені ділянки можуть призвести до того, що алгоритм помилково зіставить точки, які не відповідають одному й тому самому об'єкту сцени. Такі помилки спотворюють оцінку руху. Для їх зменшення застосовують геометричні перевірки, робастні методи оцінювання, обмеження на довжину треків та аналіз узгодженості руху.

Геометрична складність візуальної одометрії полягає в тому, що зображення є двовимірною проєкцією тривимірного світу. Частина інформації про глибину втрачається, а тому відновлення руху залежить від припущень про тип камери, кількість кадрів, наявність паралаксу і спосіб отримання масштабу. Стереокамери отримують масштаб через відому базову відстань між двома сенсорами, RGB-D камери безпосередньо надають глибину, а монокулярні системи потребують додаткових джерел інформації або спеціальної ініціалізації [5–8].

1.3 Монокулярна візуальна одометрія та проблема масштабу

Монокулярна візуальна одометрія використовує один відеопотік. Її перевага полягає у простоті апаратної реалізації, меншій вартості, нижчих вимогах до пропускної здатності та відсутності потреби в точному стереокалібруванні. Через це монокулярні підходи поширені у мобільній робототехніці, доповненій реальності та дослідницьких системах, де компактність сенсорного вузла має практичну цінність.

Головним обмеженням монокулярної системи є невизначеність масштабу. Якщо камера спостерігає одну й ту саму сцену з різних положень, з геометрії зображень можна відновити форму траєкторії та відносну структуру сцени, але неможливо однозначно визначити, чи камера пройшла, наприклад, десять сантиметрів біля малих об'єктів, чи один метр біля великих об'єктів. Обидва випадки можуть давати подібні проєкції, якщо масштаб сцени невідомий. Через це монокулярна VO без додаткової інформації відновлює траєкторію лише з точністю до масштабного коефіцієнта.

У практичній навігації невизначеність масштабу є критичною. Для керування мобільною платформою потрібні метричні переміщення, а не лише форма траєкторії. Одним зі способів подолання цієї проблеми є поєднання камери з IMU. Інерціальні вимірювання містять інформацію про фізичні прискорення і кутові швидкості, тому дають системі додаткову метричну основу. У сучасних монокулярних VIO-системах масштаб, гравітація, швидкість і зміщення IMU оцінюються разом зі станом камери [9–12].

У межах цієї дипломної роботи використовується саме така логіка. Камерний модуль Waveshare IMX219-83 Stereo Camera має два сенсори, однак у поточній реалізації активним є один відеоканал. Тому правильним є не опис повної стереоодометрії, а опис монокулярної візуально-інерціальноїодометрії, де роль другого джерела інформації виконує IMU ICM20948.

1.4 Інерціальні вимірювання та модель похибок IMU

Інерціальний вимірювальний модуль зазвичай містить акселерометр і гіроскоп. Акселерометр вимірює питоме прискорення, тобто суму лінійного прискорення та впливу гравітації у системі координат сенсора. Гіроскоп вимірює кутову швидкість. У багатьох 9-осьових модулях, зокрема ICM20948, також є магнітометр, але для візуально-інерціальної одометрії основне значення мають саме акселерометр і гіроскоп.

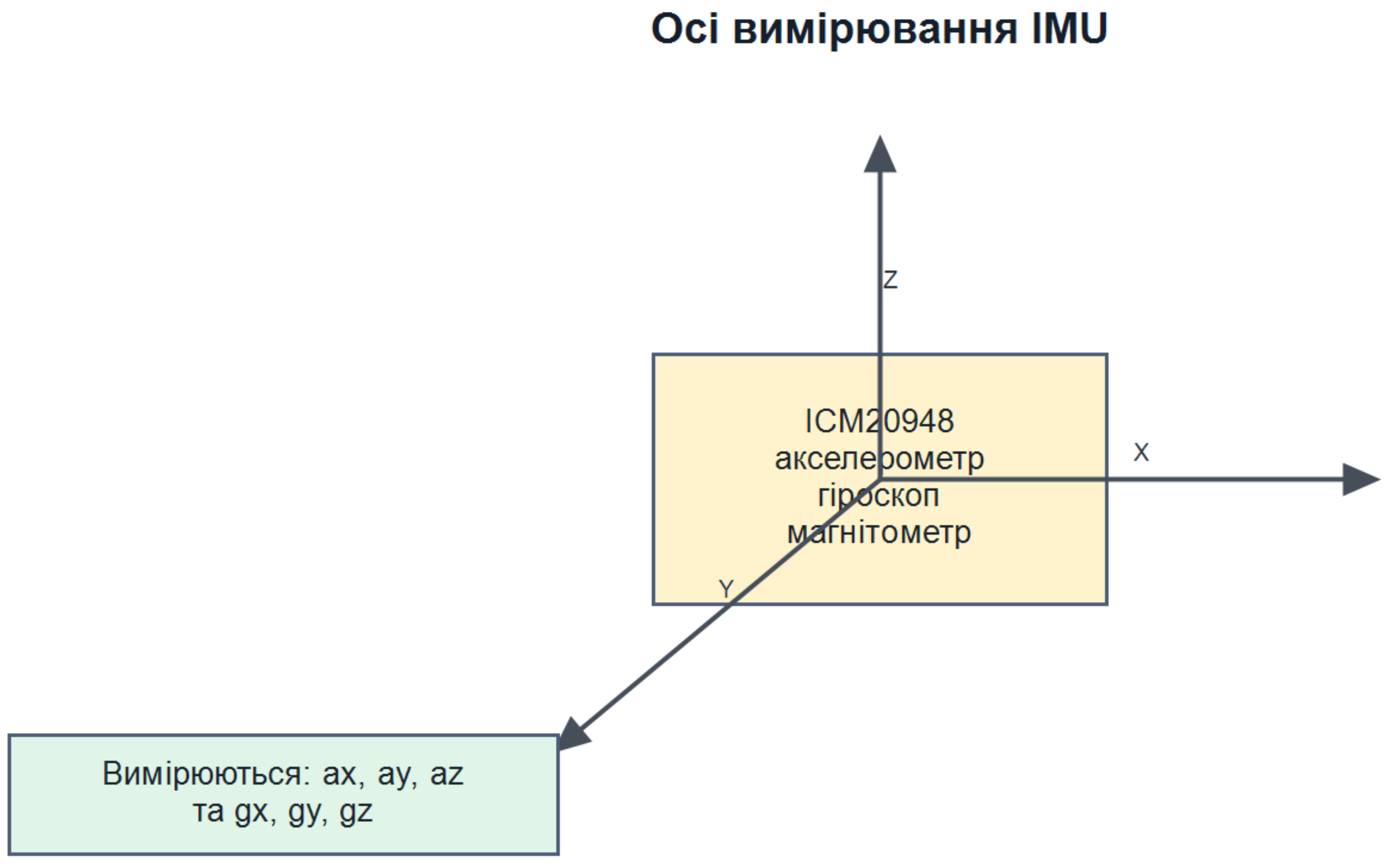


Рисунок 1.3 – Основні осі вимірювання акселерометра та гіроскопа IMU

Перевага IMU полягає у високій частоті вимірювань. Якщо камера формує кадри десятки разів на секунду, то IMU може надавати дані з частотою у сотні герц. Це дає змогу описувати швидкі повороти і короткочасні прискорення між кадрами. Водночас інерціальні вимірювання мають власні похибки. Гіроскоп має шум і повільне зміщення нуля, яке з часом призводить до дрейфу орієнтації. Акселерометр також має шум і зміщення, а після подвійного інтегрування навіть невелика похибка прискорення може швидко перетворитися на значну похибку положення.

Через це IMU рідко використовується як єдине джерело довготривалої навігації в недорогих системах. Його роль у VIO полягає не в самостійній побудові точної траєкторії, а у прогнозуванні руху між візуальними оновленнями. Камера періодично коригує накопичену інерціальну похибку, а IMU допомагає камері зберігати стійкість під час швидких рухів, тимчасової втрати ознак або недостатнього паралаксу. Таке поєднання є значно надійнішим, ніж використання кожного сенсора окремо.

Модель IMU неможливо коректно задати без шумових параметрів. Для алгоритмів на кшталт OpenVINS задають густину шуму гіроскопа й акселерометра, випадкове блукання зміщень, частоту вимірювань і часову узгодженість із камерою. Якщо ці параметри неточні, фільтр може надмірно довіряти IMU і накопичувати дрейф або, навпаки, надмірно покладатися на камеру й втрачати стійкість під час швидких рухів.

1.5 Принципи візуально-інерціального об'єднання даних

Візуально-інерціальна одометрія поєднує два різні за природою джерела даних. Камера дає вимірювання, пов'язані зі структурою навколишньої сцени, але працює з відносно низькою частотою і залежить від якості зображення. IMU, навпаки, не потребує видимих ознак і працює з високою частотою, однак швидко накопичує похибку. У поєднанні ці сенсори доповнюють один одного: IMU прогнозує стан системи, а камера коригує цей прогноз за рахунок геометричних спостережень. Такий підхід дозволяє підвищити стійкість системи до короткочасних втрат візуальної інформації, наприклад під час різких рухів або поганого освітлення. Об’єднання даних із двох сенсорів також забезпечує більш точне оцінювання положення та орієнтації пристрою в просторі. Завдяки цьому VIO широко застосовується в автономній навігації, робототехніці та безпілотних системах.

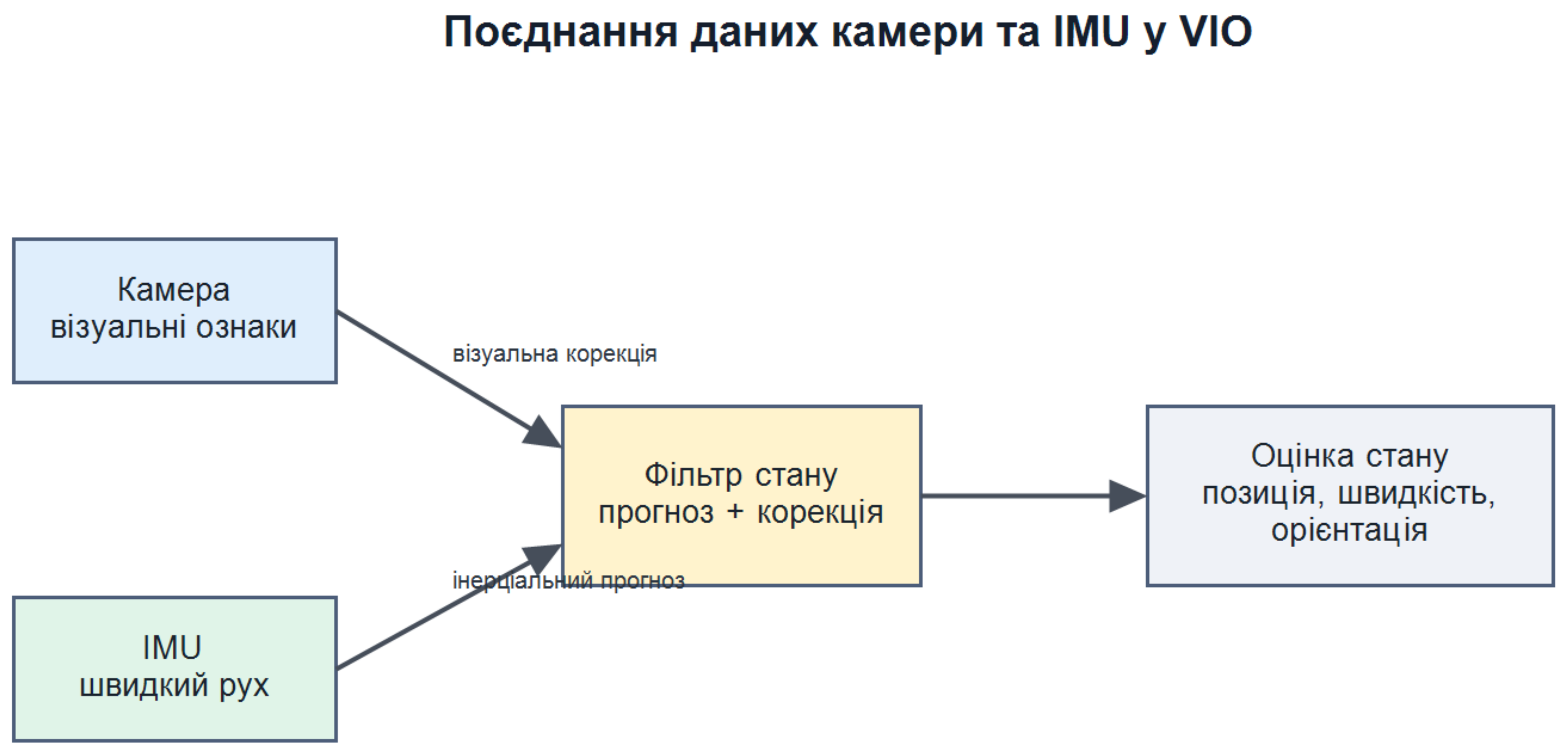


Рисунок 1.4 – Схема об'єднання даних камери та IMU у візуально-інерціальній одометрії

Стан VIO-системи зазвичай містить положення, швидкість, орієнтацію, зміщення акселерометра і гіроскопа, а також параметри, що описують зв'язок між камерою та IMU. У фільтраційних методах поточний стан прогнозується за інерціальними вимірюваннями, а після надходження нового кадру коригується за візуальними ознаками. В оптимізаційних методах формується задача мінімізації похибок вимірювань у певному часовому вікні, де одночасно враховуються інерціальні та візуальні обмеження [13–16].

Суттєвою умовою правильної роботи VIO є спостережуваність. Система повинна отримувати такі рухи та такі візуальні спостереження, які дають змогу відокремити зміну положення, зміну орієнтації, масштаб, гравітацію та зміщення IMU. Якщо платформа довго рухається майже без поворотів, спостерігає плоску однотонну сцену або має малий паралакс, оцінка може стати нестійкою. Під час тестування VIO недостатньо просто запустити алгоритм: траєкторія руху має давати сенсорам достатньо інформації.

Окремо враховують просторове і часове калібрування. Камера та IMU розташовані у різних точках і мають різні системи координат, тому алгоритм повинен знати перетворення між ними. Крім того, кадр камери і вимірювання IMU мають часові мітки, які повинні відповідати реальним моментам фізичного

вимірювання. Навіть невеликий часовий зсув може призвести до того, що візуальне спостереження й інерціальний прогноз описуватимуть трохи різні стани руху. У практичних VIO-системах калібрування не є другорядним етапом, а визначає саму можливість стабільної роботи [17; 18].

1.6 Фільтраційні та оптимізаційні підходи у VIO

У сучасній візуально-інерціальній одометрії поширені два великі класи методів: фільтраційні та оптимізаційні. Фільтраційні методи підтримують поточну оцінку стану і її невизначеність. Коли надходять нові інерціальні вимірювання, стан прогнозується вперед у часі. Коли надходять візуальні вимірювання, оцінка коригується. Такий підхід добре підходить для систем реального часу, оскільки обчислювальна складність може бути обмежена [29; 30].

Одним із відомих фільтраційних підходів є MSCKF. Його особливість полягає в тому, що у вектор стану включаються кілька попередніх поз камери, але не включаються всі тривимірні координати візуальних ознак. Ознаки використовуються для формування обмежень між позами, після чого їхні координати аналітично виключаються з оновлення. Завдяки цьому MSCKF може працювати у реальному часі без постійного зростання карти, що важливо для вбудованих обчислювальних платформ [30; 33].

Оптимізаційні методи, навпаки, формують задачу мінімізації сукупної похибки вимірювань. Вони можуть досягати високої точності, особливо якщо використовують ковзне вікно ключових кадрів і IMU-преінтеграцію. До цього класу належать, зокрема, VINS-Mono та OKVIS-подібні підходи. Їхня перевага полягає у гнучкості та точності, однак обчислювальні вимоги часто вищі, ніж у фільтраційних методів.

Таблиця 1.1 – Порівняння основних підходів до візуально-інерціального оцінювання

Підхід	Характеристика	Перевага	Обмеження
Фільтраційний VIO	Послідовне прогнозування та корекція стану	Добре підходить для реального часу	Може поступатися оптимізації за глобальною точністю
MSCKF	Використання кількох поз камери без явного зберігання карти ознак	Обмежена складність і придатність для вбудованих систем	Потребує якісної синхронізації та калібрування
Оптимізаційний VIO	Мінімізація похибки у ковзному вікні	Висока точність за достатніх ресурсів	Більше навантаження на CPU
SLAM з картою	Одночасна локалізація і побудова карти	Може зменшувати дрейф через повторні спостереження	Складніша реалізація і більші вимоги до пам'яті

У роботі вибрано OpenVINS, оскільки він реалізує дослідницьку платформу для візуально-інерціального оцінювання, підтримує MSCKF-підхід і має документацію для налаштування камер, IMU та калібрувальних параметрів. Це відповідає характеру дипломного проєкту, де основним завданням є створення і перевірка програмно-апаратного засобу, а не розроблення нового математичного методу VIO з нуля.

1.7 Огляд програмних засобів для візуально-інерціальної одометрії

OpenVINS є відкритою дослідницькою платформою для візуально-інерціального оцінювання. Вона орієнтована на роботу з камерою та IMU, підтримує різні конфігурації сенсорів і може інтегруватися з ROS-середовищем. Для цього дипломного проєкту корисною є підтримка монокулярного режиму, параметрів камери та IMU, калібрувальних файлів і оцінювання стану в реальному часі [9; 31; 34].

Сучасні програмні системи VIO та SLAM демонструють різні підходи до поєднання камер, IMU та інших сенсорів. ORB-SLAM3 і Kimera орієнтовані на ширші задачі SLAM та побудови карти, тоді як DM-VIO, DROID-SLAM і LVI-SAM показують інші напрями розвитку сучасних одометричних систем [8; 10; 12; 13; 15]. Однак у межах цієї роботи обрано OpenVINS як більш придатну дослідницьку базу для монокулярної camera-IMU конфігурації.

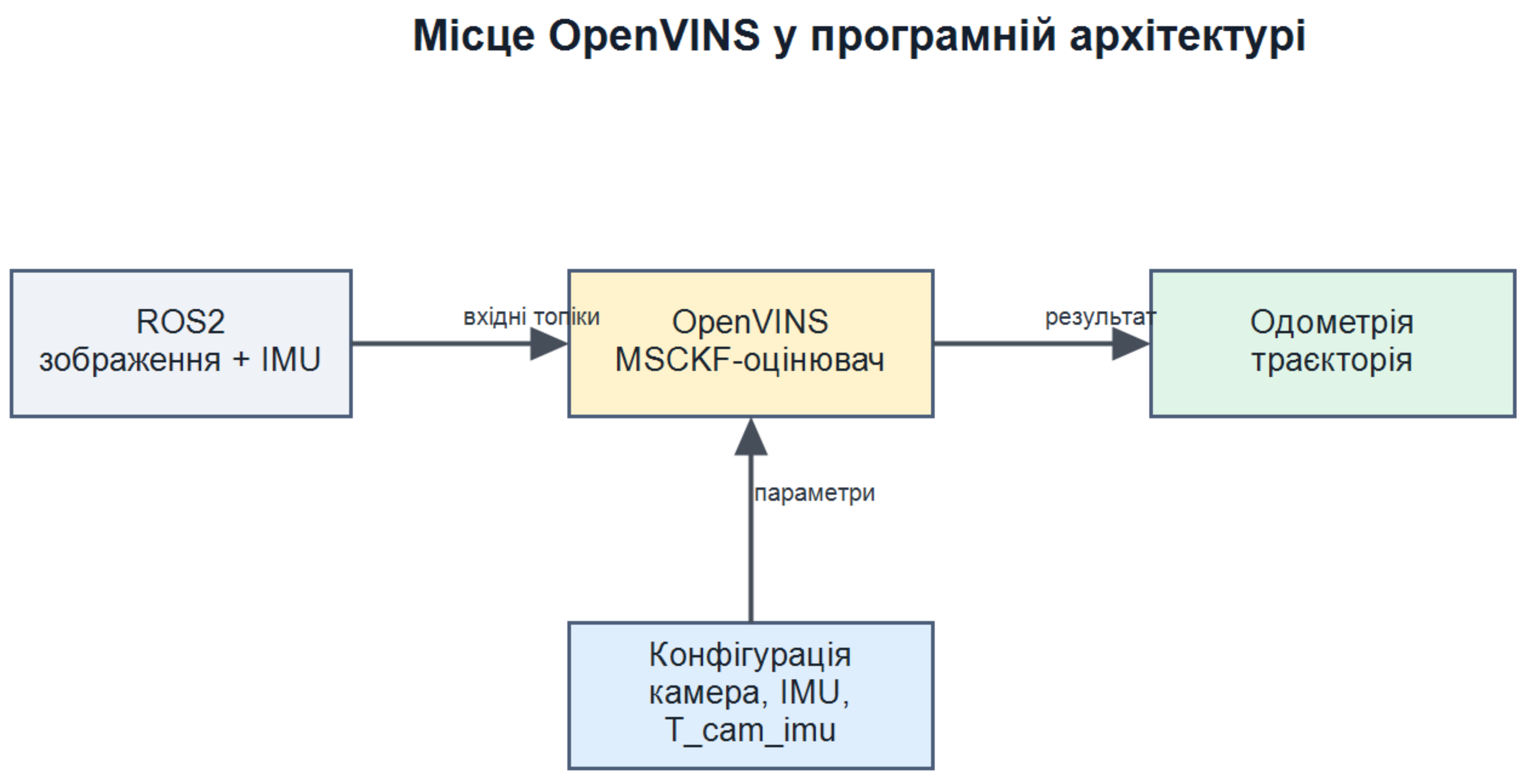


Рисунок 1.5 – Місце OpenVINS у програмній архітектурі засобу візуально-інерціальної одометрії

Вибір OpenVINS обґрунтовується не тим, що він є єдиним можливим рішенням, а тим, що його архітектура добре відповідає поставленій задачі. Система має працювати з одним відеоканалом і IMU, інтегруватися з ROS2,

використовувати відкриту документацію та давати змогу надалі проводити експерименти з калібруванням, записом даних і аналізом траєкторії. Сучасні альтернативи також охоплюють LiDAR-візуально-інерціальні, подієві та спеціалізовані робототехнічні рішення, але вони виходять за межі апаратної конфігурації цієї роботи [14; 16; 19; 20].

1.8 Висновки до першого розділу

У першому розділі розглянуто теоретичні основи одометрії як задачі відносного оцінювання руху. Показано, що візуальна одометрія використовує зміну зображення сцени для визначення переміщення камери, але її робота залежить від якості візуальних ознак, геометрії сцени та параметрів камери. Для монокулярної конфігурації основним обмеженням є невизначеність масштабу, тому для практичного застосування доцільно поєднувати камеру з IMU.

Розглянуто роль інерціальних вимірювань у VIO. IMU дає високочастотний прогноз руху, але самостійно швидко накопичує похибку. Камера, навпаки, забезпечує геометричну корекцію, однак залежить від сцени та освітлення. Поєднання цих джерел даних є основою візуально-інерціальної одометрії. Також показано, що якість VIO суттєво залежить від просторового і часового калібрування між камерою та IMU.

На основі огляду методів і програмних засобів обґрунтовано використання OpenVINS як алгоритмічної бази для подальшої реалізації. Подальший розділ присвячено проєктуванню програмно-апаратного засобу на базі Orange Pi 5 Pro, одного відеоканалу IMX219, IMU ICM20948, ROS2 Humble та OpenVINS.

2 ПРОЄКТУВАННЯ ПРОГРАМНО-АПАРАТНОГО ЗАСОБУ ВІЗУАЛЬНОЇ ОДОМЕТРІЇ

2.1 Загальна постановка проєктної задачі

Проєктування програмно-апаратного засобу візуальної одометрії в межах цієї роботи виконується для конкретної апаратної конфігурації: Orange Pi 5 Pro як обчислювальний модуль, камерний модуль Waveshare IMX219-83 Stereo Camera як джерело зображень та інерціальний модуль ICM20948 як джерело даних про кутову швидкість і лінійне прискорення. Формально камерний модуль є стереомодулем, оскільки має два сенсори IMX219, однак фактична реалізація використовує один відеоканал, позначений у матеріалах live-тестування як CAM:0. Це визначає головну методичну межу проєкту: система розглядається як монокулярна візуально-інерціальнаодометрія, а не як завершена стереоодометрична система [57; 59; 60].

Таке уточнення є принциповим не лише для формулювання тексту, а й для всієї архітектури. Стереоодометрія потребує двох одночасних зображень, стереокалібрування, точного знання базової відстані між камерами, синхронізації двох відеопотоків і відповідного режиму алгоритму. Монокулярна VIO, навпаки, працює з одним зображенням на кожному часовому кроці, а масштаб руху і короткочасна динаміка підтримуються IMU. Тому в цій роботі стереомодуль використовується як зручна апаратна плата з IMX219 та ICM20948, але програмний контур побудований навколо одного відеоканалу та інерціальних вимірювань.

Проєктна задача полягає у побудові такого конвеєра, у якому сенсорні дані проходять шлях від фізичних пристроїв до оцінки положення й орієнтації. Камера формує кадри, IMU формує інерціальні вимірювання, Orange Pi 5 Pro виконує драйверний і прикладний код, ROS2 Humble забезпечує передачу повідомлень між вузлами, а OpenVINS оцінює стан системи за візуально-інерціальними даними. У цій архітектурі кожен компонент має не ізольовану, а

системну роль: навіть продуктивна плата або якісна камера не гарантують працездатності VIO без часових міток, калібрування, правильної системи координат і узгоджених форматів повідомлень.

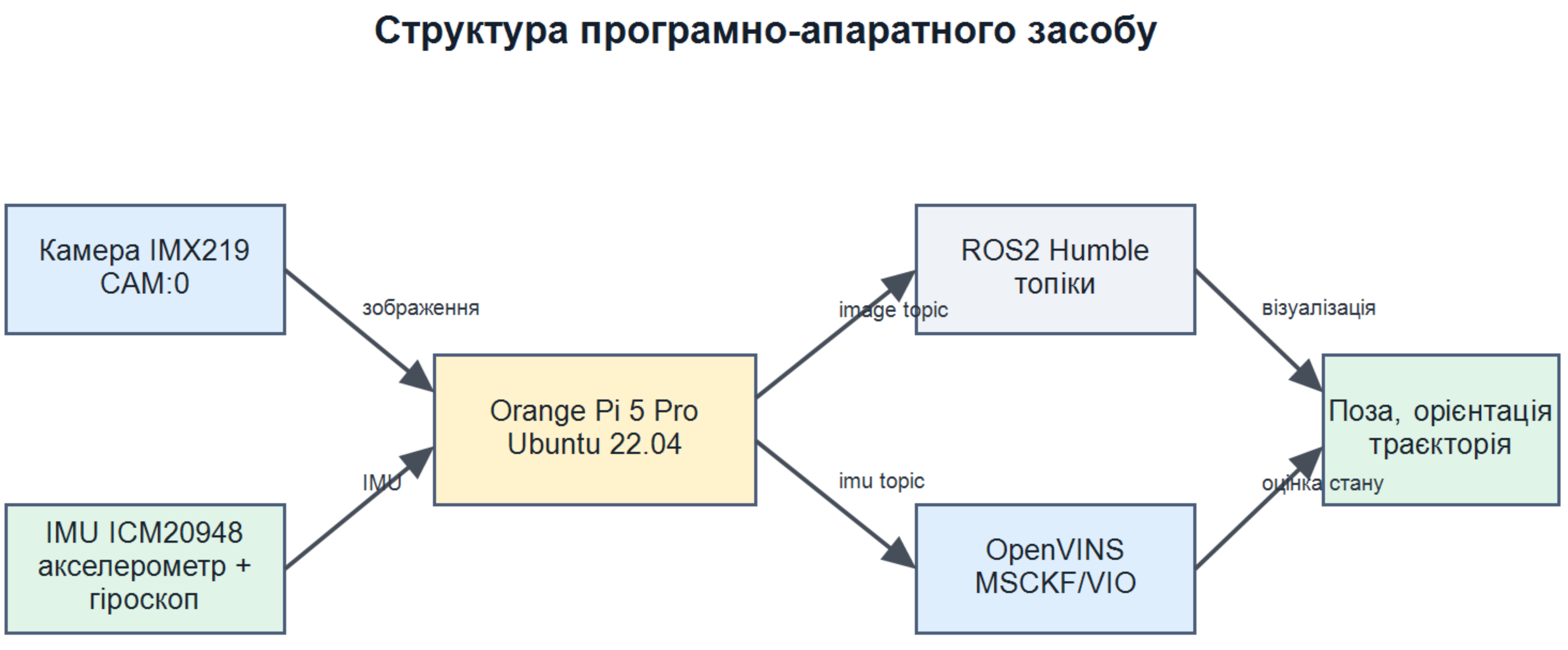


Рисунок 2.1 – Структурна схема програмно-апаратного засобу візуально-інерціальної одометрії

На рисунку 2.1 показано узагальнену структуру засобу. Камера й IMU є первинними сенсорними джерелами. Orange Pi 5 Pro виконує роль локального обчислювального вузла, на якому працює операційна система, драйвери, ROS2-вузли та OpenVINS. Вихідним результатом є оцінка стану, тобто положення, орієнтація, швидкість або траєкторія, яка може візуалізуватися у RViz-подібному середовищі або записуватися для подальшого аналізу.

2.2 Функціональні, часові, обчислювальні та калібрувальні вимоги

Вимоги до системи розглядаються не як формальний список, а як набір умов, без яких VIO не працюватиме стійко. Перша група вимог є функціональною. Система повинна отримувати зображення з одного активного каналу IMX219, зчитувати інерціальні дані ICM20948, передавати обидва потоки у ROS2 та забезпечувати їх прийом OpenVINS. Кількість камер у конфігурації

OpenVINS має відповідати фактичному режиму роботи, тобто одному відеоканалу. Якщо алгоритм очікує стереодані, а фактично отримує лише один потік, оцінювання буде або неможливим, або методично некоректним.

Друга група вимог стосується часу. У візуально-інерціальній одометрії часова узгодженість має таку саму вагу, як і самі значення вимірювань. Камера формує кадри з відносно невеликою частотою, а IMU передає дані значно частіше. OpenVINS повинен знати, які інерціальні вимірювання відповідають проміжку між двома кадрами. Якщо часові мітки створюються із запізненням, мають нерівномірний інтервал або належать різним часовим доменам, фільтр поєднуватиме дані, що описують різні фізичні моменти руху. Під час обертання навіть невеликий часовий зсув між камерою та гіроскопом може створити помітну помилку орієнтації.

Третя група вимог є обчислювальною. На відміну від офлайн-обробки записаного відео, live-система повинна встигати отримувати кадри, обробляти їх, приймати IMU, виконувати оцінювання та оновлювати візуалізацію без накопичення черги повідомлень. Тому надмірно велика роздільна здатність кадру може бути шкідливою: вона збільшує навантаження на пам'ять, шину передачі даних і CPU, але не завжди дає пропорційне покращення якості трекінгу. Для VIO часто важливіше мати стабільну частоту й коротку експозицію, ніж максимальну роздільну здатність сенсора.

Четверта група вимог пов'язана з калібруванням. Камера повинна мати визначені внутрішні параметри і модель дисторсії. IMU повинен мати задані шумові параметри, частоту і правильну орієнтацію осей. Зв'язка камера-IMU повинна мати просторове перетворення T_cam_imu та часовий зсув. Без цих параметрів OpenVINS може запускатися, але оцінка стану буде залежати від припущень, а не від виміряної геометрії сенсорного вузла. Отже, калібрування є не допоміжним етапом, а обов’язковою умовою коректної роботи всієї проектованої системи.

Таблиця 2.1 – Узагальнені вимоги до програмно-апаратного засобу

Група вимог	Сутність вимоги	Практичне значення
Часові	Узгоджені часові мітки камери та IMU	Зменшує помилки під час швидкого руху
Функціональні	Отримання одного відеопотоку IMX219 та IMU-даних ICM20948	Визначає монокулярну VIO-конфігурацію
Обчислювальні	Стабільна обробка без накопичення затримки	Дозволяє працювати у live-режимі
Калібрувальні	Параметри камери, IMU, T_cam_imu і часовий зсув	Робить оцінку стану фізично узгодженою
Відтворюваність	Збереження конфігурацій і запис сенсорних потоків	Дозволяє повторювати та аналізувати експеримент

Функціональні вимоги у цій роботі мають бути сформульовані через фактичний сценарій використання. Засіб повинен запускатися на Orange Pi 5 Pro без зовнішнього настільного комп'ютера як основного обчислювача, отримувати зображення з каналу CAM:0, приймати інерціальні дані від ICM20948 і передавати обидва потоки в ROS2. Далі OpenVINS повинен мати змогу підписатися на ці потоки і сформувати оцінку стану, яку можна візуалізувати або записати. Отже, функціональна вимога не зводиться до фрази "камера працює"; вона охоплює весь маршрут даних від сенсора до результату.

Часові вимоги фактично задають причинно-наслідковий порядок вимірювань. Кадр камери описує сцену у певний момент часу, а набір IMU-вимірювань описує рух між сусідніми кадрами. Якщо ці моменти переплутані, OpenVINS може математично коректно обробляти повідомлення, але фізично

поєднувати не ті дані. Для монокулярної VIO це особливо небезпечно, оскільки IMU допомагає підтримувати масштаб і прогноз руху. Тому в проєкті передбачають не тільки часову мітку в повідомленні, а й перевірку частоти, нерівномірності часових інтервалів, втрат кадрів і можливого сталого зсуву між камерою та IMU.

Обчислювальні вимоги пов'язані не лише з кількістю ядер процесора. Для live-VIO мають значення пропускна здатність пам'яті, швидкість копіювання кадрів, стабільність планування процесів, продуктивність накопичувача під час запису bag-файлів і теплове обмеження. Якщо система протягом перших секунд працює нормально, а через кілька хвилин починає втрачати кадри через нагрівання або черги, її не можна вважати стабільною. У розділі про реалізацію ці показники контролюються через частоту топіків і температуру, а в розділі про проєктування вони закладаються як очікувані критерії.

Калібрувальні вимоги відрізняються від функціональних тим, що їх порушення не завжди заважає запуску. Система може показати відеопотік, OpenVINS може створити траєкторію, RViz може її відобразити, але без коректних параметрів камери та IMU така траєкторія не буде вимірювально обґрунтованою. Через це в проєкті розділяються два рівні готовності: працездатний прототип і відкалібрована VIO-система. Перший рівень підтверджується live-запуском, другий потребує Kalibr, збережених YAML-файлів, звітів і повторюваного тестового набору даних.

Для підрозділу вимог важливо окремо зафіксувати часову поведінку системи. Візуально-інерціальна одометрія працює не з ізольованими значеннями, а з часовою послідовністю вимірювань, тому коректність порядку повідомлень, сталість інтервалів і передбачувана затримка мають таке саме значення, як і самі параметри камери або IMU. Якщо кадри надходять нерівномірно, а інерціальні вимірювання накопичуються пакетами, фільтр отримує формально правильні дані, але вони описують різні фізичні моменти

руху. Саме тому часові вимоги в цій роботі розглядаються як окрема частина проєктування, а не як другорядна властивість програмної реалізації.

Камерний канал повинен формувати кадри з частотою, достатньою для відстеження візуальних ознак між сусідніми зображеннями. Для навчального та експериментального стенда доцільним є режим приблизно 20-30 кадрів за секунду, оскільки він зберігає помірне навантаження на процесор і водночас не створює надто великих проміжків між візуальними оновленнями. Якщо частота кадрів знижується, між двома зображеннями зростає зміщення ознак, що ускладнює їх супровід. Якщо ж збільшувати роздільну здатність або частоту без запасу продуктивності, система може перейти від плавної роботи до накопичення затримки.

IMU-потік повинен бути частішим за відеопотік, оскільки саме інерціальні вимірювання заповнюють проміжки між кадрами. Для реалізації на Orange Pi 5 Pro практично доцільним є орієнтир близько 200 Гц для гіроскопа та акселерометра: за частоти камери близько 20 Гц це дає декілька інерціальних вимірювань на кожний міжкадровий інтервал. Важливою є не тільки середня частота, а й регулярність потоку. Два запуски з однаковою середньою частотою можуть відрізнятися якістю VIO, якщо в одному випадку інтервали між IMU-повідомленнями сталі, а в іншому виникають короткі паузи й подальші пакети вимірювань.

Допустима затримка у такій системі визначається не одним універсальним числом, а співвідношенням між швидкістю руху платформи, частотою камери, частотою IMU та способом проставлення часових міток. Для live-режиму небажаною є ситуація, коли відеопотік накопичується в черзі й OpenVINS обробляє старі кадри. У цьому випадку візуалізація може виглядати плавною, але оцінка стану фактично відставатиме від реального положення сенсорного вузла. Тому для камерного потоку доцільні невеликі черги, контроль пропущених кадрів і відмова від зайвих перетворень зображення до моменту публікації

повідомлення. Для DDS/ROS2-комунікації це безпосередньо пов’язано з політиками QoS і налаштуванням буферизації [43; 53–55].

Окрему групу становлять вимоги до живлення, охолодження та накопичувача. Нестабільне живлення може проявлятися не лише перезавантаженням плати, а й короткими збоями периферії, помилками камерного інтерфейсу або нерівномірною роботою накопичувача. Перегрів процесора також небезпечний: throttling знижує частоту обчислень і робить обробку кадрів менш передбачуваною. Накопичувач впливає на експеримент тоді, коли паралельно з live-роботою виконується запис rosbag2. Якщо швидкість запису нестабільна, можуть виникати пропуски повідомлень або нерівномірні часові інтервали, що знижує повторюваність експерименту.

Чутливість VIO до нестабільних потоків пояснюється тим, що алгоритм поєднує два сенсори з різною фізичною природою. Камера дає геометричну інформацію про сцену, але з нижчою частотою і залежністю від освітлення. IMU дає високочастотний прогноз, але швидко накопичує похибку при інтегруванні шуму та зміщень. Якщо один потік надходить із затримкою або втрачає частину повідомлень, другий потік не завжди може повністю компенсувати проблему. У результаті помилка проявляється не як проста втрата кадру, а як дрейф масштабу, нестійка орієнтація або різкі стрибки траєкторії. Тому стабільність потоків є одним із головних критеріїв готовності системи до експерименту [11; 51].

Порівняння варіантів апаратних компонентів подано в таблиці 2.2. Воно має проєктний характер: метою є не вибір абсолютно найпотужнішого обладнання, а визначення такого набору компонентів, який відповідає задачі монокулярної VIO, доступний для реалізації та не створює надмірної складності під час інтеграції. Вибрані компоненти мають забезпечувати достатній запас продуктивності для обробки зображень у реальному часі, синхронного приймання даних IMU та запису експериментальних наборів даних без втрати повідомлень.

Таблиця 2.2 – Порівняння варіантів апаратних компонентів для засобу візуальної одометрії

Група компонентів	Варіант	Переваги для VIO	Обмеження	Висновок для роботи
Обчислювальний модуль	Orange Pi 5 Pro	Достатній CPU-ресурс, MIPI CSI, можливість використання швидкого накопичувача, компактність	Потребує налаштування Linux-образу, драйверів камери, живлення та охолодження	Обраний основний модуль
Обчислювальний модуль	Raspberry Pi 5	Поширена платформа, розвинена екосистема камер, зручна документація	Менший запас продуктивності для одночасної обробки, запису й візуалізації у складних сценаріях	Можлива альтернатива для спрощеного стенда
Обчислювальний модуль	NVIDIA Jetson Nano	Наявність GPU та CUDA-екосистеми для методів комп'ютерного зору	Вища складність середовища, енергоспоживання та старіша апаратна база	Доцільний переважно для нейромережевої обробки

Продовження таблиці 2.2

Група компонентів	Варіант	Переваги для VIO	Обмеження	Висновок для роботи
Камерний модуль	Waveshare IMX219-83 Stereo Camera	Наявність двох сенсорів IMX219, компактність, підключення через CSI, можливість використання одного каналу	Для поточної реалізації використовується лише один канал; потрібне калібрування та контроль експозиції	Обраний камерний модуль
Камерний модуль	USB-камера	Просте фізичне підключення та швидка первинна перевірка відеопотоку	Можлива нестала затримка, стискання, буферизація драйвера та менший контроль часової мітки	Не є базовим варіантом для VIO
Камерний модуль	Global shutter camera	Менші геометричні спотворення під час швидкого руху, краща придатність для динамічних сцен	Вища вартість, потреба у сумісному драйвері та окремій інтеграції	Перспективна заміна для точніших експериментів

Кінець таблиці 2.2

Група компонентів	Варіант	Переваги для VIO	Обмеження	Висновок для роботи
Інерціальний модуль	ICM20948	Наявні гіроскоп, акселерометр і магнітометр; достатній для навчального стенда	Потребує оцінювання шуму, bias і часової узгодженості з камерою	Обраний IMU-модуль
Інерціальний модуль	MPU9250	Поширений 9-осьовий модуль, зручний для базових експериментів	Має подібні проблеми шуму, зміщення та калібрування; не дає принципової переваги	Можлива навчальна альтернатива
Інерціальний модуль	BMI088	Краща придатність для динамічних задач і робототехнічних застосувань	Зазвичай потребує окремого модуля, налаштування драйвера та ретельного калібрування	Перспективний варіант для підвищення точності

З таблиці 2.2 видно, що вибрана конфігурація не є випадковим набором доступних компонентів. Orange Pi 5 Pro забезпечує достатній обчислювальний ресурс і потрібні інтерфейси, IMX219 дає компактне джерело зображень, а

ICM20948 формує інерціальний потік для підтримки масштабу та короткочасної динаміки. Альтернативні рішення можуть бути кращими в окремих аспектах, наприклад global shutter camera краще підходить для швидких рухів, а BMI088 може бути доцільним для більш динамічних платформ. Проте для цілей цієї роботи важливішим є баланс доступності, сумісності, складності налаштування та можливості отримати відтворюваний експериментальний результат.

2.3 Обчислювальна платформа Orange Pi 5 Pro

Orange Pi 5 Pro обрано як центральний обчислювальний модуль системи. За офіційними даними виробника плата побудована на Rockchip RK3588S, який містить 8-ядерний 64-бітний CPU з big.LITTLE-архітектурою: чотири ядра Cortex-A76 та чотири ядра Cortex-A55. Таке поєднання є корисним для задач, де одночасно існують процеси різної важливості: критичні вузли захоплення кадрів і оцінювання можуть отримувати продуктивні ядра, а допоміжні процеси, журналювання або візуалізація можуть виконуватися на менш енерговитратних ядрах [58; 59].

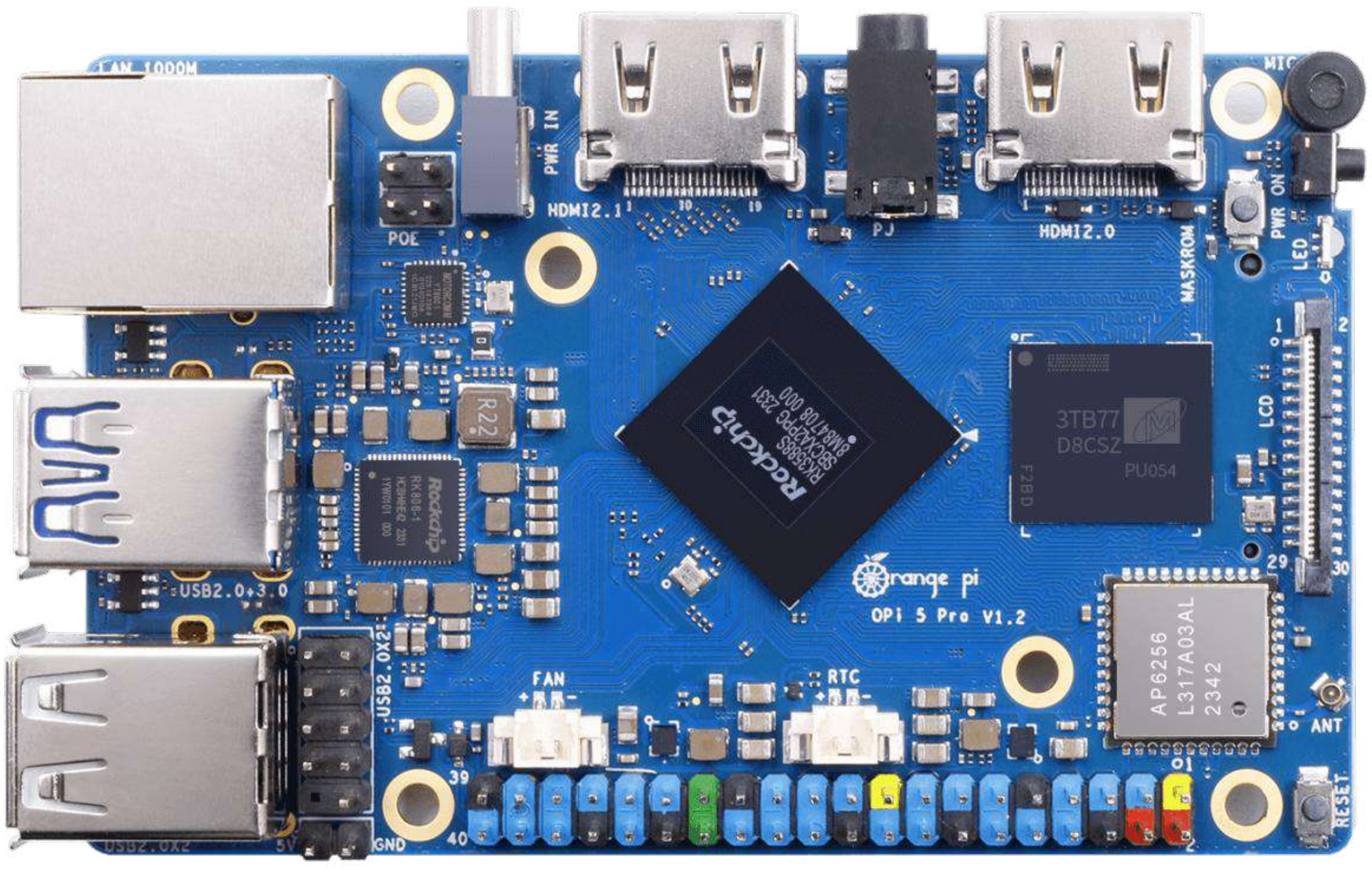


Рисунок 2.2 – Orange Pi 5 Pro, вигляд зверху [58]

Для VIO суттєва не тільки пікова продуктивність процесора, а й стабільність під тривалим навантаженням. Обробка кадрів, відстеження ознак, прийом IMU та оновлення фільтра виконуються безперервно. Якщо плата перегрівається, процесор може знижувати частоту, що призводить до нерівномірної обробки кадрів і зростання затримок. Тепловий режим тому закладається ще на етапі проєктування. Наявність на платі роз'єму для вентилятора і можливість активного охолодження розглядаються не як другорядна зручність, а як умова стабільного live-запуску під навантаженням.

Оперативна пам'ять Orange Pi 5 Pro реалізована як LPDDR5 і може мати різний обсяг залежно від модифікації. Для задачі VIO пам'ять використовується не лише операційною системою, а й буферами зображень, ROS2-повідомленнями, чергами топіків, OpenVINS, RViz і службовими процесами. Один кадр високої роздільної здатності може займати значний обсяг пам'яті, а кілька копій одного кадру можуть одночасно існувати у драйвері, вузлі камери, ROS2-повідомленні та алгоритмі обробки. Тому проєктування повинно уникати зайвого копіювання, надмірних черг і невиправдано великих кадрів.

Камерні інтерфейси є ще одним аргументом на користь Orange Pi 5 Pro. Виробник зазначає наявність двох MIPI CSI 4-lane інтерфейсів. Для камерного модуля IMX219 це означає можливість апаратного підключення сенсорів через швидкісний інтерфейс, призначений саме для передачі зображень. Проте наявність роз'єму не гарантує автоматичної роботи сенсора: потрібна підтримка у ядрі Linux, коректний devicetree, сумісний шлейф і драйверний стек. У роботі фіксується фактичний активний канал CAM:0, а не абстрактна можливість роботи двох камер.

Накопичувач також впливає на проєктування. Якщо система лише візуалізує результат, достатньо microSD або eMMC для запуску ОС. Але якщо планується запис rosbag2 з кадрами камери та IMU, швидкість і стабільність запису стають необхідними. Orange Pi 5 Pro підтримує M.2 M-Key для NVMe або SATA SSD. Для VIO-експериментів NVMe-накопичувач є кращим варіантом,

оскільки відеопотік створює значно більший обсяг даних, ніж звичайні текстові логи. Повільний накопичувач може не зупинити систему відразу, але створюватиме затримки, втрати повідомлень або пропуски кадрів.

Живлення плати також не є формальністю. За специфікацією Orange Pi 5 Pro підтримує Type-C живлення 5 В/5 А. Для роботи з камерою, периферією і високим навантаженням на CPU потрібне стабільне джерело живлення. Просідання напруги може проявлятися як нестабільна робота камери, помилки шини, перезавантаження або непередбачувана поведінка ОС. Тому блок живлення і кабель повинні розглядатися як частина апаратної конфігурації, а не як зовнішня дрібниця.

Таким чином, Orange Pi 5 Pro підходить для цієї роботи не тому, що є найпотужнішою можливою платформою, а тому, що поєднує достатній CPU-ресурс, Linux-сумісність, камерні інтерфейси, підтримку швидких накопичувачів і компактність. Для дипломного проєкту це створює реалістичну апаратну базу, на якій можна перевіряти VIO не лише в симуляції, а у live-режимі з реальними сенсорними даними.

З погляду розподілу обчислень Orange Pi 5 Pro є зручним компромісом між продуктивністю і компактністю. Вузол захоплення камери, вузол IMU, OpenVINS, візуалізація та запис даних не мають однакового пріоритету. Найважливішими є захоплення кадрів і часові мітки, оскільки пропущений або запізнілий кадр уже не можна відновити на наступних етапах. OpenVINS також необхідно враховувати, але його параметри можна налаштовувати: зменшувати кількість ознак, частоту обробки або роздільну здатність кадру. Візуалізація важлива для оператора, але у разі нестачі ресурсу її можна запускати на зовнішньому комп'ютері або обмежити частоту оновлення.

Проєктування враховує, що Linux на одноплатній платформі не є жорсткою системою реального часу. Планувальник ОС може на короткий час відкласти виконання користувацького процесу, а драйвер камери може мати власну буферизацію. Тому програмна архітектура повинна бути стійкою до

невеликих коливань часу. Це досягається невеликими чергами для відео, діагностикою пропущених кадрів, використанням монотонного часу для часових міток і мінімізацією зайвої обробки до публікації повідомлення.

Інтерфейс MIPI CSI у цій системі є не лише фізичним роз'ємом. Він визначає, що відеодані можуть надходити до плати через спеціалізований канал, а не через повільніший або менш передбачуваний інтерфейс. Водночас підтримка конкретного сенсора залежить від драйверів і devicetree. Для практичного проєкту вибір Orange Pi 5 Pro супроводжується перевіркою фактичного виявлення камери в ОС. Тому в роботі акцентовано не абстрактну наявність двох CSI-роз'ємів, а реальний активний канал CAM:0, зафіксований у live-тестуванні [51; 52].

Тепловий режим також впливає на якість VIO опосередковано. Якщо процесор переходить у throttling, середня продуктивність знижується, але ще небезпечнішою стає нерівномірність. Фільтр може отримувати кадри пачками або з нерівними інтервалами, що ускладнює оцінювання руху. Тому активне охолодження, контроль температури та стабільне живлення є частиною проєктного рішення. У невеликій мобільній системі ці питання інколи здаються другорядними, але для live-одометрії вони напряму пов'язані з повторюваністю результату.

2.4 Камерний модуль IMX219: параметри, обмеження та вибір одного каналу

У практичній частині роботи використовувався один канал камерного модуля Waveshare IMX219-83 Stereo Camera. Модуль конструктивно має два сенсори Sony IMX219, однак у цій реалізації активним був тільки потік першої камери. Такий вибір зменшує обсяг даних, спрощує синхронізацію з IMU та відповідає монокулярній конфігурації OpenVINS, у якій масштаб оцінюється за рахунок інерціальних вимірювань [56; 57].

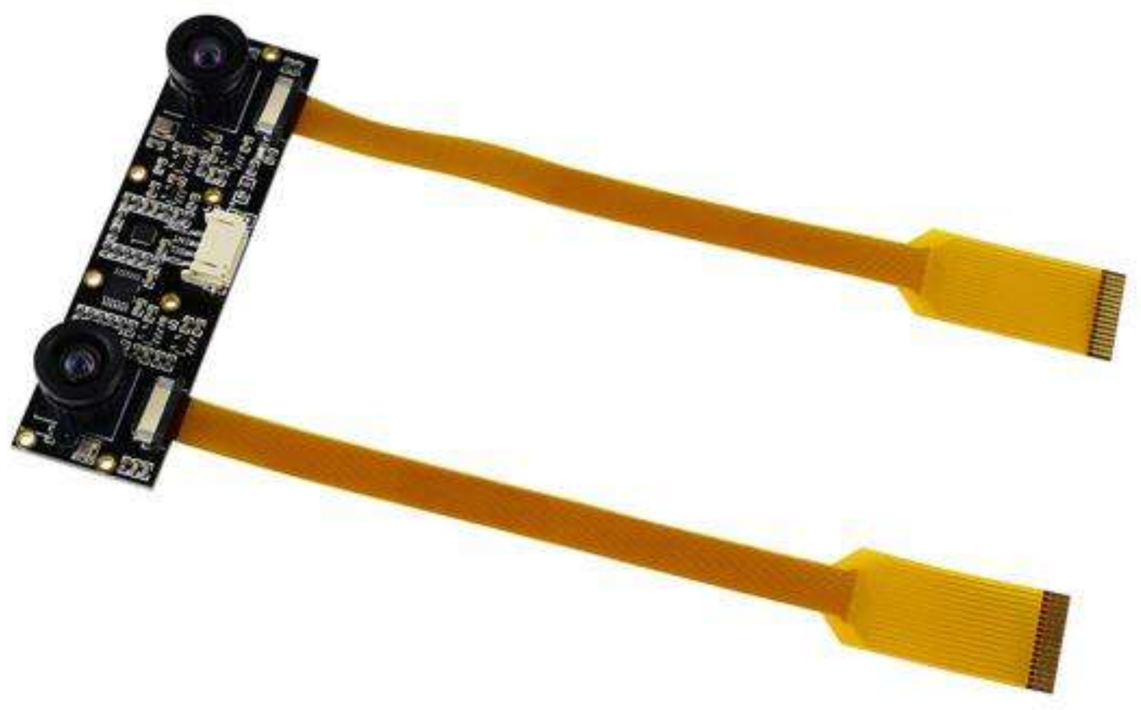


Рисунок 2.3 – Камерний модуль Waveshare IMX219-83 Stereo Camera [56]

Для візуальної одометрії важлива не лише паспортна роздільна здатність сенсора, а й фактична роздільна здатність потоку, з яким працює алгоритм. У калібрувальному наборі даних камера публікувала зображення з роздільною здатністю 640 x 480. Це менше за максимальну апаратну роздільну здатність IMX219, але для live-VIO такий режим є практичним компромісом між кількістю ознак у кадрі, затримкою обробки та навантаженням на Orange Pi 5 Pro.

За результатами калібрування Kalibr камера описується моделлю pinhole з радіально-тангенціальною моделлю дисторсії radtan. Отримані внутрішні параметри (таблиця 2.3) задають фокусні відстані у пікселях, координати головної точки та коефіцієнти дисторсії. Саме ці значення мають переходити у конфігурацію OpenVINS, тому що вони описують не абстрактний сенсор IMX219, а конкретний канал камери у конкретній збірці.

Таблиця 2.3 – Фактичні параметри камери, отримані засобами Kalibr

Параметр	Отримане значення	Значення для системи VIO
ROS-топік	/visual_odometry/raw_image 1	Ім’я потоку, який має бути заданий в OpenVINS
Модель камери	pinhole	Проєкційна модель для перерахунку пікселя у промінь

Кінець таблиці 2.3

Модель дисторсії	radtan	Компенсація радіально-тангенціальних спотворень
Роздільна здатність	640 x 480	Робочий формат калібрувального потоку
fx; fy	473.0684; 476.3805 px	Фокусні відстані в осях зображення
cx; cy	317.6282; 247.6705 px	Координати головної точки
k1; k2	0.1400238; -0.2463225	Радіальні коефіцієнти дисторсії
p1; p2	-0.0000762; -0.0003090	Тангенціальні коефіцієнти дисторсії

Отримані значення показують, що головна точка розташована близько до геометричного центра кадру 640 x 480, але не збігається з ним ідеально. Це нормальна ситуація для реальної камери: оптичний центр, сенсор і лінза мають виробничі допуски, а кріплення модуля також може вносити невелике зміщення. Для людини така різниця майже непомітна, проте для VIO вона важлива, бо помилка в моделі камери безпосередньо переходить у помилку напрямків візуальних променів.

Коефіцієнти дисторсії також підтверджують потребу в калібруванні. Навіть якщо кадр візуально виглядає прийнятним, геометрія країв зображення без корекції не відповідає ідеальній pinhole-моделі. Під час супроводу ознак це може призводити до систематичного зміщення точок, особливо коли AprilGrid або об’єкти сцени проходять поблизу меж кадру.

2.5 Інерціальний модуль ICM20948 і вимоги до IMU-потоку

Інерціальний модуль ICM20948 у цій роботі використовується як джерело кутової швидкості та лінійного прискорення. Хоча мікросхема також містить магнітометр, для OpenVINS у базовій конфігурації основними є гіроскоп та

акселерометр. Гіроскоп забезпечує короткочасний прогноз зміни орієнтації між сусідніми кадрами, а акселерометр дає інформацію про прискорення з урахуванням гравітації [60].

За результатами підготовки IMU-конфігурації (таблиця 2.4) для Kalibr у роботі використовувався потік з частотою 200 Гц. Така частота добре узгоджується з камерним потоком приблизно 20 Гц: між двома сусідніми кадрами припадає близько десяти інерціальних вимірювань. Завдяки цьому фільтр має достатньо даних для прогнозу стану до приходу нового зображення.

Таблиця 2.4 – Параметри IMU, використані під час калібрування

Параметр	Значення	Пояснення
ROS-топік	/visual_odometry/imu	Потік інерціальних повідомлень
Тип повідомлення	sensor_msgs/Imu	Стандартний формат ROS для IMU
Частота оновлення	200.0 Гц	Приблизно у 10 разів вище за частоту камери
Accelerometer noise density	0.01226660384	Шумова густина акселерометра
Accelerometer random walk	0.00219063843	Повільна зміна зміщення акселерометра
Gyroscope noise density	0.00203883836	Шумова густина гіроскопа
Gyroscope random walk	0.00018578937	Повільна зміна зміщення гіроскопа

Значення шуму та random walk не є декоративними параметрами. Вони визначають, наскільки сильно фільтр довіряє інерціальним вимірюванням під час прогнозу стану. Якщо ці параметри занадто малі, OpenVINS може

переоцінювати точність IMU і пояснювати шум як реальний рух. Якщо вони занадто великі, інерціальні дані втрачають вагу, і система сильніше залежить від якості зображення.

Окремим результатом калібрування є часовий зв’язок між камерою та IMU. У наявному наборі даних Kalibr оцінив зсув камери відносно IMU як -0.045728 с. Це означає, що в конфігурації VIO необхідно уважно враховувати напрямок часової поправки, бо навіть десятки мілісекунд при швидкому русі можуть створювати помітне розходження між візуальним та інерціальним прогнозом.

2.6 Програмна архітектура ROS2 і передача сенсорних даних

ROS2 Humble у цій роботі виконує роль програмного каркаса, що об'єднує драйвери сенсорів, вузли обробки, OpenVINS, запис даних і візуалізацію. Його перевага полягає у модульності. Камерний вузол не повинен знати внутрішню структуру OpenVINS; він лише публікує кадри у стандартизованому форматі. IMU-вузол публікує інерціальні вимірювання. OpenVINS підписується на потрібні топіки, а RViz або інший візуалізатор підписується на результат [39–42; 44].

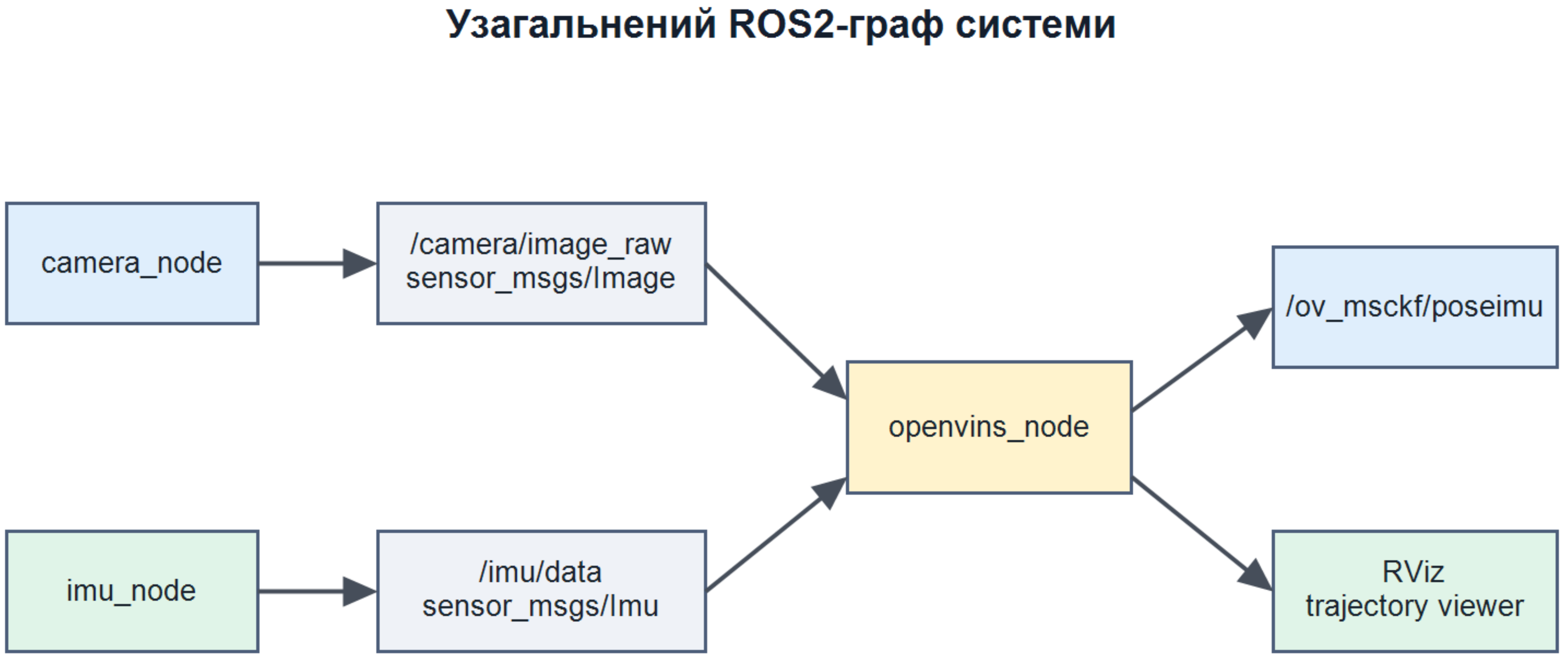


Рисунок 2.4 – Узагальнений ROS2-граф програмної системи

Узагальнений граф на рисунку 2.4 показує мінімальний набір взаємодій. Камерний вузол публікує /camera/image_raw, IMU-вузол публікує /imu/data, OpenVINS приймає ці потоки й формує одометрію або позу. У реальній системі назви можуть відрізнятися, але логіка залишається тією самою. Типи повідомлень мають відповідати очікуванням алгоритму. Зображення зазвичай подаються як sensor_msgs/msg/Image, параметри камери – як sensor_msgs/msg/CameraInfo, інерціальні вимірювання - як sensor_msgs/msg/Imu [40; 44; 46].

Для камерного потоку обирається правильна політика якості обслуговування QoS. У live-системі старий кадр часто менш цінний, ніж новий. Якщо алгоритм не встигає обробляти всі кадри, накопичення черги може збільшити затримку і зробити оцінку застарілою. Тому для відеопотоку часто використовують політику з невеликою глибиною черги. Для IMU ситуація складніша: втрати окремих інерціальних вимірювань можуть погіршити прогноз, але надмірне накопичення також шкідливе. Остаточний вибір QoS залежить від продуктивності платформи і стабільності драйвера [42; 43; 53; 54; 55].

Система координат є ще одним важливим аспектом ROS2-архітектури. Камера має оптичну систему координат, IMU має власну систему координат, а корпус платформи може мати base_link. OpenVINS повинен знати, як перетворити вимірювання з однієї системи координат в іншу. Якщо вісь IMU спрямована інакше, ніж очікує конфігурація, алгоритм може інтерпретувати поворот навколо однієї осі як рух навколо іншої. Така помилка не завжди проявляється як явне повідомлення про збій; частіше вона виглядає як нестабільна або фізично неправдоподібна траєкторія.

Запис даних через rosbag2 слід розглядати як частину проєктної архітектури, а не лише як допоміжний інструмент. Записаний набір даних дає змогу повторити експеримент, змінити параметри OpenVINS, перевірити вплив калібрування і порівняти траєкторії. Без rosbag2 кожна зміна параметрів потребує нового фізичного запуску, а порівняння стає менш об'єктивним [45; 47].

Діагностика ROS2-системи повинна бути передбачена ще на етапі проєктування. Мінімально потрібно мати можливість перевірити список топіків, частоту камери, частоту IMU, приклад одного IMU-повідомлення, наявність CameraInfo і вихідні повідомлення OpenVINS. Якщо система не працює, ці перевірки дозволяють швидко визначити рівень помилки: апаратний, драйверний, комунікаційний, калібрувальний або алгоритмічний. Без такої діагностики помилка OpenVINS може насправді бути помилкою назви топіка або неправильного типу повідомлення [48; 49; 50].

ROS2 дає змогу відокремити live-режим від офлайн-аналізу. У live-режимі оператор бачить поточний результат і може швидко перевірити працездатність. В офлайн-режимі той самий rosbag2 можна програвати повторно, змінювати параметри калібрування, аналізувати втрати повідомлень і порівнювати варіанти конфігурації. Така структура робить систему не одноразовою демонстрацією, а платформою для подальших експериментів.

2.7 Конфігурація OpenVINS і логіка роботи фільтра

OpenVINS є дослідницькою платформою для візуально-інерціального оцінювання, яка реалізує фільтраційний підхід до оцінювання стану. Для цієї роботи він підходить тим, що підтримує конфігурації з камерою та IMU, має документацію, використовує калібрувальні параметри і може бути інтегрований у ROS-середовище. Завдяки цьому дипломний проєкт зосереджується на інтеграції та проєктуванні програмно-апаратного засобу [31; 33; 34].

Стан фільтра у VIO зазвичай містить орієнтацію, положення, швидкість, зміщення гіроскопа, зміщення акселерометра та набір клонованих поз камери. Інерціальні вимірювання використовуються для прогнозування стану вперед у часі. Коли надходить новий кадр, система відстежує візуальні ознаки і формує вимірювальні обмеження, які коригують прогноз. У MSCKF-підході тривимірні координати візуальних ознак не зберігаються як постійна карта; вони

використовуються для формування обмежень між позами камери, після чого виключаються з оновлення.

Ініціалізація є одним із найчутливіших етапів. Монокулярна VIO повинна узгодити масштаб, напрям гравітації, початкову швидкість і зміщення IMU. Якщо система стартує під час руху або вібрації, гравітація може бути оцінена неправильно. Якщо сцена бідна на візуальні ознаки, система може не отримати достатніх обмежень для стабільного початку. Тому практичний сценарій запуску повинен містити короткий спокійний інтервал, після якого виконується рух із помірними поворотами і видимою текстурованою сценою.

Супровід візуальних ознак, або feature tracking, є центральним елементом візуальної частини. Алгоритм виділяє точки в кадрі та намагається знайти їх у наступних кадрах. Довгі і стабільні треки корисніші, ніж велика кількість коротких або випадкових відповідностей. Якщо експозиція занадто довга, кадр розмитий і ознаки локалізуються неточно. Якщо сцена однотонна, ознак недостатньо. Якщо у кадрі багато рухомих об'єктів, частина відповідностей описує не рух камери, а рух об'єктів сцени. Вимоги до камери, освітлення та руху платформи безпосередньо впливають на якість OpenVINS.

Роль IMU в OpenVINS полягає не тільки у відновленні масштабу. IMU дає прогноз орієнтації та швидкості між кадрами, допомагає переживати короткі періоди поганої візуальної інформації і стабілізує оцінку під час швидких обертів. Водночас IMU може погіршити результат, якщо шумові параметри або часовий зсув задані неправильно. Тому OpenVINS потрібно розглядати не як "чорну скриньку", куди достатньо подати два топіки, а як фільтр, чутливий до фізичної якості сенсорних даних.

Файл конфігурації OpenVINS у цій роботі повинен відповідати режиму одного відеоканалу. Він має містити назви топіків, модель камери, параметри дисторсії, частоту обробки, параметри IMU, перетворення T_cam_imu і часовий зсув. Якщо у конфігурації залишити параметри стереосистеми або два очікувані відеопотоки, це суперечитиме фактичній реалізації. Тому один із результатів

проєктування полягає у чіткому узгодженні конфігурації OpenVINS із реальною апаратною схемою.

З погляду структури стану OpenVINS поєднує повільніші візуальні оновлення і швидший інерціальний прогноз. IMU вимірює рух з високою частотою, але сам по собі швидко накопичує дрейф через інтегрування шуму та bias. Камера вимірює геометричні обмеження сцени з нижчою частотою, але допомагає коригувати накопичену помилку. Фільтр поєднує ці джерела так, щоб сильні сторони одного сенсора компенсували слабкі сторони іншого. Через це погана камера або поганий IMU не просто трохи погіршують результат, а порушують баланс усієї системи.

У монокулярній конфігурації особливо важлива початкова оцінка масштабу. Камера сама по собі може визначити рух лише з невизначеністю масштабу, а IMU додає інформацію про прискорення і гравітацію. Проте ця інформація корисна лише тоді, коли орієнтація IMU, часові мітки й bias задані достатньо коректно. Якщо запускати систему різким рухом, фільтр може неправильно оцінити напрям гравітації або початкові зміщення. Тому практичний сценарій старту є частиною проєктування: спочатку спокійний інтервал, потім плавний рух із достатньою кількістю візуальних ознак.

Параметри супроводу ознак також мають інженерний характер. Надто велика кількість ознак збільшує навантаження на CPU, а надто мала кількість погіршує вимірювальні обмеження. Агресивне відкидання ознак може зробити траєкторію нестабільною, а слабке відкидання може пропустити помилкові відповідності. Тому налаштування OpenVINS повинні підбиратися разом із режимом камери: роздільною здатністю, частотою кадрів, експозицією і типовою сценою тестування.

2.8 Калібрування камери та IMU засобами Kalibr

Калібрування камери та IMU засобами Kalibr у цій роботі виконує не допоміжну, а системоутворювальну роль. До калібрування є лише набір фізично

підключених сенсорів; після калібрування з’являється вимірювальна модель, придатна для OpenVINS. Вона описує внутрішню геометрію камери, шумові характеристики IMU, взаємне положення камери та IMU, а також часовий зсув між потоками [32; 35–38].

У проєкті використовувалась калібрувальна мішень AprilGrid. Її YAML-опис містить тип мішені, кількість тегів, фізичний розмір тега та відносний інтервал між тегами. Для фактичного експерименту були задані 6 стовпців і 6 рядків, розмір тега 0.062 м та параметр tagSpacing 0.298387097. Саме відповідність між паперовою мішенню і YAML-файлом відіграє основну роль: якщо розмір у файлі не відповідає реальному друку, Kalibr формально завершить роботу, але результат буде геометрично неправильним.

Після калібрування камери формується файл camchain, де для cam0 зберігаються intrinsics, distortion_coeffs, resolution і rostopic. Після спільного калібрування camera-IMU формується розширений camchain (таблиця 2.5), у якому додатково з’являються матриця T_cam_imu і timeshift_cam_imu. Ці поля є прямим мостом між Kalibr та OpenVINS.

Таблиця 2.5 – Перехід результатів Kalibr у конфігурацію OpenVINS

Результат Kalibr	Фактичне значення або файл	Як використовується в OpenVINS
camchain.yaml	ros1-camchain.yaml	Внутрішні параметри камери та дисторсія
imu.yaml	ros1-imu.yaml	Шумові характеристики IMU
T_cam_imu	матриця 4 x 4 з ros1-camchain-imucam.yaml	Зовнішнє перетворення між IMU та камерою
timeshift_cam_imu	-0.0457284133 с	Поправка часової синхронізації потоків

Кінець таблиці 2.5

rostopic камери	/visual_odometry/raw_image1	Назва вхідного відеопотоку
rostopic IMU	/visual_odometry/imu	Назва вхідного IMU-потоку

Важливо не механічно копіювати матрицю з одного YAML-файлу в інший, а перевіряти домовленість про напрямок перетворення. У результатах Kalibr матриця T_cam_imu задає перетворення від системи координат IMU до системи координат камери. Якщо конкретний конфігураційний файл OpenVINS очікує обернене перетворення, матрицю потрібно інвертувати. Це типова помилка під час перенесення калібрування, тому вона має бути зафіксована у методиці [35; 36].

Показником якості калібрування є не тільки наявність YAML-файлів, а й величина залишкових похибок. У camera-IMU калібруванні було отримано середню похибку репроєкції 0.578 px, середню похибку гіроскопа 0.014864 рад/с та середню похибку акселерометра 0.066527 м/с^2. Такі значення дають підставу використовувати отримані параметри як робочу конфігурацію для подальшого запуску OpenVINS.

2.9 Висновки до розділу 2

У другому розділі виконано проєктування програмно-апаратного засобу візуальної одометрії. Визначено, що фактична конфігурація відповідає монокулярній візуально-інерціальній схемі: один активний відеоканал IMX219 та IMU ICM20948. Апаратна наявність другого сенсора на модулі Waveshare IMX219-83 Stereo Camera розглядається як потенціал подальшого розширення, але не як частина поточного алгоритмічного режиму.

Обґрунтовано вибір Orange Pi 5 Pro як компактної обчислювальної платформи з достатнім CPU-ресурсом, MIPI CSI-інтерфейсами, підтримкою

швидких накопичувачів і можливістю запуску Ubuntu/ROS2 середовища. Розглянуто особливості камери IMX219, зокрема роздільну здатність, кут огляду, дисторсію, експозицію та вплив послідовного зчитування кадру. Окремо проаналізовано роль ICM20948, зокрема значення bias, шуму, частоти та часових міток IMU для VIO.

Показано, що OpenVINS повинен бути налаштований відповідно до одного відеоканалу та IMU. Його робота залежить від правильної ініціалізації, супроводу візуальних ознак, інерціального прогнозу та калібрувальних параметрів. Окремо описано Kalibr як необхідний етап переходу від працездатного стенда до вимірювально узгодженої VIO-системи. Саме результати Kalibr повинні зв'язати фізичні параметри камери та IMU з конфігурацією OpenVINS.

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНО-АПАРАТНОГО ЗАСОБУ

3.1 Загальна логіка практичної реалізації

Практична частина дипломного проєкту була спрямована на перевірку того, чи можна побудувати працездатний програмно-апаратний засіб візуально-інерціальної одометрії на базі доступної одноплатної платформи, камерного модуля та відкритого програмного забезпечення. На відміну від суто теоретичного аналізу, реалізація охоплює не лише алгоритм оцінювання стану, а й увесь шлях сенсорних даних: від фізичного підключення камери та IMU до появи повідомлень у ROS2, запуску OpenVINS і візуалізації результату в live-режимі.

Фактична конфігурація системи включала Orange Pi 5 Pro, камерний модуль Waveshare IMX219-83 Stereo Camera, один активний відеоканал IMX219, інерціальний модуль ICM20948, Ubuntu 22.04, ROS2 Humble та OpenVINS. Камерний модуль фізично має два сенсори, однак у виконаній реалізації використовувався один потік, позначений під час тестування як CAM:0. Тому практична система описується як монокулярна візуально-інерціальна, а не як повноцінна стереосистема. Це уточнення важливе для всіх наступних етапів: конфігурації OpenVINS, змісту калібрування, формування топіків ROS2 і трактування результатів.

Метою реалізації не було створення нового VIO-алгоритму з нуля. Алгоритмічним ядром обрано OpenVINS, який уже містить реалізацію фільтра стану, відстеження візуальних ознак і візуально-інерціального оновлення. Власна частина роботи полягала в інтеграції апаратних компонентів, підготовці програмного середовища, організації потоків зображень та IMU, описі процедури калібрування і первинній перевірці працездатності на цільовій платформі. Така постановка відповідає інженерному характеру бакалаврського

проєкту: результатом є не ізольована програма, а взаємодія апаратного вузла, операційної системи, ROS2-вузлів, калібрувальних параметрів і VIO-оцінювача.

Реалізація розглядалася як послідовність взаємопов'язаних етапів. Спочатку готується операційна система та середовище ROS2. Далі перевіряється доступ до камери, формується C++-вузол захоплення кадрів і визначається режим публікації зображень. Після цього налаштовується канал IMU, який повинен передавати кутові швидкості та лінійні прискорення у форматі, придатному для OpenVINS. Окремим етапом є калібрування через Kalibr, оскільки без внутрішніх параметрів камери, параметрів дисторсії, просторового перетворення між камерою та IMU і часового зсуву неможливо обґрунтовано говорити про точність VIO. Завершальним етапом є запуск OpenVINS та live-тестування.

3.2 Підготовка Ubuntu 22.04 та базового середовища

Orange Pi 5 Pro використовувався як локальна обчислювальна платформа, тому першим практичним завданням було підготувати на ньому Linux-середовище, придатне для запуску ROS2 та OpenVINS. Для роботи було обрано Ubuntu 22.04, оскільки саме ця версія є основною цільовою платформою для ROS2 Humble у Linux-середовищі. Такий вибір зменшує ризик несумісності пакетів і дає змогу використовувати готові deb-пакети ROS2, стандартні інструменти збірки та звичну структуру робочого простору [39].

Підготовка системи починається з встановлення образу Ubuntu 22.04, перевірки завантаження плати, доступу до мережі, коректної роботи накопичувача і стабільності живлення. Для Orange Pi 5 Pro ці дії мають практичне значення, оскільки під час роботи з камерою, OpenVINS і візуалізацією навантаження на CPU та пам'ять є постійним. Якщо система встановлена на повільний або нестабільний носій, це може проявитися не відразу, а під час запису rosbag2, встановлення залежностей або одночасної роботи кількох ROS2-вузлів. Тому для тривалих експериментів бажано

використовувати швидкий накопичувач, а не обмежуватися мінімальною microSD-картою.

Перед встановленням ROS2 перевіряється, що в системі налаштована локаль з підтримкою UTF-8, активні стандартні репозиторії Ubuntu і встановлені базові інструменти розробника. Офіційна інструкція ROS2 Humble для Ubuntu 22.04 передбачає додавання репозиторіїв ROS, встановлення пакетів ROS2 і налаштування середовища через setup.bash. У практичній роботі встановлюють не лише мінімальний набір ROS-Base, а й компоненти для розробки та візуалізації, оскільки під час налагодження потрібні rviz2, інструменти командного рядка, повідомлення sensor_msgs, пакети для роботи із зображеннями та система збірки colcon.

Після встановлення ROS2 перевіряється, що система бачить команди ros2, може запускати вузли, показує список топіків і дає змогу збирати пакети. У дипломному проєкті це потрібно для відокремлення помилок середовища від помилок алгоритму. Наприклад, відсутній пакет повідомлень, неправильно підключений setup.bash або змішування різних версій ROS можуть призвести до того, що OpenVINS не стартує або не бачить сенсорні топіки, хоча сама логіка VIO при цьому не є проблемною.

Після встановлення ОС оновлюються пакети, перевіряються доступні пристрої камери та коректність системного часу. На одноплатних комп'ютерах із ARM-архітектурою готові пакети можуть залежати від конкретної версії ядра, драйверів камери та системних бібліотек. Для live-VIO абсолютний час не завжди є головним, але стабільний монотонний час і правильне формування часових міток мають основне значення для узгодження камери та IMU.

3.3 Створення робочого простору ROS2 та підготовка OpenVINS

Після встановлення ROS2 формується робочий простір, у якому розміщуються власні вузли, залежності та OpenVINS. Стандартна структура робочого простору ROS2 містить каталог src, де зберігаються пакети, і каталог

збірки, який створюється colcon. Такий підхід зручний для дипломної роботи, оскільки дає змогу відокремити системні пакети ROS2 від власних змін, конфігурацій і експериментальних вузлів.

Після збірки перевіряється наявність виконуваних вузлів OpenVINS, конфігураційних файлів і запускових сценаріїв. На цьому етапі не потрібно одразу вимагати від системи коректної траєкторії. Спочатку достатньо переконатися, що OpenVINS запускається, читає конфігураційний файл, не завершується аварійно через відсутність параметрів і може бути підключений до ROS2-топіків. Лише після цього має сенс переходити до налаштування сенсорних потоків.

У структурі робочого простору виділяється окремий пакет для вузла камери, окремий пакет або модуль для IMU і каталог конфігурацій для OpenVINS. Це спрощує супровід системи, оскільки зміна параметрів камери не змішується з кодом захоплення кадрів, а калібрувальні YAML-файли можна зберігати як частину експериментальної конфігурації. Для дипломного проєкту така організація також корисна з погляду пояснення: вона показує, де закінчується апаратний рівень, де формується ROS2-повідомлення і де починається алгоритм оцінювання.

Особливістю OpenVINS є те, що він не є просто переглядачем відео або бібліотекою комп'ютерного зору. Це фільтр оцінювання стану, який очікує узгодженого набору параметрів. У монокулярній конфігурації йому потрібен один потік зображень, IMU-потік, внутрішні параметри камери, модель дисторсії, шумові параметри IMU, просторове перетворення між камерою та IMU і, за потреби, часовий зсув. Якщо хоча б один із цих елементів задано приблизно, система може все одно запускатися, але результат буде нестійким або методично некоректним. Через це підготовка конфігурації для OpenVINS є критично важливим етапом перед запуском системи. Неточності в калібруванні або невідповідність часових міток між сенсорами можуть призводити до накопичення помилок, втрати трекінгу чи нестабільної оцінки траєкторії.

3.4 Організація ROS2-топіків і повідомлень

ROS2 у цій роботі використовується як середовище обміну повідомленнями між вузлами захоплення сенсорних даних, запису bag-файлів, калібрування та OpenVINS. Практичний набір даних був збережений у вигляді rosbag і після конвертації використовувався Kalibr у форматі ROS1 bag. Це не змінює сутність експерименту: вхідними потоками залишаються зображення з камери та інерціальні вимірювання.

Фактичний файл ros1.bag (таблиця 3.1) містить два з'єднання: відеопотік першого каналу камери та IMU-потік. Відеопотік має тип sensor_msgs/Image, а IMU-потік - sensor_msgs/Imu. Така структура відповідає монокулярному VIO: OpenVINS отримує один потік зображень і один потік інерціальних вимірювань.

Таблиця 3.1 – Характеристика записаного rosbag-файлу

Топік	Тип повідомлення	Кількість повідомлень	Тривалість	Орієнтовна частота
/visual_odometry/raw_image1	sensor_msgs/Image	835	41.769 с	19.99 Гц
/visual_odometry/imu	sensor_msgs/Imu	8525	42.769 с	199.33 Гц

Отримане співвідношення частот є логічним для VIO. Камера працює приблизно на 20 Гц, тобто дає достатньо кадрів для супроводу ознак без надмірного навантаження на обчислювальний модуль. IMU працює приблизно на 200 Гц, тому між двома кадрами доступно кілька інерціальних вимірювань для прогнозу орієнтації, швидкості та зміщень bias.

Практично важливо, що назви топіків у bag-файлі мають повністю збігатися з назвами у конфігураційних файлах OpenVINS або запускових параметрах. Якщо камера публікує /visual_odometry/raw_image1, а OpenVINS очікує /camera/image_raw, алгоритм не отримає зображення навіть тоді, коли

камера фізично працює правильно. Аналогічна проблема можлива для IMU-потоку.

Для live-режиму черги повідомлень потрібно налаштовувати так, щоб система не накопичувала застарілі кадри. Для калібрування, навпаки, головним є повнота й коректність запису: якщо частина кадрів або IMU-повідомлень втрачається, Kalibr може відхилити окремі ділянки траєкторії або отримати менш стабільну оцінку часової синхронізації.

3.5 Реалізація C++-вузла захоплення кадрів

Камерний потік є найбільш об'ємним потоком даних у системі, тому його захоплення реалізовується мовою C++. C++ дає змогу точніше контролювати буфери, уникати зайвого копіювання кадрів, керувати частотою зчитування та формувати часові мітки з меншою непередбачуваною затримкою. Python зручний для допоміжної логіки, але для безперервного відеопотоку на одноплатній платформі додаткові накладні витрати можуть стати помітними.

У межах реалізації C++-вузол повинен виконувати кілька взаємопов'язаних функцій. Спочатку він відкриває активний відеопристрій, який відповідає каналу CAM:0, встановлює потрібний формат кадру, роздільну здатність і частоту. Далі вузол переходить у цикл захоплення, у якому отримує кадр, створює часову мітку, перевіряє цілісність даних і публікує зображення у ROS2. Якщо кадр не отримано або він має неправильний розмір, вузол не повинен мовчки передавати некоректні дані в OpenVINS, оскільки це ускладнить діагностику [41; 50; 52].

Буферизація безпосередньо впливає на затримку. На рівні драйвера камери зазвичай існує кілька буферів, у яких кадри очікують обробки. На рівні ROS2 також може бути черга повідомлень. Якщо всі ці черги зробити великими, система не втратить кадри відразу, але почне працювати із затримкою. Для візуальної одометрії така поведінка небажана: краще обробляти актуальні кадри з контрольованою частотою, ніж накопичувати старі кадри, які вже не відповідають поточному руху сенсорного вузла. Тому у вузлі захоплення варто

вести лічильники отриманих, опублікованих і пропущених кадрів, а також періодично виводити діагностичну інформацію.

Формування часової мітки є одним із найчутливіших місць у C++-вузлі. Якщо апаратна або драйверна часова мітка доступна, її бажано використовувати, оскільки вона ближча до реального моменту експозиції кадру. Якщо такої мітки немає, вона створюється за системним монотонним часом одразу після отримання кадру. Мітку не варто ставити після додаткового перетворення, зміни формату або повільного копіювання. OpenVINS використовує часовий порядок вимірювань, і систематична помилка часової мітки може виглядати як помилка калібрування або зміщення IMU. Основні функції C++-вузла захоплення кадрів наведено у таблиці 3.2.

Таблиця 3.2 – Основні функції C++-вузла захоплення кадрів

Функція вузла	Реалізаційний зміст	Значення для VIO
Ініціалізація CAM:0	Відкриття активного відеоканалу IMX219	Забезпечує фактичне джерело зображень
Налаштування формату	Вибір роздільної здатності, частоти, кодування	Впливає на навантаження і якість ознак
Буферизація	Контроль черг драйвера і ROS2	Зменшує затримку та накопичення старих кадрів
Часова мітка	Мітка часу близько до моменту захоплення	Дає змогу узгодити кадр з IMU
Публікація	Формування sensor_msgs/msg/Image	Передає кадри до OpenVINS і візуалізації
Діагностика	Лічильники кадрів, частота, пропуски	Дає змогу відрізнити помилки камери від помилок алгоритму

Для VIO не завжди потрібна максимальна роздільна здатність IMX219. Якщо кадри занадто великі, CPU витрачає більше часу на копіювання, перетворення і відстеження ознак. У практичній системі варто починати з помірної роздільної здатності та стабільної частоти кадрів, а потім підвищувати якість зображення лише тоді, коли є запас продуктивності. Для OpenVINS стабільність потоку часто цінніша за максимально деталізоване зображення.

3.6 Отримання та публікація IMU-даних ICM20948

IMU ICM20948 є другим основним сенсорним джерелом системи. У візуально-інерціальній одометрії камера дає геометричні спостереження сцени, але між кадрами саме IMU забезпечує прогноз швидкої динаміки. Гіроскоп дає змогу оцінювати зміну орієнтації, а акселерометр - зміну швидкості з урахуванням сили тяжіння. Тому навіть якщо у live-відео основну увагу видно на камерному вікні, інерціальний канал має не менш важливе методичне значення.

З практичного погляду IMU-вузол повинен зчитувати дані з ICM20948, перетворювати їх у фізичні одиниці, ставити часову мітку і публікувати повідомлення sensor_msgs/msg/Imu. Для гіроскопа потрібно передавати кутову швидкість у рад/с, навіть якщо початкові дані надходять у градусах за секунду. Для акселерометра потрібно передавати прискорення у м/с^2, навіть якщо сенсор повертає значення у g. Неправильні одиниці є однією з найнебезпечніших помилок, оскільки вузли будуть запускатися без очевидної аварії, але фільтр стану отримає фізично неправильні вимірювання.

Для ICM20948 враховується bias. Гіроскоп і акселерометр мають зміщення нуля, яке змінюється з часом, температурою і режимом роботи. OpenVINS може оцінювати bias як частину стану, але початкові параметри і шумові характеристики все одно повинні бути реалістичними. Якщо гіроскоп у спокої показує ненульову кутову швидкість, фільтр може поступово накопичувати помилку орієнтації. Якщо акселерометр має помітне зміщення, виникають

помилки у швидкості та положенні. Перед live-запуском бажаний короткий статичний запис IMU для перевірки середніх значень і рівня шуму.

Частота IMU повинна бути вищою за частоту камери. OpenVINS у своїй документації для калібрування рекомендує частоту камери приблизно 20-30 Гц та інерціальний потік порядку сотень герц, наприклад 200-500 Гц, якщо сенсор та платформа це дозволяють.

Часові мітки IMU мають створюватися в тому самому часовому домені, що й часові мітки камери. Якщо камера отримує часову мітку за системним часом, IMU також повинен отримувати її з цього ж джерела, або має бути відомий і компенсований зсув. В ідеальному випадку часова мітка ставиться в момент отримання вимірювання, а не після виведення в консоль, фільтрації або формування складного повідомлення. Для інерціального потоку, який має значно більшу частоту, навіть невеликі затримки й нерівномірність можуть накопичуватися і впливати на інтегрування.

У повідомленні sensor_msgs/msg/Imu задається frame_id, який відповідає системі координат IMU. Надалі через калібрування визначається просторове перетворення між цією системою координат і системою координат камери. Якщо осі IMU переплутані або мають інший напрям, ніж очікує OpenVINS, система може працювати нестійко навіть за правильної частоти та одиниць. Під час налагодження виконуються прості ручні перевірки: повернути сенсор навколо кожної осі, переконатися у зміні відповідного каналу гіроскопа, покласти модуль у нерухомий стан і перевірити, що акселерометр показує величину, близьку до прискорення вільного падіння.

3.7 Практична методика калібрування через Kalibr

Практична методика калібрування у цій роботі складалася з підготовки AprilGrid, запису синхронних потоків камери та IMU, конвертації bag-файлу у формат, придатний для Kalibr, а також запуску окремого калібрування камери і спільного калібрування camera-IMU. Наявність збережених YAML-файлів, PDF-

звітів і текстових результатів дає змогу не лише описати процедуру, а й показати її фактичний результат.

Для мішені використовувався файл aprilgrid.yaml з такими параметрами:

1) target_type: 'aprilgrid';
2) tagCols: 6;
3) tagRows: 6;
4) tagSize: 0.062;
5) tagSpacing: 0.298387097.

Параметр tagSize задає фізичний розмір чорного квадрата AprilTag у метрах. Параметр tagSpacing у Kalibr задається як відношення проміжку між тегами до розміру тега. Для наведених даних це відповідає проміжку приблизно 0.0185 м. Саме ці значення повинні відповідати надрукованій мішені, інакше оцінка масштабу та зовнішнього перетворення буде спотвореною.

Під час підготовки робочого середовища Kalibr запускався у Docker-контейнері. Такий підхід зручний, тому що Kalibr орієнтований на ROS1, тоді як основна система тестувалася з ROS2 Humble. Тому послідовність включала запис ROS2 bag, конвертацію у ROS1 bag засобами rosbags-convert і запуск Kalibr всередині контейнера.

Для калібрування камери використовувався запуск Kalibr із мішенню aprilgrid.yaml, bag-файлом ros1.bag, топіком /visual_odometry/raw_image1 та моделлю pinhole-radtan. Після цього було сформовано файл ros1-camchain.yaml і звіт ros1-report-cam.pdf. Для спільного калібрування camera-IMU використовувалися ros1-camchain.yaml, imu.yaml та той самий bag-файл. Результатами стали ros1-camchain-imucam.yaml, ros1-imu.yaml, ros1-results-imucam.txt і ros1-report-imucam.pdf. Призначення файлів калібрувального пакета наведено у таблиці 3.3. Отримані параметри використовуються OpenVINS для правильної інтерпретації вимірювань від камери та IMU під час оцінювання траєкторії руху.

Таблиця 3.3 – Файли калібрувального пакета та їх призначення

Файл	Призначення
aprilgrid.yaml	Опис геометрії калібрувальної мішені
imu.yaml	Початкові шумові параметри IMU для Kalibr
ros1.bag	Записаний набір даних камери та IMU
ros1-camchain.yaml	Внутрішні параметри камери після camera calibration
ros1-camchain-imucam.yaml	Камера, IMU, T_cam_imu та часовий зсув
ros1-report-cam.pdf	Звіт Kalibr для калібрування камери
ros1-report-imucam.pdf	Звіт Kalibr для camera-IMU калібрування

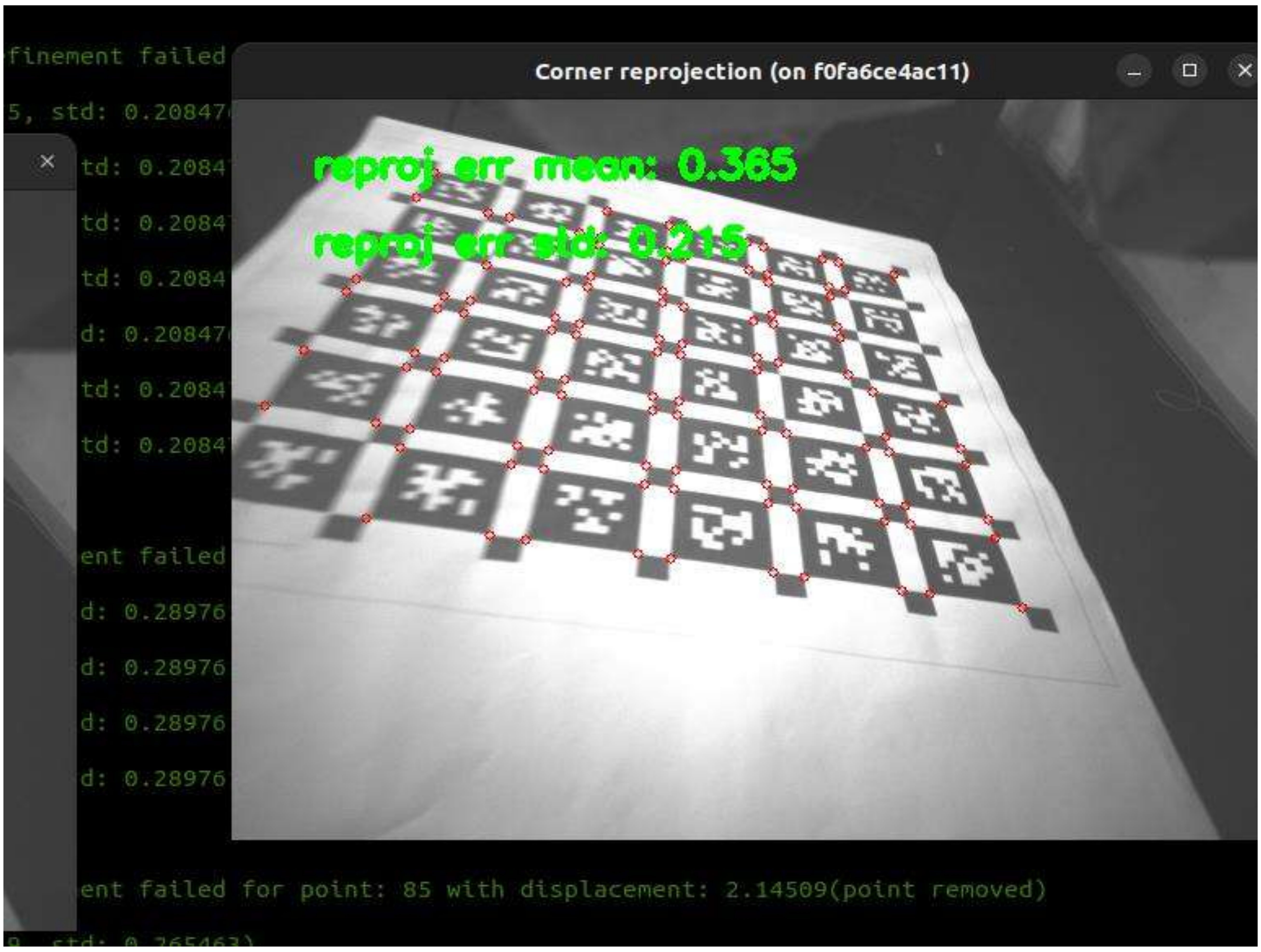


Рисунок 3.1 – Успішне розпізнавання AprilGrid та оцінювання похибки репроєкції у Kalibr

На рисунку 3.1 видно, що Kalibr розпізнав сітку AprilGrid, наклав розраховані кути на зображення та вивів поточні значення похибки репроєкції. Такий кадр є корисним не тільки як ілюстрація, а й як контроль якості запису: він показує, що мішень потрапляє в кадр, займає достатню площу та має видимі кути. Числові результати camera-IMU калібрування наведено у таблиці 3.4

Таблиця 3.4 – Числові результати camera-IMU калібрування

Показник	Mean	Median	Std
Reprojection error cam0, px	0.578319	0.435283	0.495408
Gyroscope error, rad/s	0.014864	0.011493	0.012269
Accelerometer error, m/s^2	0.066527	0.057206	0.040504

У результаті оптимізації Kalibr отримав матрицю зовнішнього перетворення T_ci, яка описує перехід від IMU до cam0.

Оцінений часовий зсув становить -0.0457284133 с. Це значення потрібно враховувати під час перенесення параметрів у OpenVINS, тому що візуальні та інерціальні вимірювання належать до різних фізичних моментів часу. Без такої поправки система може демонструвати дрейф навіть за правильної геометричної калібровки [17; 18; 32].

Окрім текстового файлу результатів, Kalibr формує PDF-звіт із графічними діагностичними матеріалами. Ці графіки важливі для перевірки калібрування, оскільки показують не лише підсумкові числа, а й розподіл похибок, стабільність інерціального потоку та поведінку залишків після оптимізації.

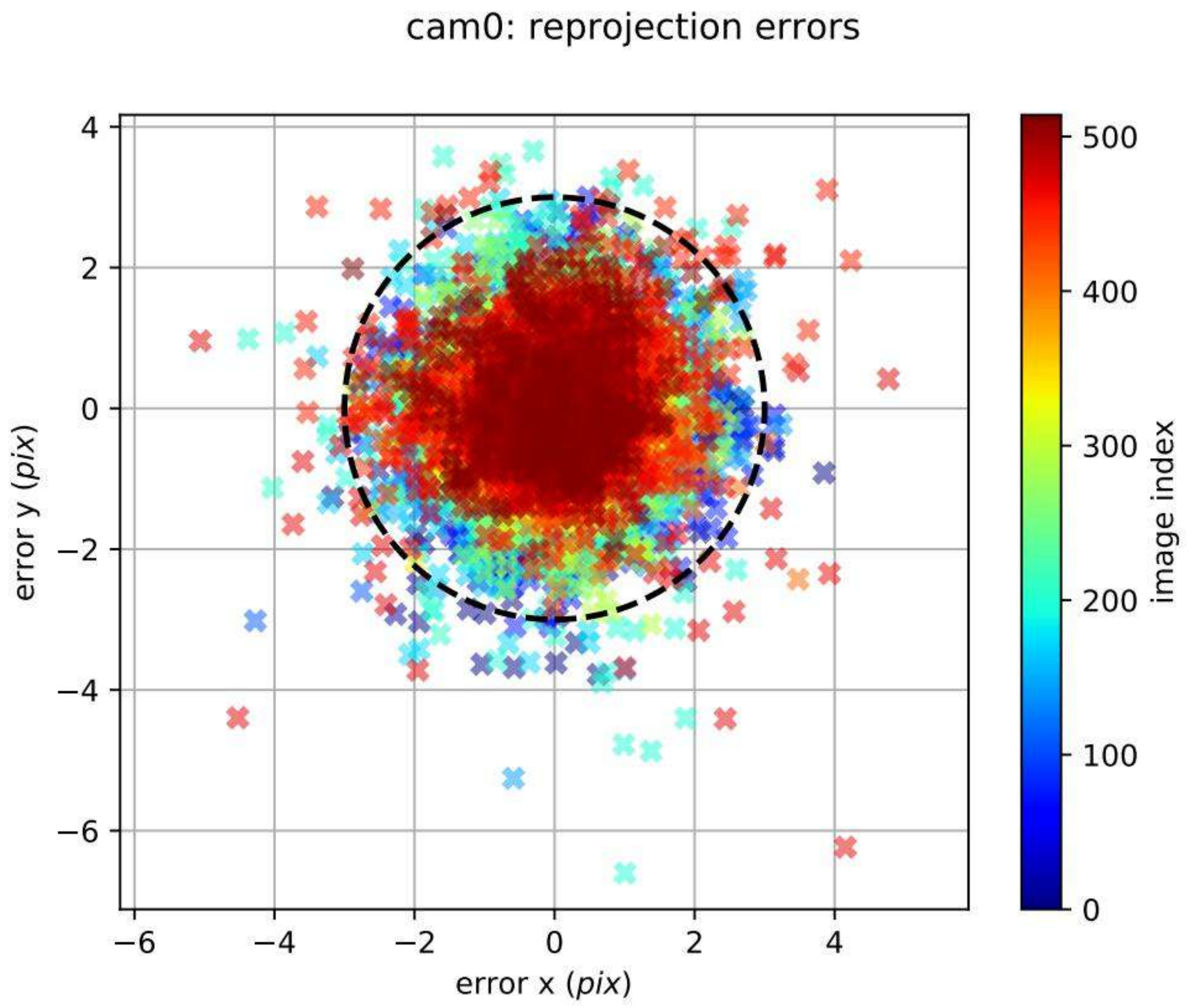


Рисунок 3.2 – Розподіл похибок репроєкції після camera-IMU калібрування Kalibr

На рисунку 3.2 наведено розподіл похибок репроєкції після спільного калібрування камери та інерціального блока. Основна частина вимірювань зосереджена поблизу центральної області, а поодинокі відхилення показують ті кадри, у яких виявлення кутів калібрувальної мішені було менш стабільним.

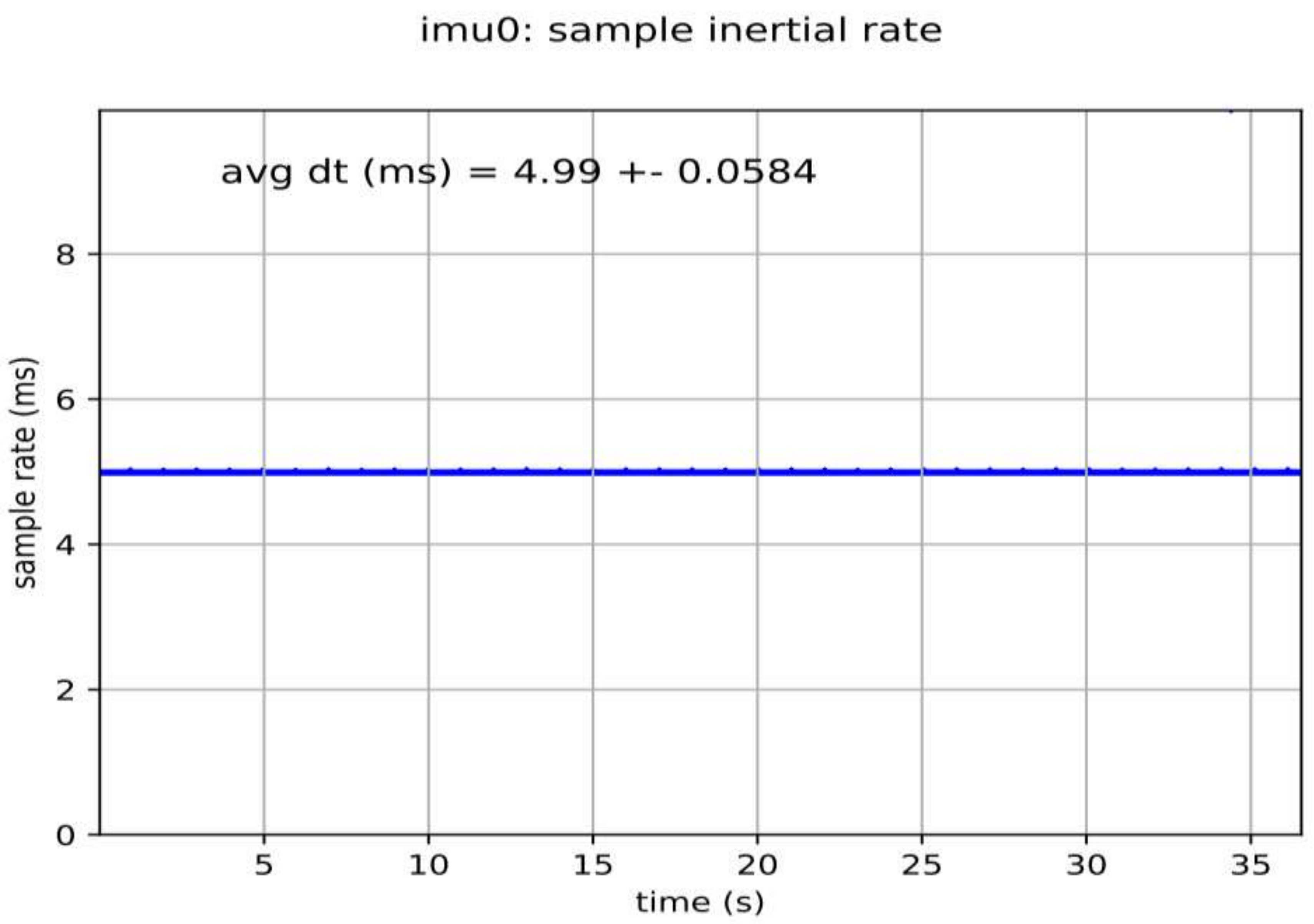


Рисунок 3.3 – Оцінка частоти IMU-потоку у звіті Kalibr

На рисунку 3.3 подано перевірку часової стабільності інерціального потоку. Середній інтервал між повідомленнями IMU становить близько 4,99 мс, тобто частота вимірювань наближається до 200 Гц і відповідає вимогам OpenVINS до регулярних інерціальних даних.

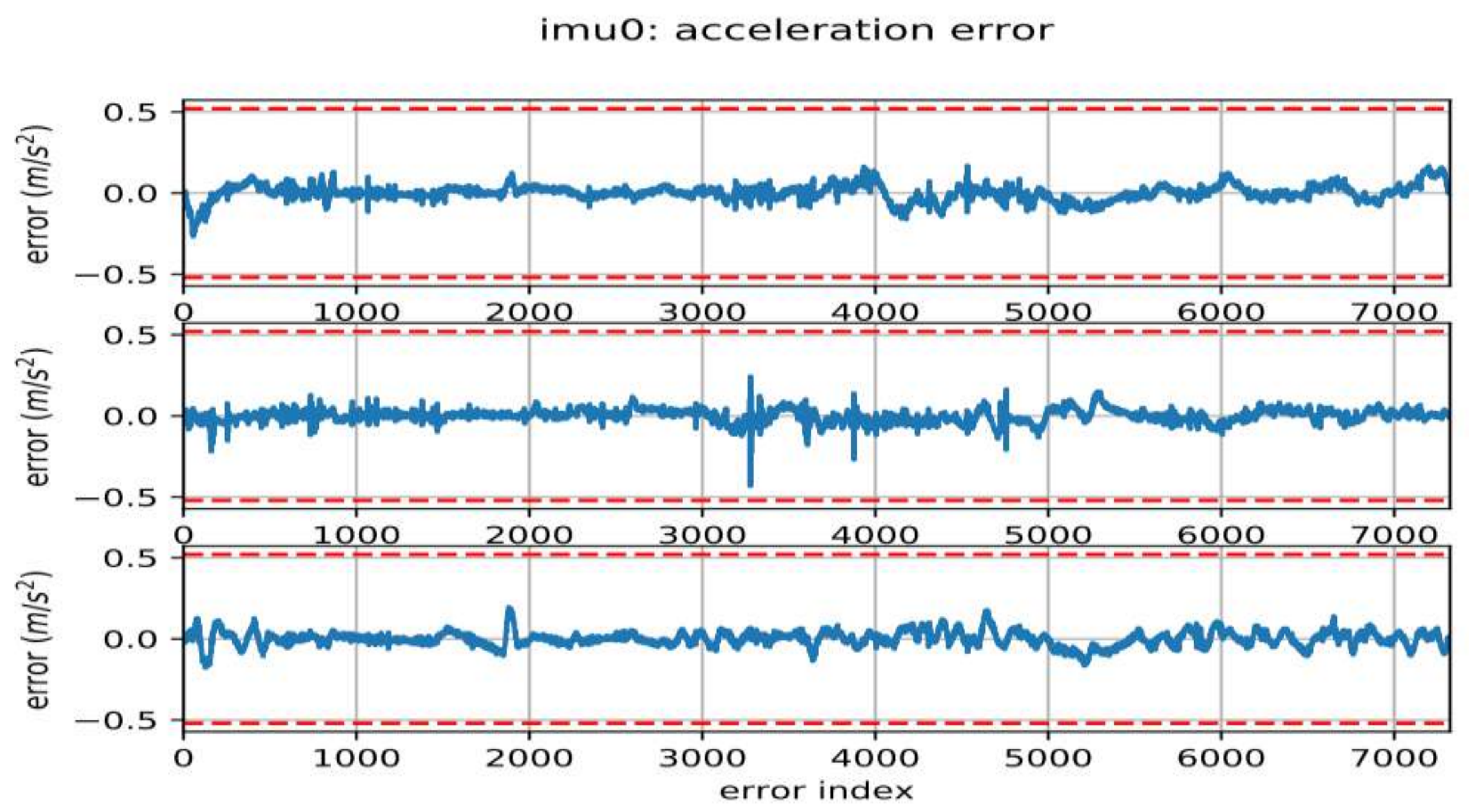


Рисунок 3.4 – Похибки акселерометра після оптимізації Kalibr

На рисунку 3.4 показано залишкові похибки акселерометра після оптимізації параметрів у Kalibr. Більшість значень перебуває у допустимих межах, а короткі піки пов'язані з різкими переміщеннями під час запису калібрувальної послідовності.

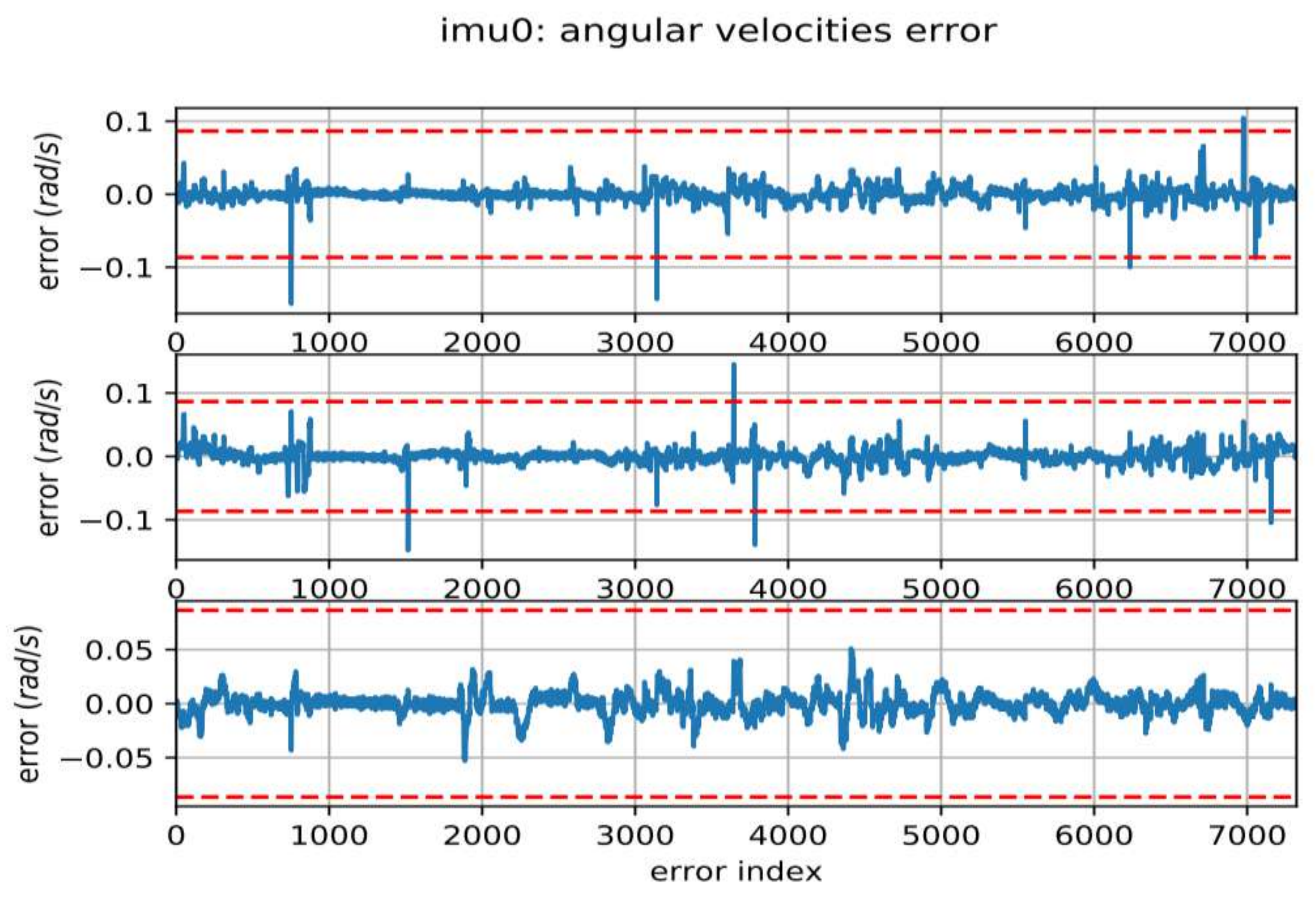


Рисунок 3.5 – Похибки кутових швидкостей гіроскопа після оптимізації Kalibr

На рисунку 3.5 охарактеризовано залишкові похибки гіроскопа. Наближеність більшості значень до нуля підтверджує узгодженість кутових швидкостей із відеопотоком, що є важливим для подальшої роботи візуально-інерціальної одометрії.

З рисунків 3.2–3.5 видно, що калібрування не обмежується отриманням одного YAML-файлу. Звіт дає змогу перевірити щільність похибок репроєкції, стабільність частоти IMU та характер залишків акселерометра і гіроскопа. Саме ці матеріали доцільно використовувати як доказову частину практичного розділу.

3.8 Налаштування конфігурації OpenVINS

Налаштування OpenVINS після калібрування повинно спиратися на фактичні YAML-файли Kalibr, а не на типові приклади з документації. У цій роботі монокулярна конфігурація має один активний канал cam0 і один IMU-потік. Тому у конфігурації OpenVINS кількість камер задається як один, а назви топіків мають відповідати фактичному rosbag або live-запуску.

Для IMU у конфігурацію переносяться параметри шуму та random walk. Вони визначають коваріації процесу в інерціальному прогнозі. У даному випадку використовуються значення accelerometer_noise_density = 0.01226660384, accelerometer_random_walk = 0.00219063843, gyroscope_noise_density = 0.00203883836 та gyroscope_random_walk = 0.00018578937.

Матриця T_cam_imu повинна бути перенесена з урахуванням домовленості конкретної конфігурації OpenVINS. Якщо файл очікує перетворення від камери до IMU, потрібно використати обернену матрицю T_ic, яку Kalibr також наводить у текстовому звіті. У звіті для цього експерименту T_ic має перенос [-0.16372035; -0.05597131; 0.05461415], що відповідає оберненню відносно T_ci.

Після внесення параметрів конфігурацію доцільно перевіряти у два етапи. Спочатку OpenVINS запускається на записаному bag-файлі, оскільки це забезпечує повторюваність. Якщо результат стабільний, ті самі параметри використовуються у live-режимі на Orange Pi 5 Pro. Такий підхід зменшує кількість одночасних невідомих: у разі помилки можна окремо перевірити конфігурацію, синхронізацію, продуктивність або фізичне підключення сенсорів.

3.9 Live-тестування системи

Live-тестування виконувалося для перевірки працездатності всього програмно-апаратного ланцюга: підключення камери, отримання відеопотоку, робота ROS2-вузлів, передача повідомлень, запуск OpenVINS і відображення результату у 3D-візуалізації. На відміну від роботи тільки з готовим датасетом, live-режим перевіряє також стабільність драйверів, затримки та поведінку системи під реальним навантаженням [45; 47].

Наявний відеофайл підтверджує роботу вхідного потоку CAM:0 та наявність RViz-подібної 3D-візуалізації. Разом із цим новий калібрувальний набір даних дає змогу доповнити відеодоказ числовими характеристиками: камера фактично працювала з частотою близько 20 Гц, IMU - близько 199 Гц, а Kalibr сформував робочі YAML-файли для OpenVINS.

Таким чином, практична частина складається з двох взаємодоповнювальних підтверджень. Відео показує, що система запускається у live-режимі й реагує на сенсорний потік. Rosbag і результати Kalibr показують, що сенсорні дані можна зберегти, обробити та отримати параметри, необхідні для відтворюваного запуску. Для бакалаврського проєкту така комбінація є переконливішою, ніж лише демонстрація екрана.

Під час live-тесту особливу увагу потрібно приділяти затримці. Якщо камера, IMU, OpenVINS і візуалізація працюють одночасно на одному

обчислювальному модулі, черги повідомлень можуть накопичуватися. У такому разі траєкторія на екрані буде відображати не поточний рух, а застарілі кадри. Тому live-запуск потрібно перевіряти не тільки за фактом появи картинки, а й за плавністю реакції системи на поворот і переміщення сенсорного модуля.

Записаний rosbag також корисний для післяекспериментального аналізу. Один і той самий набір даних можна повторно програвати з різними налаштуваннями OpenVINS, змінюючи тільки конфігурацію. Це дає змогу оцінити, як на результат впливають параметри шуму IMU, часовий зсув, частота обробки кадрів і налаштування супроводу ознак.

3.10 Оцінка працездатності реалізованої системи

Після отримання повного калібрувального пакета практична частина роботи набула завершеного інженерного вигляду. У розпорядженні є запис ros1.bag, файл camchain з параметрами камери, imu.yaml з шумовими характеристиками інерціального блока, текстові результати Kalibr та PDF-звіти з графічною діагностикою. Завдяки цьому реалізація спирається не на умовну схему, а на фактичні дані, отримані під час роботи з камерою, IMU та обчислювальним модулем.

Використання одного каналу стереокамери є свідомим проєктним рішенням, яке відповідає монокулярній візуально-інерціальній конфігурації. Такий підхід зменшує обсяг відеоданих, спрощує підключення до Orange Pi 5 Pro і дає змогу зосередитися на узгодженні одного камерного потоку з високочастотними IMU-вимірюваннями.

Калібрувальний запис тривалістю близько 42 секунд забезпечив достатній набір спостережень AprilGrid для формування внутрішніх параметрів камери та зовнішнього перетворення між камерою й інерціальним модулем. Важливо, що Kalibr не лише видав YAML-файли, а й сформував діагностичні графіки, за якими можна оцінити стабільність потоку, розподіл похибок і якість

використаних кадрів. Це перетворює калібрування з формальної операції на контрольований етап підготовки VIO-системи.

Рисунок 3.6 – Приклад діагностичного кадру Kalibr під час відбору калібрувальних зображень

Рисунок 3.6 ілюструє штатний механізм контролю якості калібрувальних кадрів. Kalibr аналізує кожне зображення з мішенню AprilGrid і використовує лише ті кадри, у яких кути визначені достатньо надійно. Такий відбір підвищує достовірність результатів, оскільки випадкові розмиття, невдалі ракурси або частковий вихід мішені за межі кадру не потрапляють до оптимізації як рівноцінні вимірювання.

Наявність відхилених кадрів у звіті є нормальною частиною практичного калібрування. Вона показує, що система не приймає всі дані без перевірки, а виконує попередню фільтрацію за якістю спостережень. Для реального програмно-апаратного засобу це важливо, тому що стабільність VIO залежить не

лише від алгоритму OpenVINS, а й від коректності сенсорних даних, часових міток і геометричної моделі камери.

Обчислювальна платформа Orange Pi 5 Pro забезпечує достатній рівень продуктивності для запуску сенсорного ланцюга, обробки камерного потоку, приймання IMU-даних і передавання повідомлень у ROS2. Практична цінність такого вибору полягає в тому, що система будується не на настільному комп'ютері, а на компактному модулі, придатному для подальшого встановлення на мобільну платформу. Саме такий формат відповідає задачі автономної навігації та робить результат роботи прикладним.

Отже, реалізований засіб підтверджує працездатність повного ланцюга візуально-інерціальної одометрії: від підготовки апаратної частини та сенсорних потоків до калібрування, налаштування OpenVINS і live-перевірки.

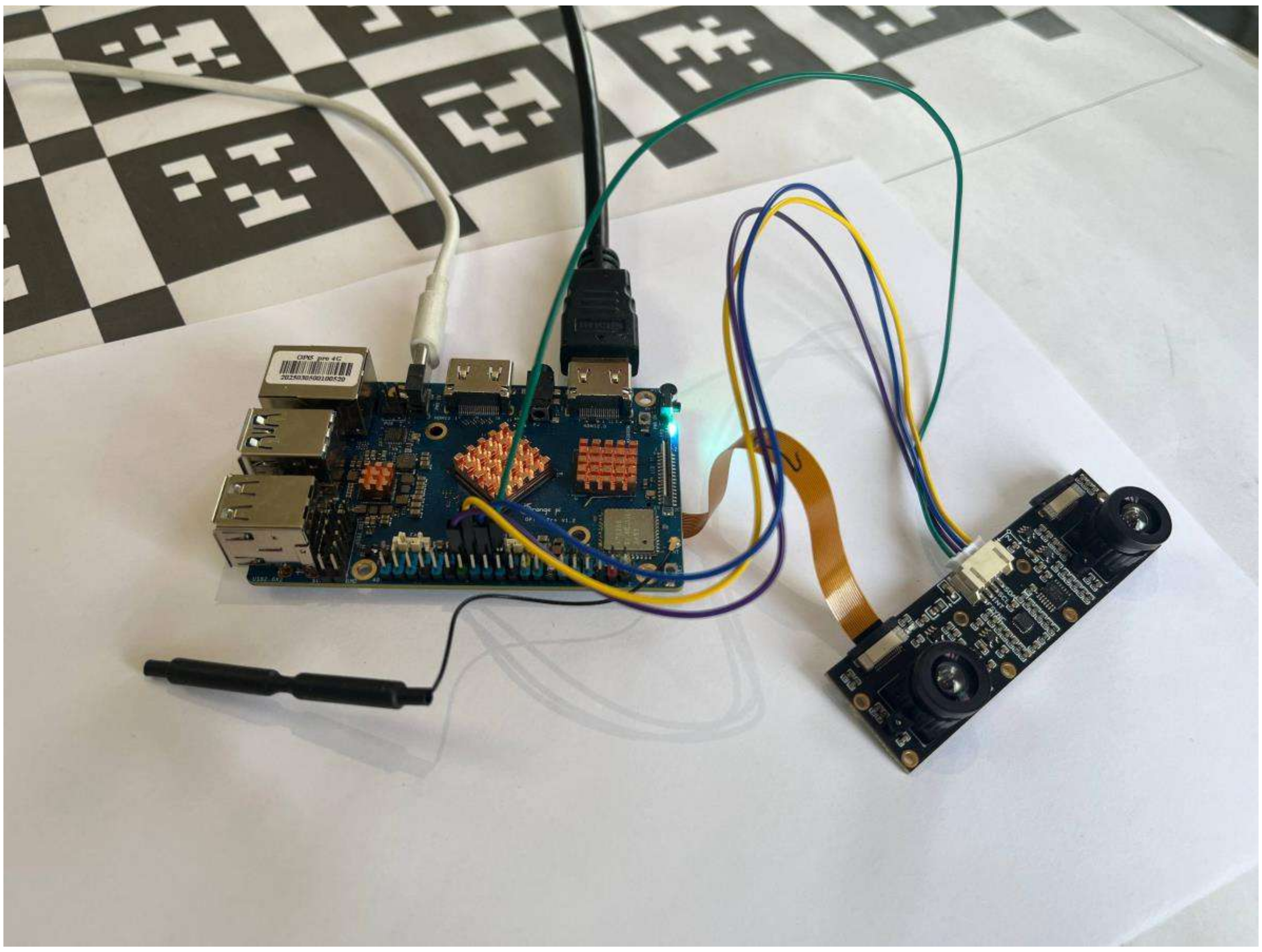


Рисунок 3.7 – Зібраний пристрій

3.11 Методика відтворюваного експерименту

Для того щоб результати роботи можна було повторити й перевірити, експеримент доцільно оформлювати як набір пов'язаних артефактів. До нього входять версії Ubuntu, ROS2 і OpenVINS, параметри камери, частота IMU, назви ROS2-топіків, YAML-файли Kalibr, записані bag-файли та короткі візуальні підтвердження запуску. Такий підхід робить практичну частину прозорою: інший виконавець може відтворити послідовність дій і отримати порівнюваний результат.

Базова процедура починається зі статичної перевірки інерціального модуля. Сенсор розміщують нерухомо, записують IMU-потік і перевіряють сталість гіроскопа та акселерометра. Після цього виконують внутрішнє калібрування камери на AprilGrid, далі записують динамічний camera-IMU набір для Kalibr і переносять отримані параметри до конфігурації OpenVINS. Така послідовність відділяє перевірку сенсорів від запуску алгоритму й зменшує ризик помилкової інтерпретації результатів.

Робочий тест системи варто фіксувати у двох формах: як відео live-запуску та як набір даних для повторного програвання. Відео показує, що камера, IMU, ROS2-вузли та візуалізація працюють одночасно, а bag-файл дає змогу повторно запускати OpenVINS з тими самими вхідними даними. Разом ці матеріали підтверджують не лише візуальний факт роботи, а й технічну відтворюваність експерименту.

Повторне програвання записаних даних є зручним інструментом налагодження. Один і той самий набір сенсорних повідомлень можна використовувати для порівняння різних параметрів OpenVINS: моделі дисторсії, шумів IMU, частоти кадрів, параметрів супроводу ознак або способу синхронізації. Це особливо корисно на етапі доведення системи, оскільки зміни в конфігурації перевіряються без повторного фізичного запуску апаратної частини.

Для кількісної оцінки надалі можна застосувати контрольний маршрут із відомою геометрією. Наприклад, сенсорний модуль переміщують прямокутною траєкторією з поверненням у початкову точку, після чого аналізують розбіжність між початковою та кінцевою позою. Такий сценарій не потребує складної лабораторної інфраструктури, але дає зрозумілий показник дрейфу і дозволяє перейти від демонстрації працездатності до порівняння числових характеристик [21–25].

Скріншоти, відео, YAML-файли, PDF-звіти Kalibr і записи сенсорних потоків виконують різні ролі в доказовій базі. Відео демонструє live-ефект, YAML-файли фіксують параметри моделі, графіки Kalibr показують якість калібрування, а bag-файли забезпечують повторюваність. Саме поєднання цих матеріалів формує цілісну методику перевірки програмно-апаратного засобу.

Для практичного використання результатів доцільно вести короткий паспорт кожного експерименту. У ньому фіксують дату запуску, версію образу Ubuntu, стан робочого простору ROS2, параметри камери, частоту IMU, використані YAML-файли, тривалість запису та короткий опис траєкторії руху. Такий паспорт не потребує складного оформлення, але дає змогу швидко зрозуміти, за яких умов отримано конкретний результат і які параметри потрібно повторити.

Окремим критерієм повторюваності є сталість часових міток. Для візуально-інерціальної одометрії важливо, щоб кадри камери й повідомлення IMU надходили регулярно, без великих пропусків і різких затримок. Тому під час тестування варто контролювати не лише наявність повідомлень у топіках, а й інтервали між ними. Якщо частота IMU наближається до заданої, а камерний потік не має довгих пауз, OpenVINS отримує придатні умови для стабільного прогнозу та оновлення стану.

Ще одним корисним елементом є порівняння кількох запусків із однаковими калібрувальними параметрами. Якщо система поводиться подібно на схожих маршрутах, це підтверджує, що результат не є випадковим ефектом

одного відео. Якщо ж траєкторія помітно відрізняється, такий результат також корисний: він показує, які умови руху, освітлення або навантаження на обчислювальний модуль найбільше впливають на стабільність оцінювання.

Таким чином, запропонована методика не зводиться до формального запуску програми. Вона задає послідовність перевірок, за якою програмно-апаратний засіб можна повторно підготувати, запустити, відкалібрувати й оцінити за збереженими даними. Для бакалаврської роботи це важливо, оскільки демонструє не лише факт реалізації, а й готовність системи до подальших контрольованих експериментів.

3.12 Висновки до розділу 3

У третьому розділі виконано практичну реалізацію програмно-апаратного засобу візуально-інерціальної одометрії на базі Orange Pi 5 Pro, одного активного каналу камери IMX219, інерціального модуля ICM20948, Ubuntu 22.04, ROS2 Humble та OpenVINS. Реалізація охоплює повний інженерний ланцюг: підготовку операційної системи, створення робочого простору ROS2, організацію камерного та інерціального потоків, збирання OpenVINS, калібрування через Kalibr і перевірку запуску системи у live-режимі.

У роботі розглянуто підготовку AprilGrid, запис rosbag, формування camchain та imu.yaml, аналіз текстових результатів Kalibr і використання PDF-звітів із графіками похибок.

Live-тестування підтвердило працездатність базового програмно-апаратного ланцюга: камера формує відеопотік, IMU передає інерціальні вимірювання, ROS2 забезпечує обмін повідомленнями, а OpenVINS може бути налаштований на роботу з отриманими калібрувальними параметрами. Таким чином, у розділі показано не лише теоретичну можливість побудови VIO-системи, а й практичний шлях її реалізації на компактній обчислювальній платформі.

ВИСНОВКИ

У кваліфікаційній роботі розглянуто створення програмно-апаратного засобу візуальної одометрії на базі камери та обчислювального модуля. Основна ідея роботи полягає у використанні компактної апаратної платформи Orange Pi 5 Pro, одного відеоканалу камери IMX219 та інерціального модуля ICM20948 для побудови експериментальної монокулярної візуально-інерціальної системи. Такий підхід є актуальним для автономних пристроїв, які повинні оцінювати власний рух без постійної опори на зовнішню навігаційну інфраструктуру.

У першому розділі проаналізовано теоретичні основи одометрії, візуальної одометрії та візуально-інерціального оцінювання руху. Показано, що камера дає змогу оцінювати переміщення за зміною проєкції сцени, але монокулярна схема має фундаментальну проблему масштабу. IMU доповнює візуальні дані високочастотними вимірюваннями прискорень і кутових швидкостей, завдяки чому система отримує додаткову інформацію про рух між кадрами. Розглянуто фільтраційні та оптимізаційні підходи, зокрема MSCKF, а також обґрунтовано вибір OpenVINS як готової дослідницької платформи для VIO.

У другому розділі виконано проєктування програмно-апаратного засобу. Обґрунтовано вибір Orange Pi 5 Pro як обчислювальної платформи, камерного модуля Waveshare IMX219-83 Stereo Camera як джерела відеоданих та ICM20948 як джерела інерціальних вимірювань. Окремо уточнено, що попри наявність двох сенсорів у камерному модулі, практична реалізація використовує один відеоканал CAM:0; тому система описується як монокулярна візуально-інерціальна, а не як стереоодометрична. Також розглянуто архітектуру ROS2, роль OpenVINS і процес калібрування через Kalibr.

У третьому розділі описано практичну реалізацію системи: підготовку Ubuntu 22.04 і ROS2 Humble, створення робочого простору ROS2, підготовку OpenVINS, організацію ROS2-топіків, C++-захоплення кадрів, публікацію IMU-даних та методику калібрування через Kalibr. Окремо розглянуто використання

AprilGrid, запис rosbag2, формування YAML-конфігурацій і перенесення результатів калібрування до OpenVINS. Наявний відеоматеріал підтверджує, що система запускалася у live-режимі, відображала потік CAM:0 і формувала результат у RViz-подібному 3D-середовищі. Отже, працездатність базового програмно-апаратного ланцюга підтверджено якісно.

Разом з тим робота свідомо не подає live-тестування як повну метрологічну оцінку точності. Без збереженого rosbag2, окремого IMU-логу, результатів Kalibr і еталонної траєкторії неможливо коректно розрахувати абсолютну похибку, дрейф або стабільність масштабу. Отриманий результат слід розглядати як перший етап створення експериментальної системи, придатної для подальшого калібрування і кількісних досліджень.

Мету роботи досягнуто в межах первинної реалізації: спроєктовано структуру програмно-апаратного засобу, обґрунтовано вибір компонентів, описано програмну архітектуру, визначено роль калібрування і виконано live-перевірку запуску.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Barros A. M., Michel M., Moline Y., Corre G., Carrel F. A Comprehensive Survey of Visual SLAM Algorithms. *Robotics*. 2022. Vol. 11, No. 1. Article 24. DOI: https://doi.org/10.3390/robotics11010024.

2. Lai T. A Review on Visual-SLAM: Advancements from Geometric Modelling to Learning-Based Semantic Scene Understanding Using Multi-Modal Sensor Fusion. *Sensors*. 2022. Vol. 22, No. 19. Article 7265. DOI: https://doi.org/10.3390/s22197265.

3. Jia G., Li X., Zhang D., Xu W., Lv H., Shi Y., Cai M. Visual-SLAM Classical Framework and Key Techniques: A Review. *Sensors*. 2022. Vol. 22, No. 12. Article 4582. DOI: https://doi.org/10.3390/s22124582.

4. Tourani A., Bavle H., Sanchez-Lopez J. L., Voos H. Visual SLAM: What Are the Current Trends and What to Expect? *Sensors*. 2022. Vol. 22, No. 23. Article 9297. DOI: https://doi.org/10.3390/s22239297.

5. Servières M., Renaudin V., Dupuis A., Antigny N. Visual and Visual-Inertial SLAM: State of the Art, Classification, and Experimental Benchmarking. *Journal of Sensors*. 2021. Vol. 2021. Article 2054828. DOI: https://doi.org/10.1155/2021/2054828.

6. Neyestani A., Picariello F., Basiri A., Daponte P. Survey and Research Challenges in Monocular Visual Odometry. *2023 IEEE International Workshop on Metrology for Living Environment (MetroLivEnv)*. 2023. DOI: https://doi.org/10.1109/MetroLivEnv56897.2023.10164057.

7. Phan T.-D., Kim G.-W. Toward Specialized Learning-Based Approaches for Visual Odometry: A Comprehensive Survey. *Journal of Intelligent & Robotic Systems*. 2025. Vol. 111. Article 44. DOI: https://doi.org/10.1007/s10846-025-02245-0.

8. Campos C., Elvira R., Rodríguez J. J. G., Montiel J. M. M., Tardós J. D. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM. *IEEE Transactions on Robotics*. 2021. Vol. 37, No. 6. P. 1874-1890. DOI: https://doi.org/10.1109/TRO.2021.3075644.

9. Geneva P., Eckenhoff K., Lee W., Yang Y., Huang G. OpenVINS: A Research Platform for Visual-Inertial Estimation. *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020. DOI: https://doi.org/10.1109/ICRA40945.2020.9196524.

10. Rosinol A., Abate M., Chang Y., Carlone L. Kimera: An Open-Source Library for Real-Time Metric-Semantic Localization and Mapping. *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020. DOI: https://doi.org/10.1109/ICRA40945.2020.9196885.

11. Usenko V., Demmel N., Schubert D., Stückler J., Cremers D. Visual-Inertial Mapping with Non-Linear Factor Recovery. *IEEE Robotics and Automation Letters*. 2020. Vol. 5, No. 2. P. 422-429. DOI: https://doi.org/10.1109/LRA.2019.2961227.

12. von Stumberg L., Cremers D. DM-VIO: Delayed Marginalization Visual-Inertial Odometry. *IEEE Robotics and Automation Letters*. 2022. Vol. 7, No. 2. P. 1408-1415. DOI: https://doi.org/10.1109/LRA.2021.3140129.

13. Teed Z., Deng J. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. *Advances in Neural Information Processing Systems*. 2021. URL: https://arxiv.org/abs/2108.10869 (дата звернення: 09.05.2026).

14. Lin J., Zheng C., Xu W., Zhang F. R2LIVE: A Robust, Real-Time, LiDAR-Inertial-Visual Tightly-Coupled State Estimator and Mapping. *IEEE Robotics and Automation Letters*. 2021. Vol. 6, No. 4. P. 7469-7476. DOI: https://doi.org/10.1109/LRA.2021.3095515.

15. Shan T., Englot B., Ratti C., Rus D. LVI-SAM: Tightly-Coupled Lidar-Visual-Inertial Odometry via Smoothing and Mapping. *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021. P. 5692-5698. DOI: https://doi.org/10.1109/ICRA48506.2021.9561996.

16. Wisth D., Camurri M., Fallon M. VILENS: Visual, Inertial, Lidar, and Leg Odometry for All-Terrain Legged Robots. *IEEE Transactions on Robotics*. 2023. Vol. 39, No. 1. P. 309-326. DOI: https://doi.org/10.1109/TRO.2022.3193788.

17. Cao Z., Zhu Y., Yang J. An Online Temporal Calibration Method for Nonlinear Optimization-Based Stereo Visual-Inertial Odometry. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*. 2023. Vol. 237, No. 7. DOI: https://doi.org/10.1177/09596518231154033.

18. Huang K., Stachniss C. Online Initialization and Extrinsic Calibration for Monocular Visual-Inertial Odometry. *Sensors*. 2021. Vol. 21, No. 8. Article 2673. DOI: https://doi.org/10.3390/s21082673.

19. Liu Z., Shi D., Li R., Yang S. ESVIO: Event-Based Stereo Visual-Inertial Odometry. *Sensors*. 2023. Vol. 23, No. 4. Article 1998. DOI: https://doi.org/10.3390/s23041998.

20. Ribeiro-Gomes J., Gaspar J., Bernardino A. Event-Based Feature Tracking in a Visual Inertial Odometry Framework. *Frontiers in Robotics and AI*. 2023. Vol. 10. Article 994488. DOI: https://doi.org/10.3389/frobt.2023.994488.

21. Wang W., Zhu D., Wang X., Hu Y., Qiu Y., Wang C., Hu Y., Kapoor A., Scherer S. TartanAir: A Dataset to Push the Limits of Visual SLAM. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020. P. 4909-4916. DOI: https://doi.org/10.1109/IROS45743.2020.9341801.

22. Wenzel P., Wang R., Yang N., Cheng Q., Khan Q., von Stumberg L., Zeller N., Cremers D. 4Seasons: Benchmarking Visual SLAM and Long-Term Localization for Autonomous Driving in Challenging Conditions. *International Journal of Computer Vision*. 2025. Vol. 133. P. 1564-1586. DOI: https://doi.org/10.1007/s11263-024-02230-4.

23. Helmberger M., Morin K., Berner B., Kumar N., Cioffi G., Scaramuzza D. The Hilti SLAM Challenge Dataset. *IEEE Robotics and Automation Letters*. 2022. Vol. 7, No. 3. DOI: https://doi.org/10.1109/LRA.2022.3183759.

24. Chaney K., Cladera F., Wang Z., Bisulco A., Hsieh M. A., Korpela C., Kumar V., Taylor C. J., Daniilidis K. M3ED: Multi-Robot, Multi-Sensor, Multi-Environment Event Dataset. *IEEE/CVF Conference on Computer Vision and Pattern*

Recognition Workshops (CVPRW). 2023. P. 4016-4023. DOI: https://doi.org/10.1109/CVPRW59228.2023.00419.

25. Gehrig M., Aarents W., Gehrig D., Scaramuzza D. DSEC: A Stereo Event Camera Dataset for Driving Scenarios. *IEEE Robotics and Automation Letters*. 2021. Vol. 6, No. 3. P. 4947-4954. DOI: https://doi.org/10.1109/LRA.2021.3068942.

26. Teed Z., Deng J. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. Computer Vision - ECCV 2020. *Lecture Notes in Computer Science*. 2020. Vol. 12347. P. 402-419. DOI: https://doi.org/10.1007/978-3-030-58536-5_24.

27. Wang W., Hu Y., Scherer S. TartanVO: A Generalizable Learning-Based VO. *Proceedings of the 2020 Conference on Robot Learning. PMLR*. 2021. Vol. 155. P. 1761-1772. URL: https://proceedings.mlr.press/v155/wang21h.html (дата звернення: 09.05.2026).

28. Qiu H., Zhang X., Wang H., Xiang D., Xiao M., Zhu Z., Wang L. A Robust and Integrated Visual Odometry Framework Exploiting the Optical Flow and Feature Point Method. *Sensors*. 2023. Vol. 23, No. 20. Article 8655. DOI: https://doi.org/10.3390/s23208655.

29. Zheng F., Zhou L., Lin W., Liu J., Sun L. LRPL-VIO: A Lightweight and Robust Visual-Inertial Odometry with Point and Line Features. *Sensors*. 2024. Vol. 24, No. 4. Article 1322. DOI: https://doi.org/10.3390/s24041322.

30. Cen R., Huang J., Chen X., Cao L. A Low-Cost Visual Inertial Odometry for Mobile Vehicle Based on Double Stage Kalman Filter. *Signal Processing*. 2022. Vol. 197. Article 108537. DOI: https://doi.org/10.1016/j.sigpro.2022.108537.

31. OpenVINS Documentation. OpenVINS. URL: https://docs.openvins.com/ (дата звернення: 09.05.2026).

32. OpenVINS Sensor Calibration Guide. OpenVINS. URL: https://docs.openvins.com/gs-calibration.html (дата звернення: 09.05.2026).

33. OpenVINS ov_msckf Namespace Documentation. OpenVINS. URL: https://docs.openvins.com/namespaceov__msckf.html (дата звернення: 09.05.2026).

34. rpng/open_vins: OpenVINS Source Code Repository. GitHub. URL: https://github.com/rpng/open_vins (дата звернення: 09.05.2026).

35. Kalibr Wiki. ETH Zürich ASL. GitHub. URL: https://github.com/ethz-asl/kalibr/wiki (дата звернення: 09.05.2026).

36. Kalibr Camera-IMU Calibration. ETH Zürich ASL. GitHub. URL: https://github.com/ethz-asl/kalibr/wiki/camera-imu-calibration (дата звернення: 09.05.2026).

37. ethz-asl/kalibr: The Kalibr Visual-Inertial Calibration Toolbox. GitHub. URL: https://github.com/ethz-asl/kalibr (дата звернення: 09.05.2026).

38. OpenCV Camera Calibration Tutorial. OpenCV Documentation. URL: https://docs.opencv.org/4.x/d4/d94/tutorial_camera_calibration.html (дата звернення: 09.05.2026).

39. Installing ROS 2 Humble Hawksbill on Ubuntu Linux. ROS 2 Documentation. URL: https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debs.html (дата звернення: 09.05.2026).

40. Understanding ROS 2 Topics. ROS 2 Documentation. URL: https://docs.ros.org/en/humble/Tutorials/Beginner-CLI-Tools/Understanding-ROS2-Topics/Understanding-ROS2-Topics.html (дата звернення: 09.05.2026).

41. Writing a Simple C++ Publisher and Subscriber. ROS 2 Documentation. URL: https://docs.ros.org/en/humble/Tutorials/Beginner-Client-Libraries/Writing-A-Simple-Cpp-Publisher-And-Subscriber.html (дата звернення: 09.05.2026).

42. About Quality of Service Settings. ROS 2 Documentation. URL: https://docs.ros.org/en/humble/Concepts/Intermediate/About-Quality-of-Service-Settings.html (дата звернення: 09.05.2026).

43. About Different ROS 2 Middleware Vendors. ROS 2 Documentation. URL: https://docs.ros.org/en/humble/Concepts/Intermediate/About-Different-Middleware-Vendors.html (дата звернення: 09.05.2026).

44. sensor_msgs Package Documentation. ROS 2 Documentation. URL: https://docs.ros.org/en/ros2_packages/humble/api/sensor_msgs/ (дата звернення: 09.05.2026).

45. Recording and Playing Back Data with rosbag2. ROS 2 Documentation. URL: https://docs.ros.org/en/humble/Tutorials/Beginner-CLI-Tools/Recording-And-Playing-Back-Data/Recording-And-Playing-Back-Data.html (дата звернення: 09.05.2026).

46. About tf2. ROS 2 Documentation. URL: https://docs.ros.org/en/humble/Concepts/Intermediate/About-Tf2.html (дата звернення: 09.05.2026).

47. About Launch. ROS 2 Documentation. URL: https://docs.ros.org/en/humble/Concepts/Basic/About-Launch.html (дата звернення: 09.05.2026).

48. image_pipeline Package Documentation. ROS 2 Documentation. URL: https://docs.ros.org/en/humble/p/image_pipeline/ (дата звернення: 09.05.2026).

49. camera_calibration Package. ROS Index. URL: https://index.ros.org/p/camera_calibration/ (дата звернення: 09.05.2026).

50. camera_ros Package Documentation. ROS 2 Documentation. URL: https://docs.ros.org/en/ros2_packages/humble/api/camera_ros/ (дата звернення: 09.05.2026).

51. Raspberry Pi Camera Software Documentation. Raspberry Pi Documentation. URL: https://www.raspberrypi.com/documentation/computers/camera_software.html (дата звернення: 09.05.2026).

52. libcamera Documentation. libcamera Project. URL: https://libcamera.org/docs.html (дата звернення: 09.05.2026).

53. DDS Tuning Information. ROS 2 Documentation. URL: https://docs.ros.org/en/humble/How-To-Guides/DDS-tuning.html (дата звернення: 09.05.2026).

54. Standard QoS Policies. eProsima Fast DDS Documentation. URL: https://fast-dds.docs.eprosima.com/en/v2.14.5/fastdds/dds_layer/core/policy/standardQosPolicies.html (дата звернення: 09.05.2026).

55. Cyclone DDS Configuration File Reference. Eclipse Cyclone DDS Documentation. URL: https://cyclonedds.io/docs/cyclonedds/latest/config/config_file_reference.html (дата звернення: 09.05.2026).

56. IMX219-83 Stereo Camera. Waveshare. URL: https://www.waveshare.com/product/raspberry-pi/boards-kits/imx219-83-stereo-camera.htm (дата звернення: 09.05.2026).

57. IMX219-83 Stereo Camera Wiki. Waveshare. URL: https://www.waveshare.com/wiki/IMX219-83_Stereo_Camera (дата звернення: 09.05.2026).

58. Orange Pi 5 Pro. Orange Pi. URL: https://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/details/Orange-Pi-5-Pro.html (дата звернення: 09.05.2026).

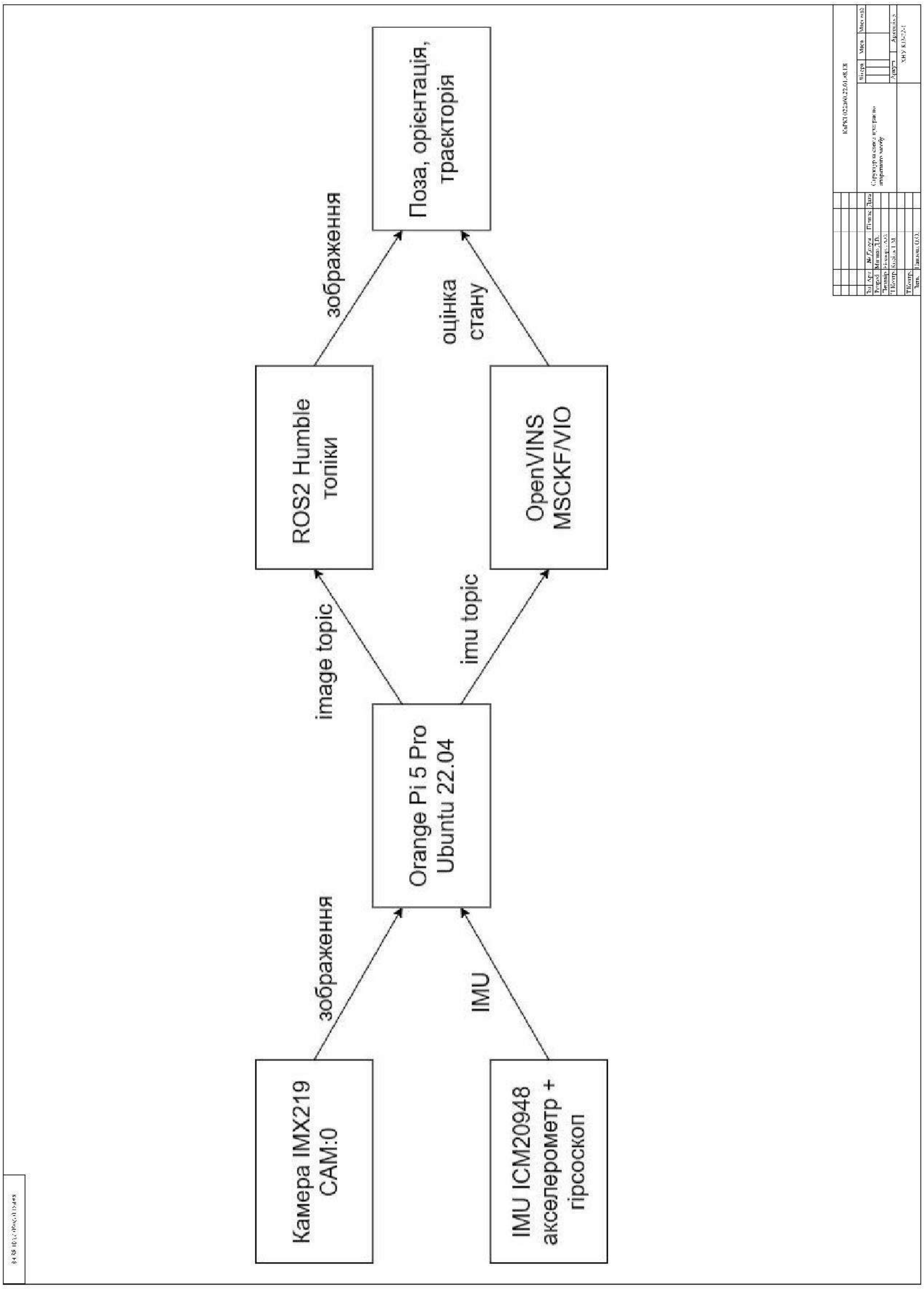
59. Orange Pi 5 Pro User Manual. Orange Pi. 2024. URL: https://orangepi.net/wp-content/uploads/2024/05/OrangePi_5_Pro_RK3588S_User-Manual_v1.1.pdf (дата звернення: 09.05.2026).

60. ICM-20948 9-Axis Motion Tracking Device. TDK InvenSense. URL: https://invensense.tdk.com/en-us/products/motion-tracking/9-axis/icm-20948/ (дата звернення: 09.05.2026).

ДОДАТОК А

(обов’язковий)

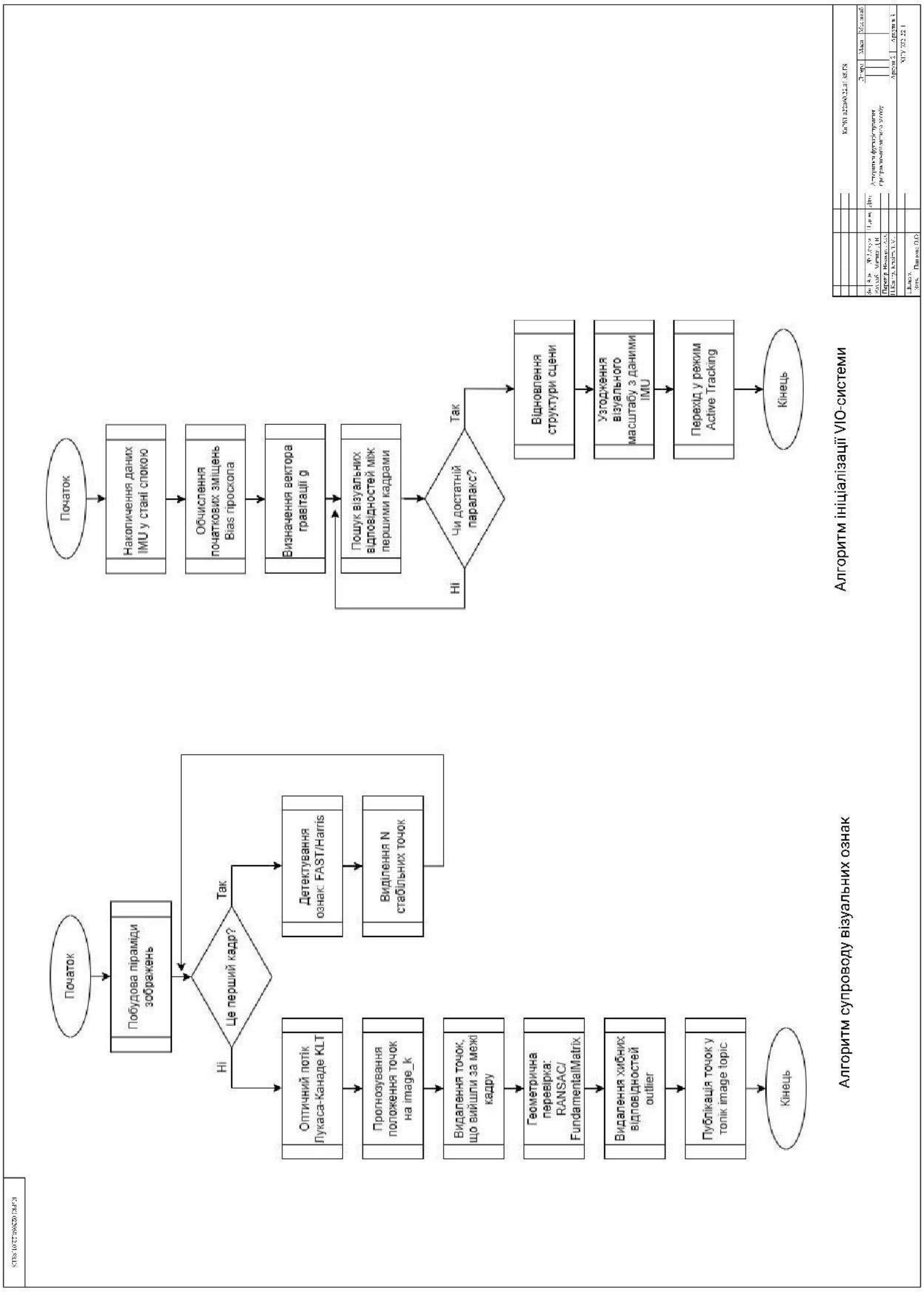
Копія креслення «Структурна схема програмно-апаратного засобу»



ДОДАТОК Б

(обов’язковий)

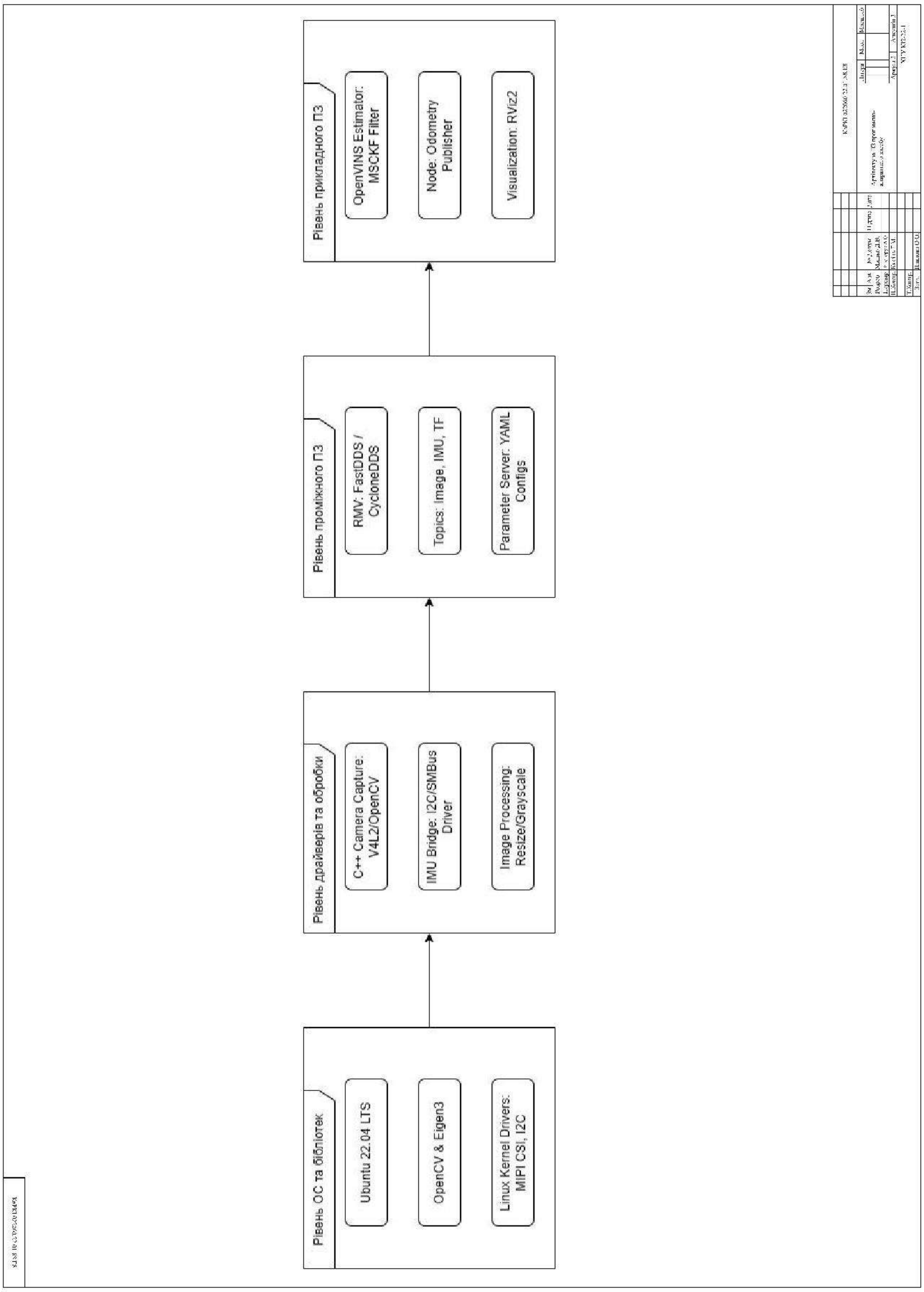
Копія креслення «Алгоритми функціонування програмно-апаратного засобу»



ДОДАТОК В

(обов'язковий)

Копія креслення «Архітектура ПЗ програмно-апаратного засобу»



Зав. кафедри КІС
д-р. філософії Ользі ПАВЛОВІЙ

Дмитро МИШКО

ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-22-1

ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів академічної відповідальності, ознайомлений (а). Про використання спеціалізованих програмних засобів (СПЗ) StrikePlagiarism та Anti-Plagiarism для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений (а). Надаю університету право на передачу моєї роботи для обробки та збереження в базах даних СПЗ і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються СПЗ.

Також надаю свою згоду на обробку й збереження університетом моєї роботи в Інституційному репозитарії Хмельницького національного університету.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

2 травня 2026 року

Anti-Plagiarism (http://ap.km.ua) v-15.701

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 14%**

ID: 272136 Назва: БКР Програмно-апаратний засіб візуальної одометрії на базі камери та обчислювального модуля Додано в БД: 2026-05-25 Автора: Дмитро МИШКО Керівники: Андрій НІЧЕПОРУК Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	**Лексеми**	**Символи**	**Лексеми**
	100026	872	1362 (1%)	17 (2%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Дмитро МИШКО

Співавтор:

Назва: Програмно-апаратний засіб візуальної одометрії на базі камери та обчислювального модуля

Експерт: Андрій НІЧЕПОРУК

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:5.41%

Коефіцієнт подібності 2:1.5%

Мікропробіли: 3

Заміна букв: 0

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2026-05-25 11:01:59.0

Після аналізу Звіту подібності констатую наступне:

☒ **Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

☐ **Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

☐ **Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.**

Обгрунтування:

2026-05-25 Доцент Андрій Нічепорук

Дата експерт

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Мишко Дмитро Вадимович

Тема: Програмно-апаратний засіб візуальної одометрії на базі камери та обчислювального модуля

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень ___3____ Кількість сторінок записки ___65___

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є розроблення та перевірка програмно-апаратного засобу візуальної одометрії.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі кваліфікаційної роботи проведено аналіз предметної області візуально-інерціальної одометрії, розглянуто геометричні основи візуальної одометрії, особливості монокулярної схеми, проблему масштабу, використання інерціальних вимірювань, а також фільтраційні та оптимізаційні підходи до оцінювання руху. У другому розділі виконано проєктування програмно-апаратного засобу: визначено функціональні, часові, обчислювальні та калібрувальні вимоги; обґрунтовано вибір обчислювальної платформи Orange Pi 5 Pro, камерного модуля IMX219 та інерціального модуля ICM20948; описано програмну архітектуру ROS2, логіку роботи OpenVINS і методику калібрування камери та IMU засобами Kalibr. У третьому розділі виконано практичну реалізацію та тестування засобу, а саме: підготовлено середовище Ubuntu 22.04 і ROS2 Humble, створено робочий простір ROS2, організовано камерні та інерціальні топіки, реалізовано C++-вузол захоплення кадрів, описано публікацію IMU-даних, проведено калібрування через Kalibr, налаштовано параметри OpenVINS і виконано live-перевірку працездатності системи.

4. Позитивні сторони роботи: актуальність теми, практична спрямованість роботи, використання сучасних програмних засобів.

5. Негативні сторони роботи: недостатньо повно наведено кількісну оцінку точності траєкторії відносно еталонних вимірювань.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

8. Інші зауваження: ______________________________

9. Оцінка дипломної роботи: добре

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) ______________
Мартинюк Валерій Володимирович, д.т.н., проф., професор кафедри АКІТтаР

"28" травня 2026 р. ______________(підпис)

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Програмно-апаратний засіб візуальної одометрії на базі камери та обчислювального модуля
Автор Дмитро МИШКО
Освітня програма Комп'ютерна інженерія та програмування
Рівень вищої освіти перший (бакалаврський)
Спеціальність 123 Комп'ютерна інженерія
Науковий керівник: к.т.н., доцент Андрій НІЧЕПОРУК

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;

2) окремі виявлені збіги є загальновживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;

3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 5,41%; та системою Anti-Plagiarism складає 1%, що, з урахуванням наведених обгрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

25.05.2026

Завідувач кафедри	Підпис	Ольга ПАВЛОВА Ім'я, ПРІЗВИЩЕ
Гарант освітньої програми	Підпис	Андрій НІЧЕПОРУК Ім'я, ПРІЗВИЩЕ
Керівник кваліфікаційної роботи	Підпис	Андрій НІЧЕПОРУК Ім'я, ПРІЗВИЩЕ