Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мережних технологій факультету інформатики

АЛГОРИТМ ОПТИЧНОЇ НАВІГАЦІЇ ДРОНА

Текстова частина
магістерської роботи
за спеціальністю „Інженерія програмного забезпечення” 121

Керівник магістерської роботи
д.т.н., проф. Глибовець А.М.

_____
(підпис)

“____” _____ 2025 р.

Виконала студентка
Бойко Х.А.
“____” _____ 2025 р.

Київ 2025

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра мережних технологій факультету інформатики

ЗАТВЕРДЖУЮ
Зав. кафедри інформатики
к.ф-м.н., доц. Гороховський С.С.
_____
(підпис)
“____” _____ 2025 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на магістерську роботу

студенту 2 р.н. магістерської програми Інженерія Програмного Забезпечення Бойко Христині Анатоліївні

Розробити Алгоритм оптичної навігації дрона

Зміст текстової частини до магістерської роботи:

Зміст
Анотація
Вступ
1 Огляд предметної області
2 Розробка алгоритму оптичної навігації
Висновки
Список літератури
Додатки (за необхідністю)

Дата видачі “____” _____ 2025 р.

Керівник
А.М. Глибовець, доктор технічних наук, професор _____
(підпис)

Завдання отримала
Х.А. Бойко _____
(підпис)

Тема: Оптична навігація

Календарний план виконання роботи:

№ п/п	Назва етапу дипломного проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на дипломну роботу	04.11.2024	
2.	Аналіз технічної літератури за темою оптичної навігації БПЛА	07.02.2025	
3.	Написання тексту теоретичної частини магістерської роботи	14.03.2025	
4.	Реалізація попередньої обробки карти	21.03.2025	
5.	Розробка алгоритму оптичної навігації за кадрами з камери дрона та підготовленою картою	11.04.2025	
6.	Реалізація алгоритму оптичної навігації з ORB та SuperPoint + SuperGlue підходами виявлення і зіставлення ключових точок	02.05.2025	
7.	Тестування алгоритму оптичної навігації й аналіз отриманих результатів	16.05.2025	
8.	Написання тексту практичної частини магістерської роботи	30.05.2025	
9.	Створення презентації для доповіді	02.06.2025	
10.	Коригування й остаточне оформлення роботи	05.06.2025	
11.	Захист магістерської роботи	09.06.2025	

Студентка Бойко Х.А.

Керівник Глибовець А.М. “____” _____ 2025 р.

Зміст

Анотація 6
Вступ 7
Основна частина 11
Розділ 1. Огляд предметної області 11
Оптична навігація БПЛА 11
Класифікація типів навігації 11
Завдання та методи оптичної навігації 11
Візуальні сенсори 13
Інші сенсори 13
Злиття сенсорних даних 15
Архітектура навігаційної системи 17
Вимоги до навігаційних систем БПЛА 18
Навігаційні системи на основі карт 19
Навігаційна система, незалежна від карт 19
Навігаційна система, залежна від карт 20
Навігаційна система, що створює карту 22
Візуальна одометрія 24
Види візуальної одометрії 24
Приклад сучасної реалізації візуальної одометрії 25
Застосування комп’ютерного зору для оптичної навігації 27
Алгоритми для виявлення та опису ключових точок 27
Методи пошуку відповідностей між ознаками 31
Геометричні перетворення 32
Нейромережеві методи для виявлення та зіставлення ключових точок 34
Сучасні рішення 38
Висновки до розділу 1 42
Розділ 2. Розробка алгоритму оптичної навігації 43
Обґрунтування вибору засобів розробки 43
Попередня обробка карти місцевості для навігації 44
Розбиття карти на фрагменти 44

Отримання візуальних ознак з фрагментів карти 44
Алгоритм навігації за підготовленою картою та кадрами з дрона 46
Створення кадру з дрона 47
Отримання візуальних ознак з кадру дрона 47
Вибір фрагментів для порівняння з кадром дрона 49
Зіставлення дескрипторів між фрагментом карти і кадром 49
Визначення позиції дрона 50
Стратегії руху до цілі 51
Завершення руху 51
Результати роботи 52
Висновки до розділу 2 54
Висновки 55
Список використаних джерел 56
Список використаних зображень 60

Анотація

Робота присвячена розробці алгоритму оптичної навігації дрона за зображеннями з камери та попередньо обробленим ортофото карти місцевості. Було реалізовано підготовку карти: розбиття на фрагменти, обчислення їхніх піксельних та GPS координат, виділення ключових точок і дескрипторів двома підходами: класичним методом ORB та нейромережевим методом SuperPoint + SuperGlue. Усі візуальні ознаки й координати збережено у стисненому форматі для пошуку відповідностей з кадрами.

Під час навігації кожен кадр з камери дрона симулюється з випадковим кутом обертання, з нього витягуються візуальні ознаки, які порівнюються з ознаками фрагментів карти. Позиція дрона визначається через обчислення гомографії між кадром і картою. У разі невдалої локалізації дрон продовжує рух у напрямку до цільової точки, використовуючи кут між стартовими та цільовими GPS-координатами. У системі можна встановлювати стартову зону запуску дрона та точку цілі на карті, фіксувати кожен крок місії, візуалізувати реальну та оцінену траєкторії на карті.

Алгоритм було реалізовано мовою Python із використанням бібліотеки OpenCV для методу ORB, попередньо натренованих моделей SuperPoint і SuperGlue [19], ортофото карти із платформи OpenAerialMap [18].

Ключові слова: оптична навігація, ключові точки, дескриптори, нейронні мережі, ORB, SuperPoint, SuperGlue, гомографія.

Вступ

Актуальність теми

Сучасна російсько-українська війна показала вразливість безпілотних літальних апаратів (БПЛА) до засобів радіоелектронної боротьби (РЕБ). Глушіння та спуфінг (підміна) GPS-сигналів призводить до помилок у позиціонуванні, втрати керованості та зниження ефективності бойових місій. Для протидії цим загрозам застосовуються апаратні рішення, зокрема оптоволоконне керування та GPS-фільтри. Оптоволоконні дрони захищені від глушіння, проте мають обмежену дальність польоту, маневреність і вантажопідйомність [1]. GPS-фільтри, встановлені між приймачем та польотним контролером, аналізують координати в режимі реального часу та блокують спотворені сигнали. Завдяки такому пристрою в українських БПЛА «Привид» під час спуфінгу автоматично активується інерційна навігація до відновлення коректного GPS-сигналу [2].

Оскільки апаратні засоби мають обмеження, дедалі більшої ваги набуває оптична навігація — підхід, що визначає положення дрона за зображеннями з бортових камер і є стійким до РЕБ. Вже існують приклади застосування оптичної навігації. Наприклад, комплекс MONOLITH U1 від 360-DRONE виконує повністю автономний політ дрона, використовуючи дані про рельєф, зони дії ППО та РЕБ, супутникові мапи висот, системи ситуаційної обізнаності. У комплексі реалізована візуальна одометрія для орієнтації за зображеннями місцевості [3]. У грудні 2024 року Міністерство цифрової трансформації України представило навігаційну систему для БПЛА AI4UAV, яка реалізує візуальну навігацію для польотів у будь-який час доби, з картами або без, як заміну GNSS [4].

Паралельно розвивається автономія управління. Автоматизація польоту та використання штучного інтелекту в українських безпілотних системах спрямовані на зниження залежності від навичок операторів і постійного зв’язку. Навіть базова автономна навігація підвищує успішність ударів з 10–20% до 70–80%, зменшуючи втрати техніки та потребу в повторних місіях. На полі бою поступово впроваджуються окремі AI-функції, такі як навігація, розпізнавання

цілей, аналіз відео та текстів. Також активно триває розробка компактних модулів для різних типів дронів, які можна швидко адаптувати до умов фронту [5].

Яскравим прикладом автономної навігації стала операція «Павутина», проведена 1 червня 2025 року Службою безпеки України, у якій 117 FPV-дронів з вибухівкою одночасно атакували 4 російські аеродроми стратегічної авіації. Дрони були заздалегідь завезені на територію Російської Федерації, заховані у вантажівках під дахами мобільних дерев'яних будинків, які відкривалися дистанційно в момент запуску. Для автоматичного розпізнавання і наведення на цілі використовувався штучний інтелект, натренований на реальних зразках літаків [20], а для зв'язку — російські телекомунікаційні мережі [21]. У результаті атаки було уражено 41 літак, зокрема Ту-95МС, Ту-22М3, Ту-160 і А-50У, що становить приблизно 34% стратегічної авіації РФ [22].

Таким чином розробка алгоритмів оптичної навігації для дронів є надзвичайно актуальним завданням в умовах сучасної війни.

Мета та завдання магістерської роботи

Мета: Розробити алгоритм оптичної навігації дрона через аналіз зображень з камери та попередньо обробленої карти із використанням класичних та нейромережевих методів виявлення та зіставлення ключових точок.

Завдання: Проаналізувати сучасні методи оптичної навігації та методи зіставлення ознак. Виконати попередню обробку ортофото карти. Розробити алгоритм оптичної навігації БПЛА за картою та кадрами з використанням ORB і SuperPoint + SuperGlue. Забезпечити стійку навігацію до цільової точки.

Об'єкт дослідження

Оптична навігація БПЛА в умовах недоступності GPS-сигналу.

Предмет дослідження

Алгоритми виявлення, опису та зіставлення ключових точок між зображеннями з камери дрона і попередньо обробленими фрагментами карти для визначення позиції дрона.

Методи дослідження

1. Алгоритми комп'ютерного зору: ORB та гомографія.
2. Нейромережеві методи: SuperPoint для виявлення ознак, SuperGlue для зіставлення ознак.
3. Методи просторової трансформації та геоприв'язки: афінні перетворення, проєктивна гомографія.
4. Методи візуалізації та аналізу траєкторій.

Джерела дослідження

- Публікації з наукових журналів: Drones, Sensors.
- Репозиторії відкритого коду з реалізацією моделей SuperPoint і SuperGlue.
- Ортофото з OpenAerialMap, технічна документація бібліотек OpenCV, PyTorch, Rasterio.

Наукова новизна одержаних результатів

- Реалізовано попередню обробку ортофото карти для оптичної навігації.
- Проведено порівняння двох підходів для локалізації без GPS: ORB і SuperPoint + SuperGlue.
- Розроблено механізм вибору фрагментів карти для зіставлення з кадром.
- Запропоновано стратегії руху до цільової точки з урахуванням невизначеності позиції.

Практичне значення одержаних результатів

Розроблений алгоритм може бути інтегрованим в навігаційні модулі дронів для автономного позиціювання та навігації в GPS-недоступних умовах.

Постановка задачі

1. Провести аналіз сучасних підходів до оптичної навігації БПЛА, включаючи класичні та нейромережеві методи виявлення та зіставлення ключових точок.
2. Розробити метод підготовки ортофото карти: розбиття на фрагменти, визначення їхніх координат, обчислення дескрипторів.
3. Реалізувати метод отримання кадрів із камери дрона та їх обробки.
4. Зіставляти кадри дрона з фрагментами карти для визначення позиції дрона за допомогою ORB та SuperPoint + SuperGlue.
5. Оцінити точність локалізації дрона різними методами зіставлення ознак.
6. Реалізувати навігацію дрона у напрямку руху до цілі.
7. Побудувати візуалізацію траєкторії дрона на основі оцінених позицій та порівняти її з реальною траєкторією.

Структура роботи

Магістерська робота містить анотацію, вступ, два основні розділи, висновки, список використаних джерел.

У першому розділі досліджено предметну область: розглянуто класифікацію навігаційних систем, особливо приділено увагу підходам оптичній навігації, навігації на основі карт, висвітлено методи виявлення та опису ключових точок та алгоритми зіставлення ознак, зокрема нейромережеві. Також проаналізовано сучасні практичні рішення для військових цілей.

У другому розділі описано реалізований алгоритм оптичної навігації, процес попередньої обробки ортофото карти, механізм симуляції кадрів з камери дрона, їх обробку та зіставлення з фрагментами карти. Представлено метод локалізації та стратегію навігації дрона до цілі. Наведено результати експериментів із порівнянням траєкторій, точності та швидкодії різних підходів до виявлення і зіставлення ключових точок.

Основна частина

Розділ 1. Огляд предметної області

Оптична навігація БПЛА

Класифікація типів навігації

Розрізняють три основні типи навігації: інерційна, супутникова та візуальна. Супутникова система (GPS) забезпечує глобальне покриття, високу точність, має низьку вартість, проте є вразливою до перешкод, потребує відкритого простору для надійного сигналу й залежить від кількості доступних супутників, що впливає на точність позиціонування. Інерційна система (INS) використовує такі пристрої, як акселерометр, гіроскоп, для визначення прискорення і кутових швидкостей БПЛА. Недоліком цієї системи є втрата точності у визначенні положення, швидкості, орієнтації дрона через накопичення похибок у вимірюваннях. Візуальна система є стійкою до перешкод, недорогою і має високу роздільну здатність, що дає змогу точно визначати місцезнаходження дрона і будувати карту середовища. Однак через обмежений діапазон БПЛА має перебувати близько до цілі для точнішої локалізації. Ефективність системи залежить від умов освітлення: тіні, відблиски, темрява можуть ускладнювати розпізнавання ознак. Окрім того, візуальна система не працюватиме в однотипних середовищах без характерних орієнтирів (пустеля, сніговий ландшафт). Через комбінацію методів і поєднання даних з супутникової, інерційної і оптичної систем можна значно покращити навігацію БПЛА [6].

Завдання та методи оптичної навігації

Основними завданнями навігації є локалізація, картографування і планування маршруту. Спочатку БПЛА визначає своє місцезнаходження. Далі створюється візуальна карта для уникнення перешкод і пошуку оптимальних зон для посадки. За допомогою алгоритму оптимізації планується найкоротший маршрут [6].

Система навігації складається з трьох підсистем: оцінки положення БПЛА, виявлення й уникнення перешкод, візуального сервоуправління [6].

Локалізація та картографування

Оцінка позиції й орієнтації БПЛА здійснюється на основі даних з GPS, IMU, візуальних, лазерних та ультразвукових сенсорів. У складних умовах, де сигнал GPS нестабільний або відсутній (наприклад, у приміщеннях чи щільній забудові), ефективність супутникової навігації значно знижується. Для точнішого позиціонування зазвичай комбінують GPS із даними інерційної навігаційної системи (INS) за допомогою фільтра Калмана. Через недоліки таких систем саме візуальні методи стають ключовими для навігації в GPS-недоступному середовищі. Візуальні системи працюють на основі зображень, що надходять із монокулярних, стереокамер або панорамних камер. Оцінка положення може бути вирішена двома основними підходами: візуальною одометрією (VO) й одночасною локалізацією і картографуванням на основі візуалізації (VSLAM). VO визначає траєкторію БПЛА покадрово, обчислюючи траєкторію в кожен момент часу й не зберігаючи попередніх координат. Цей метод базується на виявленні ключових точок і співставленні ознак між кадрами. Алгоритми VSLAM полягають в одночасній побудові послідовної карти середовища й визначенні положення БПЛА на карті. Прикладами таких алгоритмів є паралельне відстеження та картографування (PTAM) і монолокалізація та картографування (MonoSLAM) [6].

Виявлення та уникнення перешкод

Виявлення та уникнення перешкод передбачає виявлення бар'єрів й обчислення відстаней між БПЛА та об'єктами за допомогою візуальних сенсорів і сенсорів дальності, таких як радар, ультразвук, інфрачервоне випромінювання. Для вирішення цієї задачі існують два основні підходи: на основі оптичного потоку та SLAM. Через обробку зображень й оптичний потік можна створити локальну карту та визначити глибину сцени для уникнення перешкод [6].

Зображення з камер є джерелом зворотного зв’язку для візуального сервоуправління, що відповідає за стабільність БПЛА під час польоту та його рух за маршрутом через керування командами маневрування. Для стабілізації застосовують різні системи керування внутрішнім контуром, зокрема PID-

регулятори, оптимальне керування, ковзний режим, нечітку логіку та каскадне керування [6].

Планування маршруту

Існує кілька підходів до планування, серед яких виділяють глобальне, локальне та методи глибокого навчання для оптимізації маршруту. Глобальне планування використовує орографічні карти та алгоритми A*, D* або метаевристичні підходи (наприклад, генетичні алгоритми, мурашині системи, імітацію відпалу) для побудови оптимального шляху на основі наперед заданої карти. Локальне планування орієнтоване на адаптацію до змін навколишнього середовища в реальному часі та ґрунтується на методах, такі як штучні потенційні поля й нечітка логіка. Методи глибокого навчання, зокрема навчання з підкріпленням, застосовуються для створення стратегій навігації в частково відомому або динамічному середовищі, де агент навчається через зворотній зв’язок і приймає рішення без повної карти [6].

Візуальні сенсори

У системах оптичної навігації використовують різні типи візуальних сенсорів: монокулярні камери, стереокамери, RGB-D і камери типу «риб’яче око». Завдяки універсальності та низькій вартості монокулярні камери найчастіше застосовуються в компактних системах, але вони не здатні отримати інформацію про глибину сцени. Стереокамери оцінюють глибину зображення на основі принципу паралаксу. RGB-D камери передають як карти глибини, так і кольорові зображення завдяки інфрачервоним сенсорам, проте через невеликий діапазон дії вони обмежені приміщеннями. Ширококутні камери типу «риб'яче око» охоплюють значну частину простору, що сприяє кращому уникненню перешкод у складних умовах [6].

Інші сенсори

Глобальна супутникова навігаційна система (GNSS) є основним джерелом абсолютного позиціювання для БПЛА. Принцип дії GNSS базується на вимірюванні псевдодальності — часу проходження сигналу від супутника до

приймача, помноженого на швидкість світла. Для визначення тривимірного положення необхідно щонайменше чотири супутники. Точність позиціювання погіршується за несприятливої геометрії супутників або при низькій висоті польоту, де більша ймовірність затінення антени. Тому GNSS-сенсори рідко використовуються окремо: вони об’єднуються з іншими сенсорами для формування повного навігаційного вектора.

Інерційний вимірювальний пристрій (IMU) використовує гіроскопи та акселерометри для оцінки кутових швидкостей і прискорень БПЛА, що дає змогу визначати його положення та орієнтацію методом числення шляху (dead-reckoning). Основною проблемою IMU є накопичення похибок через систематичні та випадкові помилки сенсорів, зокрема зсуви, масштабні коефіцієнти, перехресні зв'язки та шум. Для підвищення точності IMU зазвичай інтегрують з іншими сенсорами, наприклад GNSS. Крен (roll) і тангаж (pitch) оцінюються на основі даних акселерометрів, тоді як курс (yaw) визначається за допомогою магнітометрів. Саме тому інерційні сенсори часто об'єднуються з магнітометрами в системи AHRS (Attitude and Heading Reference System), які формують повноцінну орієнтацію дрона в просторі. Найчастіше в БПЛА застосовуються недорогі MEMS-IMU, які мають вищу нестабільність зміщення та шум, що призводить до швидкого зростання похибок при відмові GNSS.

Магнітометри (компаси) використовуються для визначення курсу БПЛА через вимірювання локального магнітного поля. Спочатку дані проєктуються на горизонтальну площину з урахуванням крену та тангажу, після чого обчислюється курс за допомогою арктангенса відношення горизонтальних компонент магнітного вектора. Остаточне значення курсу коригується з урахуванням локального магнітного відхилення. Магнітометри чутливі до електромагнітних завад, тому мають бути правильно розташовані.

Альтиметри, зокрема барометричні, оцінюють висоту за різницею атмосферного тиску між поточним положенням і початковим. Також застосовують мініатюрні радарні або лазерні альтиметри для навігації за рельєфом (Terrain-Aided

Navigation, TAN). Цей метод порівнює серію висотних вимірювань із цифровою моделлю рельєфу для визначення положення, що особливо актуально в умовах відсутності GPS. Новітні реалізації TAN використовують штучні нейронні мережі, як-от B-сплайни та LSTM, для адаптивного налаштування EKF-фільтра на складних ділянках місцевості [7].

Злиття сенсорних даних

Навігація БПЛА залежить від об'єднання даних із різних сенсорів: інерційних (акселерометри, гіроскопи), супутникових (GNSS), візуальних (камери), лазерних (LiDAR), магнітометрів тощо. Жоден із сенсорів окремо не дає повної та достовірної інформації: наприклад, GNSS може бути недоступною в приміщеннях або при глушінні сигналу, візуальні сенсори не працюють у темряві, а гіроскопи схильні до накопичення похибки. Тому для підвищення точності та надійності навігації застосовується злиття даних з кількох джерел у межах єдиної фільтраційної моделі, за допомогою якої компенсуються похибки окремих сенсорів і покращується загальна оцінка стану системи. Залежно від складності задачі, доступних обчислювальних ресурсів та вимог до точності, використовуються різні підходи: комплементарні фільтри (CF, NLCF), фільтри Калмана (KF, EKF, UKF), методи Монте-Карло (фільтр частинок) [7].

Комплементарні фільтри – це простий і обчислювально дешевий спосіб злиття незалежних вимірювань з шумом однієї і тієї ж змінної. Вони широко застосовуються в системах орієнтації та курсової стабілізації (Attitude and Heading Reference Systems) для БПЛА. Наприклад, акселерометри можуть давати стабільну оцінку кута крену й тангажу протягом тривалого періоду часу, але з високочастотним шумом. Натомість гіроскопи накопичують помилки під час інтегрування. Комплементарна фільтрація поєднує сигнали від різних сенсорів за допомогою комбінації фільтрів, налаштованих параметром τ: оцінки крену та тангажу з акселерометра проходять через фільтр низьких частот, а вимірювання з гіроскопа – через фільтр високих частот [7].

Фільтр Калмана (Kalman Filter) – це алгоритм, який оптимально оцінює стан системи на основі серії вимірювань з шумом із різних сенсорів. Цей метод рекурсивно використовує байєсівський підхід, щоб максимізувати спільну ймовірність станів і вимірювань. Фільтр Калмана виконується в два етапи: передбачення і вимірювання. Етап передбачення прогнозує вектор стану навігації та його невизначеність шляхом прогону динаміки системи через модель, а етап вимірювання інтегрує дані із сенсорів для коригування передбачення та отримання оптимальної оцінки. Оптимальність досягається зваженим об'єднанням вимірювань і прогнозом, де ваги визначаються відповідними рівнями невизначеності [7].

Популярними варіантами фільтра Калмана є розширений фільтр Калмана (Extended Kalman Filter), ітеративний розширений фільтр Калмана (It-EKF), сигма-точковий фільтр Калмана (Sigma point Kalman Filter), інваріантний фільтр Калмана (IEKF). Розширений фільтр Калмана здатний працювати з нелінійними системами, лінеаризуючи моделі стану та вимірювань за допомогою ряду Тейлора і матриці часткових похідних (Якобі). Такий підхід широко застосовується у багатьох практичних навігаційних задачах. Водночас алгоритм має обмеження, адже він передбачає гаусівський характер шумів і майже лінійну поведінку системи, що може призводити до значних похибок під час динамічних маневрів БПЛА. Для зменшення помилок розроблено вдосконалені підходи: ітеративний EKF (It-EKF), який багаторазово уточнює модель навколо оновленої оцінки стану, та геометрично обґрунтовані модифікації, що враховують інваріантність до обертань і трансляцій у системах з симетріями. EKF обмежений тим, що використовує лише перший порядок розкладу в ряд Тейлора, що призводить до втрати точності при сильній нелінійності. Натомість сигма-точковий фільтр Калмана (SPKF) застосовує набір спеціально обраних точок – сигма-точок, які апроксимують розподіл і статистику випадкової змінної без необхідності обчислення похідних. Цей підхід досягає більш точного опису середнього значення та коваріації після проходження через нелінійне перетворення. Одним із найпоширеніших варіантів є нековзний фільтр Калмана

(Unscented Kalman Filter), що застосовує метод нековзної трансформації (Unscented Transform). UT оцінює статистику нелінійного перетворення випадкової змінної, використовуючи лише скінчену кількість сигма-точок. UKF не потребує аналітичного диференціювання, дає більш точне наближення й особливо добре працює з негаусівськими шумами. Алгоритм UKF складається з послідовності кроків: генерації сигма-точок, їхнє перетворення через динамічну модель, обчислення статистик стану та вимірювання, обчислення коваріацій та оновлення оцінки. Інваріантний фільтр Калмана (Invariant Extended Kalman Filter) враховує симетрії (інваріантності) системи, такі як обертання чи переміщення, що покращує оцінку в умовах складної геометрії [7].

Фільтр частинок (Particle Filter) належить до класу методів Монте-Карло. На відміну від фільтрів Калмана, які описують розподіл ймовірностей стану через середнє та коваріацію, фільтр частинок представляє його множиною зразків (частинок), кожна з яких є гіпотезою про справжній стан системи. Такий підхід дає змогу оцінювати довільні (у тому числі складні або мультимодальні) розподіли, що робить PF дуже гнучким і придатним до систем з високою нелінійністю чи сильними спотвореннями вимірювань. Алгоритм PF передбачає генерацію частинок відповідно до моделі переходу стану, оцінку кожної частинки на основі ймовірності відповідних вимірювань, а також фазу ресемплінгу — повторного вибору частинок із більшою вагою. Це дозволяє фокусувати обчислення на більш правдоподібних ділянках простору станів, за аналогією до механізму "виживання найкращих" у природі. Основним викликом є потреба в значній кількості частинок для досягнення високої точності, що обмежує застосування PF у малих БПЛА з обмеженими обчислювальними ресурсами [7].

Архітектура навігаційної системи

Архітектура навігаційної системи БПЛА визначає спосіб об’єднання даних з різних сенсорів (GPS, IMU, камери) для точної оцінки положення. Основними є каскадна, централізована, федеративна й гібридна архітектури, які відрізняються підходом обробки сенсорної інформації. За ступенем інтеграції сенсорів розрізняють слабкозв’язані, тіснозв’язані та глибоко інтегровані системи

GNSS/INS. Найпоширенішими серед них є слабкозв'язані конфігурації з EKF-фільтром. В умовах відсутності або поганої якості супутникового сигналу провідною технологією навігації стала візуально-інерційна одометрія (Visual Inertial Odometry, VIO), що поєднує дані з камер та інерційних сенсорів [7].

Вимоги до навігаційних систем БПЛА

Ідеальна навігаційна система має відповідати вимогам високої точності, доступності, простоти встановлення та обслуговування, гнучкості масштабування, економічної доцільності та мінімальної обчислювальної складності. На точність вимірювань впливають як сенсори, так і зовнішні умови. Перешкоди та динамічні зміни в середовищі можуть знижувати її. Візуальні навігаційні системи на основі камер є чутливими до середовищ без характерних елементів, можуть мати помилки у відстеженні ознак. Швидкі рухи і обертання БПЛА призводять до розмиття зображень, що ускладнює процес локалізації та стабільного навігаційного контролю. Для широкого впровадження навігаційних систем варто використовувати технології, які не потребують спеціалізованого обладнання та є легкодоступними. Хоча GPS-чипи недорогі, вони мають похибки до кількох метрів, тому для точнішого планування маршруту потрібні 3D-карти. Побудова таких карт ускладнюється кінематичними обмеженнями БПЛА, а обробка тривимірних даних потребує більше обчислювальних ресурсів і пам'яті. Безпечний політ БПЛА потребує обробки великого обсягу даних з сенсорів у реальному часі, що підвищує обчислювальну складність. Тому навігація в умовах обмежених ресурсів й енергоспоживання батареї є серйозним викликом. Універсальність навігаційних рішень полягає у використанні однакового апаратного забезпечення та алгоритмів. Однак різні завдання мають власні вимоги до точності, ваги, вартості та середовища, тому для них не завжди можна застосувати один підхід. Попри складнощі з поєднанням даних від різних сенсорів, їх інтеграція, особливо IMU та візуальних сенсорів, підвищує точність позиціювання та навігацію [6].

Навігаційні системи на основі карт

Навігаційні системи на основі карт дають змогу БПЛА уникати перешкод і планувати маршрут через заздалегідь визначену карту. Карти можуть бути різної деталізації — від повних 3D-моделей до спрощених схем. Такі системи поділяються на незалежні від карт, залежні від карт та ті, що створюють карту під час польоту [6].

Навігаційна система, незалежна від карт

У системі, яка незалежна від карт, положення БПЛА визначається на основі зображень. Навігація здійснюється через вилучення характерних ознак, зокрема за допомогою методів оптичного потоку та відстеження ознак [6].

Оптичний потік — це метод комп'ютерного зору для оцінки руху об'єктів між зображеннями, заснований на припущенні, що яскравість пікселя не змінюється попри його переміщення. Обчислення оптичного потоку полягає у мінімізації різниці яскравості між відповідними пікселями в послідовних кадрах [8].

Існують глобальні, локальні та гібридні методи оптичного потоку [8].

Глобальні підходи обчислюють потік, одночасно враховуючи відповідність яскравості між кадрами та згладжування поля руху, щоб зберегти його узгодженість по всьому зображенню. Класичним прикладом є метод Горна-Шунка, який мінімізує функцію з регуляризацією, що поєднує відхилення яскравості між кадрами та згладжування векторного поля потоку. Цей підхід є менш чутливим до шуму, але потребує значних обчислювальних ресурсів і часто втрачає точність поблизу розривів руху [8].

Локальні методи аналізують рух у межах невеликих перекривних областей зображення (patches), де припускається сталість руху в межах кожного такого вікна. Один із найбільш відомих підходів — метод Лукаса-Канаде, який використовує апроксимацію за допомогою ряду Тейлора для побудови субпіксельних оцінок. Локальні методи добре працюють у випадках з малими переміщеннями та низьким рівнем шуму, однак можуть втрачати точність при складному або нерівномірному русі [8].

Гібридні підходи поєднують переваги локального аналізу з глобальною оптимізацією. У методах, як «Lucas/Kanade Meets Horn/Schunck: Combining Local and Global Optic Flow Methods» [9], локальні градієнти агрегуються на певній області, що дає змогу врахувати сталість яскравості в глобальній енергетичній моделі. Комбінація підходів поєднує точність локальних методів оцінки руху зі стабільністю та просторовою узгодженістю, характерною для глобальних підходів. Такі методи ефективні в складних сценах із різноспрямованими рухами, шумами або низькою текстурою [8].

Відстеження ознак є ще одним незалежним від карт методом навігації, який ґрунтується на виявленні характерних локальних елементів зображення та їхньому зіставленні між послідовними кадрами або з еталонними шаблонами. Цей метод стабільно працює в динамічному середовищі, зокрема за умов зміни освітлення, масштабу, ракурсу та просторового положення об’єктів. Завдяки використанню стійких дескрипторів та алгоритмів зіставлення, відстеження ознак має високу точність локалізації. Прикладами реалізації цього підходу є поведінкова навігація, візуальне повернення додому (visual homing), безпечна посадка БПЛА без GPS за допомогою природних орієнтирів, контроль положення в разі раптової втрати цілі. Відстеження ознак може використовуватись як основа для таких функцій, як наведення, орієнтування, уникнення перешкод та автономне приземлення [6].

Навігаційна система, залежна від карт

Система з картою передбачає попередню модель середовища, яка використовується для локалізації, планування маршруту та уникнення перешкод. Переважно використовуються два типи карт: октодерево (octree) та карти зайнятості (occupancy grid map) [6].

Октодерево – це ієрархічна структура даних, що використовується для представлення тривимірних об'єктів шляхом рекурсивного розбиття простору. Весь об’ємний простір подається у вигляді масиву з $2^n \times 2^n \times 2^n$ вокселів і ділиться на вісім рівних частин — октантів. Якщо частина об'єму повністю належить

об'єкту (усі значення — 1), вона позначається як BLACK; якщо повністю порожня — як WHITE; якщо містить змішані значення — вона розділяється далі й позначається як GRAY (внутрішній вузол). Результатом є дерево з 8-ма нащадками на кожному вузлі, яке представляє всю структуру об'єкта на різних рівнях деталізації. Структура компактно зберігає тривимірну інформацію, що важливо для візуалізації, картографії та навігації в складних середовищах [10].

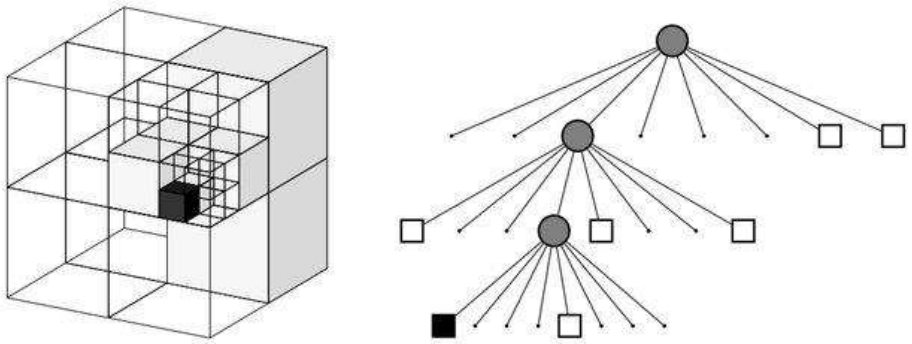

Рис. 1 Октодерево [1]

Для 2D-представлення використовується квадродерево — структура, аналогічна октодереву, але призначена для двовимірного простору. Весь простір зберігається у вигляді квадратної матриці $2^n \times 2^n$ пікселів, яка рекурсивно ділиться на чотири однакові частини — квадранти: північний-західний (NW), північний-східний (NE), південний-західний (SW) і південний-східний (SE). Якщо блок цілком належить об'єкту (усі значення — 1), він позначається як BLACK; якщо порожній — WHITE; якщо містить змішані значення — ділиться далі та позначається як GRAY. На відміну від октодерева, де вузол має 8 нащадків, у квадродереві кожен внутрішній вузол має 4. Ця структура особливо ефективна для представлення двовимірних бінарних зображень, підтримує виконання операцій об'єднання, перетину, зберігання даних з змінною роздільністю та завдяки адаптивній глибині дерева займає менший обсяг пам’яті [10].

2D та 3D моделі обираються залежно від складності середовища. Для відкритих або простих приміщень достатньо двовимірних карт, у той час як складні міські середовища з нерегулярними об’єктами потребують повноцінного тривимірного представлення. 3D-карти можуть моделювати такі складні перешкоди, як дерева,

тунелі, будівлі або нерівності рельєфу. У 3D моделях важливо не лише визначити наявність об’єкта, а й правильно врахувати його висоту, форму та положення. Для цього використовують ймовірнісні карти зайнятості, де кожній точці простору надається значення ймовірності наявності перешкоди [6].

Карта може бути побудована на основі даних від камер, стереозору, лазерних сканерів або об’ємних сенсорів. Отримані дані фільтруються, групуються та оновлюються для зменшення шумів і підвищення точності моделі середовища. Наприклад, застосовуються методи групування скан-ліній або об'єднання позитивних і негативних вимірювань з метою поступового уточнення карти [6].

Також існують підходи, які інтегрують візуальні зображення з картографічними API (наприклад, Google Maps), де поточне зображення порівнюється з фрагментами карти для уточнення координат або орієнтації дрона. У цьому випадку можуть застосовуватись алгоритми зіставлення ознак, такі як нормалізована крос-кореляція, фільтрація контурів та алгоритми пошуку відповідностей [6].

Навігаційна система, що створює карту

Через складний рельєф і змінні умови середовища (наприклад, бурі чи дощі), навігація за готовою картою може бути ненадійною. Альтернативою є створення карт у процесі польоту. Завдяки розвитку методів одночасної локалізації та побудови карти (SLAM), особливо візуального SLAM, ці підходи стали популярними в автономних БПЛА. Залежно від способу обробки зображень алгоритми SLAM класифікуються на непрямі, прямі та гібридні [6].

Непрямі підходи спершу виділяють характерні ознаки із зображень (наприклад, кути або унікальні точки), а потім використовують їх для оцінки руху та побудови карти. Такі ознаки інваріантні до обертання, масштабу та освітлення, а також стійкі до шуму та розмиття. Непрямий підхід будує точну, хоча й розріджену карту сцени, де ключові елементи визначають структуру простору. Розвиток цього методу призвів до появи більш складних систем, які поєднують

відстеження та картографування в окремих потоках і досягають високої точності локалізації навіть у складних середовищах без GPS [6].

Прямі підходи до побудови карт працюють безпосередньо з інтенсивністю пікселів зображень, не виділяючи окремих ознак, на відміну від непрямих методів. Вони оптимізують положення камери та структуру сцени, враховуючи всю візуальну інформацію, що робить їх стійкими до шуму, зміни освітлення та геометричних викривлень. Завдяки цьому прямі методи здатні відтворювати щільні або напівщільні карти середовища, хоча потребують більше обчислювальних ресурсів [6].

Гібридні підходи поєднують сильні сторони прямих і непрямих методів: спочатку карта ознак створюється за допомогою непрямих технік, а потім положення камери уточнюється прямими методами, які використовують яскравість пікселів. Одним із прикладів є напівпряма візуальна одометрія (semi-direct visual odometry), де побудова карти та оцінка руху виконуються паралельно, тому система працює в реальному часі навіть на обмежених апаратних ресурсах [6].

Візуальна одометрія

Візуальна одометрія (Visual Odometry) — це метод оцінки положення та орієнтації БПЛА через аналіз змін на послідовних кадрах із камери. Цей підхід відновлює локальну траєкторію без використання GPS і є частиною більш загальної задачі – структури із руху (Structure From Motion), що полягає у 3D-реконструкції навколишнього середовища та визначенні послідовності поз камери. На відміну від SFM, що виконується як офлайн-оптимізація з високими обчислювальними ресурсами, візуальна одометрія зосереджується на покадровому оцінюванні локальної 3D-траєкторії камери в режимі реального часу без побудови глобальної карти. Візуальна одометрія може слугувати проміжним етапом у складніших системах одночасної локалізації та картографування (Visual-SLAM) [7].

Види візуальної одометрії

У залежності від типу обробки візуальних даних для оцінки руху, алгоритми візуальної одометрії поділяються на такі категорії: на основі ознак (feature-based), на основі вигляду (appearance-based) і гібридні.

У методах на основі ознак вхідні зображення обробляються фільтрами для виявлення характерних точок (плям і кутів), з яких обираються мінімуми й максимуми як найнадійніші ознаки. Далі ці ключові точки зіставляються та відстежуються на послідовних кадрах. Оцінка руху обчислюється перетворенням $T_{k,k-1}$, яке описує зміну положення камери між кадрами за відповідними ознаками f_{k-1}, f_k. Алгоритми відрізняються за типом відповідностей між ознаками: 2D-2D, коли порівнюються координати точок на площині зображення; 3D-3D, коли кожна ознака має просторові координати (наприклад, отримані з камер глибини); і 3D-2D, коли тривимірна точка з кадру проектується у двовимірний простір. Для визначення трансформації найчастіше використовується оптимізація типу Гауса–Ньютона, яка мінімізує похибку між фактичним положенням ознак на зображенні та їхньою проєкцією з 3D-простору. Методи на основі ознак вразливі до помилкових зіставлень, тому застосовуються алгоритми для видалення викидів, наприклад, RANSAC.

Методи на основі вигляду визначають рух камери за інтенсивністю пікселів, тому ефективно працюють з менш текстурованими та розмитими зображеннями. Гібридні методи поєднують переваги попередніх підходів. Наприклад, напівпряма візуальна одометрія використовує інтенсивності пікселів лише для ключових кадрів, а для інших – ознаки.

Візуальну одометрію поєднують з інерційними сенсорами (IMU), використовуючи прискорення й кутові швидкості, для відстеження візуальних ознак і вирішення проблеми масштабної неоднозначності у випадку монокулярних камер [7].

Приклад сучасної реалізації візуальної одометрії

У статті «A Visual Odometry Pipeline for Real-Time UAS Geopositioning» було представлено підхід до геопозиціювання БПЛА в реальному часі без використання GPS, що обробляє зображення з монокулярної камери, спрямованої вертикально вниз. Запропоноване рішення реалізує абсолютне 3D-геопозиціювання, що є стійким до сезонних, інфраструктурних змін, до варіацій висоти та орієнтації дрона під час польоту.

Система складається з трьох ключових компонентів: геопросторової структури даних (квадродерева) для збереження орієнтирів, отриманих з ортофото (наприклад, з Google Maps), алгоритму зіставлення орієнтирів та механізму керуванням положення БПЛА, який враховує повороти та зміну масштабу для точнішого зіставлення орієнтирів між референсним фото та фото з БПЛА.

Система працює наступним чином. На підготовчому етапі референсні ортофото обробляються офлайн, з них автоматично виділяються орієнтири з географічною прив’язкою, які зберігаються у структурі квадродерева для швидкого пошуку. Під час польоту зображення, отримані з дрона, обробляються в реальному часі — з них також виділяються ключові точки, що потім зіставляються з базою. У дослідженні порівнювалися два алгоритми зіставлення орієнтирів: традиційний ORB і нейромережевий SuperGlue. Щоб зменшити похибки вертикального

позиціювання, було застосовано GIS-фільтрацію орієнтирів за допомогою даних OpenStreetMap, яка прибрала точки, розташовані на дахах будівель.

Для точнішого зіставлення орієнтирів між зображеннями, отриманими з БПЛА, та референсними ортофото, у системі реалізовано модуль керування положенням на основі псевдоперспективного перетворення. Таке геометричне перетворення враховує обертання, зсув і масштаб, що є лінійно залежним від висоти польоту дрона для камери з фіксованою фокусною відстанню. Завдяки модулю керування положенням БПЛА може змінювати нахил і висоту, при цьому масштаб зображень буде перетворений автоматично для коректного зіставлення з картою.

Результати експериментів підтверджують ефективність підходу: середня абсолютна похибка геопозиціювання (MAE) становила до 7.1 метра по горизонталі та до 4.5 метра по вертикалі. GIS-фільтрація зменшила середньоквадратичну похибку по висоті з 6.22 м до 1.89 м, а максимальну похибку — з 29.5 м до 6.4 м. Алгоритм SuperGlue показав кращі результати на застарілих або змінених ортофото, оскільки знаходив більшу кількість релевантних відповідностей, однак поступився ORB за швидкістю — 3.4 кадри в секунду проти 6.58 відповідно. Тому ORB доцільно використовувати в задачах, де пріоритетом є обчислювальна швидкість, а SuperGlue — у випадках, де критичною є точність за невизначених або змінних умов.

У порівнянні з підходами типу ORB-SLAM3 запропонований метод має низку переваг: забезпечує абсолютне позиціювання в реальному часі без дрейфу й не потребує точної початкової ініціалізації [11].

Застосування комп'ютерного зору для оптичної навігації

Алгоритми для виявлення та опису ключових точок

Алгоритм SIFT (Scale Invariant Feature Transform) виявляє й описує локальні ознаки на зображеннях, які є інваріантними до масштабу, повороту та частково стійкими до зміни освітлення, кута огляду й шуму. SIFT складається з чотирьох етапів: виявлення екстремумів у масштабному просторі, точної локалізації ключових точок, призначення їм орієнтації та формування дескрипторів.

Для виявлення ключових точок, інваріантних до масштабу, зображення обробляється гаусовими фільтрами. Таким чином будується піраміда масштабів, яка складається з послідовних октав. Кожна октава містить набір згладжених зображень з різними значеннями параметра σ, що логарифмічно зростають у діапазоні від σ до 2σ ($\sigma, k\sigma, k^2\sigma, \ldots, 2\sigma$). Після кожної октави зображення зменшується вдвічі, і побудова триває на новому рівні масштабу.

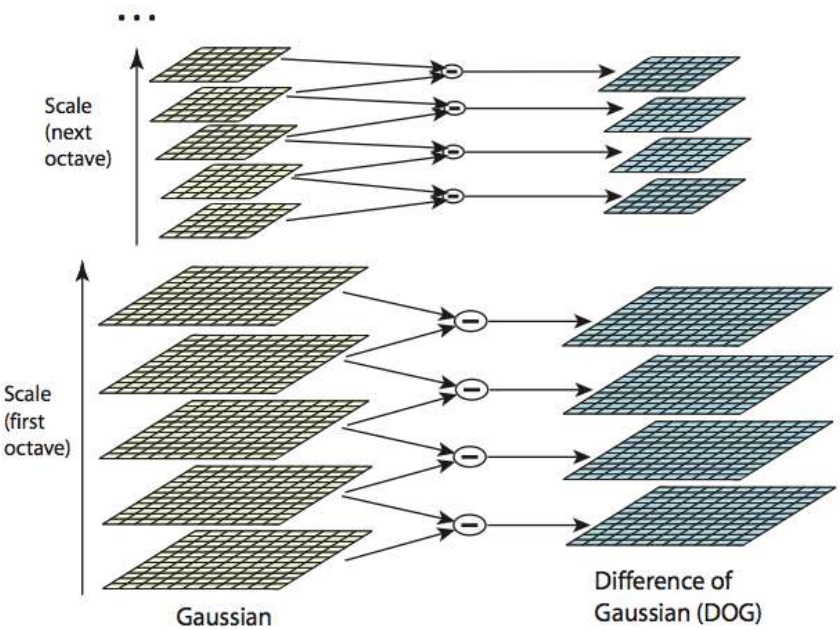

Рис. 2 Побудова піраміди масштабів у SIFT [2]

Для кожної октави обчислюється різниця гаусіанів (Difference of Gaussians, DoG) як різниця між сусідніми рівнями згладжених зображень.

$$D(x, y, \sigma) = \big(G(x, y, k\sigma) - G(x, y, \sigma)\big) * I(x, y), \text{ де}$$

$G(x, y, \sigma)$ – гаусова функція: $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$,

$I(x, y)$ – зображення,

σ – параметр згладжування,

k – сталий множник між рівнями масштабу в октаві

($k = 2^{1/s}$, де s — кількість рівнів масштабу в одній октаві).

У результаті отримується тривимірне представлення (x, y, σ), у якому шукаються локальні екстремуми. Кожен піксель порівнюється зі своїми 26 сусідами (8 сусідами на поточному зображенні, 9 сусідами у вищому масштабі та 9 сусідами в нижньому). Піксель вважається ключовою точкою-кандидатом, якщо є або максимумом, або мінімумом серед усіх цих сусідів. Для уточнення ключових точок використовується апроксимація розкладу Тейлора функції DoG в околі точки. Якщо отримане значення контрасту замале, або якщо точка розташована на краю, вона відкидається.

Кожній ключовій точці призначається орієнтація, обчислена через градієнти інтенсивності в її околі. Будується гістограма напрямків градієнтів, і напрямок із найбільшим значенням вважається основним. Якщо присутні інші піки з висотою понад 80% від головного, створюються додаткові копії ключової точки з відповідними орієнтаціями для збереження інваріантності до повороту.

У прямокутному околі навколо ключової точки (з урахуванням масштабу й орієнтації) обчислюються локальні градієнти. Область ділиться на сітку 4×4, у кожній з 16 підобластей створюється гістограма градієнтів з 8 напрямками. Загалом формується вектор довжини 128, який і є дескриптором ключової точки. Для стійкості до змін контрасту та освітлення вектор нормалізується, а значення, що перевищують 0.2, обрізаються й нормалізуються повторно.

Після побудови дескрипторів SIFT виконує пошук найближчого сусіда для знаходження відповідностей між зображеннями. Для зменшення кількості помилкових відповідностей перевіряється відношення відстаней до першого

найближчого та другого найближчого сусіда. Якщо це відношення менше 0.8, відповідність вважається надійною. Після накопичення достатньої кількості відповідностей, вони групуються за допомогою перетворення Гафа, у якому кожна відповідність голосує за позу об'єкта, що включає координати, масштаб й орієнтацію. Утворені кластери із трьох і більше узгоджених голосів перевіряються геометрично: оцінюється афінне перетворення між тренувальним і тестовим зображенням методом найменших квадратів. Якщо результати цієї перевірки показують внутрішню узгодженість відповідностей, процес розпізнавання завершується успішно [14].

Алгоритм ORB (Oriented FAST and Rotated BRIEF) був запропонований як альтернатива SIFT для виявлення та опису ключових точок. Його спеціально розроблено для роботи на пристроях з обмеженими обчислювальними ресурсами, з інваріантністю до обертання та стійкістю до шуму. ORB поєднує швидкий детектор FAST (Features from Accelerated Segment Test), доповнений обчисленням орієнтації, з модифікованим дескриптором BRIEF (rBRIEF), орієнтованим за напрямком ключової точки.

Оскільки FAST не створює ознак у різних масштабах, зображення попередньо обробляється у вигляді масштабної піраміди. На кожному рівні піраміди застосовується детектор FAST, який виявляє характерні точки зображення, використовуючи поріг інтенсивності між центральним пікселем і його сусідами. FAST не оцінює вираженість виявлених точок як кутів і помилково виділяє точки на краях зображення. Тому автори ORB запропонували сортувати FAST-дескриптори за метрикою кутів Гарріса й фільтрувати їх, відкидаючи слабкі ознаки. Оскільки базовий FAST не є інваріантним до обертання, в алгоритмі було додано обчислення орієнтації ключових точок за допомогою методу центроїда інтенсивності. У цьому підході вважається, що розподіл яскравості в околі точки зміщений відносно її геометричного центру.

Центроїд $C = (\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}})$ визначається через моменти яскравості зображення:

$$m_{pq} = \sum_{x,y} x^p y^q \, I(x, y),$$

де $I(x, y)$ є інтенсивністю пікселя. Орієнтація оцінюється як напрям вектора від центра області до центру мас яскравості: $\theta = atan2(m_{01}, m_{10})$.

BRIEF (Binary Robust Independent Elementary Features) є бінарним дескриптором, який створює вектор ознак через порівняння яскравостей пар пікселів у згладженій області зображення. Кожен біт відповідає результату бінарного тесту: якщо інтенсивність у точці x менша за інтенсивність у точці y, то записується 1, інакше – 0. Повний дескриптор формується як послідовність із n, зазвичай 256, таких тестів:

$$f_n(p) := \sum_{1 \le i \le n} 2^{i-1} \tau(p; x_i, y_i)$$

Щоб зменшити залежність від шуму, зображення згладжується перед тестуванням. Для подолання чутливості до обертання було запропоновано steered BRIEF, який враховує напрямок ключової точки. Координати пар пікселів з бінарних тестів обертаються відповідно до обчисленої орієнтації за допомогою матриці повороту R_θ: $S_\theta = R_\theta S$, де S – це матриця розміром 2×n, яка містить координати пар для тестів.

Однак додане обертання призводить до зниження дисперсії та зростання кореляції між окремими тестами, тому дескриптор є менш дискримінативним. Автори ORB вирішують цю проблему, впроваджуючи навчання нового набору бінарних тестів. Усі можливі пари точок у патчі розміром 31×31 пікселів формують повний простір можливих тестів. Кожен тест виконується на тренувальній множині з 300 тисяч ключових точок, і за критеріями високої дисперсії (середнє близьке до 0.5) та низької кореляції з іншими тестами відбирається 256 найкращих через жадібний пошук. Таким чином було реалізовано дескриптор rBRIEF (rotation-aware BRIEF), який має кращий розподіл бітів і вищу дискримінативність у порівнянні зі стандартним й орієнтованим BRIEF.

Для пошуку найближчих сусідів у великих наборах зображень ORB використовує структуру Locality Sensitive Hashing (LSH). Дескриптори

хешуються в кілька таблиць за підмножинами їхніх бітів, і пошук виконується у відповідних комірках хеш-таблиці, які мають спільну сигнатуру. Відстань між дескрипторами (схожість векторів) обчислюється за метрикою Геммінга. Щоб покращити точність і зменшити обсяг пам'яті, у ORB застосовується модифікований multi-probe LSH, що перевіряє також сусідні комірки. Завдяки низькій кореляції бітів у rBRIEF, хеш-функції розподіляють дескриптори більш рівномірно по комірках, що пришвидшує пошук.

Експерименти показали, що ORB є приблизно в 10 разів швидшим за SURF і в понад 300 разів швидшим за SIFT на зображеннях розміром 640×480. При цьому якість зіставлення ознак залишалася на високому рівні. Алгоритм зміг обробити понад 2 мільйони ознак на 2686 зображеннях за 42 секунди, що підтвердило його здатність працювати в реальному часі [24].

Методи пошуку відповідностей між ознаками

Brute-Force Matching – це метод, який порівнює кожен дескриптор з одного зображення з усіма дескрипторами іншого за визначеною метрикою відстані (наприклад, евклідовою або Геммінга) й обирає найближчу пару. Це простий підхід, який дає точні результати, проте є обчислювально складним при великій кількості ознак.

У методі найближчих сусідів (K-Nearest Neighbors) для кожного ключового дескриптора шукаються k найближчих з іншого зображення. Далі може застосовуватися тест відношення Лоу (Lowe's ratio test) для фільтрування хибних відповідностей і вибору найкращого збігу. Якщо $\frac{distance_1}{distance_2} < t$, де t – це порогове значення, то відповідність вважається надійною.

RANSAC (Random Sample Consensus) – це ітеративний метод, який шукає параметри математичної моделі для набору даних, що мають викиди (outliers). У задачах пошуку відповідностей RANSAC використовується для визначення надійного геометричного перетворення між зображеннями. Багаторазово алгоритм вибирає випадкові підмножини відповідностей, обчислює перетворення та оцінює, скільки точок із повного набору узгоджуються з

моделлю (inliers). У результаті обирається перетворення з найбільшою кількістю відповідностей (inliers). Оскільки RANSAC є стійким до шуму, то суттєво покращує точність зіставлення. Однак цей підхід є обчислювально складним і вимагає налаштування параметрів, таких як допустима похибка (поріг) і кількість ітерацій [25].

Геометричні перетворення

Гомографія – це проєктивне перетворення, що описує, як точка на одному зображенні переходить у точку на іншому за допомогою матриці H. Щоб визначити гомографію, потрібно щонайменше 4 пари відповідних точок.

Нехай $x = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$, $x' = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$ і $H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}$, тоді:

$$c\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Після усунення масштабного коефіцієнта c, задачу можна сформулювати як однорідну систему лінійних рівнянь у вигляді:

$$A \cdot h = 0,$$

де A — матриця коефіцієнтів, побудована з координат точок, h — вектор з усіх елементів матриці H. Якщо задано лише 4 невироджені пари точок, то задача має єдиний розв’язок. У випадку більшої кількості точок система розв’язується методом найменших квадратів, який знаходить таку матрицю H, що мінімізує відхилення від системи рівнянь [26]:

$$min_h ||Ah||_2 \; rank(A) = 3$$

Гомографія може використовуватися для оцінки положення камери за площинними точками, наприклад у доповненій реальності з маркером, для виправлення перспективи, склеювання панорам при обертанні камери [27].

Ще одним геометричним перетворенням є афінне, яке поєднує лінійне перетворення та зсув, тому його можна представити у вигляді множення на матрицю та додавання вектора:

$$T = A \cdot \begin{bmatrix} x \\ y \end{bmatrix} + B,$$

де A – матриця лінійного перетворення, B – вектор зсуву [28].

Це перетворення зберігає прямі лінії, паралельність і відношення відстаней вздовж прямих. На відміну від гомографії, яка має 8 ступенів вільності та враховує перспективні спотворення, афінне перетворення має лише 6 ступенів вільності й не здатне коректно відтворювати перспективу. Афінні перетворення застосовуються для вирівнювання зображень, відстеження об’єктів, у робототехніці – для суміщення даних з різних сенсорів для підвищення точності локалізації та побудови карти [29].

Нейромережеві методи для виявлення та зіставлення ключових точок

SuperPoint – це повністю згорткова нейронна мережа, яка за один прямий прохід обробляє повнорозмірне зображення, обчислюючи ключові точки та відповідні дескриптори фіксованої довжини. Модель містить спільний енкодер та два декодери: декодер для виявлення ключових точок і декодер для генерації дескрипторів. Енкодер, який складається із згорткових шарів, шарів об'єднання (pooling layers) для просторового зменшення зображення та нелінійних функцій активації, перетворює вхідне зображення розміром H × W у тензор розміром $H_c \times W_c \times F$ ($H_c = H / 8$, $W_c = W / 8$, F – кількість каналів). Декодери мають окремі параметри, але навчаються одночасно й працюють з однією спільною картою ознак, що формується енкодером. Декодер для виявлення ключових точок обчислює ймовірності для кожної області 8×8 пікселів, чи вона містить ключову точку. Для цього декодер формує тензор розміром $H_c \times W_c \times 65$, де 65 – це кількість каналів, що відповідають локальним областям 8×8 пікселів й один додатковий клас «не точка». До каналів застосовується softmax, після чого відкидається клас «не точка» й тензор перетворюється у карту ймовірностей ключових точок розміру H × W. Декодер дескрипторів формує напівщільну карту дескрипторів у тензор розміром $H_c \times W_c \times D$ (D – розмір вектора-дескриптора). Для відновлення повної карти дескрипторів застосовується бікубічна інтерполяція, і далі дескриптори нормалізуються за L2, щоб усі вектори були одиничної довжини.

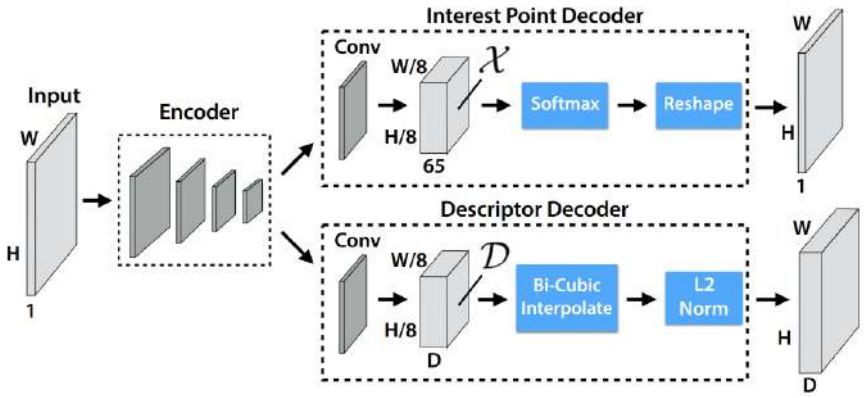

Рис. 3 Архітектура SuperPoint [15]

SuperPoint навчалася у два етапи. Спочатку було натреновано повністю згорткову нейронну мережу MagicPoint на синтетичному наборі даних Synthetic Shapes, який складається з простих геометричних фігур. Щоб покращити узагальнення моделі на реальних зображеннях, автори використали метод Homographic Adaptation, в якому вхідне зображення деформується багато разів, застосовується детектор до різних спотворених версій і результати агрегуються. Нехай I – вхідне зображення, f_θ – функція виявлення ключових точок детектором, x – знайдені ключові точки, тоді $x = f_\theta(I)$. Ідеальний детектор повинен бути коваріантним до гомографій, тобто має виконуватися умова: $H_x = f_\theta(H(I))$, $x = H^{-1}f_\theta(H(I))$. На практиці детектор не є ідеально коваріантним, тому результати детектора на випадкових гомографіях усереднюються за формулою:

$$F(I; f_\theta) = \frac{1}{N_h} \sum_{i=1}^{N_h} H_i^{-1} f_\theta\left(H_i(I)\right)$$

Після застосування методу Homographic Adaptation модель називається SuperPoint. Модель тренувалася на наборі зображень MS-COCO. Для цього використовувалися пари синтетично спотворених зображень, які мають псевдо-розмічені координати ключових точок й істинні відповідності між точками, отримані через випадково згенеровану гомографію H, що пов’язує два зображення. Загальна функція втрат – це сума втрат для детектора ключових точок L_p та для дескриптора L_d. Функція втрат L_p реалізована як крос-ентропія, що застосовується до кожної клітинки зменшеного простору з ознаками і визначає ймовірність наявності ключової точки в кожній області 8×8 пікселів. За основу функції втрат L_d взято функцію hinge loss, яка мінімізує відстань між дескрипторами відповідних точок і водночас збільшує її для невідповідних пар.

Для кожної пари зображень одночасно оптимізуються функції втрат детектора та дескриптора, баланс між якими регулюється коефіцієнтом λ:

$$L(X, X', D, D'; Y, Y', S) = L_p(X, Y) + L_p(X', Y') + \lambda L_d(D, D', S)$$

SuperPoint перевершує класичні методи при зміні освітлення та досягає результатів на рівні SIFT при зміні ракурсу [15].

Підходи до зіставлення ознак зазвичай обчислюють відстані між дескрипторами окремо, не враховуючи контексту інших ключових точок. У складних умовах, з повторюваними структурами, змінами освітлення, ці методи не є надійними.

Нейронна мережа SuperGlue навчена встановлювати відповідності між локальними ознаками. Модель складається з графової нейромережі з механізмом уваги (Attentional Graph Neural Network) й оптимального шару зіставлення (Optimal Matching Layer). Вхідними даними для SuperGlue є позиції та візуальні дескриптори ключових точок з двох зображень, які кодуються у вектор великої розмірності за допомогою багатошарового персептрона. Повний граф, вершинами якого є ключові точки обох зображень, має два типи неорієнтованих ребер: ребра всередині зображення (self edges) і ребра між зображеннями (cross edges). По всіх ребрах інформація поширюється через механізм передачі повідомлень (message passing). На кожному l шарі графова нейромережа оновлює представлення ключової точки x_i, поєднуючи його з агрегованими повідомленнями від сусідніх точок:

$$x_i^{(l+1)} = x_i^{(l)} + MLP([x_i^{(l)} || m_{E\to i}]),$$

де $||$ – конкатенація векторів, MLP – багатошаровий персептрон, $m_{E\to i}$ – агрегація повідомлень від усіх точок j, з'єднаних із i ребром типу E (E_{self} або E_{cross}).

Графова нейромережа має L шарів, які чергуються ребрами всередині одного зображення (self ребрами) та ребрами між двома зображеннями (cross ребрами). Self ребра використовують самоувагу (self-attention), а cross ребра – перехресну увагу (cross-attention). На шарі нейромережі кожна точка i формує запит q_i, за яким вибирає відповідні значення v_j інших точок на основі їхніх ключів k_j. Створюється повідомлення як зважене середнє за формулою:

$$m_{E\to i} = \sum_{j:(i,j)\in E} \alpha_{ij} v_j, \qquad \alpha_{ij} = softmax_j(q_i^T k_j),$$

де α_{ij} – вага уваги, що визначає важливість точки j для точки i.

Ключі, запити та значення отримуються як лінійні проєкції глибоких ознак, після чого на основі оновлених представлень формуються дескриптори.

Оптимальний шар зіставлення створює матрицю часткового призначення. Схожість між усіма парами ключових точок обчислюється як скалярний добуток їхніх дескрипторів, у результаті чого будується матриця оцінок. Для обробки точок без відповідностей до цієї матриці додається рядок і стовпець (dustbins) зі спільним параметром, куди призначаються такі точки без пар. Зіставлення точок розглядається як задача оптимального транспорту з ентропійною регуляризацією між двома множинами точок, яка розв'язується алгоритмом Сінхорна. Алгоритм ітеративно нормалізує експоненційно зважену матрицю оцінок по рядках і стовпцях, подібно до операції softmax. Після завершення обчислень система відкидає всі допоміжні елементи й отримує остаточну матрицю відповідностей.

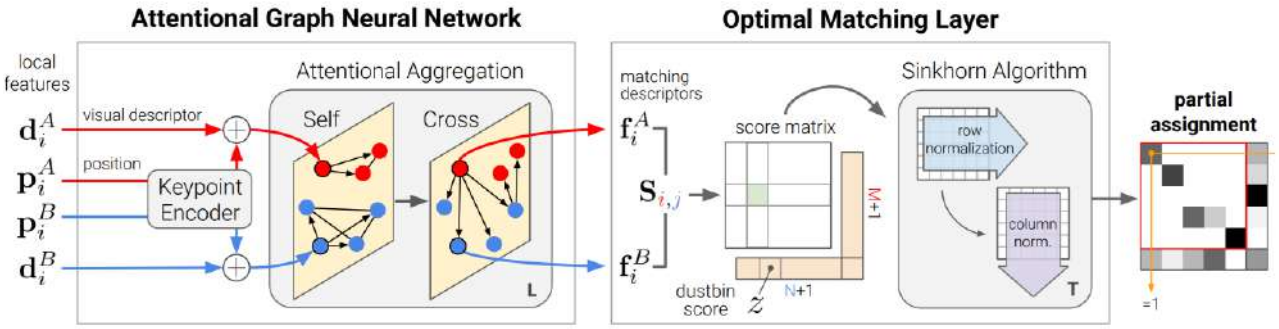

Рис. 4 Архітектура SuperGlue [23]

Модель SuperGlue навчалася в режимі з учителем на основі істинних відповідностей між точками, отриманими за допомогою поз та карт глибини або гомографій. Як функцію втрат було використано від'ємну логарифмічну правдоподібність (log-likelihood) матриці часткового призначення. Таким чином було враховано як правильні зіставлення, так і не невідповідні або незіставлені ключові точки.

SuperGlue сумісна з будь-якими дескрипторами та функціонує на сценах в приміщенні й відкритому просторі. На GPU GTX 1080 модель працює в реальному часі зі швидкістю до 15 кадрів на секунду при обробці зображень з 1024 ключовими точками [23].

Сучасні рішення

Українська система донаведення «ЗІР» здатна самостійно виявляти цілі та вражати їх у режимі автоматичного керування. За допомогою штучного інтелекту й комп'ютерного зору вона розпізнає такі типи цілей, як піхота, засоби ППО, артилерія, бронетехніка, танки тощо. «ЗІР» легко встановлюється на будь-який тип ударних дронів (літак, крило або квадрокоптер), має власну камеру й не потребує модифікації пульта чи станції керування. Система існує у двох версіях: ZIR base, яка забезпечує наведення на відстані до 1000 м, автономний політ у зоні РЕБ до 3000 м, стійкість до вітру до 12 м/с і точність ураження до 1 м, та ZIR AI, яка додатково автоматично розпізнає різні цілі на відстані 150–800 м, підтримує ручний або автоматичний вибір цілі й здатна одночасно відстежувати кілька об'єктів. Інтегрована з ArduPilot система може виконувати повністю автономні місії без зв'язку, використовуючи лише оптичну навігацію. Модель штучного інтелекту навчена на відкритих даних і вміє ефективно вражати навіть замасковані цілі (наприклад, БТР в посадках). «ЗІР» знижує вимоги до підготовки пілотів, є стійкою до засобів РЕБ, долає радіогоризонт й може вражати рухомі цілі на великій глибині фронту [12].

БПЛА Vector від компанії Quantum Systems — це сучасний розвідувальний дрон, який активно використовується в Україні з 2022 року. Завдяки оптичній навігації та програмному забезпеченню Silvus дрон працює в складних умовах глушіння GPS, автоматично перемикаючись на альтернативні частоти. Vector має модульну архітектуру: змінні носові частини та оптичні системи легко адаптуються під конкретні бойові задачі. Дрон оснащений камерами Boson для роботи вдень і вночі й обчислювальними модулями на базі Nvidia Jetson Orin з ШІ, які обробляють відеодані в реальному часі, навіть за умов відсутності супутникової навігації. Quantum Systems планує розгорнути Receptor AI — програмне забезпечення для автономного виявлення та класифікації цілей за допомогою алгоритмів машинного навчання, натренованих на українських даних. Незважаючи на автоматизацію, компанія підкреслює, що ключові етичні й стратегічні рішення має ухвалювати людина [5].

Американська компанія Skydio постачає в Україну дрони X10D, створені з урахуванням досвіду війни та особливостей українського поля бою. Раніше модель X2 мала серйозні проблеми з перешкодами РЕБ через фіксовані частоти, що обмежувало її застосування. Протягом 2022–2024 років команда Skydio понад 30 разів відвідувала Україну, щоб розробити вдосконалену версію дрона на основі зібраних відгуків. X10D оснащений системою автоматичного розпізнавання цілей, що поєднує глибоке навчання та комп'ютерний зір. X10D має найсучаснішу тепловізійну камеру, здатну виявляти теплові сліди та визначати активність техніки. Система візуально-інерційної одометрії (VIO) з шістьма ширококутними камерами створює 3D-мапу для уникнення перешкод та навігації без GPS на висоті до 300 метрів. Дрони інтегруються в систему Delta, передаючи розвіддані в режимі реального часу, що сприяє точнішому ураженню цілей артилерією. Skydio вже передала близько 1000 дронів, які активно використовуються для тактичної розвідки та знищення ворожої техніки [5].

Серед українських ударних дронів дальнього радіусу дії особливе місце посідає «Лютий», який здійснив до 80% успішних атак на нафтопереробні заводи в глибині Росії. Цей дрон розроблений повністю в Україні й має часткову автономність: може проходити складні маршрути, обходячи ППО та зони активної РЕБ. Автономна навігація використовує дані із сенсорів, алгоритми машинного зору, попередньо завантажені супутникові й топографічні карти, завдяки чому «Лютий» може точно орієнтуватися без GPS. Хоча частина маршруту виконується автономно, фінальний підхід до цілі виконується вручну з високим рівнем координації. Перед кожною місією оператор отримує детальний план із описом цілі, систем ППО, супутникових знімків, маршрутів обходу перешкод і погодинної синхронізації. Враховуються такі фактори, як напрям вітру, наявність інженерних конструкцій і складність точного заходу на ціль [5].

Американська компанія Shield AI постачає Україні дрони V-BAT з системою автономної навігації, яка працює без GPS. У 2024 році дрони пройшли випробування в умовах інтенсивної РЕБ: V-BAT успішно пролетів понад 130 км у глибину території противника, виявив батарею ЗРК «Бук» і передав координати

для удару HIMARS. Безпілотник використовує програмне забезпечення Hivemind, яке за допомогою алгоритму факторних графів визначає положення, орієнтацію і швидкість у реальному часі, інтегруючи дані з камер, GPS й інерційних сенсорів. Система адаптує маршрут із врахуванням перешкод, що особливо важливо для тривалих розвідувальних місій. Дрон може працювати 8–10 годин і відстежувати цілі протягом усього польоту [5].

Американська компанія Vermeer розробила оптико-навігаційну систему, яка поєднує в собі методи візуальної та інерційної одометрії. Система складається з 1 - 4 камер, підключених до обчислювального блоку на базі NVIDIA з вбудованим програмним забезпеченням, а також жорсткого диска із заздалегідь завантаженою 3D-картою рельєфу місцевості. Додатково використовується інерційний вимірювальний модуль. Нейронна мережа системи зіставляє зображення з камери з локальною картою й визначає місцезнаходження БПЛА, аналізуючи рух пікселів. Пристрій підключається до дрона через GPS-кабель і живлення, фактично замінюючи традиційний GPS-модуль. Завдяки цьому безпілотник стає менш вразливим до засобів радіоелектронної боротьби [13].

Українська компанія The Fourth Law серійно виробляє модулі з ШІ для FPV-дронів із функцією автоматичного наведення на ціль. Модуль складається з камери, компактного комп’ютера з програмним забезпеченням та є сумісним із більшістю дронів. Система активно використовується на фронті: оператор обирає ціль, далі ШІ бере керування на себе, відстежуючи рух об’єкта, контролюючи траєкторію дрона та коригуючи межі цілі в реальному часі. У майбутньому The Fourth Law планує реалізувати повністю автономного дрона, здатного виконувати місію без участі оператора — із автоматичним скиданням боєприпасів, виявленням цілей нейромережею, навігацією без GPS на основі карт й оптичного зору, автоматизованим зльотом і посадкою [5].

Чеська компанія LPP Holding розробила автономні бойові БПЛА з візуальною навігацією на основі штучного інтелекту. Дрони запускаються з мобільної катапульти, яка легко розгортається та дозволяє оператору оперативно відходити

з місця запуску. Усі обчислення виконуються на борту в замкненій архітектурі, тому БПЛА є повністю стійким до будь-яких засобів РЕБ [16]. Бортова навігаційна система об'єднує дані з сенсорів, порівнює зображення з офлайн-супутниковими знімками в реальному часі й коригує курс за допомогою фільтра Калмана. У фінальній фазі польоту дрон використовує ШІ для оптичного відстеження, щоб розпізнавати об’єкти й точно наводитися на ціль. Розробка, яка тривала менше року, охоплювала навчання ШІ розпізнавати об’єкти й орієнтуватися за картографічними даними без GPS, а також удосконалення аеродинаміки літальних апаратів і модульних ударних боєголовок. Компанія LPP Holding вже випробувала дрони в бойових умовах в Україні й восени минулого року розпочала перші поставки [17].

Висновки до розділу 1

У цьому розділі було розглянуто сучасні підходи до навігації безпілотних літальних апаратів, зокрема акцентовано увагу на оптичних методах, що забезпечують орієнтацію в умовах відсутності або приглушення GPS-сигналу. Проаналізовано основні типи навігації та їх поєднання в гібридних архітектурах. Розглянуто ключові компоненти навігаційних систем, включаючи візуальні й інші сенсори, фільтраційні методи для злиття даних, архітектури систем та вимоги до точності й автономності. Особливу увагу приділено методам оптичної навігації: навігації на основі карт, візуальній одометрії, SLAM, а також алгоритмам виявлення, опису та зіставлення ключових точок, як класичним (SIFT, ORB), так і нейромережевим (SuperPoint, SuperGlue). Розгляд сучасних практичних рішень підкреслив стратегічну важливість оптичної навігації БПЛА. Теоретичний аналіз допоміг обрати підходи для розробки власної системи навігації за кадрами з камери дрона та картою місцевості.

Розділ 2. Розробка алгоритму оптичної навігації

Обґрунтування вибору засобів розробки

Для реалізації алгоритму оптичної навігації було обрано мову Python, яка підтримує бібліотеки для комп'ютерного зору, геообробки, візуалізації. Розробка виконувалася в середовищі Google Colab, яке надає можливість обирати між CPU та GPU для виконання обчислень, візуалізовувати результати і зберігати їх на Google Drive. Бібліотеку OpenCV було використано для таких операцій, як читання, обробка зображень, побудова гомографії, знаходження ключових точок методом ORB. Для нейромережевого підходу було взято попередньо натреновані моделі SuperPoint і SuperGlue, реалізовані у фреймворку PyTorch.

Ортофото карти місцевості у форматі GeoTIFF було завантажене з платформи OpenAerialMap. Її розмір становить 19757×19854 пікселів, що відповідає площі приблизно 987,92×992,76 метрів. Карта була оброблена з використанням бібліотек Rasterio для читання геоданих і трансформування координат і NumPy масивних обчислень координат, векторизації обробки. Координати, ключові точки й дескриптори фрагментів карти зберігалися у форматі NumPy NPZ для подальшого зчитування й зіставлення з кадрами. Для візуалізації побудованих траєкторій та графіків порівняння оціненої і реальної позиції використовувалась бібліотека Matplotlib.

Попередня обробка карти місцевості для навігації

Розбиття карти на фрагменти

Для візуальної навігації БПЛА було попередньо оброблено ортофото місцевості у форматі GeoTIFF, з якого отримуються основні параметри: ширина, висота, афінне перетворення та координатна система (Coordinate Reference System). Карту було поділено на квадратні фрагменти, розмір яких визначається через поле зору дрона в метрах і щільність пікселів карти на метр. Значення поля зору вибрано таким чином, щоб кожен фрагмент приблизно відповідав реальному зображенню, яке дрон може отримати за один кадр на типовій висоті польоту. Розмір фрагмента додатково зменшено коефіцієнтом масштабування, щоб при зміні висоти польоту, масштабу зйомки та кута огляду зберігалося зіставлення з картою. Якщо розмір фрагмента не кратний розміру ортофото, тоді до зображення додавалося заповнення чорним фоном, щоб усі частини карти були однаковими за розміром. Для кожного фрагмента були обчислені й збережені у файл такі дані, як GPS-координати верхнього лівого кута, нижнього правого кута та центру, піксельні координати, індекси (порядкові номери фрагмента). Також записувалося, чи є фрагмент частково за межами оригінальної карти, тобто чи має додаткове заповнення. Зображення частин карти збережено у форматі .jpg із вказаними індексами в назвах файлів.

Отримання візуальних ознак з фрагментів карти

Задача локалізації дрона за кадрами й картою полягає у виявленні та зіставленні візуальних ознак. У цій роботі для порівняння було реалізовано два підходи: ORB (Oriented FAST and Rotated BRIEF) і пара нейронних мереж SuperPoint та SuperGlue. Алгоритм ORB обрано через високу швидкодію, малу обчислювальну складність та здатність працювати на пристроях з обмеженими ресурсами. Цей метод виявляє ключові точки з урахуванням орієнтації, тому є інваріантним до поворотів зображення, і створює бінарні дескриптори, які швидко зіставляються за відстанню Геммінга. Нейронні мережі SuperPoint і SuperGlue використано як сучасний нейромережевий підхід для виявлення та зіставлення ознак. Модель SuperPoint натренована знаходити ключові точки та створювати стійкі

дескриптори, які зберігають властивості при зміні масштабу, освітлення чи кута огляду. SuperGlue є графовою нейромережею, яка зіставляє ознаки з урахуванням просторового контексту, що зменшує кількість помилкових відповідностей.

Для витягування ознак із фрагментів карти використовувалися зображення у форматі .jpg, а географічна інформація бралася з метаданих GeoTIFF-файлу. Зображення частин карти перетворювались у відтінки сірого. Для моделі SuperPoint додатково нормалізувалися до діапазону [0, 1] і подавалися у вигляді тензора. У підході ORB ключові точки виявлялися зі встановленим обмеженням (до 1024 ознак), після чого для кожної з них обчислювався бінарний дескриптор. У методі SuperPoint модель повертала координати ключових точок, дескриптори розміром 256 та оцінки надійності точок (scores). В обох випадках піксельні координати ключових точок трансформувалися у GPS-координати за афінним перетворенням. У стисненому форматі .npz збережено результати обробки карти: назви зображень, індекси фрагментів, піксельні та GPS-координати фрагментів і ключових точок на них, дескриптори. На рисунку 5 представлено ключові точки фрагмента карти, отримані через ORB і SuperPoint.

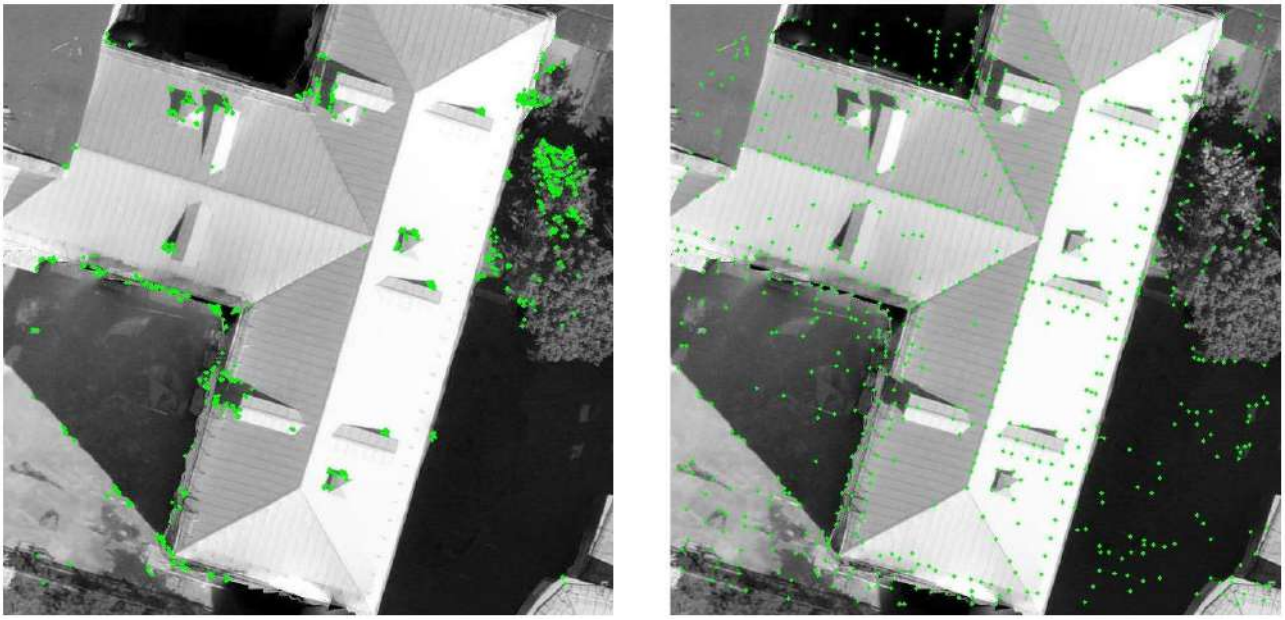

Рис. 5 Ключові точки фрагмента карти, отримані через ORB (зліва) і SuperPoint (справа)

Алгоритм навігації за підготовленою картою та кадрами з дрона

Основна ідея реалізованого алгоритму полягає у визначенні поточного положення дрона через порівняння дескрипторів зображення із камери дрона й дескрипторів фрагментів заздалегідь підготовленої карти. Спочатку задається стартова позиція дрона на карті та завантажуються попередньо обчислені ключові точки та дескриптори всіх фрагментів. На кожному кроці алгоритму створюється кадр з дрона – це сформоване на частині карти зображення, яке відповідає поточній реальній позиції і зберігається в .jpg файлі. Підхід ORB або SuperPoint обробляє кадр, виділяючи ключові точки та генеруючи їхні дескриптори, які потім порівнюються з дескрипторами частин карти. У підході ORB бінарні дескриптори зіставляються за відстанню Геммінга й фільтруються за тестом відношення Лоу. У нейромережевому підході зіставлення відбувається по-іншому: ключові точки, дескриптори й оцінки надійності точок з кадру та фрагментів карти передаються в SuperGlue. Навчена модель аналізує просторові відношення між точками й визначає відповідності між ознаками.

У випадку, коли знайдено достатню кількість валідних відповідностей, між кадром і фрагментами будується гомографія через RANSAC, яка уточнює положення дрона з урахуванням перспективних спотворень. Якщо кількість відповідностей є недостатньою для побудови гомографії, дрон не може точно локалізуватися. Тоді дрон продовжує рух у напрямку до цілі, розрахованому за кутом між останнім оціненим положенням і цільовою точкою. Якщо такого положення немає, використовується заздалегідь визначений напрямок руху від стартової позиції до цілі.

Під час переміщення фіксується як реальна траєкторія, так і оцінена через візуальне зіставлення. Алгоритм завершує роботу після досягнення цільової позиції або після перевищення максимально допустимого часу польоту. Таким чином в умовах без GPS дрон орієнтується на місцевості через зіставлення векторних ознак кадру та карти й рухається у напрямку до цілі.

Створення кадру з дрона

В умовах симуляції кадр створюється штучно: з ортофото вирізається область заданого розміру, центрована на поточних реальних координатах дрона в пікселях. Щоб моделювати зміну кута огляду камери, зображення випадково обертається на кут із заданого діапазону. Для уникнення втрат при обертанні береться область більшого розміру й обрізається до цільового розміру. У результаті формується зображення, яке імітує знімок з камери дрона над поточною зоною його польоту (див. рисунок 9).

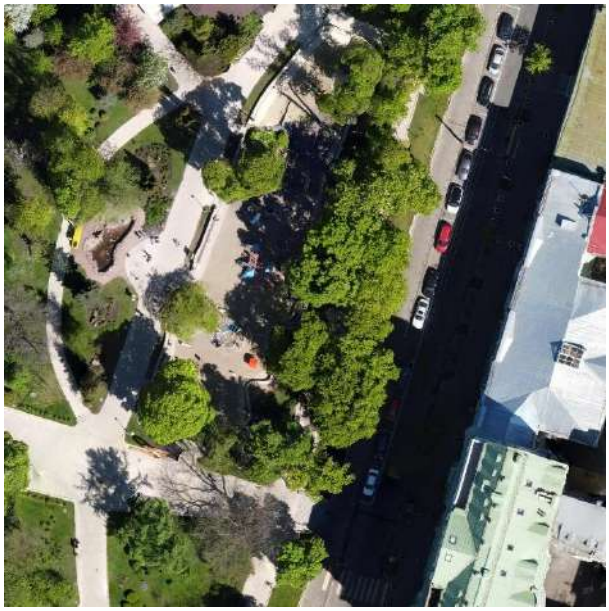

Рис. 6 Штучно створений кадр з дрона

Отримання візуальних ознак з кадру дрона

Подібно до обробки частин карти, для отримання візуальних ознак з кадру дрона використовується два підходи: класичний ORB та нейромережевий SuperPoint. У методі ORB кадр попередньо конвертується у відтінки сірого, після чого алгоритм detectAndCompute виявляє ключові точки та обчислює бінарні дескриптори. Координати ключових точок зберігаються у вигляді масиву (x, y). З SuperPoint кадр також перетворюється у відтінки сірого, нормалізується та подається на вхід нейромережі, яка повертає координати ключових точок, дескриптори розмірності 256 й оцінки надійності кожної точки. Далі отримані ознаки використовуються для пошуку відповідностей із частинами карти для визначення положення дрона. Код обробки кадру представлено на рис. 7, 8.

```python
def extract_features_orb(frame, orb):
    """
    Extracts keypoints and descriptors from a drone frame using the ORB algorithm.

    Parameters:
      frame (np.ndarray): Input color image (BGR).
      orb (cv2.ORB): An instance of cv2.ORB_create().

    Returns:
      kpts (np.ndarray): Array of keypoints, shape (N, 2) with (x, y) coordinates.
      desc (np.ndarray): Corresponding descriptors, shape (N, 32), dtype=uint8.
    """
    # Convert the input image to grayscale
    img_gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Detect keypoints and compute descriptors
    keypoints, descriptors = orb.detectAndCompute(img_gray, None)

    # Convert keypoints to NumPy array of (x, y) coordinates
    if keypoints is not None and len(keypoints) > 0:
      kpts = np.array([[kp.pt[0], kp.pt[1]] for kp in keypoints], dtype=np.float32)
    else:
      kpts = np.empty((0, 2), dtype=np.float32)
      descriptors = np.empty((0, 32), dtype=np.uint8)

    return kpts, descriptors
```

Рис. 7 Метод для отримання візуальних ознак із зображення за допомогою ORB

```python
def extract_features_superpoint(frame, superpoint):
    """
    Extracts SuperPoint keypoints, descriptors, and confidence scores from a drone frame.

    Parameters:
      frame (np.ndarray): Input color image (BGR).
      superpoint (nn.Module): A PyTorch model instance of the SuperPoint network.

    Returns:
      kpts (np.ndarray): Array of keypoints, shape (N, 2) with (x, y) coordinates.
      desc (np.ndarray): Corresponding descriptors, shape (N, 256), float32.
      scores (np.ndarray): Confidence scores for each keypoint, shape (N,).
    """
    # Convert the input BGR image to grayscale
    frame_gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Resize the grayscale image to ensure valid shape
    frame_gray = cv2.resize(frame_gray, (frame_gray.shape[1], frame_gray.shape[0]))

    # Normalize the grayscale image to range [0, 1] and convert to a 4D tensor
    frame_tensor = torch.from_numpy(frame_gray / 255.).float()[None, None].to(device)

    # Disable gradient calculation for inference
    with torch.no_grad():
      result = superpoint({'image': frame_tensor})

    # Extract keypoints as (x, y) coordinates from the output
    kpts = result['keypoints'][0].cpu().numpy()

    # Extract descriptors and transpose to shape (N, 256)
    desc = result['descriptors'][0].cpu().numpy().T

    # Extract confidence scores for each keypoint
    scores = result['scores'][0].cpu().numpy()

    return kpts, desc, scores
```

Рис. 8 Метод для отримання візуальних ознак із зображення за допомогою SuperPoint

Вибір фрагментів для порівняння з кадром дрона

Щоб зменшити обчислювальні витрати та врахувати напрям руху дрона, кадр зіставляється не з усіма фрагментами карти, а з обмеженим набором заздалегідь відібраних кандидатів. На першому кроці пошук здійснюється в межах початкової зони площею m × m фрагментів з центром у стартовій позиції. Коли вже була виконана хоча б одна локалізація, набір фрагментів-кандидатів обирається динамічно. Для цього оцінюється відстань, яку дрон може пройти за останній часовий інтервал й обчислюється основний напрямок руху – це кут між оціненим положенням дрона та цільовою точкою. Далі генеруються основний і з відхиленням напрями, для кожного з яких розраховується очікувана позиція дрона. Навколо кожної такої точки вибираються фрагменти в межах заданого радіуса, що й утворює підмножину кандидатів. При попередній невдалій локалізації кандидатами стають всі доступні фрагменти карти.

Зіставлення дескрипторів між фрагментом карти і кадром

Підхід ORB

Зіставлення дескрипторів виконується окремо для кожного фрагмента-кандидата алгоритмом BFMatcher за Hamming-відстанню та пошуком двох найближчих сусідів. Також застосовується тест відношення Лоу, за яким зберігаються лише ті відповідності, де відстань до найближчого сусіда значно менша, ніж до другого. Далі відбираються n найкращих відповідностей на фрагмент. Кожна знайдена відповідність трансформується з локальних координат фрагмента в глобальні координати карти. Щоб уникнути надмірної концентрації точок у невеликій області, додатково фільтруються дублікати за просторовою сіткою: якщо точка карти вже зустрічалася в околі, то вона ігнорується. Усі валідні відповідності з усіх фрагментів-кандидатів збираються в єдиний масив точок кадру та відповідних точок карти, які згодом використовуються для побудови гомографії.

Підхід SuperPoint + SuperGlue

У нейромережевому підході відповідності між кадром і фрагментами карти визначаються одночасно, у батчі. Ключові точки та дескриптори з кадру дрона, а також з усіх фрагментів-кандидатів, передаються в модель SuperGlue, яка

навчається знаходити відповідності з урахуванням просторових зв’язків і контексту. Для уніфікації розмірів у батчі ключові точки та дескриптори фрагментів доповнюються до максимальної довжини. У вигляді тензорів до SuperGlue передаються дані: координати ключових точок, дескриптори та оцінки надійності. Модель повертає індекси пар відповідностей, де для кожної точки з кадру вказано відповідну точку на карті або -1, якщо відповідність не знайдена. Усі знайдені відповідності трансформуються з координат фрагмента до глобальних координат карти. На відміну від ORB, відповідності не розбиваються за фрагментами, і не застосовується фільтрація за просторовою сіткою, оскільки модель сама навчається уникати дублів і вибирати найнадійніші пари.

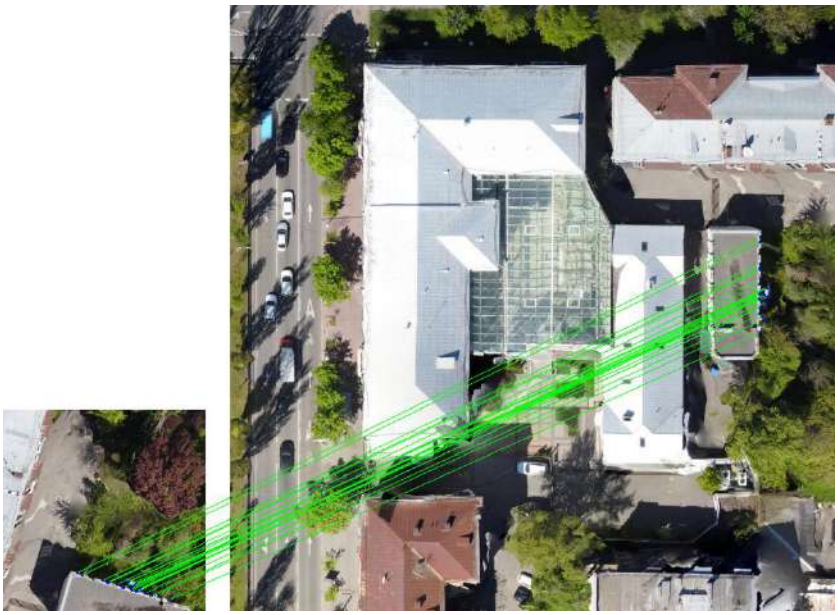

Рис. 9 Збіги ключових точок (ORB) між кадром дрона та фрагментом карти

Визначення позиції дрона

Для визначення позиції дрона на карті будується гомографія між точками кадру з камери дрона та відповідними точками на фрагментах карти. Якщо існує хоча б чотири відповідності, тоді за допомогою методу findHomography із використанням RANSAC з бібліотеки OpenCV обчислюється матриця гомографії, яка описує проєктивне перетворення між двома площинами. Через цю матрицю центр кадру проектується на карту, і таким чином визначається його положення в піксельних координатах на зображенні ортофото. Піксельні координати трансформуються в метричну систему проєкції GeoTIFF за

допомогою афінного перетворення, а потім – у глобальні GPS-координати (широту і довготу) через перетворення координатної системи. Отже, локалізація дрона в умовах відсутності сигналу GPS здійснюється на основі візуального зіставлення, побудови матриці гомографії та перетворення координат.

Стратегії руху до цілі

Дрон поступово наближається до цільової точки через постійну локалізацію та вибір напрямку руху. Якщо знайдено достатню кількість відповідностей між кадром і фрагментами карти, тоді обчислюється гомографія та оцінюється поточна позиція дрона. Після цього кут до цілі визначається як азимут між координатами оціненої позиції та цілі. Реальне положення дрона зміщується відповідно до цього напрямку з урахуванням часу, витраченого на поточний крок, і заданої швидкості. При невдалій локалізації дрон все одно продовжує рух. У такому випадку стратегія залежить від наявності попередньої оцінки положення. Якщо існує така оцінка, береться кут між останньою оціненою позицією та цільовою точкою. Якщо ж оціненого положення немає, дрон рухається у фіксованому напрямку, заданому на початку місії як напрямок від стартової позиції до цілі. Такий підхід гарантує поступовий рух до цілі навіть при тимчасовій втраті візуальної інформації або перебування в областях із низькою кількістю відповідностей.

Завершення руху

Рух дрона завершується, якщо досягнуто цільову точку або вичерпано граничний час місії. Досягнення цілі визначається за допомогою географічної відстані між оціненою позицією дрона та заданими GPS-координатами цілі.

Результати роботи

У межах роботи було повністю реалізовано алгоритм оптичної навігації дрона за кадрами з камери та попередньо обробленою картою місцевості. Для виявлення та зіставлення ознак використано два підходи: класичний (ORB) та нейромережевий (SuperPoint + SuperGlue).

Алгоритм з підходом ORB було запущено на CPU в Google Colab, а з підходом SuperPoint + SuperGlue – на L4 GPU в Google Colab.

У результаті 30 запусків алгоритму для кожного з двох методів було отримано такі середні показники:

ORB

- Середній час витягування візуальних ознак з кадру – 0,113 с ± 0,029
- Середній час зіставлення візуальних ознак кадру з ознаками обраних фрагментів карти – 0,313 с ± 0,048
- Середній час визначення положення дрона – 0,507 с ± 0,152

У всіх виконаннях алгоритму ціль було досягнуто.

SuperPoint + SuperGlue

- Середній час витягування візуальних ознак з кадру – 6,287 с ± 0,209
- Середній час зіставлення візуальних ознак кадру з ознаками обраних фрагментів карти – 14,087 с ± 9,957
- Середній час визначення положення дрона – 20,901 с ± 9,846

У всіх виконаннях алгоритму ціль було досягнуто.

Незважаючи на триваліший час обробки у випадку нейромережевого підходу, за рахунок адаптивного вибору частоти обробки кадрів відповідно до швидкості БПЛА, алгоритм успішно досягав цілі у всіх експериментах.

Таким чином алгоритм з підходом ORB можна використовувати для БПЛА зі швидкістю до 300 км/год, а з підходом SuperPoint + SuperGlue – до 20 км/год.

При використанні обчислювальних платформ на зразок Raspberry Pi, рекомендовано зменшити роздільну здатність кадрів у ~2,5 рази або компілювати код на C++ для підвищення швидкодії.

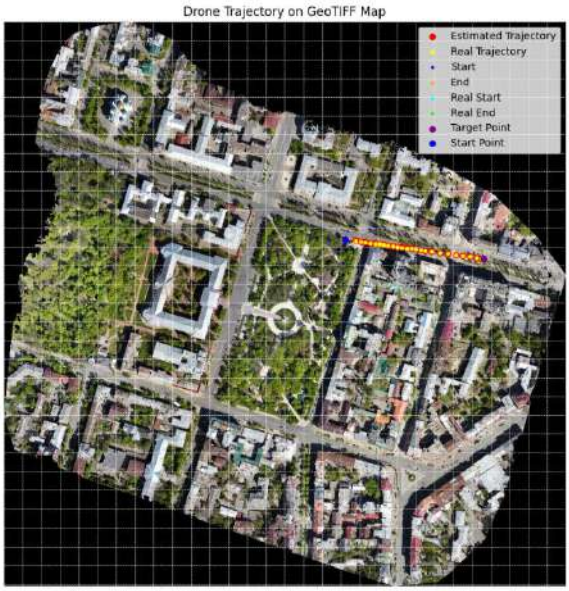

Рис. 10 Приклад відпрацювання алгоритму з підходом ORB

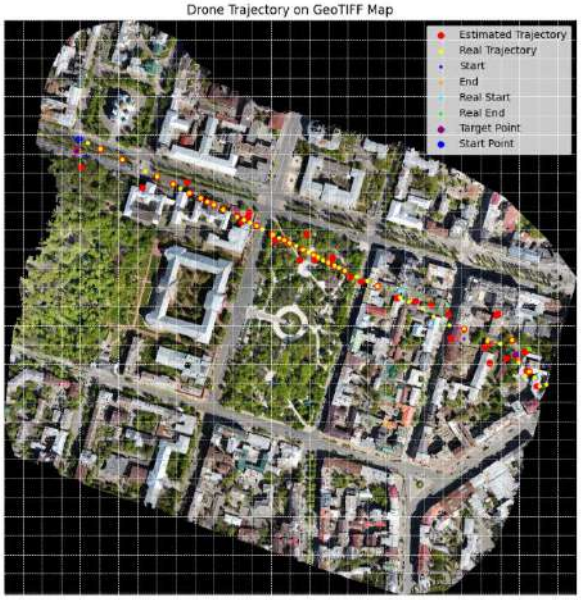

Рис. 11 Приклад відпрацювання алгоритму з підходом SuperPoint + SuperGlue

Висновки до розділу 2

Реалізовано та протестовано повноцінний алгоритм оптичної навігації дрона в умовах відсутності GPS-сигналу. Було виконано попередню обробку ортофото карти: розбиття на фрагменти, обчислення GPS-координат, виявлення ключових точок і дескрипторів двома підходами: ORB та SuperPoint + SuperGlue. Розроблений алгоритм виконує зіставлення ознак кадру з камери дрона з ознаками обраних фрагментів карти, оцінює положення дрона через гомографію та реалізує поступовий рух до цілі, навіть у разі невдалої локалізації. Для обох підходів реалізовано повний цикл навігації: створення кадру, його обробка, пошук відповідностей, визначення позиції, оновлення напрямку руху та завершення місії.

Експерименти довели надійність обох методів: при всіх запусках дрон успішно досягав цільової точки. Підхід ORB показав високу швидкодію, що дозволяє використовувати його на БПЛА зі швидкістю до 300 км/год, тоді як нейромережевий підхід мав більшу точність відповідностей (знайдено у 1,77 разів більше збігів по ключових точках на кадрі та карті), але потребував більше часу на обробку (в 137 разів), тому є доцільним для повільніших апаратів (до 20 км/год) або потужніших обчислювальних платформ.

Отримані результати підтверджують можливість використання алгоритму як складової автономної системи навігації дронів у GPS-недоступних середовищах.

Висновки

У магістерській роботі реалізовано алгоритм оптичної навігації дрона в умовах відсутності GPS-сигналу. Розроблене рішення поєднує попередню обробку ортофото карти, виділення ключових точок і дескрипторів, а також побудову гомографії для визначення положення дрона за кадрами з бортової камери.

Для виявлення та зіставлення ознак використано два підходи: класичний ORB та нейромережевий SuperPoint + SuperGlue. Обидва методи показали високу надійність у досягненні цільової точки, при цьому ORB продемонстрував вищу швидкодію, а нейромережевий підхід – кращу точність відповідностей.

Результати експериментів засвідчили, що розроблений алгоритм може бути інтегрований у системи автономної навігації БПЛА, особливо в умовах радіоелектронної боротьби або втрати GPS. Запропоноване рішення є перспективним для подальшого вдосконалення з урахуванням обмежених обчислювальних ресурсів реальних платформ.

Список використаних джерел

1. Богдан Мірошниченко, «На них не діє РЕБ. Як Україна розгортає виробництво дронів "на оптоволокні», 10 січня 2025, http://epravda.com.ua/oborona/shcho-take-droni-na-optovolokni-ta-yak-voni-zminyuyut-vse-na-poli-boyu-801851 (дата звернення: 02.03.2025)
2. «Наш захист від спуфінгу!», 31 липня 2023, https://pryvyd.com.ua/gps-filter/ (дата звернення: 07.01.2025)
3. Наталія Кушнерська, «Розумні дрони та швидка розвідка: які інновації пропонують українські розробники для фронту», 14 лютого 2025, https://24tv.ua/ukrayinski-rozrobniki-prezentuvali-novi-viyskovi-tehnologiyi_n2753131 (дата звернення: 15.03.2025)
4. Владислав Паливода, «Автономний БПЛА без GPS та GNSS з візуальною навігаційною системою», 20 грудня 2024, https://speka.media/v-ukrayini-pokazali-sistemu-yaka-dozvolyaje-dronam-litati-vnoci-bez-navigaciyi-shho-vidomo-py76kq, https://youtu.be/0oRSCjdpHfM?si=M1Z_Ocj8BSukOsl8 (дата звернення: 25.01.2025)
5. Kateryna Bondar, «Ukraine's Future Vision and Current Capabilities for Waging AI-Enabled Autonomous Warfare», 6 березня 2025, https://www.csis.org/analysis/ukraines-future-vision-and-current-capabilities-waging-ai-enabled-autonomous-warfare (дата звернення: 13.03.2025)
6. Muhammad Yeasir Arafat, Muhammad Morshed Alam, Sangman Moh, «Vision-Based Navigation Techniques for Unmanned Aerial Vehicles: Review and Challenges», 27 січня 2023, https://www.mdpi.com/2504-446X/7/2/89 (дата звернення: 15.01.2025)
7. Suraj Bijjahalli, Roberto Sabatini , Alessandro Gardi, «Advances in intelligent and autonomous navigation systems for small UAS», червень 2020, https://www.researchgate.net/publication/342027772_Advances_in_Intellig

ent_and_Autonomous_Navigation_Systems_for_small_UAS (дата звернення: 21.02.2025)

8. Richard Szeliski, «Computer Vision: Algorithms and Applications», 18 серпня 2010, https://vim.ustc.edu.cn/_upload/article/files/d4/87/71e9467745a5a7b8e80e94007d1b/4cd69b21-85d3-43ba-9935-fd9ae33da82b.pdf (дата звернення: 19.02.2025)
9. Andres Bruhn, Joachim Weickert, Christoph Schnorr, «Lucas/Kanade Meets Horn/Schunck: Combining Local and Global Optic Flow Methods», жовтень 2004, https://www.mia.uni-saarland.de/Publications/bruhn-ijcv05c.pdf (дата звернення: 03.02.2025)
10. Hanan Samet, «The Quadtree and Related Hierarchical Data Structures», червень 1984, https://users.umiacs.umd.edu/~ramanid/cmsc878R/p187-samet.pdf (дата звернення: 27.02.2025)
11. Jianli Wei, Alper Yilmaz, «A Visual Odometry Pipeline for Real-Time UAS Geopositioning», 5 вересня 2023, https://www.mdpi.com/2504-446X/7/9/569 (дата звернення: 09.03.2025)
12. «ЗІР. Система розпізнавання та ураження цілей», https://zir-system.com/ (дата звернення: 11.03.2025)
13. Кирило Кушніков, «Ukrainian Drones Use Vermeer Optical Navigation System», 27 лютого 2025, https://militarnyi.com/en/news/ukrainian-drones-use-vermeer-optical-navigation-system/ (дата звернення: 24.03.2025)
14. David G. Lowe, «Distinctive Image Features from Scale-Invariant Keypoints», 5 січня 2004, https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf (дата звернення: 18.01.2025)
15. Daniel DeTone, Tomasz Malisiewicz, Andrew Rabinovich, «SuperPoint: Self-Supervised Interest Point Detection and Description», 19 квітня 2018, https://arxiv.org/pdf/1712.07629v4 (дата звернення: 01.03.2025)

16. LPP Unmanned systems, «Autonomous UAV system», https://www.lpp-uas.com/ (дата звернення: 20.05.2025)
17. Pavel Otto, «Umělá inteligence místo operátora. České drony obstály ve válce na Ukrajině», 14 травня 2025, https://www.e15.cz/byznys/technologie-a-media/umela-inteligence-misto-operatora-ceske-drony-obstaly-ve-valce-na-ukrajine-1424473 (дата звернення: 16.05.2025)
18. OpenAerialMap, https://map.openaerialmap.org/ (дата звернення: 10.03.2025)
19. SuperGluePretrainedNetwork, https://github.com/magicleap/SuperGluePretrainedNetwork (дата звернення: 22.04.2025)
20. «Як СБУ організувала атаку на аеродроми РФ? Нові деталі», 2 червня 2025, https://www.radiosvoboda.org/a/operatsiya-sbu-pavutyna-detali/33431885.html (дата звернення: 02.06.2025)
21. «Павутина Малюка: що відомо про вражаючу спецоперацію СБУ, під час якої дрони з фур уразили більше 40 російських літаків», 1 червня 2025, https://nv.ua/ukr/ukraine/events/specopreaciya-pavutina-yak-droni-z-fur-urazili-desyatki-rosiyskih-litakiv-strategichnoji-aviaciji-50518604.html (дата звернення: 02.06.2025)
22. Станіслав Погорілов, «Малюк підбив підсумки спецоперації СБУ зі знищення стратегічних літаків РФ», 2 червня 2025, https://www.pravda.com.ua/news/2025/06/2/7515181/ (дата звернення: 02.06.2025)
23. Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, Andrew Rabinovich, «SuperGlue: Learning Feature Matching with Graph Neural Networks», 28 березня 2020, https://arxiv.org/pdf/1911.11763 (дата звернення: 18.04.2025)
24. Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary Bradski, «ORB: an efficient alternative to SIFT or SURF», 6-13 листопада 2011,

https://www.researchgate.net/publication/221111151_ORB_an_efficient_alternative_to_SIFT_or_SURF (дата звернення: 04.04.2025)

25. «Feature Matching in Computer Vision: Techniques and Applications», 30 липня 2024, https://www.geeksforgeeks.org/feature-matching-in-computer-vision-techniques-and-applications/ (дата звернення: 08.04.2025)
26. Siddharth Agarwal, «Homography - And how to calculate it?», 26 січня 2020, https://medium.com/all-things-about-robotics-and-computer-vision/homography-and-how-to-calculate-it-8abf3a13ddc5 (дата звернення: 11.04.2025)
27. «Basic concepts of the homography explained with code», https://docs.opencv.org/4.x/d9/dab/tutorial_homography.html (дата звернення: 13.04.2025)
28. «Affine Transformations», https://docs.opencv.org/4.x/d4/d61/tutorial_warp_affine.html (дата звернення: 13.04.2025)
29. Michal Aibin, «Differences Between Homography and Affine Transformation», 18 березня 2024, https://www.baeldung.com/cs/homography-vs-affine-transformation (дата звернення: 13.04.2025)

Список використаних зображень

1. з1 – [Електронний ресурс] – Режим доступу: https://www.researchgate.net/profile/Syed-Izzat-Ullah/publication/369862695/figure/fig2/AS:1143128116894388 4@1687175753195/An-illustration-of-the-Octomap-octree-representation-9_W640.jpg
2. з2 – [Електронний ресурс] – Режим доступу: https://leohung.github.io/images/ammai/03/%E8%9E%A2%E5%B9%95%E5%BF%AB%E7%85%A7-2012-03-13-%E4%B8%8B%E5%8D%888.34.58.png