

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Факультет програмної інженерії та бізнесу

Кафедра інженерії програмного забезпечення

Пояснювальна записка до дипломного проекту

магістра

(освітній ступінь)

на тему «Модель створення об'єктів доповненої реальності з візуальною
одометрією»

XAI.603.667п1.121.156335.200

Виконав: студент 6 курсу групи № 667п1
Спеціальність 121 – Інженерія програмного
забезпечення

(код та найменування)

Освітня програма Хмарні обчислення та
Інтернет речей

(найменування)

Заїка В.І.

(прізвище й ініціали студента)

Керівник Конорев Б.М.

(прізвище та ініціали)

Рецензент Іващенко Г.С.

(прізвище та ініціали)

Харків – 2020

Міністерство світи і науки України
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Факультет програмної інженерії та бізнесу
(повне найменування)

Кафедра інженерії програмного забезпечення
(повне найменування)

Рівень вищої освіти другий (магістерський)

Спеціальність 121 – інженерія програмного забезпечення
(код та найменування)

Освітня програма хмарні обчислення та Інтернет речей
(найменування)

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис) (ініціали та прізвище)
“ ____ ” _____ 2020 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ

Заїки Володимиру Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема дипломного проекту Модель створення об'єктів доповненої реальності з візуальною одометрією

керівник дипломного проекту Конорев Борис Михайлович, д.т.н, професор
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “ ____ ” ____ 2020 року № ____

2. Термін подання студентом роботи _____

3. Вихідні дані до роботи: мобільного застосунку для формування об'єктів доповненої реальності

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

1 дослідити сучасні методи та підходи формування доповненої реальності;

2 розробити модель створення об'єктів доповненої реальності на основі механізмів візуальної одометрії;

3 розробити прототип мобільного застосунку для формування об'єктів доповненої реальності з використанням розробленої моделі;

4 провести тестування прототипу мобільного застосунку для формування доповненої реальності.

5. Перелік графічного матеріалу

РПЗ – стор. 86, рисунків – 32 шт., таблиць – 1 шт., презентація – 18 слайдів.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Конорев Б. М., проф. каф. 603		
2	Конорев Б. М., проф. каф. 603		
3	Конорев Б. М., проф. каф. 603		

8. Нормоконтроль _____ В.А. Постернакова « ____ » _____ 2020 р.
(підпис) (ініціали та прізвище)

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту	Строк виконання етапів проекту	Примітка
1	Отримання і затвердження теми диплому	03.09.2019	
2	Аналіз предметної області	04.09.2019	
3	Постановка задачі	20.11.2019	
4	Проведення теоретичних досліджень	22.11.2019	
5	Розробка прототипу ПЗ	02.09.2020	
6	Підготовка пояснювальної записки	22.10.2020	
7	Оформлення пояснювальної записки до дипломного проекту	10.11.2020	
8	Передзахист дипломного проекту	24.11.2020	
9	Захист дипломного проекту	03.12.2020	

Студент _____

(підпис)

Заїка В.І.

(прізвище та ініціали)

Керівник роботи _____

(підпис)

Конорев Б.М.

(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломного проекту містить 86 стор., 32 рис., 1 додаток, 22 джерел.

Об'єкт дослідження – процес створення доповненої реальності.

Предмет дослідження – моделі та методи створення доповненої реальності.

Метою досліджень є підвищення ефективності методів формування доповненої реальності шляхом використання механізмів візуальної одометрії за рахунок розроблення мобільного застосунку для формування доповненої реальності.

Для досягнення мети необхідно виконати наступні завдання: дослідити сучасні методи та підходи формування доповненої реальності; розробити модель створення об'єктів доповненої реальності на основі механізмів візуальної одометрії; розробити прототип мобільного застосунку для формування об'єктів доповненої реальності з використанням розробленої моделі; провести тестування прототипу мобільного застосунку для формування доповненої реальності.

Методи досліджень. У роботі було використано методи розпізнавання образів, методи моделювання, методи системного аналізу.

Наукова новизна. Удосконалено модель створення об'єктів доповненої реальності яка на відміну від існуючих використовує механізми візуальної одометрії для одночасної локалізації та картографування об'єктів у процесі побудови доповненої реальності що дає можливість підвищити точність формування віртуальних локацій об'єктів доповненої реальності.

Практичне значення отриманих результатів. В результаті роботи був розроблений мобільний застосунок для формування доповненої реальності, який може бути використаний при роботі з різноманітними об'єктами в інтер'єрі.

ДОПОВНЕНА РЕАЛЬНІСТЬ, ВІЗУАЛЬНА ОДОМЕТРІЯ,
ЛОКАЛІЗАЦІЯ, КАРТОГРАФУВАННЯ, ANDROID OS, KOTLIN

ABSTRACT

Explanatory note to the graduate work contains 86 pp., 32 fig., 1 application, 22 sources.

The object of research is the process of creating augmented reality.

The subject of research - models and methods of creating augmented reality.

The aim of the research is to increase the efficiency of augmented reality formation methods by using the mechanisms of visual odometry by developing a mobile application for augmented reality formation.

To achieve this goal it is necessary to perform the following tasks: to explore modern methods and approaches to the formation of augmented reality; develop a model of creating augmented reality objects based on the mechanisms of visual odometry; develop a prototype of a mobile application for the formation of augmented reality objects using the developed model; to test the prototype of a mobile application for the formation of augmented reality.

Research methods. The methods of image recognition, modeling methods, methods of system analysis were used in the work.

Scientific novelty. The model of creating augmented reality objects has been improved, which, unlike the existing ones, uses the mechanisms of visual odometry for simultaneous localization and mapping of objects in the process of constructing augmented reality, which makes it possible to increase the accuracy of virtual locations of augmented reality objects.

The practical significance of the results. As a result of work the mobile application for formation of the augmented reality which can be used when working with various objects in an interior was developed.

AUGMENTED REALITY, VISUAL ODOMETRY, LOCALIZATION, MAPPING, ANDROID OS, KOTLIN

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	8
ВСТУП.....	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ	12
1.1 Засоби формування доповненої реальності.....	12
1.1.1 Класифікація доповненої реальності	15
1.1.2 Аналіз існуючих застосувань доповненої реальності	20
1.2 Комп'ютерний зір та концепція Feature detection.....	24
1.3 Огляд принципу роботи доповненої реальності на мобільній платформі	28
1.4 Порядок розміщення об'єктів доповненої реальності у навколишньому середовищі	33
1.5 Популярні бібліотеки для формування доповненої реальності	35
1.6 Постановка мети і завдань дослідження.....	40
1.7 Висновки до розділу 1	41
2 МОДЕЛЬ СТВОРЕННЯ ОБ'ЄКТІВ ДОПОВНЕНОЇ РЕАЛЬНОСТІ НА ОСНОВІ ВІЗУАЛЬНОЇ ОДОМЕТРІЇ.....	42
2.1 Методи візуальної одометрії.....	41
2.1.1 Метод розширеного фільтру Калмана	47
2.1.2 Метод FastSLAM.....	50
2.1.3 Метод DP-SLAM	51
2.2 Модель створення об'єктів доповненої реальності на основі візуальної одометрії	53
2.3 Аналіз досвіду використання технології доповненої реальності.....	58
2.4 Висновки до розділу 2	63

3 РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ ФОРМУВАННЯ ОБ’ЄКТІВ ДОПОВНЕНОЇ РЕАЛЬНОСТІ.....	66
3.1 Проектування прототипу мобільного застосунку.....	66
3.1.1 Вимоги до архітектури мобільного застосунку	66
3.1.2 Структура класів мобільного застосунку	67
3.2 Опис функціональності мобільного додатку	70
3.3 Тестування розробленого мобільного застосунку.....	72
3.3.1 Тестування зміни положення об’єкту	72
3.3.2 Тестування позиції об’єкта.....	75
3.4 Аналіз результатів досліджень	78
3.5 Висновки до розділу 3	79
ВИСНОВКИ.....	80
ПЕРЕЛІК ПОСИЛАНЬ	82
<i>ДОДАТОК А</i>	84

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

VR – англ. Virtual Reality – віртуальна реальність.

AR – англ. Augmented Reality – доповнена реальність.

VO – візуальна одометрія.

МЗ – мобільний застосунок.

ОЛК – Одночасна локалізація та картографування.

ВСТУП

Актуальність теми. Доповнена реальність (в перекладі з англійської augmented reality або AR) – це доповнення фізичного світу за допомогою цифрових даних, яке забезпечується комп'ютерними пристроями (смартфонами, планшетами та окулярами AR) в режимі реального часу [1]. Про суть поняття «доповнена реальність» говорить сама назва – це технології, які доповнюють реальність віртуальними елементами:

- відображення віртуальних об'єктів на екрані пристроїв – комп'ютерів, планшетів, телефонів, шоломів;
- перегляд віртуальних об'єктів за допомогою спеціальних окулярів і шоломів;
- візуалізація об'єктів в реальності.

Основний момент при використанні доповненої реальності – це накладення віртуальних (нереальних) об'єктів на реальність, їх комбінування. У цьому істотна відмінність доповненої реальності від віртуальної.

Доповнена реальність дозволяє доповнити світ віртуальними об'єктами, звуками, образами. Вона може бути інтерактивною, тобто:

- на віртуальні об'єкти можна впливати (наприклад, доторкнутися на екрані планшета до зображення цуценя і побачити, як він завилляє хвостом у відповідь);
- з ними можна сфотографуватися (наприклад, в цікавій масці);
- з їх допомогою можна переходити на сайти і тому подібне, варіантів інтерактивності доповненої реальності сила-силенна.

В даний час у багатьох країнах доповнена реальність є важливою складовою великого бізнесу, який прагне освоювати нові технології та залучати нових клієнтів і партнерів. Крім того, доповнена реальність широко

використовується в таких соціально-значущих сферах, як охорона здоров'я, освіта і наука.

Найпростіший варіант елементів доповненої реальності – це зображення, 3D-модель або текст, які з'являються на екрані смартфона при наведенні його камери на якусь мітку. Міткою може служити якесь зображення чи навіть предмет. Таке використання технології доповненої реальності вельми і вельми корисно. З його допомогою можна отримати інформацію про визначні пам'ятки, подивитися меню в ресторані, більш наочно познайомитися зі шкільними предметами та ін.

Крім того, доповнена реальність дозволяє впливати на об'єкт. Наприклад, можна доторкнутися до моделі лицарських обладунків, повернути їх навколо своєї осі, розібрати на складові частини і прочитати інформацію про кожен з них. Можна доторкнутися до зображення тварини, і воно оживе: заговорить, потанцює, розповість про себе. Можна відкрити дверцята віртуального автомобіля і подивитися, як він виглядає всередині.

Наступний варіант – це та інформація, яку ми можемо бачити, надівши шолом або окуляри доповненої реальності. Це виглядає набагато більш вражаюче. Сучасні шоломи та окуляри доповненої реальності доповнюють реальний світ віртуальними об'єктами, які часом настільки реальні і так добре вписуються в навколишнє оточення, що викликають бурю емоцій.

Об'єкт дослідження – процес створення доповненої реальності.

Предмет дослідження – моделі та методи створення доповненої реальності.

Метою досліджень є підвищення ефективності методів формування доповненої реальності шляхом використання механізмів візуальної одометрії за рахунок розроблення мобільного застосунку для формування доповненої реальності.

Для досягнення мети необхідно виконати наступні завдання:

– дослідити сучасні методи та підходи формування доповненої реальності;

- розробити модель створення об'єктів доповненої реальності на основі механізмів візуальної одометрії;
- розробити прототип мобільного застосунку для формування об'єктів доповненої реальності з використанням розробленої моделі;
- провести тестування прототипу мобільного застосунку для формування доповненої реальності.

Методи досліджень. У роботі було використано методи розпізнавання образів, методи моделювання, методи системного аналізу.

Наукова новизна. Удосконалено модель створення об'єктів доповненої реальності яка на відміну від існуючих використовує механізми візуальної одометрії для одночасної локалізації та картографування об'єктів у процесі побудови доповненої реальності що дає можливість підвищити точність формування віртуальних локацій об'єктів доповненої реальності.

Практичне значення отриманих результатів. В результаті роботи був розроблений мобільний застосунок для формування доповненої реальності, який може бути використаний при роботі з різноманітними об'єктами в інтер'єрі.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Засоби формування доповненої реальності

Процес формування доповненої реальності відбувається за рахунок обробки зображення з камери телефону, веб-камери чи іншого пристрою процесором, та накладання віртуальних об'єктів на це зображення.

У якості елементів AR-технологій можуть виступати відео та аудіо матеріали, 2D та 3D-моделі, текстовий контент. У системі доповненої реальності використовуються спеціальні маркери або мітки, котрі сприймаються камерою та слугують основою для позиціонування елементів AR (рисунок 1.1).



Рисунок 1.1 – Використання доповненої реальності

Якщо алгоритм програмного забезпечення розпізнав маркер на зображенні з камери, позиція цього маркеру фіксується і на його місці відображається відповідний елемент доповненої реальності. Існують також системи без використання маркерів, вони працюють відслідковуючи геометрію та взаємне місцеположення реальних об'єктів [3]. В цьому випадку будується система

координат, початок якої знаходиться у певній точці реального навколишнього середовища. Зазвичай при цьому використовуються й інші сенсори, такі як гіроскоп чи акселерометр, для орієнтування у побудованій системі координат.

Для вирішення завдання побудови доповненої реальності скористаємося бібліотекою OpenCV [3]. За допомогою неї буде проводитись пошук маркера на відеопотоці, а потім відображення віртуального об'єкта. Безпосередньо сам процес пошуку ми описувати не будемо, оскільки для цього використовуються алгоритми feature point detection, описані вище. Перш за все, нас цікавить питання побудови 3D простору по знайденої 2D гомографії.

Під гомографією розуміється матриця побудови взаємно-однозначного відображення проектованого простору на себе, що переводить точки в точки, прямі в прямі і зберігає відношення інцидентності точок і прямих, а також подвійне ставлення будь-якої четвірки колінеарних точок. Вона дозволяє приводити зображення до єдиної перспективи і геометрії.

Для побудови 3D простору нам необхідно знати дві матриці: внутрішню (intrinsic matrix) і зовнішню (extrinsic matrix). Потім, скориставшись методами OpenGL, можна без будь-яких проблем намалювати віртуальний об'єкт поверх маркера.

Внутрішня матриця (або матриця проекції) складається з параметрів використовуваної камери: фокальної відстані по двох осях (f_x , f_y) і координат центру фокуса (c_x , c_y).

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (1.1)$$

Процес знаходження параметрів камери називається її калібрацією. Для його проведення необхідно зробити знімки каліброваного шаблону (шахівниці) за допомогою використовуваної камери, перенести фотографії на комп'ютер і

провести підрахунок параметрів матриці. З метою отримання якомога більш точних параметрів слід дотримуватись таких умов:

- шаблон роздруковується на чистому аркуші формату А4;
- лист шаблону повинен лежати рівно на поверхні, краї не повинні бути загнутими, бажано відсутність будь-яких вигинів;
- розмір знімків з камери повинен бути приведений до розміру кадру відеопотоку. У деяких випадках можлива ситуація, коли дозвіл фотографій вище дозволу відео. Тому, перед калібруванням необхідно зменшити знімки до потрібного розміру;
- кількість знімків шаблону повинно бути не менше 10. Виконати їх потрібно з різних ракурсів. Чим більше буде знімків, тим точніші будуть отримані параметри матриці проєкції, а ця точність далі буде впливати на наявність / відсутність зрушень при побудові 3D об'єктів.

Зовнішня матриця (або матриця моделі) – це матриця перетворень моделі за допомогою розтягування, повороту і перенесення. Вона дозволяє однозначно задати положення об'єкта в просторі. Діагональні елементи відповідають за розтягнення моделі.

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}, \quad (1.2)$$

де елементи r відповідають за поворот об'єкту в просторі;

t – за перенесення об'єкту.

Інші елементи r – за поворот об'єкту в просторі. Елементи t відповідають за перенесення. В цілому, структура даної матриці може варіюватися в залежності від рівнянь перетворень координат, але описані ключові елементи в ній присутні завжди.

Обчислення її виробляється шляхом визначення чотирьох пар точок об'єкта та відповідного його положення в кадрі. Точки положення в кадрі – це

вершини чотирикутника, що описує (обмежує) об'єкт в кадрі. Отримати дані точки можна, якщо подіяти на крайні точки шаблону гомографією.

Щодо точок самого об'єкта варто відзначити кілька моментів:

– точки об'єкта задаються в 3D, а точки на кадрі в 2D. Відповідно, якщо задати їм нульове значення z , то початок координат на z буде зрушено щодо площині об'єкта на кадрі;

– точки об'єкта задаються таким чином, щоб було зручно працювати в 3D просторі, а саме: початок координат знаходиться прямо по центру шаблону, а одиниця довжини дорівнює половині меншої сторони. У цьому випадку, ми виключаємо залежність від конкретних розмірів шаблону в пікселях, а 3D простір буде масштабований по меншій стороні.

Сконструйовані матриці потім використовуються для обчислення вектора повороту і перенесення з подальшим занесенням в матрицю моделі. Варто зазначити, що процес обчислення матриці моделі займає кілька мілісекунд і суттєво не впливає на швидкість роботи програми. Основне навантаження йде в області аналізу відеопотоку і пошуку на ньому шаблонного зображення.

Описаний вище підхід можна застосувати до всіх бібліотек комп'ютерного зору в плані побудови доповненої реальності. З практичної точки зору він виглядає дещо складним з огляду на необхідність проведення калібрування камери для отримання коректної матриці проектування. Одним із способів вирішення цієї проблеми є створення набору параметрів різних камер; в залежності від використовуваної камери буде проводитися завантаження необхідних параметрів.

1.1.1 Класифікація доповненої реальності

Доповнена реальність поділяється на різні категорії. Кожна категорія має різне використання та застосовується в різних сценаріях для вирішення специфічних завдань. Нижче наведені основні типи доповненої реальності [2]:

- доповнена реальність на основі маркерів (Marker-based AR);

- доповнена реальність без маркерів (Marker-less AR);
 - проєкційна доповнена реальність (Projection AR);
 - доповнена реальність на основі накладання (Superimposition based AR).
- Доповнена реальність на основі маркерів також називається розпізнавання зображень або розпізнаванням на основі AR. Приклад наведено на рисунку 1.2.



Рисунок 1.2 - Приклад AR технології, що базується на маркерах

Простими словами, ця технологія використовує камеру в пристрої AR для отримання результату. QR код та 2D фотографія є прикладами візуального маркера.

Користувачі отримують результат, коли програма розпізнає зображення з камери як маркер. Технологія AR, що базується на маркерах, має наступні використання:

- виявляє об'єкт перед камерою і дає інформацію про виявлений об'єкт на екрані;

- переводить слова, які фіксує камера, за допомогою технології оптичного розпізнавання символів і показує їх у перекладеній версії;
- технологія також корисна в освіті, оскільки вона може допомогти учням перетворити свою уяву в реальність;
- допомагає у створенні 3D моделей об'єктів або архітектури без побудови об'єкту у фізичній формі.

Доповнена реальність без маркерів також називається AR на основі визначення місцезнаходження. Приклад наведено на рисунку 1.3.

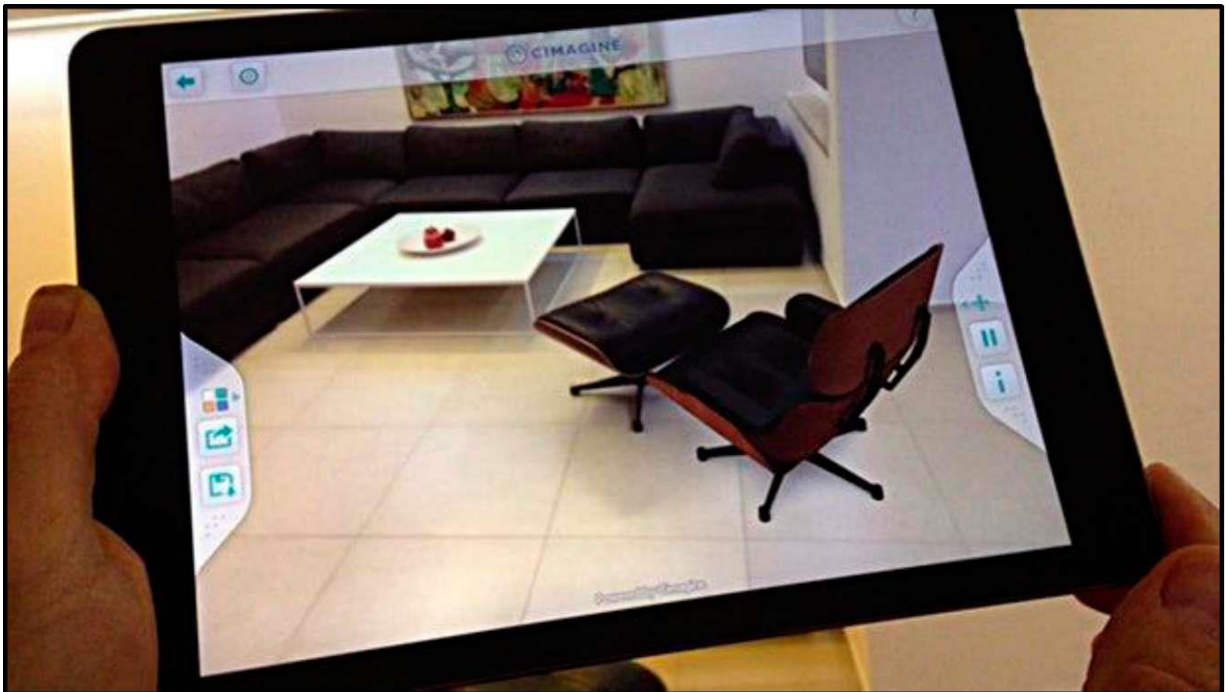


Рисунок 1.3 - Приклад доповненої реальності без маркерів

Це один з найбільш широко використовуваних і впроваджених типів технологій. Це єдиний тип, котрий не використовує ніякої системи розпізнавання існуючого об'єкту. Саме тому система не знає про об'єкт нічого, в той час як AR на основі маркерів повинно знати. Щоб забезпечити ефективний результат, ця технологія використовує різні інструменти розміщення та допомагає обчислити розташування за допомогою сенсорів пристрою. Наприклад, ви можете

помістити віртуальні меблі до будь-якої кімнати, бо системі все одно куди вони поставлені.

Інструментами визначення розташування є: GPS, цифровий компас, вимірювач швидкості і акселерометр. Ця технологія виявляється в якості допомоги ненажерливим мандрівникам у виявленні їхніх місць / напрямків і працює, визнаючи поточне місцезнаходження користувача. Вона визначає орієнтацію, читає дані через GPS і прогнозує місця, куди користувач хоче піти, і, нарешті, показує відповідну інформацію на своєму екрані смартфона користувача. Зіставлення напрямків та пошук у прилеглих місцях - це інші види використання AR на основі місцезнаходження.

Проекційна доповнена реальність заснована на функції проєкції AR, де світло від пристрою проєктується на об'єкти. Приклад наведено на рисунку 1.4.



Рисунок 1.4 - Приклад проєкційної доповненої реальності

Існують численні підходи, які роблять цю технологію більш цікавою. Один з підходів полягає в тому, щоб проєктувати світло на поверхню реального середовища і зробити його інтерактивним за допомогою сенсорів. Проєкція

також може бути виконана в повітрі за допомогою технології лазерної плазми. Це допомагає у вивченні структури та конфігурацій проєктованих зображень. Наприклад, коли світло калькулятора буде спроектовано через проєкційне пристрій AR на долоню, 12 розділів на вашій руці стануть 12 кнопками, і користувач може торкатися цих кнопок та взаємодіяти з ними.

Доповнена реальність на основі накладання є типом технології, де розпізнавання об'єктів відіграє надзвичайно важливу роль. У цій технології доповнене зображення може замінити оригінальне зображення, частково або повністю. Технологія корисна в медичній сфері. Лікар може ретельно обстежити пацієнта і дати належне лікування. Приклад доповненої реальності на основі накладання наведено на рисунку 1.5.

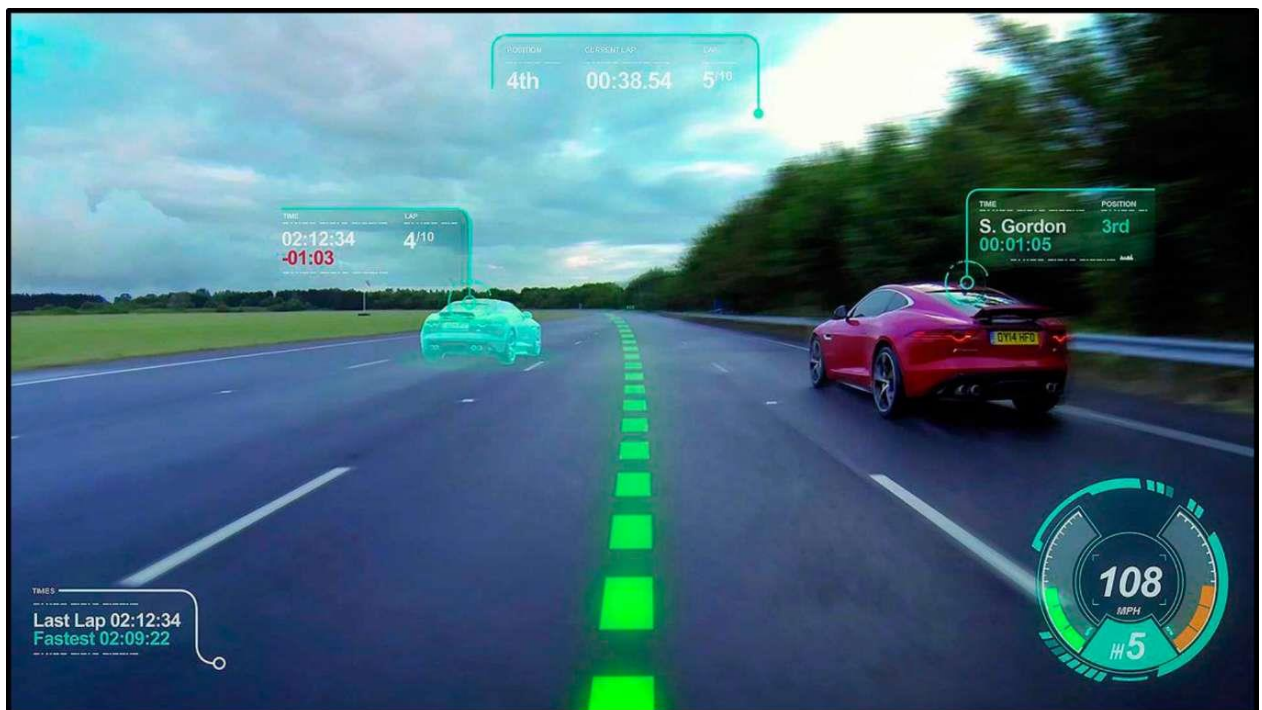


Рисунок 1.5. - Доповнена реальність на основі накладання

Технологія також є корисною у військовому застосуванні, оскільки AR на основі накладання може запропонувати кілька поглядів на ціль без будь-якого відволікання. Ця технологія була б корисною для любителів історії, оскільки

накладання стародавніх зображень поверх сучасних може надати багато інформації про минуле. Це також може допомогти зробити наукову освіту більш цікавою, наприклад, при вивченні структури кісток.

Пройшли ті часи, коли доповнена реальність обмежувалася лише науково - фантастичними романами та фільмами. Сьогодні всі користувачі смартфонів можуть застосовувати цю технологією для вирішення повсякденних справ. Поступово технологія AR розширила свої розміри і в даний час використовується в різних областях, таких як військова, медична, освітня, мистецька, архітектурна та інші. Але навіть зараз її потенціал дуже великий, бо вона відносно молода в порівнянні з іншими інформаційними та комп'ютерними технологіями.

1.1.2 Аналіз існуючих застосувань доповненої реальності

Доповнена реальність в освіті надає інформації можливість бути інтерактивною та більш придатною для сприйняття і запам'ятовування учнями та студентами. Існує багато застосування направлених на вирішення конкретних задач в сфері навчання [3]. Доповнена реальність може створити деякі дивовижні моменти під час процесу навчання. Зокрема, вона є відносно новою в цій сфері. Крім того, вона ще не стала повсякденною, тому багато людей не розуміє її переваг. Існує ряд прикладів, які можуть показати важливість AR в освіті.

Наприклад, існує додаток Dinosaur 4D+, який побудовано за допомогою доповненої реальності на основі маркерів (див. рисунок 1.6). Він може служити як посібник динозаврів, де маркерами є їх картки з зображеннями. Коли користувач відкриває мобільний додаток та наводить камеру на картку (маркер), він бачить 3D модель динозавру. Цей додаток може використовуватися на уроках біології в школі, щоб мотивувати учнів запам'ятовувати факти про динозаврів, та допомагати краще запам'ятовувати як вони виглядають. Крім цього, усі моделі анімовані та можуть супроводжуватися звуком, що створює більш глибокий досвід занурення в при використанні додатку.



Рисунок 1.6. - Мобільний додаток Dinosaur 4D+

Дивлячись на додаток з доповненою реальністю може з'явитися питання. Воно полягає в тому, чи має таке 3D відображення речей який-небудь вплив на успіх студентів у навчанні? Чи є якась користь від використання доповненої реальності в класі? Відповідь так. Крім того, що студенти бачать моделі на сенсорному екрані, вони також можуть взаємодіяти з ними: обертати, маніпулювати та масштабувати. Тобто, добре реалізована доповнена реальність дозволяє студентам бачити моделі природним шляхом. Вони можуть переміщати об'єкти і отримувати більш докладну інформацію про них, що дозволяє швидко їх розуміти. Як результат, студенти можуть легко аналізувати властивості об'єктів з більшою точністю і свободою. Саме тому інтерес до вивчення речей збільшується. Більше того, такий підхід робить класи більш цікавими та не нудними.

Також, існує додаток зі схожим принципом роботи, заснованому на маркерах, котрий дозволяє користувачам вивчати анатомію тіла людини. Назва додатку.

Anatomy 4D, а працює він з посібником по анатомії, в котрому на сторінках надруковані зображення-маркери для розпізнавання [4]. Додаток є досить складним, бо він дозволяє маніпулювати багатьма сутностями. Наприклад, маркер людини може служити для розпізнавання та відображення моделей лімфатичної, нервової, м'язової системи людини та її скелета. Інтерфейс користувача додатку Anatomy 4D наведено на рисунку 1.7.

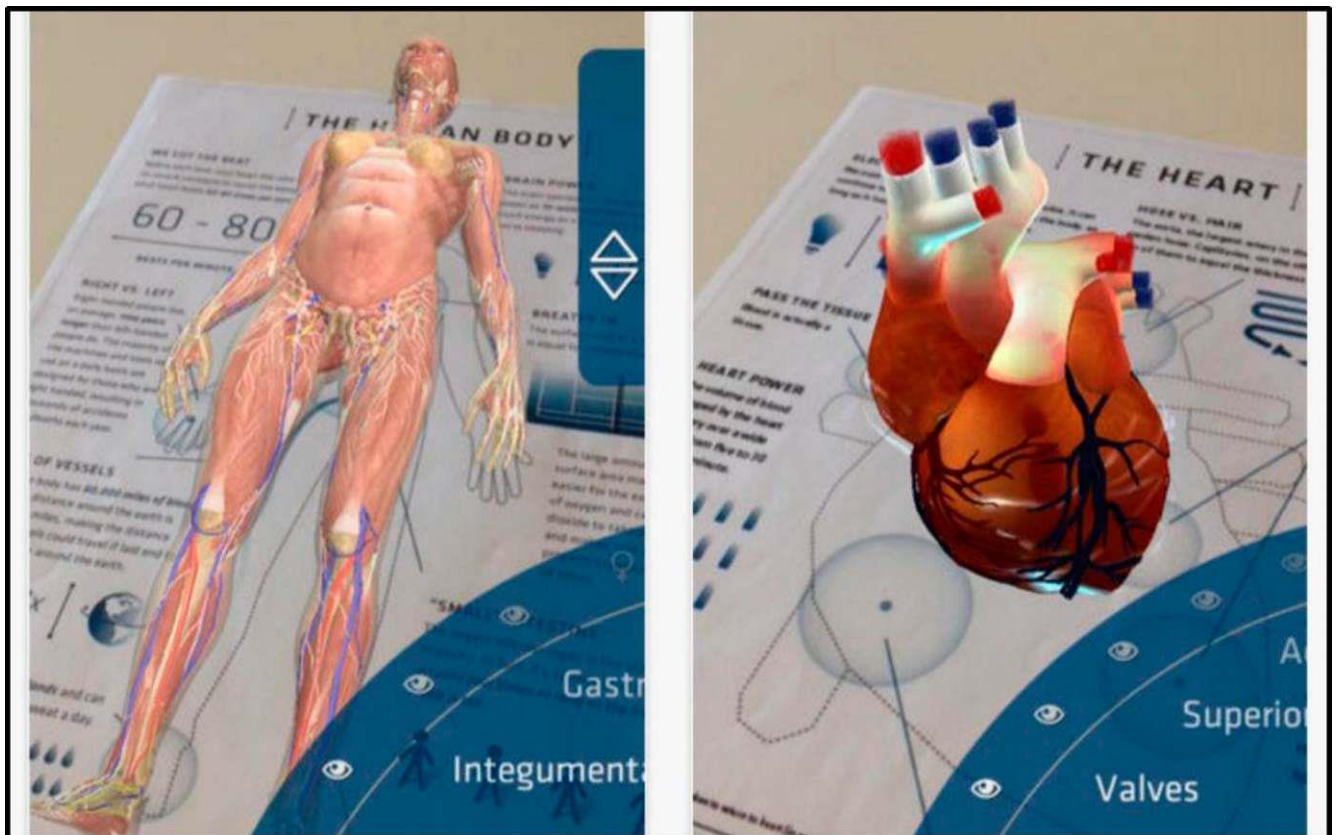


Рисунок 1.7 - Знімки екрану мобільного додатку Anatomy 4D

Іншим прикладом використання доповненої реальності на мобільних додатках є Google Translate. Принцип роботи цієї функції додатку в тому, що вона розпізнає текст, надрукований або написаний однією мовою та перекладає його на іншу задану мову. Що цікаво, ця функція працює в режимі реального часу і вона замінює текст оригінальної мови на обрану одразу на екрані девайсу, котрий застосовує користувач. Наприклад, користувач відкриває додаток та

вибирає мову оригіналу японську, а мову перекладу англійську. Після цього відкриває камеру та наводить фокус на текст, надрукований мовою оригіналу. Додаток замінює цей текст на англійський текст і створює ілюзію, нібито на поверхні цей текст надруковано англійською. Додаток використовує доповнену реальність на основі накладання. Приклад живого перекладу додатку Google Translate наведено на рисунку 1.8.



Рисунок 1.8 - Приклад живого перекладу за допомогою Google Translate

Інтелектуальний алгоритм перш за все за допомогою штучного інтелекту розпізнає символи та формує слова. Після цього доступний текст трансліюється з мови оригіналу в обрану мову. Наступний етап використовує технологію доповненої реальності, в даному випадку накладення віртуального об'єкту на реальний контекст. Наприклад, на зображенні вище контекстом є лист паперу, а віртуальний об'єкт англійського перекладу тексту накладається поверх реально надрукованих японських символів. Таке використання доповненої реальності дуже корисне в ситуації, коли користувач вивчає іноземну мову та не може надрукувати текст.

1.2 Комп'ютерний зір та концепція Feature detection

Теорія комп'ютерного зору є основоположною для розвитку технологій доповненої реальності, і використовується під час розпізнавання маркерів [2]. Основний напрямок даної дисципліни – це аналіз і обробка зображень (в тому числі і відеопотоку). Алгоритми комп'ютерного зору дозволяють виділяти ключові особливості на зображенні (кути, межі області), проводити пошук фігур і об'єктів в реальному часі, виконувати 3D реконструкцію з кількох фотографій і багато іншого.

В області доповненої реальності алгоритми комп'ютерного зору використовуються для пошуку в відеопотоці спеціальних маркерів. Залежно від завдання, в якості маркера можуть виступати як спеціально сформовані зображення, так і особи людей. Після знаходження маркера в відеопотоці і обчисленні його місця розташування, з'являється можливість побудови матриці проєкції і позиціонування віртуальних моделей. За допомогою них можна накласти віртуальний об'єкт на відеопотік таким чином, що буде досягнутий ефект присутності. Основна складність якраз і полягає в тому, щоб знайти маркер, визначити його місце розташування в кадрі і спроектувати відповідним чином віртуальну модель.

За останнє десятиліття була створена велика теоретична база в сфері обробки зображень та пошуку на ньому різних об'єктів. Перш за все, це стосується методів контурного аналізу, порівняння з шаблоном, виявлення особливостей і генетичних алгоритмів. З точки зору побудови доповненої реальності найчастіше використовуються останні два підходи. Дамо короткий опис по кожному з них.

Генетичні алгоритми – це евристичні алгоритми пошуку, які використовуються для вирішення завдань оптимізації та моделювання шляхом випадкового підбору, комбінування і варіації шуканих параметрів з використанням механізмів, що нагадують біологічну еволюцію [2].

У комп'ютерному зорі вони використовуються для пошуку об'єкта деякого заданого класу на статичному зображенні або відео потоці. Спочатку необхідно провести навчання алгоритму за допомогою двох різних наборів зображень:

- «хороші» – містять потрібний об'єкт;
- «погані» – помилкові зображення без шуканого об'єкта.

При цьому для навчання використовується велика кількість зображень, і чим їх більше – тим краще буде працювати сам алгоритм. Для кожної картинки проводиться виділення різних ключових особливостей: кордону, лінії, центральних елементів (рисунок 1.9).

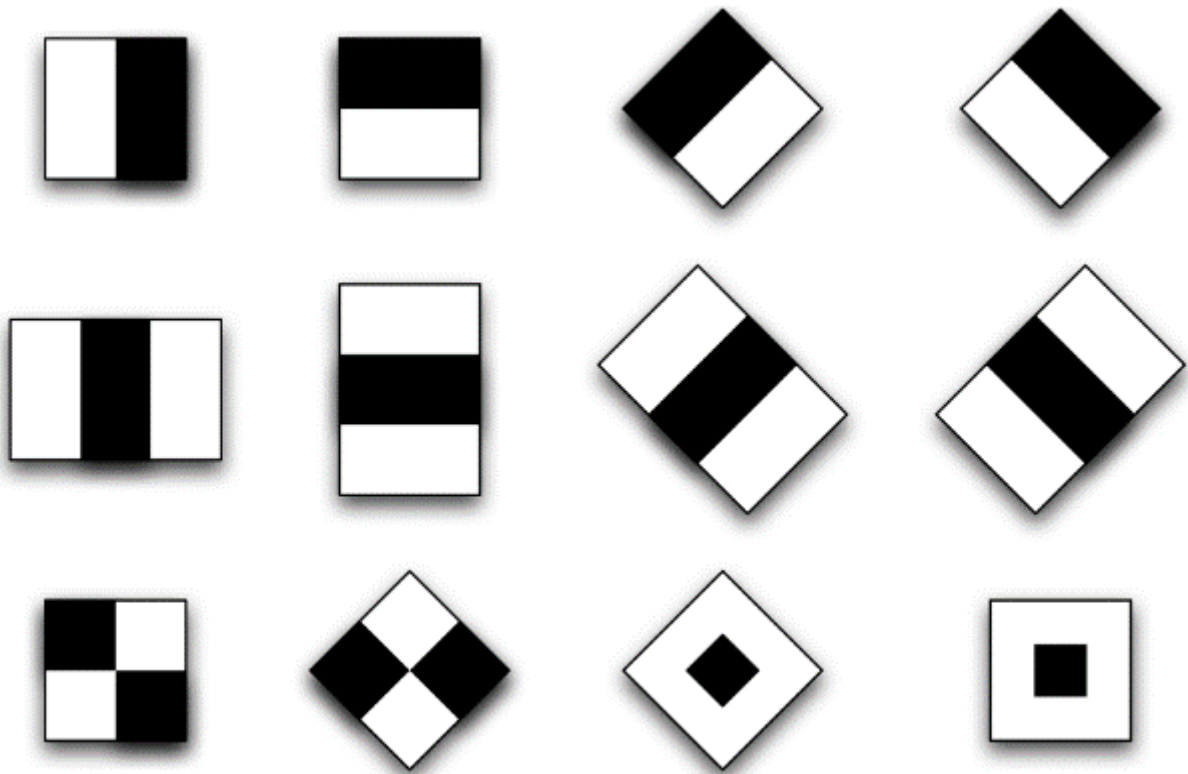


Рисунок 1.9 – Примітиви Хаара, які використовуються в алгоритмі

За ним проводиться побудова статистичної моделі, яка потім і використовується для пошуку об'єкта на зображенні.

Прикладом використання даного підходу може служити алгоритм розпізнавання осіб і очей у відеопотоці. Поступово навчаючи алгоритм, можна домогтися високих результатів знаходження заданого класу об'єктів. Однак необхідність навчання якраз і робить використання генетичних алгоритмів досить проблематичним. Для їх хорошої роботи потрібно значне число різних зображень (як «хороших», так і «поганих»), і час побудови класифікатора для кожного об'єкта може займати тривалий час.

Концепція feature detection в комп'ютерному зорі відноситься до методів, які націлені на обчислення абстракцій зображення і виділення на ньому ключових особливостей [2]. Дані особливості можуть бути як у вигляді ізольованих точок, так і кривих або пов'язаних областей. Не існує строгого визначення того, що таке ключова особливість зображення. Кожен алгоритм розуміє під цим своє (кути, межі, області тощо).

Найчастіше для пошуку маркерів використовуються алгоритми, які виконують пошук і порівняння зображень по ключових точках. Ключова точка – це певний ділянку картини, який є характерною для заданого зображення. Що саме приймається за дану точку – безпосередньо залежить від використовуваного алгоритму (рисунок 1.10).

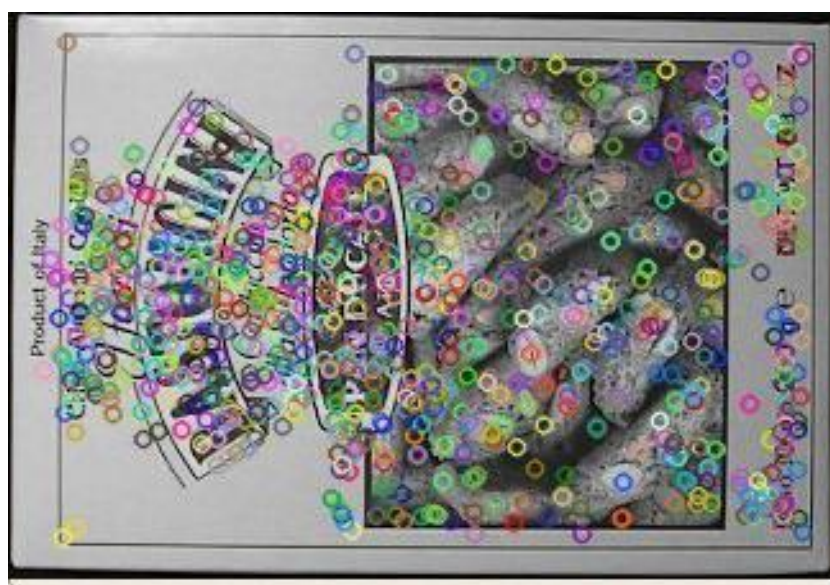


Рисунок 1.10 – Приклад ключових точок на зображенні

Для їх знаходження і подальшого порівняння використовуються три складові:

- детектор (англ. Feature detector) – здійснює пошук ключових точок на зображенні;
- дескриптор (англ. Descriptor extractor) – виробляє опис знайдених ключових точок, оцінюючи їх позиції через опис навколишніх областей;
- матчер (англ. Matcher) – здійснює побудову відповідностей між двома наборами точок.

Спочатку за допомогою детектора проводиться пошук ключових точок шаблонного (шуканого) зображення. Отримані точки потім описуються за допомогою дескриптора. Дана інформація зберігається в окремий файл (або базу даних), щоб не виконувати цей процес повторно. Ця база може зберігатися як локально на носії, так і на віддаленому сервері. При обробці відеопотоку з метою пошуку заданого шаблону описаний процес виконується для кожного кадру (за винятком збереження даних). Для встановлення відповідності між ключовими точками і дескрипторами застосовується матчер.

Різні алгоритми працюють з різною швидкістю та ефективністю. В умовах застосування їх для побудови доповненої реальності необхідно використовувати тільки ті, які показують високу швидкість роботи при досить хорошій якості відстеження позицій ключових точок. В іншому випадку ми можемо отримати помітні відставання у обробці відеоданих.

Для підвищення швидкості роботи алгоритмів feature points detection застосовуються різні способи фільтрації точок, щоб мінімізувати їх число і відсіяти зовсім погані поєднання. Таким чином, можна домогтися не тільки підвищення швидкості роботи алгоритмів, але і якості трекінгу маркерів.

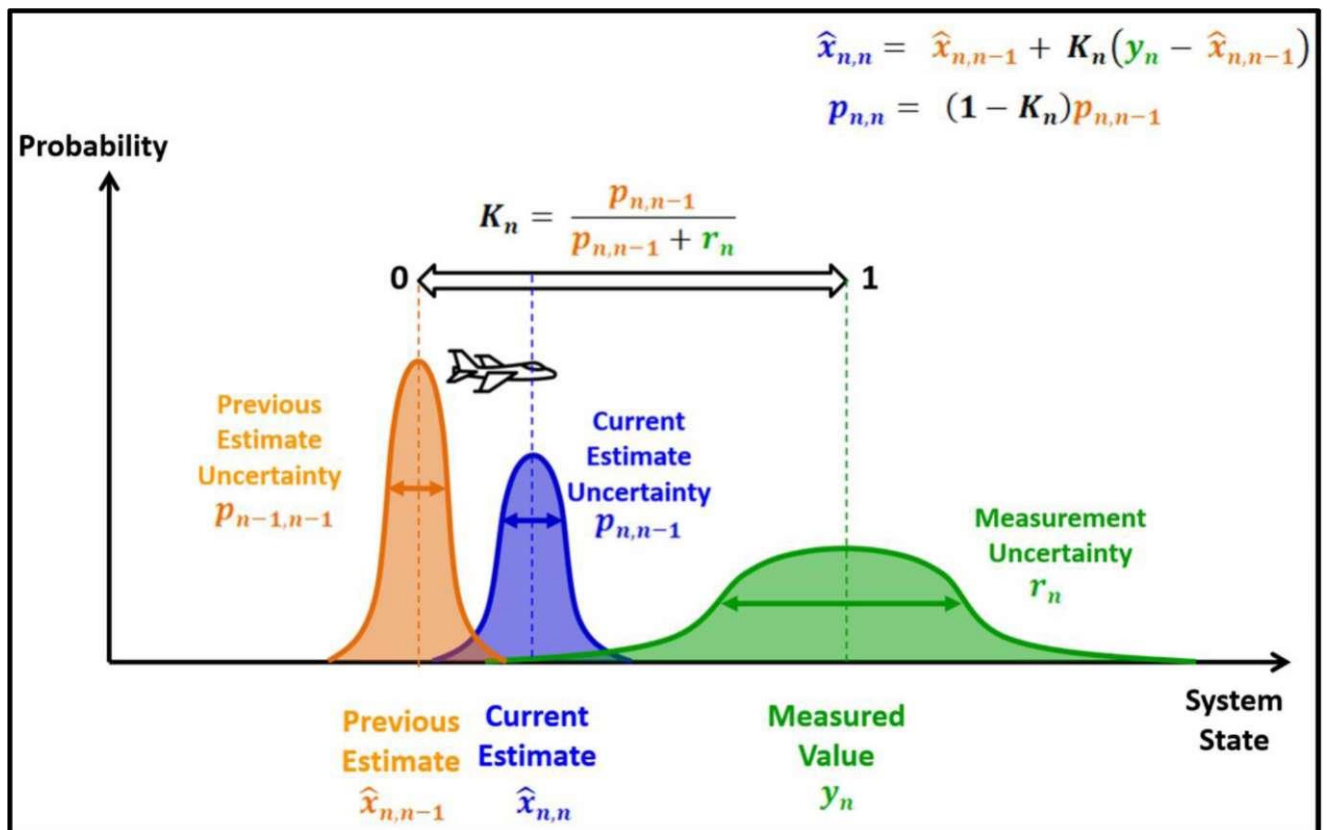
1.3 Огляд принципу роботи доповненої реальності на мобільній платформі

Для того, щоб краще зрозуміти на що здатна технологія доповненої реальності, треба зрозуміти як вона працює та які у неї є обмеження. Для вирішення задач доповненої реальності, Apple створила технологію ARKit [5]. Саме тому розглянемо її. Технічно ARKit є системою візуальної одометрії (VIO), з деяким 2D відстеженням площини. VIO означає, що програмне забезпечення відстежує вашу позицію в просторі у режимі реального часу, тобто ваше розташування в просторі перераховується між кожним оновленням кадру на дисплеї, приблизно 30 або більше разів на секунду. Ці розрахунки проводяться двічі, паралельно. Ваша поза відстежується через візуальну систему (камеру), зіставляючи точку в реальному світі з пікселем на датчику камери кожного кадру.

Також ваша поза відстежується інерційною системою (акселерометр і гіроскоп - разом називають інерційною одиницею вимірювання або IMU). Вихідні дані обох цих систем потім об'єднуються через фільтр Калмана [6], який визначає, яка з двох систем забезпечує найкращу оцінку вашої "реальної" позиції (називається Ground Truth) і передає це оновлення через ARKit SDK. Так само, як ваш одометр у вашому автомобілі відстежує відстань, на яку подорожував автомобіль, система VIO відстежує відстань, на яку iPhone перемістився у просторі. Великою перевагою, яку приносить VIO, є те, що показання IMU зчитуються приблизно 1000 разів на секунду і базуються на прискоренні (русі користувача).

Dead Reckoning використовується для вимірювання руху пристрою між показаннями IMU. Dead Reckoning - це припущення, на кшталт вимірювання дистанції за допомогою людського ока, але помилка є набагато меншою. Таким чином, візуальні та інерційні системи відстеження базуються на абсолютно різних системах вимірювань без взаємної залежності. Це означає, що фотокамера може бути закрита або може переглядати сцену з декількома оптичними

функціями (наприклад, білою стіною), а інерційна система в цей час буде брати «навантаження» на себе. В альтернативному сценарії пристрій може залишатися на місці і візуальна система буде отримувати більш стабільне розташування в просторі, ніж інерційна система. Фільтр Калмана постійно вибирає найкращу позицію з двох, тому результатом є стабільне відстеження. Приклад наведено на рисунку 1.11.



Рисунк 1.11 - Приклад роботи фільтру Калмана

В цілому, ARKit можна розділити на три компоненти. Перший - це відстеження, коли ARKit використовує трекінг реального світу, тобто технологія здатна знаходити положення користувача щодо навколишнього середовища. Апаратне забезпечення Apple використовує алгоритми комп'ютерного зору «Core Motion», що допомагає пристрою оцінювати відносну позицію за допомогою камери. У поєднанні з сенсорами та датчиками, наприклад

акселерометром, гірометром і магнітометром, «Core Motion» здатний виконувати трекінг реального світу. Другий - розуміння сцени. Це компонент ARKit, який допомагає зрозуміти компоненти навколишнього середовища, такі як горизонтальні поверхні та навколишнє освітлення. Третій - рендерінг. Це компонент у ARKit, який дозволяє відтворювати різні типи 2D/3D-контенту з відповідними текстурами. На рисунку 1.12 зображені напрями відстеження акселерометру та гіроскопу.

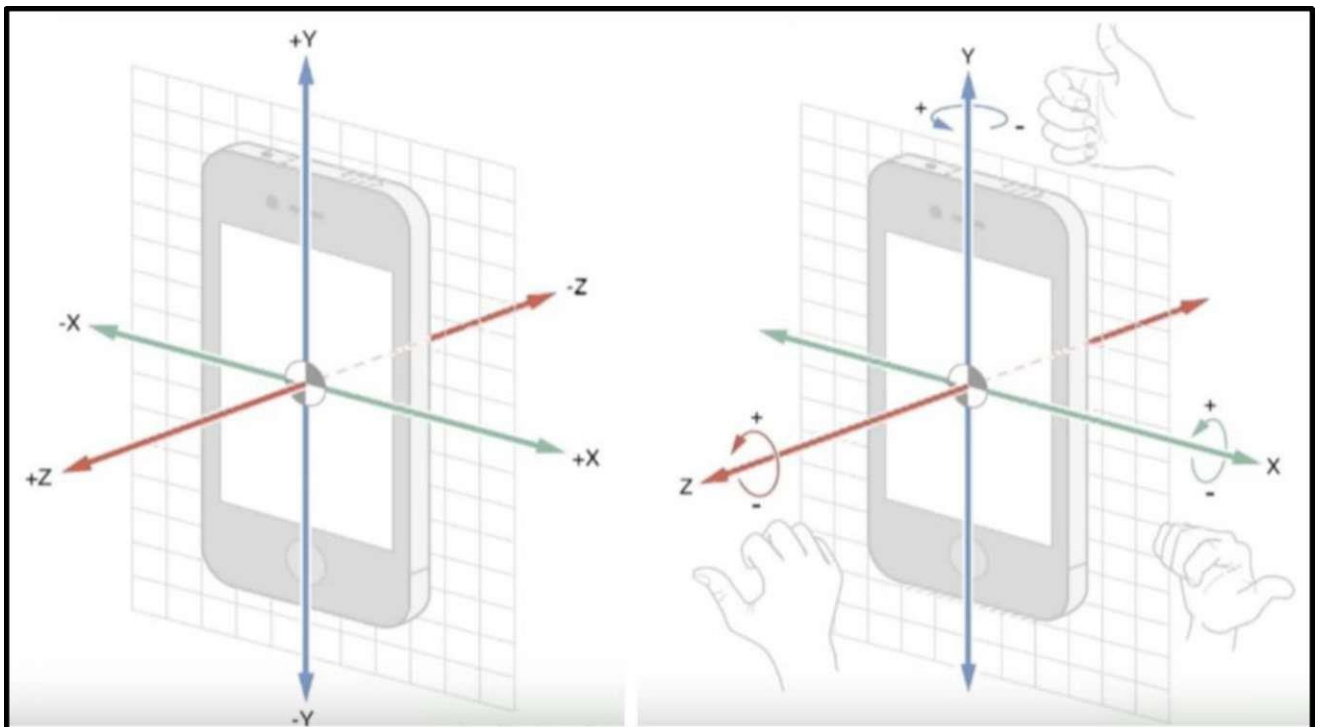


Рисунок 1.12 - Принцип роботи акселерометру та гіроскопу

Давайте розберемо ці компоненти, щоб зрозуміти, як вони працюють. Відстеження світу - це здатність ARKit зрозуміти середовище та інтерпретувати відносне розташування користувача. ARKit використовує спеціальний код для інтерпретації даних камери та даних з датчиків пристрою, таких як акселерометр, гіроскоп і магнітометр. На цьому етапі системна бібліотека «Core Motion» інтерпритує дані, отримані з сенсорів та датчиків. Дані з акселерометру дають можливість інтерпретації руху девайсу у тривимірному просторі за трьома

осями, дані гіроскопу інтерпретуються для розуміння повороту девайсу щодо одної з трьох осей. Магнітометр, в свою чергу, надає можливість зрозуміти орієнтацію девайсу щодо універсального північного полюсу.

Наступний компонент - розуміння сцени. Цей компонент складається з декількох підсистем: виявлення площини, «Hit-testing» та оцінка освітлення. ARKit розпізнає площину на основі контрастної піксельної інформації в даній сцені. Технологія намагається інтерпретувати горизонтальну поверхню. Деякі інтелектуальні функції включають злиття площин, що означає, що вона може зрозуміти кілька поверхонь, але може об'єднати ці поверхні, якщо вони розміщені на одному і тому ж об'єкті. Приклад наведено на рисунку 1.13.

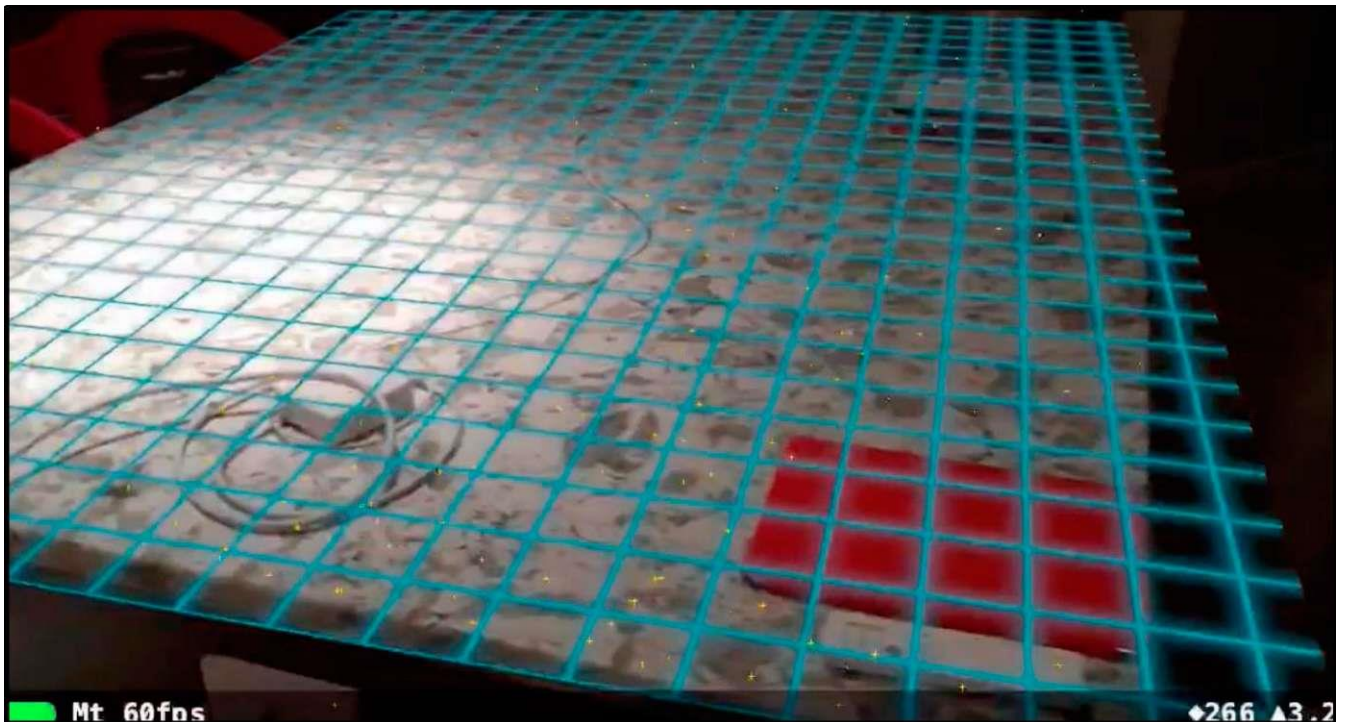


Рисунок 1.13 - Візуалізація роботи підсистеми виявлення площини

«Hit-testing» - це спосіб для ARKit зрозуміти середовище і підготуватися шляхом розуміння деяких опорних точок в просторі (див.рис. 1.14). Після цього він використовує ці точки прив'язки для позначення на них цифрових об'єктів. ARKit зазвичай рендерить опорні точки як жовті крапки. Коротко, «Hit-testing»

є камерою пристрою, посилюючи промінь для виявлення фізичного світу і сортує кожен точку за відстанню у цьому світі. Це також спосіб для ARKit зрозуміти глибину середовища.

Оцінка освітлення - це доволі складна та багатокомпонентна система. Для того, щоб контент був реалістичний, він повинен відображатися відносно навколишнього освітлення та, окрім цього, створювати тіні та інші ознаки реального освітлення. Для цього камера пристрою фіксує експозицію оточення і надсилає його до ARKit, щоб налаштувати освітлення цифрових об'єктів, які технологія відображає на екрані пристрою.

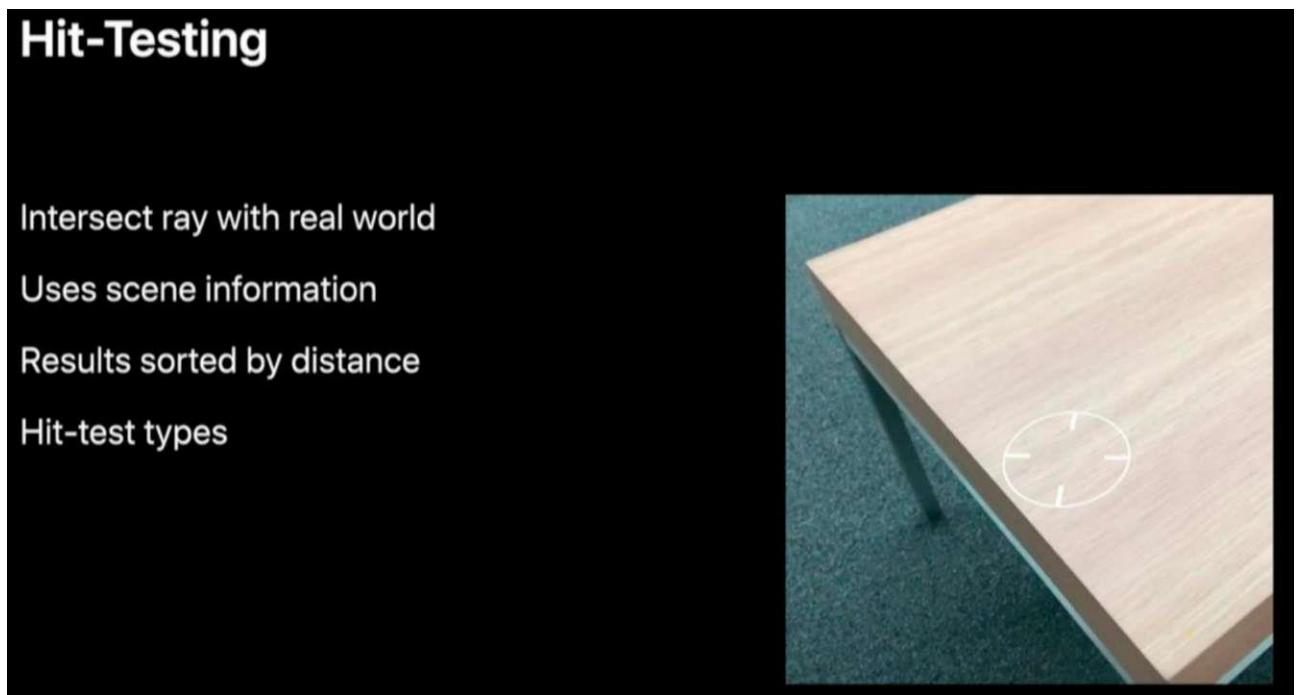


Рисунок 1.14 - Принцип роботи «Hit-testing»

Рендерінг - це останній значний компонент в технології доповненої реальності. ARKit використовує існуючі структури Apple для створення та надання активів. Розробники мобільних додатків можуть створювати контент, який ефективно використовує апаратне забезпечення Apple. Для рендерінгу

можуть використовуватися декілька різних фреймворків: SpriteKit, SceneKit та Metal. В залежності від задачі, розробник може вибрати один з них.

SpriteKit є фреймворком Apple для створення 2D візуального контенту. У поєднанні з ARKit це призводить до створення 2D компонентів, які дотримуються білбордингу. Білбординг - це, коли компоненти завжди орієнтовані на камеру. Можна створити глибину відображення з SpriteKit просто регулюючи масштаб.

SceneKit схожий на SpriteKit, але призначений для 3D-контенту і надає параметри для швидкого складання 3D-сцени, на яких розміщені складні анімації.

Metal - це спеціальна бібліотека текстуровання, яку Apple пропонує розробникам. Те, що пропонує Metal - це покращений рендерінг для пристроїв Apple порівняно з OpenGL. За допомогою Metal можна створити текстури будь-якої складності, але він більш складний, ніж SpriteKit чи SceneKit. Більш того, SpriteKit та SceneKit побудовані на основі Metal. Саме тому він використовується для розробки складних мобільних додатків чи мобільних ігор.

Сучасні пристрої володіють одними з найкращих сенсорів фотокамери та відстеження руху (акселерометр, гіроскоп) в галузі, та найпотужнішими мобільними процесорами розробленими за допомогою 7 нанометрового технологічного процесу [7]. Це дозволяє розробляти мобільні додатки та програмні системи, використовуючи складні алгоритми, котрі в режимі реального часу можуть відстежувати навколишній світ та додавати AR об'єкти до нього. Це відкриває багато можливостей для застосування технології в сфері освіти, бо вона є однією з найбільш перспективних галузей доповненої реальності та може вирішувати складні завдання мотивації навчання та допомагати мільйонам студентів на всій планеті.

1.4 Порядок розміщення об'єктів доповненої реальності у навколишньому середовищі

Рішення задачі позиціонування об'єктів доповненої реальності складається з послідовних етапів.

Етап 1. Завантаження ознак цільових об'єктів. Є один або декілька маркерів (об'ємне тіло, картинка, текст тощо), котрі необхідно знайти у навколишньому середовищі. Для кожного з цих маркерів формується набір ознак, за якими можна виявити їх на вхідному зображенні. Цими ознаками можуть бути розміщення граней, точок, колір абощо.

Етап 2. Попередня обробка зображення з камери. З певною частотою з відеопотоку камери витягується фрейм для обробки. Оскільки є зовнішні фактори, що впливають на чіткість картинки, попередня обробка включає в себе адаптацію яркості, контрасту та насиченості кольорів отриманого фрейму. В результаті з відеопотоку з певною частотою отримаємо по зображенню, підготовленому для подальших дій.

Етап 3. Виявлення ознак маркерів на зображенні. Після попередньої обробки, фрейм передається алгоритму розпізнавання образів для співставлення ознак кожного маркеру на картинці. Якщо маркер знайдено – фіксується його місцеположення на фреймі.

Етап 4. Формування координатних осей для доповненої реальності. На основі позицій знайдених на фреймі маркерів формується система координат. При її формуванні враховується місцеположення кожного маркеру та кут нахилу камери до нього.

Етап 5. Відображення об'єктів доповненої реальності (рендеринг). Створена система координат слугує основою для розміщення віртуальних об'єктів поверх зображення навколишнього середовища, отриманого з камери. Їх позиції відносно маркерів визначаються розробником, а кут нахилу до камери залежить від куту повороту координатних осей.

Завдання дослідження полягає у аналізі існуючих рішень для формування доповненої реальності та інтеграції алгоритму одночасного відстеження місцеположення. Це надасть змогу використовувати доповнену реальність без застосування попередньо визначених маркерів.

1.5 Популярні бібліотеки для формування доповненої реальності

Vuforia SDK – це найпопулярніша в світі платформа доповненої реальності [4], що включає в себе комплект засобів розробки (SDK) для створення додатків доповненої реальності. Vuforia використовує алгоритми комп'ютерного зору для виявлення та відстеження плоских зображень (міток), а також простих тривимірних об'єктів в режимі реального часу.

Vuforia має свій інтерфейс API, що дозволяє писати код на мовах C++, Java, Objective-C і .Net. Vuforia SDK можна встановити в якості розширення для таких IDE як Android Studio, XCode, Visual Studio і вести розробку програмного забезпечення під конкретну мобільну платформу (наприклад, Windows Phone, Android або iOS), або використовувати розширення Vuforia для ігрового движка Unity і розробляти кросплатформенні мобільні додатки.

Розглянемо методи взаємодії Vuforia Engine (двигуна Vuforia) з реальним світом. Під інтерфейсами доповненої реальності розуміють елемент(и) через які відбувається взаємодія реального (фізичного) світу з Vuforia Engine і користувачем, які переносять рівень інтерактивності людина-машина на новий рівень [6]. Vuforia має декілька методів розпізнавання (вводу даних) візуальної інформації, яка зчитується з камери пристрою, та оброблюється Vuforia Engine.

– Маркери (Image Targets) — це реальні плоскі зображення нанесені на будь-яку сприятливу, пласку поверхню. Маркер не потребує спеціальних чорних та білих зон, для того щоб він міг відстежуватись у Vuforia Engine. Маркери зручно використовувати де завгодно, але у найближчий час їх скоріше за все будуть використовувати для підказок, інформаційних довідок, для створення інтерактивних книжок та посібників, візуалізувати будь-які поняття і тд.

– Мітка VuMark – більш розвинута версія звичайних маркерів. Мітка VuMark має контур, кордон, вільну зону, елементи коду і може включати в себе логотип (рисунок 1.4). Програмний модуль Vuforia шукає контур мітки на

зображенні, що отримується з камери пристрою, і, в разі виявлення, зчитує елементи коду. Елементи коду діляться на «світлі» і «темні» і в сукупності являють собою двійковий код, при розшифровці якого можна отримати унікальний ідентифікатор виробу, посилання на інтернет-сторінку з довідковою інформацією про нього та інше.

– Мульти-маркери (Multi-Targets) — аналогічний до звичайних маркерів, але з їх використанням відбувається відстеження більше одного маркера на одному реальному об'єкті, наприклад на кубіку. Це особливо корисно для надвеликих фізичних об'єктів на які накладається доповнена реальність, які потрібно охопити з усіх сторін, наприклад: будинки, пам'ятники, музеї і т.д.

– Циліндричні маркери (Cylinder Targets) – реальні плоскі зображення нанесені на циліндричні поверхні. Вони дозволяють відстежувати маркери у формі скручених конусів та циліндрів. Такі маркери використовуються на чашках, пляшках, широких стопах та ін.

– Розпізнавання фізичних об'єктів у 3D (Object Scanner) – розпізнає 3D об'єкти та робить з них спрощені геометричні скелети-примітиви (об'єкти-мітки). Це дозволяє у майбутньому ідентифікувати конкретний фізичний об'єкт у фізичному просторі. Для створення нового маркера-об'єкта спочатку сканують існуючий об'єкт Object Scanner(ом), після чого автоматично генерується Object Data File. Сгенерований файл завантажується у Vuforia Target Manager (можливо завантажити максимум до 20 Object Data File).

Положення віртуальних предметів, які доповнюють реальний простір, має бути пов'язане з певним нерухомим реальним об'єктом, відносно якого вони будуть розміщуватися. Цим об'єктом є так звана мітка.

Програмний модуль Vuforia аналізує зображення, що отримується з камери пристрою, і намагається знайти мітку. Як тільки мітка виявлена, проводиться оцінка її нахилу, що дозволяє визначити положення користувача, який тримає пристрій, щодо цієї позначки. Знаючи положення користувача щодо мітки і положення віртуальних предметів в системі координат, пов'язаної з міткою,

можна вирахувати місцеположення і кути нахилу віртуальних предметів щодо користувача і вивести їх на екран мобільного пристрою.



Рисунок 1.15 – Маркери VuMark

Є безкоштовна версія для розробників з постійно видим водняним знаком на екрані, та платна для готових додатків – 99 доларів у місяць за один мобільний додаток.

Бібліотека EasyAR – безкоштовна і проста у використанні альтернатива Vuforia від китайських розробників [4]. Підтримувані платформи: Android, iOS, UWP, Windows, OS X і Unity.

Остання версія EasyAR включає наступні функції:

- розпізнавання зображень;
- розпізнавання 3D-об'єктів;
- сприйняття навколишнього середовища з одночасним орієнтуванням на місцевості;
- хмарне розпізнавання;
- робота на смарт-окулярах;
- хмарне розгортання додатків.

Бібліотека повністю безкоштовна. Щоб почати роботу з EasyAR, потрібно лише зареєструвати обліковий запис і згенерувати ключ плагіна для конкретного додатку, що розробляється. EasyAR легко інтегрується. Документація детально описана, а приклади інтуїтивно зрозумілі.

На відміну від інших інструментів, EasyAR не має менеджера міток, розпізнає об'єкти виходячи з 3D-моделі, яка повинна лежати всередині проекту, і генерує цільові зображення під час виконання програми. Використовуючи EasyAR потрібно писати більше коду, і трохи інакше формувати проект в IDE. Всією поведінкою керує розробник – він повинен визначити власну поведінку для події розпізнання / втрати об'єкта.

Бібліотека розпізнає об'єкти за умови гарного освітлення, та все ж відсоток успішних виявлень міток нижчий за Vuforia.

Є платна версія – одноразовий платіж у 500 доларів за мобільний додаток. Платна версія включає у себе такі можливості, як:

- відслідковування 3D об'єктів;
- одночасне відслідковування різних типів маркерів;
- запис відео з екрану.

Бібліотека Wikitude SDK є основним продуктом австрійської компанії. Вперше запущена у жовтні 2008 р. SDK включає в себе розпізнавання та відстеження зображень, рендеринг 3D-моделей, відео-накладання, AR-локацію та технологію ОЛК (одночасна локалізація та картографування) [5], яка дозволяє розпізнавати та відстежувати об'єкти, а також здійснювати миттєве відстеження положення пристрою без використання маркеру. Крос-платформний SDK доступний для операційних систем Android і iOS, а також оптимізований для кількох пристроїв із чудовими окулярами [6].

З особливостей можна виділити те, що на відміну від Vuforia 3D-об'єкти в базу завантажуються у вигляді відеороликів. Цільовий маркер генерується з відео. Для того, щоб маркер успішно був згенерований, необхідно відзняти його на однотонному тлі.

Нещодавно Wikitude випустила повністю нове потужне ОЛК-рішення для додатків доповненої реальності Wikitude SDK 7.

Wikitude SDK має в арсеналі такі функції: відмінне розпізнавання і відстеження зображень, технологію тривимірного стеження на базі ОЛК, GEO Data (покращена робота з даними з географічною прив'язкою), хмарне розпізнавання (дозволяє зберігати бази даних зображень в хмарному сховищі).

У прикладах, наданих компанією Wikitude, можна побачити роботу наступних функцій:

- виконання нативного коду і JavaScript;
- приклади розпізнавання зображень;
- функціонуючі 3D-моделі;
- представлення точок інтересу і механізму завантаження;
- широкий спектр демо-проектів;
- простота запуску доповненої реальності з будь-якого URL.

Для доповненої реальності на основі місцезнаходження, позиція об'єктів на екрані мобільного пристрою розраховується зі врахуванням позиції користувача (за допомогою GPS або Wi-Fi), напрямку, у якому користувач дивиться на об'єкт (за допомогою компаса) та акселерометром.

У 2017 році Wikitude запустила технологію ОЛК. Instant Tracking, перша функція, що використовує ОЛК, дозволяє розробникам легко створювати середовища та відображати вміст додаткової реальності без потреби в маркерах. Проте при цьому, вона спирається на декілька згенерованих зображень-міток. Розпізнавання об'єктів – це останнє доповнення на основі ОЛК, із запуском SDK 7. Ідея відстеження об'єктів дуже схожа на відстеження зображень, але замість розпізнавання зображень у навколишньому середовищі об'єкт Tracker може працювати з тривимірними структурами та предметами. Є безкоштовна версія бібліотеки для розробників, з водяними знаками на екрані. А вартість повної версії для випуску додатків доволі висока – від 2490 до 4490 євро на рік.

1.6 Постановка мети і завдань дослідження

Недоліком перерахованих вище рішень є той факт, що для розміщення об'єктів доповненої реальності використовуються мітки, задані заздалегідь або згенеровані з відеопотоку під час виконання програми. Це накладає певні обмеження на можливості побудови доповненої реальності у одноманітному середовищі. Тому існує необхідність у пошуку альтернативного підходу до орієнтації пристрою у навколишньому середовищі.

Саме тому у цій роботі було вирішено розробити модель формування доповненої на основі механізмів візуальної одометрії, а також створити прототип мобільного застосунку для формування доповненої реальності з використанням розробленої моделі і виконати дослідження з порівнянням результатів. Це має надати змогу розширити можливості доповненої реальності та зробити роботу з нею більш стабільною.

Метою досліджень є підвищення ефективності методів формування доповненої реальності шляхом використання механізмів візуальної одометрії за рахунок розроблення мобільного застосунку для формування доповненої реальності.

Для досягнення мети необхідно виконати наступні завдання:

- дослідити сучасні методи та підходи формування доповненої реальності;
- розробити модель створення об'єктів доповненої реальності на основі механізмів візуальної одометрії;
- розробити прототип мобільного застосунку для формування об'єктів доповненої реальності з використанням розробленої моделі;
- провести тестування прототипу мобільного застосунку для формування доповненої реальності.

1.7 Висновки до розділу 1

У цьому розділі описано поняття та процес побудови доповненої реальності. Описано поняття «комп'ютерний зір» та «feature detection». Розглянуто різні існуючі бібліотеки для побудови доповненої реальності. Як результат, обрано бібліотеку Wikitude для подальшої інтеграції через вичерпність її документації та стабільну роботу. Сформульовано мету й завдання дипломного проекту.

2 МОДЕЛЬ СТВОРЕННЯ ОБ'ЄКТІВ ДОПОВНЕНОЇ РЕАЛЬНОСТІ НА ОСНОВІ ВІЗУАЛЬНОЇ ОДОМЕТРІЇ

2.1 Методи візуальної одометрії

При розробці систем управління пристроями часто постає необхідність знати карту навколишнього простору і положення, наприклад, щоб уникати зіткнення з перешкодами.

Одночасна локалізація та картографування (ОЛК) (англ. simultaneous localization and mapping) є алгоритмічною обчислювальною задачею побудови і оновлення мапи невідомого оточення з одночасним відстеженням місцеположення у процесі руху [8].

Алгоритми ОЛК обмежуються наявними ресурсами, таким чином не можуть бути абсолютно досконалыми, бо досягають оперативної доступності. Опубліковані методи і підходи реалізовані в безпілотних автомобілях, безпілотних літаючих засобах (рисунок 2.1), автономних підводних апаратах, планетоходах, згодом виникли в побутових роботах і навіть всередині людського тіла.

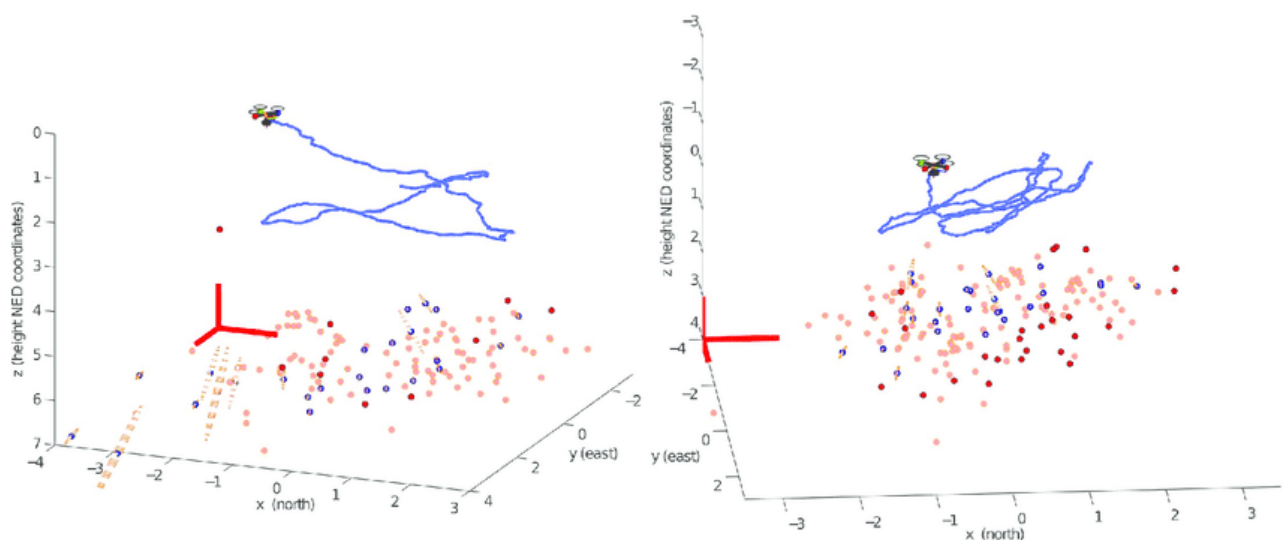


Рисунок 2.1 – Використання ОЛК на літаючому засобі

ОЛК – це не якийсь певний алгоритм або набір ПЗ – це концепція, загальна методологія для вирішення двох завдань:

- побудова карти дослідженого простору;
- побудова траєкторії руху робота на мапі.

Треба відзначити, що повністю дана задача як і раніше не вирішена і до сих пір ведуться дослідження.

Перший принциповий момент – ОЛК не припускає будь-яких знань про середовище – ні міток на місцевості / ні попередньої карти немає – всі рішення будуються тільки на результатах вимірювань датчиків (зазвичай це rgb- і далекомірні камери, але, буває, і додаткове обладнання гіроскопа, gps-датчика і т.д.).

Другий не менш важливий момент – середовище вважається статичним. Це означає, що якщо робот проїхав повз фікуса, ніхто за його спиною цей же самий фікус рухати не стане. В кадрі жодного стороннього руху немає – ніхто в кадрі не ходить. Освітлення радикально не змінюється – тобто тіні по стінах не стрибають. Такі умови складно гарантувати в природному середовищі – шелестить листя або кішка запросто може дезорієнтувати робота в просторі.

Постановка задачі виглядає наступним чином:

Дана послідовність даних спостереження сенсору o_t за дискретні проміжки часу t , задачею ОЛК є розрахувати і визначити розташування агента x_t і мапу оточення m_t . Всі величини зазвичай ймовірнісні, тому необхідно обчислити:

$$P(x_t | o_{1:t}, m_t).$$

Застосування формули Баєса дає основу для послідовного оновлення апостеріорного розташування, при даній мапі і функції переходу $P(x_t, x_{t-1})$

$$P(x_t | o_{1:t}, m_t) = \sum_{m_{t-1}} P(o_t | x_t, m_t) L_t,$$

$$L_t = \sum_{x_{t-1}} \frac{P(x_t|x_{t-1})P(x_{t-1}|m_t, o_{1:t-1})}{Z}. \quad (2.1)$$

Аналогічно, мапа може оновлюватися послідовно наступним шляхом:

$$P(m_t|x_{t-1}, o_{1:t}) = \sum_{x_t} \sum_{m_t} P(m_t|x_t, m_{t-1}, o_t) K_t, \\ K_t = P(m_{t-1}, x_t|o_{1:t-1}, m_{t-1}). \quad (2.2)$$

Як для більшості задач наближення, рішення можна знайти при наближенні двох змінних, до локального оптимального рішення, шляхом почергового оновлення двох рівнянь у формі EM-алгоритму.

У загальному випадку ОЛК можна описати як послідовність кроків:

- сканування навколишнього простору;
- визначення зміщення на основі порівняння поточного кадру з попереднім;
- виділення на поточному кадрі особливостей-міток;
- зіставлення міток поточного кадру з мітками отриманими за всю історію спостережень;
- оновлення на основі цієї інформації положення пристрою за всю історію спостережень;
- перевірка на петлі – чи не проходимо ми повторно по одній і тій же місцевості;
- вирівнювання загальної карти світу (відштовхуючись від положення міток і пристрою за всю історію спостережень).

До статистичних методів, що використовуються для задач апроксимації наведених вище, належать: фільтр Калмана, фільтр часток (який є методом Монте-Карло) і узгоджене сканування діапазонних даних. Вони дозволяють визначити оцінку функції апостеріорної ймовірності для позиції робота і параметрів мапи. Техніки оцінювання приналежності до множини в основному

засновуються на поширенні сталого інтервалу. Вони забезпечують множину, яка містить позицію робота і множину апроксимації мапи. «Bundle adjustment» є наступною популярною технікою у ОЛК, який використовує дані зображень поєднані з оцінками розташування робота і орієнтирів на місцевості, тим самим підвищуючи точність мапи. Вона використовується в комерційних ОЛК системах, таких як проект Tango компанії Google.

Нові алгоритми ОЛК досі потребують активного дослідження і пошуку, і часто обумовлені різними вимогами і припущеннями щодо типів карт, дачив і моделей. Більшість ОЛК систем можна розглядати як комбінації виборів кожного з цих аспектів.

Побудова топологічних мап це метод представлення довколишнього світу, який узагальнює структуру (тобто, топологію) оточення замість створення геометрично точної мапи. Топологічні методи ОЛК використовуються для підвищення загальної узгодженості метричних алгоритмів ОЛК.

На противагу тому, для топологічного представлення світу існують мапи у вигляді поверхонь, які використовують масиви оцифрованих елементів (зазвичай квадратних, або гексагональних клітин), і роблять припущення яка клітина зайнята в конкретний момент. Зазвичай клітини вважаються статистично незалежними, аби спростити обчислення.

Різноманітні алгоритми ОЛК реалізовані у відкритих бібліотеках Robot Operating System, які часто використовуються з Point Cloud Library для побудови 3D карт або візуальних ефектів у OpenCV.

ОЛК складається з декількох частин: виділення мітки, об'єднання даних, стан оцінювання ситуації, оновлення стану та орієнтирів [9]. Існує багато способів виконати кожен з цих кроків. Це означає, що деякі кроки цього процесу можуть бути виконані різними алгоритмами. Ідея полягає в тому, що можна використовувати певну реалізацію і розширити її, використовуючи власний новий підхід до цих алгоритмів. Розглянемо на прикладі мобільного робота в приміщенні. ОЛК застосовується як для 2D, так і для 3D руху. Ми будемо розглядати тільки 2D рух.

В даний час існує декілька основних підходів до вирішення цих завдань: розширений фільтр Калмана (extended Kalman filter, EKF), FastSLAM, DP-SLAM [10].

В останні десятиліття XX-го століття практично єдиним методом вирішення подібних завдань був розширений фільтр Калмана. Основним недоліком даного підходу є квадратична залежність складності алгоритму від кількості спостережуваних орієнтирів (практично не перевищує декількох сотень орієнтирів). В даний час існує і активно розвивається альтернативний підхід, названий FastSLAM, в основі якого лежить так званий фільтр частинок (Particle Filter, Monte Carlo methods). На відміну від EKF в FastSLAM одна велика карта розглядається як сукупність локальних підкарт, що дозволяє прибрати залежність орієнтирів один від одного і таким чином значно скоротити час перерахунку оцінки стану системи.

Проте у кожного з цих методів є свої обмеження і недоліки, що ще раз наголошує на необхідності вдосконалення алгоритмів картографії місцевості автономними мобільними роботами.

Структурний подання карти місцевості. Перше питання, на яке необхідно відповісти, вирішуючи завдання дослідження місцевості – що буде являти собою карта і в якому вигляді її зберігати. Адже перш ніж вивчати місцевість потрібно сформулювати уявлення про те, як найкращим чином описати навколишнє середовище, в якій буде функціонувати робот. Найчастіше середу умовно можна віднести до однієї з двох груп – з численними яскраво вираженими орієнтирами і з відсутністю таких.

Як приклади середовища першої категорії можна привести поле з окремими деревами. Іншим прикладом можуть послужити порожнини будівельних конструкцій, будь то коридори, кімнати або вентиляційна система, де як орієнтири можуть виступати кути. До другої групи можна віднести середу, де орієнтирів дуже мало або вони важко розрізняються, або занадто великі і не можуть бути використані в якості матеріальної точки.

Якщо в досліджуваному середовищі немає можливості знайти орієнтири, то її простіше всього представляти у вигляді сітки із заповненням осередків, зайнятих перешкодами.

Зберігати таку карту зручно у вигляді масиву, де елементи, що відображають стан перешкод, мають значення 1, а всі інші – 0. Таке уявлення карти застосовується, наприклад, в алгоритмі DP-SLAM.

У разі, коли в досліджуваній місцевості є численні орієнтири, карта являє собою масив оцінок їх розташування. Розмірність масиву $D_{im} = R * m$, де R - розмірність простору, а m – кількість орієнтирів. Для зберігання структури такої карти найпростіше використовувати картографічну базу даних, яка відображає стан орієнтирів, їх унікальні властивості і взаємозв'язки. Матриця оцінок стану динамічної системи на основі розширеного Фільтри Калмана використовує саме цей варіант подання карти.

2.1.1 Метод розширеного фільтру Калмана

Фільтр Калмана – рекурсивний фільтр, що оцінює вектор стану динамічної системи, використовуючи ряд неповних і зашумлених вимірювань [10]. Названий на честь Річарда Калмана. Фільтр Калмана призначений для рекурсивного дооцінювання вектора стану апріорно відомої динамічної системи, тобто для розрахунку поточного стану системи необхідно знати поточне вимірювання, а також попередній стан самого фільтра. Стан фільтра знаходиться в двох змінних:

X_k – оцінка вектора стану динамічної системи в момент часу k ;

P_k – коваріаційна матриця помилок (міра точності оцінювання вектора стану) в момент часу k .

Розширений фільтр Калмана (EKF) дуже схожий на простий фільтр Калмана, за тим винятком, що він може бути використаний в нелінійних процесах.

ЕКФ – це один з найбільш поширених методів рішення задачі ОЛК. Він дозволяє не тільки уточнювати оцінку становища робота на мапі, але і положення всіх виявлених орієнтирів. Зазвичай процес оцінки стану системи, в контексті ОЛК, розбивають на три етапи.

Етап 1. Оновлення оцінки стану системи на основі одометричних даних.

Етап 2. Оновлення оцінки стану системи на основі повторно виявлених орієнтирів.

Етап 3. Додавання нових орієнтирів в систему.

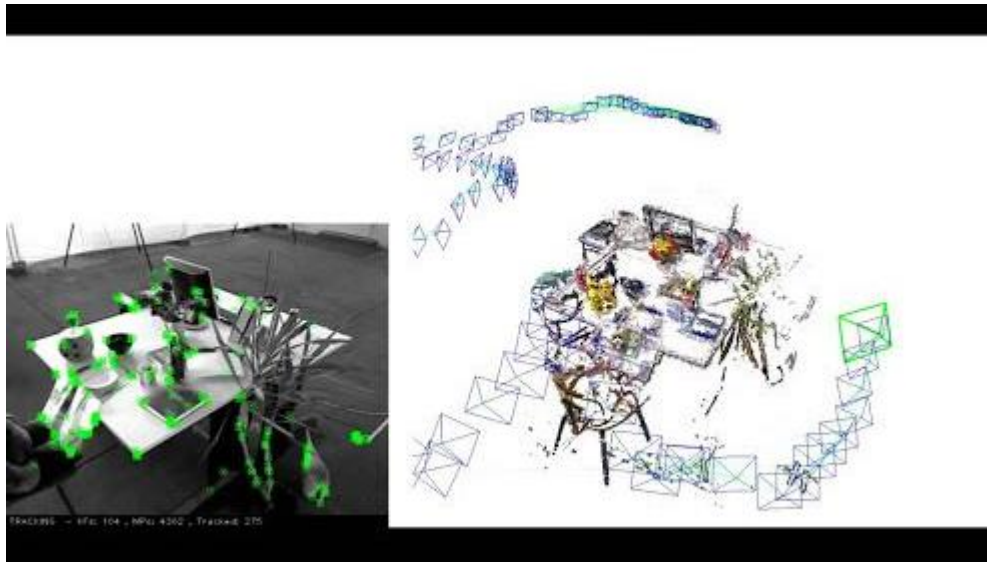


Рисунок 2.2 – Приклад результатів виконання ОЛК за ЕКФ

На кожному кроці у нас є набір раніше отриманих особливостей і дані, які щойно надійшли (дальномірні і RGB). На підставі нових і попередніх кадрів ми можемо визначити зміщення камери (використовуючи методи візуальної одометра) і передбачити нову позицію камери. З нового кадру ми можемо виділити місце розташування особливостей і обчислити положення камери щодо них [12]. На основі різниці між двома цими оцінками позицій камери оновлюються ймовірності / ваги для всіх особливостей і коригуються пози-траєкторія руху робота. Як структури для зберігання інформації про світ і

траєкторії руху робота – використовується розростається згодом матриця коваріації, що містить на кожному кроці вичерпну інформацію про наш поточному поданні світу.

Loop closure – виявлення петлею [12]. Відокремлено від завдань ОЛК стоїть питання як відслідковувати ситуації коли камера повертається туди, де вже побувала (в літературі це називається loop closure - замикання, петля). Одне з рішень – так звана корзина "слів". Кожному кадру ставиться у відповідність дескриптор, який вираховується на основі візуальних особливостей зображення. Для зберігання інформації про зображеннях, на основі даних для навчання, формується словник у вигляді дерева, що містить "слова" для подання дескрипторів і їх ваги (відображають наскільки часто вони зустрічалися в наборі зображень для навчання). Для формування "слова" проводиться пошук візуальних особливостей на основі даних для тренінгу і їх подальша угруповання (за допомогою методу k-середнього). Кожен наступний рівень дерева виходить шляхом повторення даної операції з дескриптором батьківського вузла. В результаті, при аналізі ново-даних, що надійшли проводиться швидке розкладання ідентифікатора поточного кадру в лінійну комбінацію вузлів дерева, де в ролі коефіцієнтів виступають їх ваги - ймовірності.

Як уже згадувалося вище, в даний час найбільш популярно уявлення проблеми ОЛК у вигляді графа, де вершини і ребра представляють позицію робота і розташування особливостей сцени. iSam – одна з відкритих реалізацій, побудована за цим принципом [12]. У iSam використовується двочастковий граф складається з вузлів-поз, що містять результати обчислень координат робота і вузлів-факторів, що містять результати оцінок візуальної одометра, що відбиває зрушення між двома послідовними позами. Додатково використовується безліч вузлів-особливостей, що містять обчислені координати особливостей представлених на сцені. Вузли-поз і вузли-особливості не можуть бути пов'язані один з одним безпосередньо, а тільки через вузли-фактори. Як розширення можуть бути використані вузли-якоря - особливості спостерігаються різними роботами (одночасно) або особливості які бачить один

і той же робот протягом декількох незалежних подорожей. Ядро системи проводить оптимізацію даного графа вирівнюючи все вузли на основі нових даних. Можлива робота і без особливостей і без вузлів-якорів - тільки на основі результатів візуальної одометра.

При всій своїй привабливості, ЕКФ проте має свої недоліки, до яких можна віднести в першу чергу обмеження на кількість орієнтирів в системі. Пов'язано це з тим, що матриця P має розмірність m на m , де m – кількість виявлених орієнтирів. На кожному етапі оновлення матриці P , повинен бути оновлений кожен її елемент, в зв'язку з чим складність алгоритму $O(m^2)$. Таким чином, ЕКФ застосовується в ситуації, коли середовище має не дуже велику кількість (кілька сотень) легко помітних орієнтирів.

2.1.2 Метод FastSLAM

У описі застосування розширеного фільтра Калмана для вирішення завдання картографії місцевості, була позначена основна проблема, що обмежує можливості даного підходу, а саме неможливість його використання на картах, які налічують більше кількох сотень орієнтирів.

Саме для вирішення цієї проблеми в 2002 році Montemerlo, Trun, Koller, і Wegbreit розробили новий підхід до вирішення завдання ОЛК [10]. FastSLAM розділяє завдання локалізації та картографії на безліч підзадач, використовуючи незалежність стану окремих елементів моделі ОЛК. В основі алгоритму FastSLAM лежить застосування Байєсової мережі. З цього слідує, що всі спостереження незалежні один від одного. Насправді, єдине, що їх пов'язує – це помилка визначення положення робота. Таким чином, якщо положення робота визначено абсолютно точно, то в цьому випадку не повинно існувати жодних залежностей між окремими спостереженнями. Звичайно, в реальності положення робота ніколи не відомо абсолютно точно, в цьому і полягає сама сутність проблеми ОЛК, але, тим не менше, сам факт незалежності орієнтирів один від

одного в достатній мірі мотивував творців FastSLAM обробляти кожен орієнтир окремо.

При використанні цього алгоритму, задача ОЛК розділена на кілька підзавдань, і жодна з оцінок стану орієнтира не залежить від попередніх. Цей факт, в свою чергу, дозволяє вирішити проблему поліноміальної складності алгоритму на основі розширеного фільтра Калмана і уникнути її в FastSLAM. Єдина ціна, яку доводиться платити за таке спрощення – це можливість падіння точності, пов'язана з ігноруванням кореляції помилок оцінки положень орієнтирів.

FastSLAM одночасно відстежує кілька можливих маршрутів, в той час як розширений фільтр Калмана не зберігає навіть одного, а лише працює з положенням робота – останнім кроком поточного маршруту. В оригінальному вигляді, FastSLAM зберігає маршрут, але в обчисленнях використовує тільки попередній крок.

В даний час в науковому світі для побудови систем локальної навігації і картографії застосовується як підхід на основі розширеного фільтра Калмана, так і на основі FastSLAM. Вибір того чи іншого методу залежить від необхідної точності системи і її швидкодії, від параметрів середовища, таких як наявність орієнтирів, їх кількість і фізичні характеристики і від багатьох інших факторів. В останні роки все більшого поширення набуває алгоритм FastSLAM і поступово витісняє розширений фільтр Калмана.

2.1.3 Метод DP-SLAM

DP-SLAM – реалізація, що використовує свідчення лазерного далекоміра і фільтр частинок для зберігання гіпотез про становище робота і конфігурації навколишнього його сцени [10]. Фільтр частинок (послідовний метод Монте-Карло) – рекурсивний алгоритм для чисельного вирішення проблем оцінювання (фільтрації, згладжування), особливо для нелінійних випадків.

Метод полягає в тому, що в фільтрі підтримується безліч гіпотез про поточний стан робота. У початковий момент генерується випадковий набір гіпотез. В ході роботи алгоритму деякі з них будуть відсіюватися через невідповідності вторинними ознаками системи, які можуть бути виміряні більш точно. Після завершення циклу сканування вибирається найбільш вірогідна з залишених в фільтрі гіпотез.

Для зберігання і структурування великого числа проміжних карт авторами була реалізована деревоподібна структура (рисунк 2.3). Вся інформація, необхідна для вирішення обох підзадач, локалізації і коригування карти, поміщається в одному фільтр частинок.

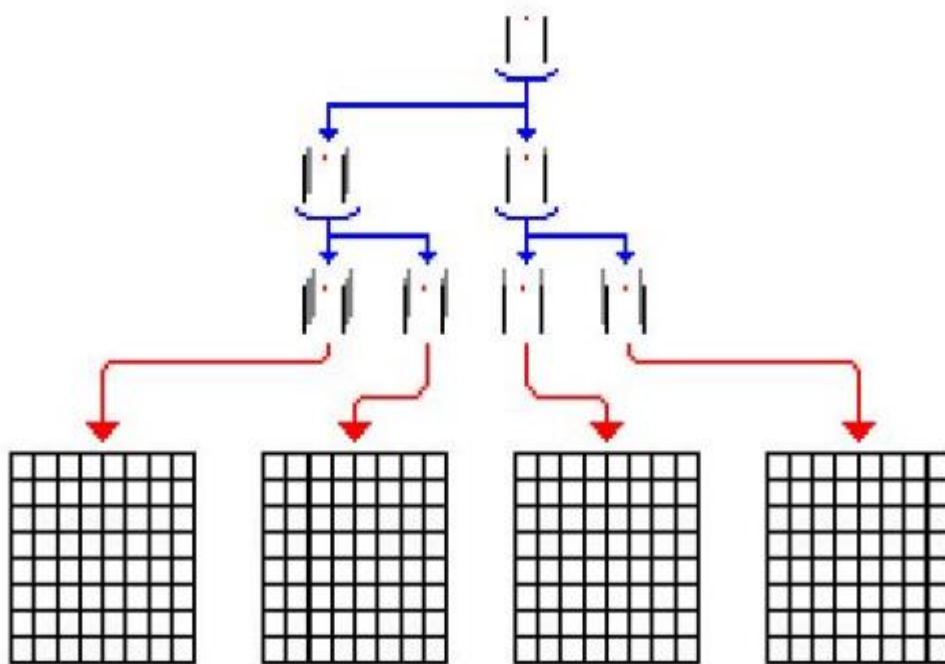


Рисунок 2.3 – Деревоподібна система зберігання "гіпотетичних" карт

Розглянемо дві реалізації алгоритму DP-SLAM. Перша реалізація одночасно підтримує до 9000 можливих гіпотез, і відсіває тільки ті з них, які в силу невиконання яких-небудь умов абсолютно точно не можуть відповідати реальному стану системи. Другий підхід на кожному кроці алгоритму вибирає тільки одну найвірогіднішу гіпотезу і оновлює карту виходячи з неї. У цьому

випадку має перевагу реалізація, яка одночасно підтримує велику кількість гіпотез, адже досить імовірна ситуація, при якій в силу шумів самої різної природи найбільш імовірною стане гіпотеза, досить сильно відрізняється від реального стану системи.

2.2 Модель створення об'єктів доповненої реальності на основі візуальної одометрії

Існуючі бібліотеки для формування доповненої реальності спираються на мітки у навколишньому середовищі. Деякі з них заявляють, що підтримують ОЛК, але на досвіді вони просто генерують мітки з початкової сцени. Тобто, перед розміщенням об'єкта доповненої реальності, бібліотека генерує декілька зображень-міток і далі спирається на них. Таким чином, практично ми бачимо роботу без міток. Але фактично ці мітки просто генеруються при першому запуску. Далі це часто призводить до погрішностей у роботі програмного забезпечення, неплавної появи віртуальних об'єктів, якщо вони зникли з поля зору, а потім знову з'явилися.

Задача полягає у тому, щоб інтегрувати методи одночасного картографування та локалізації для досягнення побудови доповненої реальності без використання маркерів. Це дасть можливість більш плавно відобразити віртуальні об'єкти та будувати великі віртуальні локації поверх зображення реального світу з використанням орієнтування у локації. При цьому, будуть використані як зображення з камери, так і сенсори смартфона, такі як акселерометр, гіроскоп та педометр.

Дано послідовність зображень I_s , знятих камерою, що рухається з відомими внутрішніми параметрами K у моменти часу t_s , $s = 1, 2 \dots N$. Дано евклідові утворення R_s , t_s , що описують положення камери для I_s . Нехай x_i – це множина особливих точок, які можна побачити на I_s . Для кожної x_i дано $x_{i,j} \in R^2$ – упорядкована множина проекції особливої точки x_i на I_j , $j \in t_s$. Нехай $x_{di} \in x_i$ – множина особливих точок, що належить динамічним об'єктам.

Необхідно розробити метод розпізнавання точок динамічних об'єктів, що визначають множину x_{di} найкращим чином.

Розробити метод реконструкції положення динамічного об'єкту, що вираховує координати $x_{di,j}$, $j \in t_s$ у просторі особливих точок динамічних об'єктів x_{di} . Орієнтуючись на ці координати, відобразити об'єкти доповненої реальності.

Будемо вважати камеру відкаліброваною, так що її внутрішня матриця K з фокусом, оптичним центром та координатами дисторсії відома. Кожна точка x_i породжує трек проєкцій $\{x_{i,j}\}$ у моменти $j = t_{i,1}, \dots, t_{i,n_i}$, $x_{i,j} \in R^2$.

Положення камери задається матрицею повороту R 3×3 , $r_{i,j} \in R$ і вектором переносу $t \in R^3$. Визначимо вектор параметрів θ , як вектор, по якому можуть бути побудовані R , t . Відображення проєкції $\pi(\theta, X)$ для точки $X \in R^3$ на камеру з положенням θ при умові одиничної матриці K записується наступним чином

$$\pi(\theta, X) = \frac{1}{r_3^T X + t_3} \begin{pmatrix} r_1^T X + t_1 \\ r_2^T X + t_2 \end{pmatrix}, \quad (2.3)$$

де r_i^T – рядок i матриці R ;

t – компонента вектору переносу, для $i = 1, 2, 3$.

Якщо матриця K не є одиничною, ми можемо домножити усі проєкції в однородному вигляді на K^{-1} .

Обчислення коефіцієнта Калмана K [9] розраховується, щоб з'ясувати, наскільки ми будемо довіряти спостережуваним орієнтирів і скільки нових даних в результаті ми хочемо з них взяти. Якщо ми побачимо, що камера повинна бути переміщена на 10 см вправо, відповідно до попередніх орієнтирів ми використовуємо коефіцієнт Калмана, щоб з'ясувати, наскільки ми дійсно виправляємо становище. Це може бути тільки 5 см, тому що ми не можемо довіряти орієнтирам повністю, а скоріше маємо знайти компроміс між зміщенням орієнтира і корекцією, в тому числі від датчиків. Це робиться з використанням невизначеності спостережуваних орієнтирів поряд з

вимірюванням якості замірів і характеристик датчиків. Якщо чіткість зміщення орієнтирів працює дійсно погано, тобто з високою похибкою, в порівнянні з показниками датчиків пристрою, ми, звичайно, не довіряємо йому, тому коефіцієнт Калмана буде низьким. Навпаки, якщо зміщення орієнтирів обчислюється чітко в порівнянні з показниками датчиків, то коефіцієнт Калмана буде високим.

Визначення динамічних точок. Якщо деяка точка x_i у полі зору камери нерухома, то її проекції $x_{i,j} \in R^2$ задовольняють рівнянню

$$\pi(\theta_{c,j}, X) = x_{i,j} + v_{i,j}, \quad (2.4)$$

де $v_{i,j}$ - випадковий шум у спостереженнях;

$\{\theta_{c,j}\}$ - положення камери у моменти часу $j = t_{i,1}, \dots, t_{i,n_i}$.

Точка класифікується як статична, якщо виконана умова (2.4) на середню похибку перепроєктування:

$$\exists X_i \in R^3 : \frac{1}{n_i} |\pi(\theta_j, X_i) - x_{i,j}| < \varepsilon, \quad (2.5)$$

для деякого параметра похибки ε .

У обраному методі фільтрації Калмана, всі позиції, відмінні від поточної, маргіналізуються після кожного кадру. Функції, які можуть бути виміряні знову в майбутньому, зберігаються. Результатом є графік, який залишається відносно компактним; він не буде рости довільно з часом і не буде рости взагалі при повторному русі в обмеженій області, додавши постійні змінні параметра тільки тоді, коли будуть досліджені нові області. Недоліком є те, що графік швидко стає повністю взаємопов'язаним, оскільки кожне усунення змінної минулої позиції призводить до заповнення новими зв'язками між кожною парою змінних функцій, до яких вона була приєднана. Тому спільний потенціал по всіх цих взаємопов'язаних змінних повинен зберігатися і оновлюватися. Розрахункові витрати на поширення спільних розподілів погано масштабуються з

урахуванням числа змінних, і це головний недолік фільтрації: в ОЛК кількість функцій на карті буде сильно обмежена. Стандартним алгоритмом фільтрації з використанням гауссових імовірнісних розподілів є ЕКФ, де щільні взаємозв'язки між ознаками проявляються в одній щільності з'єднання за функціями, що зберігаються у вектором і великій коваріаційній матриці.

Інший варіант полягає в тому, щоб зберегти підхід оптимізації ВА, дозволяючи будувати графік з нуля з плином часу, коли він росте, але для його спрощення шляхом видалення невеликої підмножини минулих позицій. У деяких додатках розумно, щоб збережені позиції були в ковзному вікні останніх положень камери, але в цілому вони представляють собою набір інтелектуальних або евристичних обраних ключових кадрів. Решта позицій, і всі пов'язані з ними вимірювання, які не маргіналізуються, як в фільтрі, а просто відкидаються – не вносять ніякого вкладу в оцінки поточної позиції. У порівнянні з фільтрацією, цей підхід призведе до створення графіка, який має більше елементів (оскільки багато минулі позиції збережені), але, що важливо для виведення, відсутність маргіналізації означає, що він буде залишатися слабко взаємозалежним. В результаті оптимізація графіка залишається відносно ефективною, навіть якщо кількість функцій на графіку і вимірювань з ключовими кадрами дуже велика. Можливість включення більшої кількості вимірів об'єктів компенсує інформацію, втрачену з відкинутих кадрів.

Реконструкція положення динамічного об'єкту. Динамічний об'єкт i складається з набору точок, представленого у деякій системі координат об'єкта $\{X_{i,j}\}$, $X_{i,j} \in R^3$. Ці точки рухаються узгоджено, і їх місцеположення у момент t задається вектором параметрів $\theta_{o,t}$. Нехай \times – операція композиції евклідових утворень, тоді $\theta_{oc,j} = \theta_{c,j} \times \theta_{o,j}$ задає утворення з координат об'єкта у координати камери, і проєкції кожної точки динамічного об'єкта задовольняють умову

$$\pi(\theta_{o,t}, X_{i,k,l}) = x_{i,kl} + v_{i,k,l} \quad \forall k = 1 \dots N_i, \quad (2.6)$$

де i – номер об'єкта, N_i – число точок об'єкта, l – момент часу.

Починаючи з $N_i \geq 5$, по двом кадрам можуть бути відновлені положення і координати точок об'єкта. Для $N_i \leq 3$ це неможливо без додаткових припущень про положення або руху об'єкта (наприклад, положення або рух в площині).

У цій роботі будемо вважати $N_i \geq 5$. Пропонується наступний алгоритм реконструкції.

- Якщо об'єкт спостерігається на двох досить віддалених кадрах, розрахувати істотну матрицю для точок об'єкта, визначити консенсусну множину точок, що рухаються узгоджено з об'єктом, і відкинути викиди, ініціалізувати координати точок об'єкта;

- Для наступних кадрів, запустити метод ОРпР [13] для точок об'єкта і визначити його позицію на кожному кадрі;

- Виконати bundle adjustment [13] для уточнення положення об'єкта і точок на кожному кадрі.

Останнім часом значно зросло розуміння загального характеру проблем локалізації та реконструкції. Зокрема, нещодавно у області досліджень «Структура з руху» (Structure from Motion, SFM) [14] був виявлений розрив в комп'ютерному баченні, принципи якого були отримані з фотограмметрії та підполі одночасної локалізації і зіставлення (ОЛК) досліджень мобільних застосунків – звідси дещо неясна двозначна термінологія. Істотний характер цих двох проблем, які оцінюють рух датчиків шляхом моделювання раніше невідомого, але статичного середовища, той же, але мотивація дослідників історично була іншою. SFM займався проблемами реконструкції 3D-сцени з невеликих наборів зображень, а проектна геометрія і оптимізація були поширеними методами рішення. З іншого боку, в ОЛК класичною проблемою є оцінка руху рухомого робота в режимі реального часу, оскільки він постійно спостерігає і відображає його невідому середу за допомогою датчиків, які можуть включати чи не включати камери. Тут на перший план вийшли послідовні методи фільтрації.

Він повністю прийняв байєсовські методи, щоб вони могли бути зрозумілі єдиною мовою і повним переходом до методологій. Але залишається той факт, що в конкретному завданні відстеження монокулярних камер в режимі реального часу найкращі системи були сильно прив'язані до одного з підходів або до іншого. Питання в тому, чому і наскільки один підхід явно перевершує інший, потребує вирішення для керівництва майбутніми дослідженнями в цій важливій галузі застосування.

2.3 Аналіз досвіду використання технології доповненої реальності

В останні роки технологія доповненої реальності зазнала небувалого розвитку. Завдяки зусиллям компаній Apple та Google ця технологія стала мати змогу використовуватись будь-де. Так, розроблені різноманітними компаніями додатки на основі технології доповненої реальності, стають затребуваними у різних сферах людської діяльності, наприклад маркетингу, медицині, авіації, туризмі, дизайні, в сфері продаж та в іграх. Все що для цього потрібно — мобільний телефон та доступ до мережі Інтернет, і то, не в усіх випадках. На сьогоднішній день застосування технології доповненої реальності в освітній галузі лише тільки починає набирати оберти. Сьогодні ця технологія знаходить все більш прихильників серед викладачів і науковців ніж декілька років тому. Але треба не забувати, що більшості молоді не цікаво вчитися по методах, які є привичними для старшого покоління.

Провівши аналіз розробок вітчизняних та закордонних виробників, можна зробити висновок, що ринок друкованої продукції з використанням сучасного програмного забезпечення та електронних засобів навчання стає все ширшим. Виробники пропонують споживачам різноманітну дитячу розважальну і навчальну літературу. Технологія доповненої реальності дозволила «оживити» сторінки всесвітньовідомої книги Льюїса Керрола «Аліса у Задзеркаллі». Друге видання, що ілюстроване Є. Гапчинською, українською художницею-живописцем, переносить маленьких дослідників в казковий світ пригод. Таким

чином, поступовий перехід від дитячих книжок-розмальовок і книжок-казок з доповненою реальністю технології застосування доповненої реальності поширюються на виробництво навчальної продукції, тобто поступово переходять від ігрової технології, до технології навчальної.

Як каже керівник напряму «Освіта» компанії Sensorama Academy, Уляна Автомонова, викладачі в Україні дуже консервативні, що значною мірою впливає на поширеність нових технологій в навчальному процесі. Для того, щоб використовувати технологію доповненої реальності в навчальному процесі, необхідно закупити обладнання для роботи. На сьогоднішній день існує більш ніж 20 високоякісних безкоштовних мобільних додатків з різних предметів, які можна інтегрувати в заняття. Прикладом може бути додаток Google Expeditions, що надає більш ніж 700 готових фотосфер з різних навчальних тем.

Відразу потрібно розуміти, що освіта - це сама по собі довгострокова інвестиція. Видимі результати проявляються роками пізніше. У школі, університеті, або іншому навчальному закладі, доповнена реальність допоможе краще засвоювати складні поняття. Гарний приклад - легені. Викладач може довгий час пояснювати функції легенів, як вони виглядають зовні і зсередини, на словах. Уявіть, що завдяки віртуальній реальності вдасться "побувати" всередині легенів, побачити все своїми очима.

Доповнена реальність підходить для початку заняття, щоб занурити в певну тему, середини - щоб візуалізувати якесь складне поняття. Або ж частина з доповненою реальністю можна використовуватися в кінці, щоб надихнути учня на виконання домашнього завдання.

Обов'язково треба виділити одну думку - по мірі розвитку технологій людина отримує більше можливостей, але в той же час втрачає деякі навички, які були у попередніх поколінь. Наприклад, навичка концентрації та усидливість. На жаль, це факт: проводячі більшість часу постійно в цифровому світі інформаційного різноманітності, що складається з нотифікацій Facebook, Telegram, бажання самовиразитися за допомогою Instagram-Селфі або ігор в кінорежисера в Youtube, люди покоління Z дуже погано управляють своєю

увагою. Якщо співрозмовник їм не цікавий, вони приділять йому в середньому 8 секунд.

Говорячи про вітчизняні розробки в сфері освіти, варто згадати «Академію розвитку інтелекту SMARTUM Україна», яка презентувала оновлений мобільний додаток з функцією доповненої реальності для курсу «Ментальна арифметика». Підручники з «Ментальної арифметики» взаємодіють з мобільним додатком при навчанні, показуючи правильність апікатури пальців, а також допомагають виконувати вправи правильно без контролю тренера. Також у підручнику є чарівні картинки для розфарбовування, які оживають в додатку і переливаються яскравими фарбами.

Видавничий дім «Освіта» — перше в Україні видавництво навчальної літератури, яке створило для школи підручники та навчально-методичні додатки до них з доповненою реальністю. Це підручник інтегрованого курсу «Я досліджую світ» для 1 класу закладів загальної середньої освіти, універсальний дидактичний матеріал до курсу «Я досліджую світ» для 1 класу. Наразі готуються до публікації два підручники «Я досліджую світ» для 2 класу а також підручник для 6 класу закладів загальної середньої освіти «Всесвітня історія. Історія України».



Рисунок 2.4 - Підручник з доповненою реальністю

В ході дослідження було виявлені 4 пункта, які гальмують розвиток використання технології доповненої реальності в Україні:

- наявність великої кількості якісного контенту з доповненою реальністю для навчання, але контенту з українською мовою надзвичайно мало;
- проблема вибору пристрою. Деякі додатки працюють лише на операційних iOS або Android останніх версій. Деякі додатки можуть бути не доступні через відмінність операційних систем, наприклад додаток працює лише на iOS, а в навчальному закладі працюють на пристроях з Android;
- висока ціна пристроїв з підтримкою Apple ARKit або Google ARCore. Їх ціна може бути в діапазоні 800-1800\$;
- небажання деяких освітян що-либo міняти в своїй діяльності, незважаючи на конкретні зміни, які впроваджує Міністерство освіти і науки. Міністерство розуміє, як важливі інновації. У деяких школах у класах інформатики досі викладають школярам Word, PowerPoint і двійковий код. Дітям набагато цікавіше дізнатися про 3D-принтери, штучний інтелект або експедиції на Марс.

Зміни в українській освіті відбудуться тоді, коли самі викладачі почнуть цікавитися і добре розбиратися в сучасних технологіях - тільки тоді вони зможуть з інтересом розповідати про це своїм учням. На мою думку саме час відходити від стандартів і робити навчання цікавим для дітей. І, врешті-решт, питати у них самих, що їм цікаво.

За кордоном технологія доповненої реальності зазнала значного розвитку в сфері освіти. Досить оригінально до вивчення іноземної мови підійшли в Японії. Видавнича група Tokyo Shoseki випустила серію англomовних самовчителів і мобільний додаток, який підтримує доповнену реальність на смартфоні. Компанія пропонує по-новому поглянути на звичний світ навчання і не списувати з рахунків старі книжки, об'єднавши звичні речі і сучасні технології. Більш практичні друковані самовчителі англійської мови з елементами доповненої реальності можуть не поступатися в плані інтерактивності типовому

вивчення іноземної мови на екрані iPad, але при цьому залишатися більш доступним рішенням для освітньої системи. Вивчаючи серію книг «Новий обрій», читачі можуть навести смартфон на відповідний розділ сторінки, щоб почути діалог або спробувати поговорити з вигаданими іноземцями. У будь-якому випадку, навик говоріння іноземною мовою при цікавій бесіді розвивається набагато швидше, ніж при прослуховуванні аудіозаписів.



Рисунок 2.5 - Додаток для вивчення іноземних мов Tokyo Shoseki

Мабуть, найбільш наочне застосування AR і VR технологій можливо в вивченні точних та інженерних наук. Проект Ханнеса Кауфмана і Бернда Мейера «PhysicsPlayground» орієнтований на моделювання фізичних експериментів в області механіки. Величезна кількість інструментів для аналізу впливу сили, маси, траєкторії, швидкості та інших характеристик об'єктів фізичного світу дозволяють детально вивчати процеси, що відбуваються і експериментувати в тривимірному віртуальному просторі, виключаючи витрати на оригінальні випробування. Програма функціонує на сучасному фізичному движку, який спочатку створювався для розробки ігор, однак його можливості з лишком забезпечують реалізацію освітніх завдань на практиці.

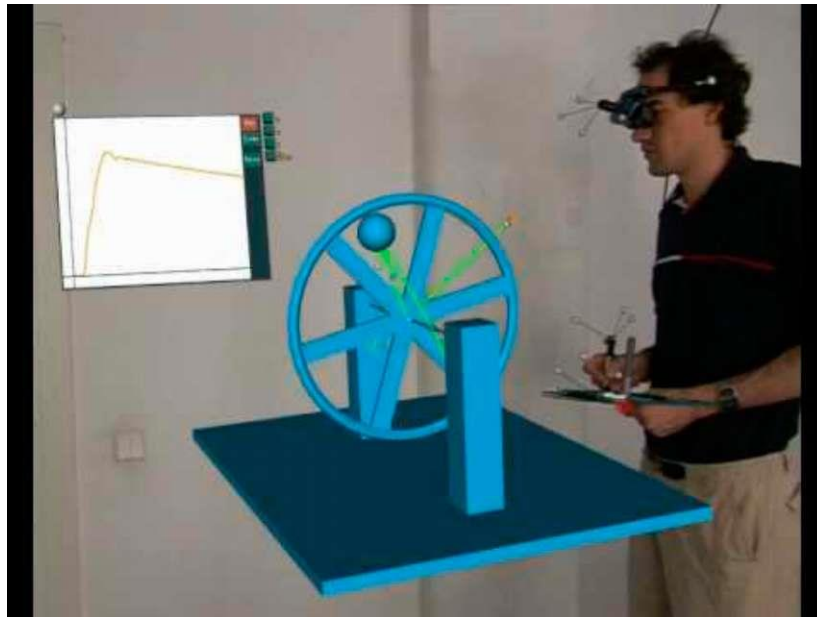


Рисунок 2.6 - Додаток «PhysicsPlayground»

Варто виділити певні моменти:

- Єльський університет протестував VR-тренування для проведення хірургічних маніпуляцій з жовчним міхуром. Автори дослідження виявили, що VR-група була на 29% швидше і в шість разів рідше допускала помилки;
- дослідження на тему: «Вплив VR на академічну діяльність», яке провели в Пекіні, показало схожі результати. Двом групам дітей викладали одну і ту ж дисципліну, але різними способами. Першій групі - традиційними методами, другій - за допомогою віртуальної реальності. Студенти першої групи набрали в середньому 73% в підсумковому тесті, студенти групи VR - 93%. Група VR також продемонструвала більш глибоке розуміння і утримання знань через два тижні.

2.4 Висновки до розділу 2

У розділі 2 розкрито сутність концепції ОЛК. Описано різні методи вирішення цієї задачі. Метод ORB-SLAM з використанням методу розширеного

фільтру Калмана найкраще підходить для використання у випадку з монокулярними камерами через його розповсюдженість та простоту.

Сформовано постановку математичного завдання розробки моделі ОЛК. Розроблено алгоритм побудови доповненої реальності з використанням відомого методу ORB-SLAM для монокулярної камери та використанням фільтру Калмана, що дозволяє розміщувати віртуальні об'єкти у невідомому середовищі без використання заздалегідь заданих міток.

3 РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ ФОРМУВАННЯ ОБ'ЄКТІВ ДОПОВНЕНОЇ РЕАЛЬНОСТІ

3.1 Проектування прототипу мобільного застосунку

Для розробки програмного забезпечення під операційну систему Android OS було використано такі мови програмування, як Java і Kotlin для розробки самого мобільного додатку, та C++ для інтеграції алгоритму ORB-SLAM. Інтеграція виконана з використанням інструментарію Android NDK. У якості бібліотеки доповненої реальності було обрано Wikitude SDK.

У якості середовища розробки використано IDE Android Studio 3.0.1 для Windows.

Вхідними даними є відеопотік з камери смартфона та дані з його датчиків, а саме:

- акселерометра для відслідковування переміщення мобільного пристрою у просторі;
- гіроскопу для реагування на зміну кутів орієнтації мобільного пристрою у просторі;
- педометра для підрахунку кількості зроблених кроків.

На виході отримуємо той же відеопотік, але вже з накладеними поверх нього об'єктами доповненої реальності. Положення об'єктів має динамічно змінюватися з урахуванням переміщення мобільного пристрою у просторі.

3.1.1 Вимоги до архітектури мобільного застосунку

Система повинна підтримувати щонайменше 90% смартфонів з операційною системою Android. Тому мінімальна версія API – 19.

Код програми має бути гнучким до змін, зокрема до правок у методі одночасної локалізації та картографування. Це призводить до потреби в

модульності системи, інтеграції засобами Android NDK для легкого використання вихідних кодів відомих реалізацій SLAM на C++.

Також важлива вимога ефективності та швидкодії процесу обробки вхідного відеопотоку та відображення об'єктів доповненої реальності. Це призвело до вибору нативних методів розробки у середовищі програмування Android Studio з використанням мов програмування Java та Kotlin.

Для керування процесом виконання програми необхідно, щоб користувач мав змогу розмістити об'єкт доповненої реальності, запустити відслідковування місцеположення пристрою, спостерігати за зміною положення віртуального об'єкта у навколишньому середовищі та зупинити відслідковування місцеположення пристрою.

3.1.2 Структура класів мобільного застосунку

Інтерфейс *TrackerListener* надає методи для відслідковування стану сцени навколишнього середовища. Метод *onInitializationPoseChanged* слугує для прослуховування події зміни початкової позиції пристрою. Метод *onStateChanged* – за зміни стану відслідковування об'єктів доповненої реальності. *onTracked* – за подію появи об'єкта у полі зору камери. Метод *onTrackingStarted* – за початок відслідковування об'єктів доповненої реальності. *onTrackingStopped* – за кінець відслідковування.

Інтерфейс *ExternalRendererListener* слугує слухачем на подію кінця процесу ініціалізації зовнішнього рендера *onRenderExtensionCreated*.

Інтерфейс *InstantTracker* надає проміжні методи для взаємодії з методами відслідковування об'єктів за допомогою зовнішньої бібліотеки *WikitudeSDK*. Основними методами є *convertScreenCoordinatesToPointCloudCoordinate* для конвертації положення камери у координати навколишнього середовища, метод *setDeviceHeightAboveGround*, що задає висоту камери над поверхнею землі та метод *setTrackingPlaneOrientation*, що задає поточне положення мобільного пристрою.

Абстрактний клас *Renderable* надає методи та поля для зберігання відтворюваних об'єктів. Матриці *projectionMatrix* та *viewMatrix* зберігають дані про положення об'єкту у просторі та його поточне положення на екрані. Метод *onDrawFrame* малює об'єкт на екрані, в той час як метод *onSurfaceCreated* викликається після першої ініціалізації об'єкту на екрані.

Клас *GLRenderer* містить відомості про усі об'єкти доповненої реальності та відображає їх. У ньому є карта *TreeMap<String, Renderable> mRenderables* з ключем – ідентифікатором об'єкта. Основні методи *onDrawFrame(GL10)* для відмальовування об'єктів на фреймі та *onSurfaceChanged(GL10, int, int)* для реагування на зміни на фреймі. Та методи керування списком об'єктів, такі як додання *setRenderableForKey*, видалення *removeRenderableForKey* та отримання *getRenderableForKey*.

Клас *OrbSLAMHelper* є проміжним між класами мобільного додатку та алгоритмічною частиною вирішення задачі SLAM, написаною на C++. Метод *glesInit* виконує підготовку до виконання локалізації та картографування. Метод *glesResize* задає розміри вхідного відеопотоку даних. Метод *initSystemWithParameters* ініціалізує систему зі шляхом до камери та калібрування. Метод *startCurrentORB* починає локалізацію та картографування SLAM.

Клас *MainActivity* є центральним класом для керування усіма складовими програми. Він реалізує інтерфейси *TrackerListener* та *ExternalRendererListener*. Містить у собі реалізації таких вищеназваних інтерфейсів так класів, як *InstantTracker*, *Driver*, *GLRenderer* та *OrbSLAMHelper*. Містить поле *mWikitudeSDK* для використання методів побудови доповненої реальності за допомогою бібліотеки *Wikitude*. Містить поле *mSurfaceView* для відображення відеопотоку з камери разом з віртуальними об'єктами.

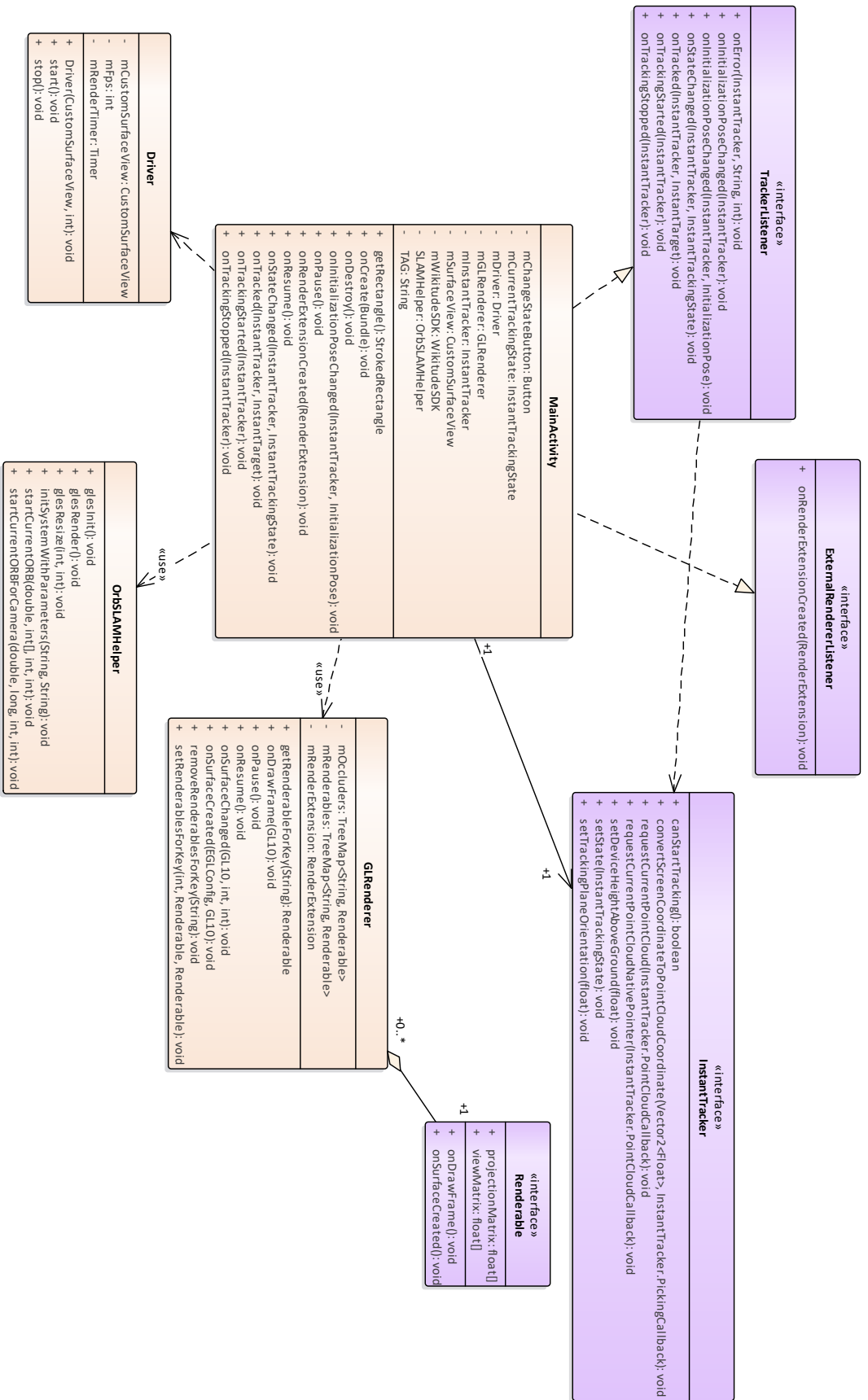


Рисунок 3.1 - Діаграми класів

3.2 Опис функціональності мобільного додатку

Для демонстрації роботи мобільного застосунку було обрано смартфон Xiaomi Redmi 3 Pro з операційною системою MIUI 9.5.2 на базі Android 5.1.1.

Після відкриття застосунку на екрані з'являється вікно з відеопотоком з головної камери (рисунок 3.2). Зверху екрану знаходиться повзунок для введення висоти камери над поверхнею землі. Шкала – від 10 см до 2 м. В залежності від введеної висоти, на екрані змінюється розмір об'єкта доповненої реальності. У даному випадку об'єктом виступає квадратний полігон висотою та шириною у 1 м.



Рисунок 3.2 – Головний екран мобільного додатку

Зелений колір свідчить про готовність застосунку до фіксації об'єкта у навколишньому середовищі. Для початку роботи алгоритму одночасної локалізації та відображення полігону у встановленому місці, внизу екрану знаходиться кнопка (рис. 3.3). Спочатку вона знаходиться у стані «Розмістити об'єкт».

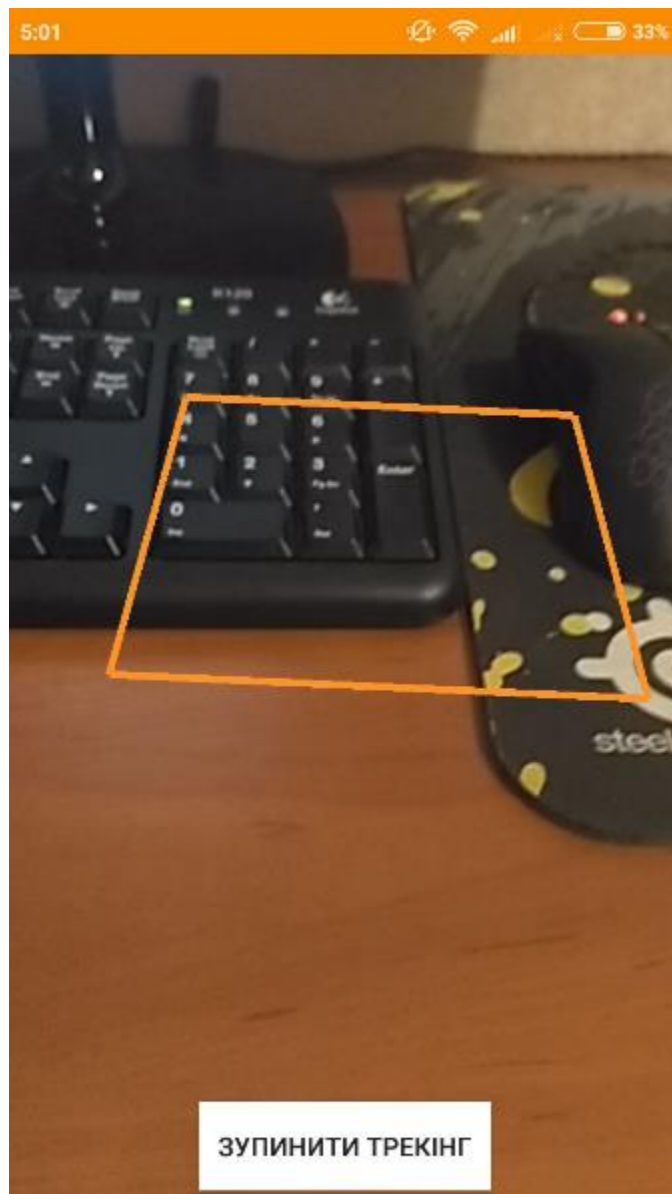


Рисунок 3.3 – Розпочато трекінг об'єкту доповненої реальності

Після натискання на кнопку, вона приймає стан «Зупинити трекінг» (рисунок 3.3). Полігон приймає оранжевий колір. Це свідчить про те, що об'єкт було зафіксовано у навколишньому середовищі. Якщо повертати камерою у різні

сторони, об'єкт буде залишатися на своєму місці відносно навколишнього середовища. При цьому у верхній частині екрану зникає повзунок для введення висоти камери над поверхнею. Це зумовлено тим, що розмір об'єкта вже зафіксовано і він має залишатися незмінним.

Якщо натиснути на кнопку «Зупинити трекінг», програма зупинить відслідковування положення телефону у просторі, об'єкт доповненої реальності знову прийме зелений колір, а у верхній частині екрану з'явиться повзунок для введення висоти над поверхнею землі. Програма повернулася у початковий стан.

3.3 Тестування розробленого мобільного застосунку

Для випробовування застосунку та порівняння результатів з аналогічними рішеннями з відслідковуванням міток, було вирішено провести ряд досліджень.

Користувач мав розмістити об'єкт доповненої реальності у відкритій місцевості. Після цього протестувати зміну положення об'єкту, спочатку не випускаючи його з поля зору. Таким чином перевірити, чи зафіксовано об'єкт. Далі перевірити позицію об'єкта після плавного відведення камери убік до повного зникнення його меж. І повернути камеру на вихідну позицію, перевіривши, що об'єкт знову з'явився у полі зору на тому самому місці.

Було пройдено ряд маршрутів зі зникненням та відображенням об'єкта з поля зору.

3.3.1 Тестування зміни положення об'єкту

Для досліджень з відслідковуванням міток було обрано показовий демонстраційний додаток від бібліотеки EasyAR з технологією InstantTracking. На відкритій вуличній місцевості розміщено об'єкт доповненої реальності – паралелепіпед, розфарбований градієнтом для кращого виділення граней та ребер (рисунок 3.4).

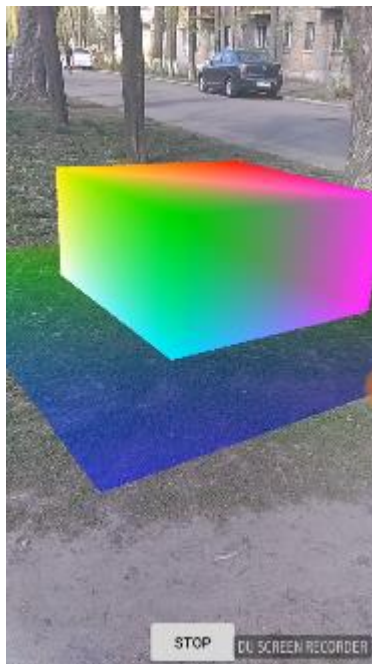


Рисунок 3.4 – Ініціалізовано доповнену реальність

Далі було змінено кут огляду об'єкта доповненої реальності (рисунок 3.5). Як видно, з цим програма легко впоралася. Після цього, відводимо камеру вбік, щоб об'єкт ненадовго пропав з поля зору, та повертаємо назад, щоб мало бути видно лише край об'єкта. Як видно на рисунку 3.6, і з цим програма добре відпрацювала.

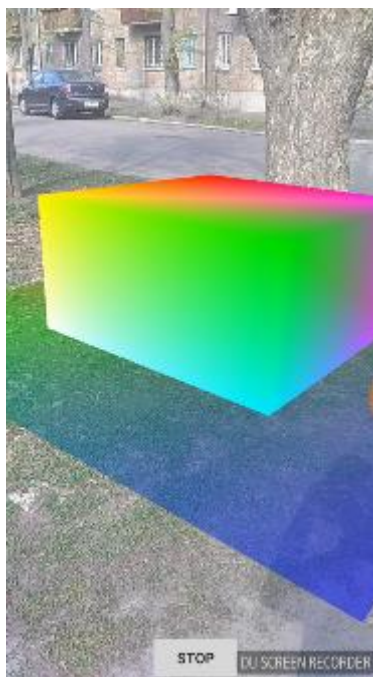


Рисунок 3.5 – Зміна кута огляду



Рисунок 3.6 – Об’єкт доповненої реальності ледь видно

Фінальний тест – проходимо декілька метрів вбік, а потім вертаємося на вихідну позицію. У цьому випадку об’єкт пропав з поля зору і не з’явився на екрані одразу, як очікувалось (рисунок 3.7).



Рисунок 3.7 – Результат виконання після зникнення об’єкта з поля зору

3.3.2 Тестування позиції об'єкта

Для досліджень з використанням одночасної локалізації було проведено ті ж тести на відкритій вуличній місцевості. Об'єкт доповненої реальності ініціалізовано, як показано на рисунку 3.8.



Рисунок 3.8 – Зафіксовано об'єкт доповненої реальності

Далі було змінено кут огляду об'єкта доповненої реальності (рисунок 3.9). Відносно навколишнього середовища, об'єкт залишився на тому ж місці.

Після цього, відводимо камеру вбік, щоб об'єкт ненадовго пропав з поля зору, та повертаємо назад, щоб було видно лише край об'єкта. На рисунку 3.10 бачимо, що мобільний застосунок все правильно відобразив, об'єкт на місці.

Фінальний тест – проходимо декілька метрів вбік, а потім вертаємося на вихідну позицію. У цьому випадку об'єкт пропав з поля зору і з'явився на екрані тоді, коли ми повернулися на початкову позицію (рисунок 3.11).



Рисунок 3.9 – Зміна кута огляду об'єкту



Рисунок 3.10 – Об'єкт доповненої реальності ледь видно



Рисунок 3.11 – Результат виконання після зникнення об’єкта з поля зору

3.4 Аналіз результатів досліджень

Програмний засіб з використанням візуальної одометрії не дав збій після зникнення об’єкта з поля зору, у той час як програма з технологією InstantTracking не змогла одразу відобразити об’єкт у необхідному місці.

Було пройдено декілька маршрутів, після яких об’єкт доповненої реальності залишався на місці. Для порівняння результатів було враховано кількість поворотів у маршруті, його довжину та точність позиціонування об’єктів доповненої реальності (таблиця 3.1).

Проведені дослідження показали, що застосування механізмів візуальної одометрії дає переваги над застосуванням маркерів. Було проведено тестування розробленого мобільного застосунку, яке показало, що в залежності від довжини маршруту та кількості поворотів, ПЗ відображає об’єкти доповненої реальності

з певною похибкою. Також це залежить від точності датчиків, частоти обробки зображень та чіткості зображення з камери.

Таблиця 3.1 – Точність позиціонування для різних маршрутів

Маршрут	Кількість поворотів, шт.	Довжина, м	Точність, см
1	1	3	4
2	2	5	10
3	3	8	18

3.5 Висновки до розділу 3

В даному розділі було визначено засоби розробки програмного забезпечення, вимоги до його архітектури. Для розробки використано такі мови програмування, як Java і Kotlin для розробки самого мобільного додатку для операційної системи Android OS, та C++ для інтеграції методу ORB-SLAM. Інтеграція виконана з використанням інструментарію Android NDK. У якості бібліотеки доповненої реальності було обрано Wikitude SDK.

Розроблено програмне забезпечення, що задовольняє вимогам та реалізує пропонувані рішення для доповненої реальності зі SLAM. Описано класи програми та наведено їх структурну схему.

ВИСНОВКИ

У дипломному проекті магістра було розглянуто засоби та технології побудови доповненої реальності. Найвідоміші з бібліотек – це Vuforia, Wikitude, EasyAR. Досліджено їх функціонал, особливості, зручність використання та вартість. Обрано бібліотеку Wikitude для інтеграції методу ОЛК. Розглянуто методи формування доповненої реальності на мобільних пристроях на основі маркерів та без них. Розкрито концепцію одночасної локалізації та картографування. Аргументовано, навіщо вона може бути використана у процесі формування доповненої реальності.

Недоліком існуючих рішень є той факт, що для розміщення об'єктів доповненої реальності використовуються мітки, задані заздалегідь або згенеровані з відеопотоку. Це накладає обмеження на можливості побудови доповненої реальності у одноманітному середовищі, а також при русі камери за межі віртуальних об'єктів. Тому існує необхідність у пошуку альтернативного підходу до розміщення віртуальних об'єктів та орієнтації пристрою у навколишньому середовищі.

У роботі було розроблено модель формування доповненої реальності з одночасним відстеженням місцеположення, а також створено мобільний застосунок для формування доповненої реальності з використанням розробленої моделі і виконано дослідження з порівнянням результатів.

Описано архітектурні вимоги до програмного забезпечення. Надано опис структурної схеми класів. Проведено ряд експериментів із застосуванням існуючих бібліотек та розробленого програмного забезпечення.

Результати проведених експериментів демонструють, що програмний засіб з використанням одночасної локалізації та картографування не дає збій навіть після зникнення об'єкта з поля зору, у той час як існуючі рішення не змогли одразу відобразити об'єкт у необхідному місці. Було пройдено декілька маршрутів, після яких об'єкт доповненої реальності залишався на необхідному

місці. Для порівняння результатів виконання методу, було враховано кількість поворотів у маршруті, його довжину та точність позиціонування об'єктів доповненої реальності.

Таким чином, усі поставлені завдання дипломного проекту було виконано, а поставлена мета – досягнута. Дослідження результатів виявило, що застосування методів візуальної одометрії більшості випадків дає перевагу у точності позиціонування віртуальних об'єктів у порівнянні з застосуванням маркерів.

ПЕРЕЛІК ПОСИЛАНЬ

- 1 Что такое AR (дополненная реальность) [Электрон. ресурс]. – Режим доступа: <http://tofar.ru/dopolnennaya-realnost.php>
- 2 ARToolKit API Documentation. URL: <http://artoolkit.sourceforge.net/apidoc>.
- 3 ARToolKit website. 2006, 4/20. URL: <http://www.hitl.washington.edu/artoolkit>.
- 4 Billinghurst, M., Kato, H. & Poupyrev, I. The MagicBook - Moving Seamlessly between reality and virtuality. IEEE Comput. Graph. Appl. 2001. Vol. 21, No. 3, pp. 6-8.
- 5 CellaGameS CellaGameS: SMMT library - SLAM Multimarker Tracker for Symbian. 2010. URL: <http://cellagames.com/smmt.html>
- 6 Celozzi, C., Paravati, G., Sanna, A. & Lamberti, F. A 6-DOF ARTagbased tracking system. Consumer Electronics, IEEE Transactions on 2010. Vol. 56, No. 1, pp. 203-210.
- 7 Emerging Technologies 2007. Technology Review. MIT 2007. URL: <http://www.technologyreview.com/specialreports/specialreport.aspx?id=19>.
- 8 Evans, C. Aliens and UFOs. Carlton books, 2008. pp. 163-197.
- 9 Freeman, R.M., Juliet, S.J. & Steed, A.J. A method for predicting marker tracking error. Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR). Nara-Ken New Public Hall, Nara, Japan, 13-16 Nov. 2007. pp. 157-160
- 10 Технологии и алгоритмы дополненной реальности [Электрон. ресурс]. – Режим доступа: <https://arealidea.ru/articles/tekhnologii-i-algoritmy-dlya-sozdaniya-dopolnennoy-realnosti/>
- 11 OpenCV Library [Электрон. ресурс]. – Режим доступа: <https://opencv.org/>
- 12 Обзор средств реализации AR приложений [Электрон. ресурс]. – Режим доступа: <http://nppsatek.com/blogs/2018/01/10/>

13Best augmented reality SDK for AR development for iOS and Android [Электрон. ресурс]. – Режим доступа: <https://thinkmobiles.com/blog/best-ar-sdk-review/>

148 лучших SDK дополненной реальности для iOS и Android [Электрон. ресурс]. – Режим доступа: <https://holographica.space/articles/8-best-ar-sdk-2017-9287>

15Интерфейси доповненої реальності Vuforia Engine [Электрон. ресурс]. – Режим доступа: <http://dspace.pnpu.edu.ua/bitstream/123456789/5176/1/Prokopenko.pdf>

16SLAM (method) [Электрон. ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping

17Søren Riisgaard and Morten Rufus Blas «SLAM for dummies» 2015. – 127 с.

18Р.В. Печерский, С.В. Манько «Алгоритмы локальной навигации и картографии для бортовой системы управления автономного мобильного робота», 2009. – 22 с.

19Сойфер В.А. «Методы компьютерной обработки изображений (2-е издание)» М.: Физматлит, 2003. – 782 с.

20SLAM – принципы и ссылки на open source [Электрон. ресурс]. – Режим доступа: <http://my-it-notes.com/2013/01/slam-basis-and-links-at-open-source/>

21Revisiting the PnP problem: A fast, general and optimal solution / Y. Zheng, Y. Kuang, S. Sugimoto et al. // Computer Vision (ICCV), 2013 IEEE International Conference on / IEEE. — 2013. — P. 2344–2351.

22Real-time Monocular SLAM: Why Filter? / Hauke Strasdat, J. M. M. Montiel and Andrew J. Davison // Strasdat ETAL Conference – 2010. – P. 8.

ДОДАТОК А**Лістинг фрагментів коду**

```

import turicreate as turi

datasetId = 0
def _init_(self, id):
    self.datasetId = id url =
FileManager.urlFor(self.datasetId) data =
turi.image_analysis.load_images(url)
data["type"] = data["path"].apply(lambda path: "Type1" if "type2" in
path else "Types")
data.save("typeIorType2.sframe") data.explore()
dataBuffer = turi.SFrame("typeIorType2.sframe")
trainingBuffers, testingBuffers = dataBuffer.random_split(0.9) model =
turi.image_classifier.create(trainingBuffers, target="type",
model="resnet-50")
evaluations = model.evaluate(testingBuffers)

model.save("TypeClassifier.model")
model.export_coreml("TypeClassifier.mlmodel")

....

import Foundation import CoreML import Vision

var classificationRequest: VNCoreMLRequest

func loadModel() -> VNCoreMLRequest {
    let model = try VNCoreMLModel(for: MobileNet().model)

    let request = VNCoreMLRequest(model: model, completionHandler: {
[weak self] request, error in
        self?.processClassifications(for: request, error: error)
    })
    request.imageCropAndScaleOption = .centerCrop
    self.classificationRequest = request return
    request
}

func initiateHandler(){
    DispatchQueue.global(qos: .userInitiated).async {
        let handler = VNImageRequestHandler(ciImage: ciImage,
orientation: orientation) do {

```

```

        try handler.perform([self.classificationRequest]) } catch {
            print("Failed to perform
classification.\n\(error.localizedDescription)
") }
        }
    }
}

func processClassifications(for request: VNRequest, error: Error?)
{ Di spatchQueue.main.async {
    guard let results = request.results else {
        self.classificationLabel.text = "Unable to classify
image.\n\(error!.localizedDescription)" return
    }
    let classifications = results as! [VNClassificationObservation]
}
}

....

class func loadObjectNamed(_ objectName: String,
fromSceneNamed sceneName: String) -> CAAAnimation? {
    let url = FileManager(urlForResource: sceneName, ofType: "DAE")
    #if os(OSX)
        let options: [SCNSceneSource.LoadingOption: Any]
= [SCNSceneSource.LoadingOption.convertToYUp: true] #else
        let options: [SCNSceneSource.LoadingOption: Any] =
[:] #endif
    let sceneSource = SCNSceneSource(url: url, options: options)
let animation = sceneSource?.entryWithIdentifier(animationName, withClass:
CAAAnimation.self)
    animation?.fadeInDuration =
0.3 animation?.fadeOutDuration
= 0.3 return animation
}

.....

class ARViewController: UIViewController, ARSCNViewDelegate {

    @IBOutlet var sceneView: ARSCNView!

    var nodeModel:SCNNode! let nodeName = ""

    override func viewDidLoad() { super.viewDidLoad()
sceneView.delegate = self

    sceneView.showsStatistics = true sceneView.antialiasingMode =
.multisampling4X

```

```
let scene = SCNScene() sceneView.scene = scene

let modelScene = SCNScene(named:
    "art.scnassets/cherub/cherub.dae")!

    nodeModel = modelScene.rootNode.childNode(withName: nodeName,
recursively: true) }
}
```