

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет прикладної математики**

**Кафедра програмного забезпечення комп'ютерних систем**

«На правах рукопису»  
УДК 004.89

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_ Євгенія СУЛЕМА

«\_\_» \_\_\_\_\_ 2024 р.

**Магістерська дисертація**

**на здобуття ступеня магістра**

**за освітньо-науковою програмою «Інженерія програмного забезпечення  
мультимедійних та інформаційно-пошукових систем»  
зі спеціальності 121 Інженерія програмного забезпечення**

**на тему: «Метод та програмне забезпечення для поєднання  
віртуальної та доповненої реальності на смартфонах»**

Виконав:

студент II курсу, групи КП-31мп  
Проценко Андрій Павлович \_\_\_\_\_

Керівник:

доцент кафедри ПЗКС, к.т.н.  
Юрчишин Василь Якович \_\_\_\_\_

Консультант з нормоконтролю:

доцент кафедри ПЗКС, к.т.н., доцент  
Онай Микола Володимирович \_\_\_\_\_

Рецензент:

старший викладач кафедри СПСКС ФПМ, к.т.н.  
Наливайчук Микола Васильович \_\_\_\_\_

Засвідчую, що у цій  
магістерській дисертації немає  
запозичень з праць інших  
авторів без відповідних  
посилань.

Студент \_\_\_\_\_

Київ – 2024 року

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Факультет прикладної математики**

**Кафедра програмного забезпечення комп'ютерних систем**

Рівень вищої освіти – другий (магістерський)

Спеціальність (спеціалізація) – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Інженерія програмного забезпечення  
мультимедійних та інформаційно-пошукових систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Євгенія СУЛЕМА

«\_\_» \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**  
**на магістерську дисертацію студенту**

Проценко Андрію Павловичу

1. Тема дисертації «Метод та програмне забезпечення для поєднання віртуальної та доповненої реальності на смартфонах», науковий керівник дисертації доцент кафедри ПЗКС, к.т.н. Юрчишин Василь Якович, затверджені наказом по університету від «8» листопада 2024 р. № 5007-С.
2. Термін подання студентом дисертації «17» грудня 2024 р.
3. Об'єкт дослідження: процес поєднання віртуальної та доповненої реальності на смартфонах.
4. Предмет дослідження: методи, способи та програмні засоби поєднання віртуальної та доповненої реальності на смартфонах.
5. Перелік завдань, які потрібно розробити:
  - оцінити сучасний стан проблеми, обґрунтувати актуальність напрямку досліджень, сформулювати мету та задачі дослідження;
  - провести аналіз існуючих рішень для поєднання віртуальної та доповненої реальності;
  - оцінити рівень можливостей які надають існуючі рішення в контексті поєднання віртуальної та доповненої реальності;
  - розробити метод поєднання віртуальної та доповненої реальності для смартфонів згідно з запропонованим способом;
  - виконати програмну реалізацію розробленого методу;
  - оцінити ефективність отриманого програмного рішення й окремих його компонентів у досягненні мети даної роботи;
  - описати бізнес-модель, що дозволить представити кінцеве програмне рішення для даної роботи як повноцінний програмний продукт.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу:
- схема архітектури клітини пам'яті нейронної мережі LSTM (плакат);
  - схема архітектури нейронної мережі (плакат);
  - схема зв'язків модулів програмного забезпечення (плакат);
  - схема роботи системи сканерів LiDAR (плакат);
  - схема моделювання майбутніх траєкторій камери (плакат);
  - дерево проблем та рішень (плакат).

7. Перелік публікацій:

- XVII наукова конференція магістрантів та аспірантів «Прикладна математика та комп'ютинг» (ПМК-2024).

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Онай М.В., к.т.н., доцент		

9. Дата видачі завдання «10» жовтня 2023 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Грунтовне ознайомлення з предметною галуззю	29.10.2023	
2.	Визначення структури магістерської дисертації; вивчення літератури, пошук додаткової літератури, патентний пошук	13.12.2023	
3.	Робота над першим розділом магістерської дисертації; проведення наукового дослідження	20.02.2024	
4.	Проведення наукового дослідження; робота над другим розділом магістерської дисертації; розроблення програмного забезпечення	08.04.2024	
5.	Проведення наукового дослідження; робота над статтею за результатами наукового дослідження	13.05.2024	
6.	Проведення наукового дослідження; робота над третім розділом магістерської дисертації	17.06.2024	
7.	Завершення роботи над основною частиною магістерської дисертації; підготовка ілюстративного матеріалу; підготовка матеріалів доповіді на конференції ПМК-2024	10.11.2024	
8.	Оформлення текстової і графічної частини магістерської дисертації	21.11.2024	

Студент

Андрій ПРОЦЕНКО

Науковий керівник

Василь ЮРЧИШИН

## РЕФЕРАТ

**Актуальність теми.** Зі зростанням популярності мобільних пристроїв та розширенням сфер застосування змішаної реальності постає потреба у рішеннях, здатних поєднати віртуальний та реальний світи без затримок і спотворень. Наявні інструменти та платформи часто залишаючи поза увагою точне прогнозування руху, глибинну реконструкцію сцени й інтегровану підтримку AR/VR. Саме тому актуальним є розроблення методу та програмного забезпечення, які б цілісно охопили ці завдання.

**Об'єктом дослідження** є процес процес поєднання віртуальної та доповненої реальності на смартфонах.

**Предметом дослідження** є методи, способи та програмні засоби поєднання віртуальної та доповненої реальності на смартфонах.

**Метою роботи** є створення програмного забезпечення, покращення вже існуючих методів поєднання віртуальної та доповненої реальності на смартфонах з використанням технологій машинного навчання та штучного інтелекту.

**Наукова новизна** роботи полягає у запропонованому методі передбачення рухів голови за допомогою нейронної мережі LSTM із довжиною послідовності 64 і відбіркою складних траєкторій, який досягає найвищої точності з F1-score до 96% та MAE близько 0.83–0.85 см, забезпечуючи при цьому стабільний час виконання, достатній для застосування в реальному часі.

**Практична цінність** полягає у можливості підвищити реалістичність і стабільність змішаних AR/VR-сцен на смартфонах. Запропонований підхід точно прогнозує рух пристрою, ефективно використовує глибинні дані та оптимізовано інтегрується з AR/VR-інструментами, надаючи конкурентні переваги над існуючими рішеннями.

**Апробація роботи.** Основні положення і результати даної роботи були представлені на XVII науковій конференції магістрантів та аспірантів

«Прикладна математика та комп'ютинг ПМК-2024» (Київ, 20-22 листопада 2024р.).

**Структура та обсяг роботи.** Магістерська дисертація складається зі вступу, п'яти розділів, висновків та додатків.

У вступі надано загальні відомості про сучасний стан речей в області віртуальної та доповненої реальності, зокрема на смартфонах, виконано оцінку проблем створення таких застосунків, обґрунтовано актуальність напрямку досліджень у даній роботі.

У першому розділі був зроблений огляд існуючих інструментальних рішень для поєднання віртуальної та доповненої реальності на смартфонах, проведений аналіз їх особливостей, вказані переваги та недоліки, обґрунтована необхідність розробки методу та програмного забезпечення поєднання віртуальної та доповненої реальності на смартфонах.

У другому розділі зроблений теоретичний виклад пропонованого способу для віртуальної та доповненої реальності на смартфонах, описані принципи роботи його основних частин.

У третьому розділі наведений опис засобів розробки програмного забезпечення та архітектури системи, а також функціональності, що надають її компоненти із зазначанням їх структури та наданням прикладів роботи.

У четвертому розділі проведений аналіз програмної реалізації основних частин запропонованого підходу та роботи програми в цілому. Також визначені напрями для її подальшого вдосконалення у майбутньому.

У п'ятому розділі побудовано бізнес-модель кінцевого продукту, у якій визначені основні параметри та процеси, що важливі для реалізації розробленого програмного забезпечення у вигляді комерційного програмного рішення.

У висновках підбиті підсумки про проведену роботу в рамках даної дисертації.

**Ключові слова:** мобільний застосунок, смартфон, змішана реальність, віртуальна реальність, доповнена реальність, програмне забезпечення.

## ABSTRACT

**Actuality.** With the growing popularity of mobile devices and the expanding range of applications for mixed reality, there is a pressing need for comprehensive solutions capable of integrating virtual and real worlds without delays or distortions. Existing tools and platforms often focus only on certain aspects, leaving out precise motion prediction, in-depth scene reconstruction, and integrated AR/VR support. Therefore, developing a method and software that comprehensively address these challenges and ensure a smooth, realistic, and scalable interaction experience in a mixed environment is highly relevant.

**Object of research** is the process of integrating virtual and augmented reality on smartphones.

**Subject of research** is methods, approaches, and software tools for integrating virtual and augmented reality on smartphones.

**Goal of the work** is to create software and improve existing methods of combining virtual and augmented reality on smartphones using machine learning and artificial intelligence technologies.

**The scientific novelty** of the work lies in the proposed method of predicting head movements using an LSTM neural network with a sequence length of 64 and a selection of complex trajectories, achieving the highest accuracy with an F1-score of up to 96% and an MAE of about 0.83–0.85 cm, while maintaining a stable execution time sufficient for real-time applications.

**The practical value** of the method lies in the ability to significantly enhance the realism and stability of mixed AR/VR scenes on smartphones. The proposed approach more accurately predicts device movement, efficiently utilizes depth data, and integrates optimally with AR/VR tools, providing a competitive edge over existing solutions.

**Approbation.** The main provisions and results of this work were presented at the XVII scientific conference of master's and postgraduate students "Applied Mathematics and Computing PMC-2024" (Kyiv, November 22, 2024).

**Structure and content.** The master's thesis consists of an introduction, five chapters, conclusions and appendices.

The introduction provides general information on the current state of affairs in the field of virtual and augmented reality, particularly on smartphones, evaluates the challenges of creating such applications, and substantiates the relevance of the research direction undertaken in this work.

The first chapter reviews existing tool-based solutions for integrating virtual and augmented reality on smartphones, analyzes their features, highlights their advantages and disadvantages, and justifies the necessity of developing a method and software for combining virtual and augmented reality on smartphones.

The second chapter presents the theoretical foundations of the proposed approach for virtual and augmented reality on smartphones, describing the operating principles of its main components.

The third chapter details the software development tools and system architecture, as well as the functionality provided by its components, including their structure and examples of operation.

The fourth chapter analyzes the software implementation of the main parts of the proposed approach and the program's overall operation. It also identifies directions for future improvement.

The fifth chapter constructs a business model for the final product, defining the key parameters and processes essential for implementing the developed software as a commercial solution.

The conclusions summarize the work carried out within this thesis.

**Keywords:** mobile application, smartphone, mixed reality, virtual reality, augmented reality, software, machine learning model.

## ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ .....	4
ВСТУП.....	6
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ДЛЯ ПОЄДНАННЯ ВІРТУАЛЬНОЇ ТА ДОПОВНЕНОЇ РЕАЛЬНОСТІ .....	7
1.1. Проблеми поєднання віртуальної та доповненої реальності .....	7
1.2. Аналіз існуючих інструментальних рішень .....	8
1.3. Висновки до розділу 1 .....	15
2. АНАЛІЗ МЕТОДУ ПОЄДНАННЯ ВІРТУАЛЬНОЇ ТА ДОПОВНЕНОЇ РЕАЛЬНОСТІ НА СМАРТФОНАХ .....	16
2.1. Концептуальна архітектура системи .....	16
2.2. Обробка сенсорних даних та виділення ключових ознак .....	18
2.3. Використання LSTM для прогнозування руху смартфона .....	19
2.4. Інтеграція PyTorch та ExecuTorch для мобільного інференсу .....	22
2.5. Реконструкція сцени та формування стерео-зображень .....	23
2.6. Інтеграція з Unity та AR Foundation .....	25
2.7. Платформна специфіка: використання Xcode та Rider .....	26
2.8. Оптимізація продуктивності та зменшення затримок .....	28
2.9. Забезпечення масштабованості та адаптації під різні сценарії .....	29
2.10. Висновки до розділу 2 .....	30
3. ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ СПОСОБУ ДЛЯ ГЕНЕРАЦІЇ ТЕСТІВ .....	32
3.1. Опис засобів розробки програмного забезпечення .....	32
3.2. Архітектура розробленого програмного забезпечення .....	36
3.3. Особливості програмної реалізації системи .....	42
3.4. Висновки до розділу 3 .....	44

4. АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ .....	46
4.1. Аналіз результатів роботи розробленого застосунку .....	46
4.2. Шляхи подальшого вдосконалення застосунку .....	48
4.3. Висновки до розділу 4 .....	49
5. ПОБУДОВА БІЗНЕС-МОДЕЛІ .....	51
5.1. Виділення проблеми .....	51
5.2. Зацікавлені сторони .....	52
5.3. Комерційне рішення. Основні характеристики .....	54
5.4. Конкурентні переваги рішення .....	55
5.5. Унікальна ціннісна пропозиція .....	56
5.6. Доходи та витрати .....	58
5.7. Бізнес-модель .....	60
5.8. Висновки до розділу 5 .....	63
ВИСНОВКИ .....	64
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ .....	67
ДОДАТКИ .....	70

## СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

Application Program Interface (API) – це набір правил чи інструкцій, які дозволяють різним програмам взаємодіяти одна з одною й визначають набір методів і правил для обміну даними між програмами. Це дозволяє розробникам використовувати функціональність програми чи сервісу без необхідності розкриття всіх внутрішніх деталей її реалізації [5].

Mean Absolute Error (MAE) — середня абсолютна похибка, яка використовується як міра точності прогнозу [6].

Unity – багатоплатформовий інструмент для розроблення відеоігор і застосунків, і рушій, на якому вони працюють. Створені за допомогою Unity програми працюють на настільних комп'ютерних системах, мобільних пристроях та гральних консолях у дво- та тривимірній графіці, та на пристроях віртуальної чи доповненої реальності [7].

Віртуальна реальність (англ. *Virtual reality*; VR) – це створений технічними засобами світ, який передається людині через її відчуття: зір, слух, дотик і інші [8].

Графічний процесор (англ. *Graphics processing unit*; GPU) – окремий пристрій персонального комп'ютера або ігрової приставки, виконує графічний рендеринг. Сучасні графічні процесори дуже ефективно обробляють і зображують комп'ютерну графіку, завдяки спеціалізованій конвеєрній архітектурі вони набагато ефективніші в обробці графічної інформації, ніж типовий центральний процесор [9].

Довга короткочасна пам'ять, ДКЧП (англ. *Long short-term memory*; LSTM) – це архітектура рекурентних нейронних мереж. На відміну від традиційних РНМ, мережа ДКЧП добре підходить для навчання з досвіду з метою класифікації, обробки або передбачення часових рядів в умовах, коли між важливими подіями існують часові затримки невідомої тривалості [10].

Доповнена реальність (англ. *Augmented reality*; AR) – це результат введення в поле сприйняття будь-яких сенсорних даних з метою доповнення

відомостей про оточення і поліпшення сприйняття інформації [11].

Змішана реальність (англ. *Mixed reality*; MR) – проектування тривимірних віртуальних об'єктів або голограм на фізичний простір. Дозволяє переміщатись навколо віртуального об'єкту, оглядати його з усіх боків і, за необхідності, всередині [12].

Інерційний вимірювальний пристрій, ІВП (англ. *Inertial Measurement Unit*; IMU) – це електронний пристрій, який вимірює і видає на вихід значення швидкості тіла, його орієнтації, та гравітаційних сил з якими рухається тіло, а іноді і магнітне поле, що його оточує, використовуючи для цього поєднання акселерометрів, гіроскопів, і іноді магнітометрів [13].

Лідар (англ. *Light Identification, Detection and Ranging*; LiDAR) – технологія отримання та обробки інформації про віддалені об'єкти за допомогою активних оптичних систем, що використовують явища відбиття світла і його розсіювання в прозорих і напівпрозорих середовищах [14].

Рекурентні нейронні мережі, (англ. *Recurent neural networks*; RNN) – це клас штучних нейронних мереж, у якому з'єднання між вузлами утворюють граф орієнтований у часі. Це створює внутрішній стан мережі, що дозволяє їй проявляти динамічну поведінку в часі. На відміну від нейронних мереж прямого поширення, РНМ можуть використовувати свою внутрішню пам'ять для обробки довільних послідовностей входів [15].

Система глобального позиціонування (англ. *Global Positioning System*; GPS) – сукупність радіоелектронних засобів, що дозволяє визначати положення та швидкість руху об'єкта на поверхні Землі або в атмосфері. Положення об'єкта обчислюється завдяки використанню розміщеного на ньому GPS-приймача, який приймає та обробляє сигнали супутників космічного сегменту GPS-системи глобального позиціонування [16].

Центральний процесор (англ. *Central processing unit*; CPU) – частина комп'ютера, що призначена для інтерпретації команд [17].

## ВСТУП

У сучасних умовах бурхливого розвитку цифрових технологій, коли межі між фізичним та віртуальним дедалі частіше стираються, створення нових форм взаємодії користувача з цифровим контентом стає стратегічно важливим завданням. Розширення можливостей мобільних пристроїв, удосконалення їх апаратної складової та збільшення обчислювальної потужності формують передумови для появи якісно нових рішень, здатних забезпечити глибше занурення у змішане середовище. Саме в таких реаліях виникає потреба у методах та програмних засобах, які б уніфікували підходи до поєднання різнорідних технологій, забезпечували плавність, стабільність та масштабованість досвіду взаємодії.

Водночас, стрімкий темп інновацій, високі очікування користувачів та складність організації єдиної платформи інтеграції ставлять розробників перед низкою викликів. Підвищена складність архітектурних рішень, необхідність точного прогнозування змін у просторі, оптимізація використання ресурсів та забезпечення бездоганного візуального сприйняття – усе це вимагає продуманих підходів до розробки. Відсутність систематизованого методу, що враховував би взаємодію між апаратними обмеженнями, обчислювальними витратами, мережевими особливостями та потребами кінцевих користувачів, затримує впровадження перспективних ідей у життя.

Отже, виникає необхідність у формуванні цілісного концептуального рішення: такої методології та програмного забезпечення, які б поєднували всі ланки інноваційного ланцюга – від збору даних до побудови складних інтерактивних середовищ. Лише за умови системного, адаптивного та ефективного підходу стане можливим досягти якісно нового рівня взаємодії з цифровою реальністю, відкриваючи шлях до нових сценаріїв застосування, підвищення конкурентоспроможності та задоволення вимог сучасного ринку.

# 1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ДЛЯ ПОЄДНАННЯ ВІРТУАЛЬНОЇ ТА ДОПОВНЕНОЇ РЕАЛЬНОСТІ

## 1.1. Проблеми поєднання віртуальної та доповненої реальності

У сучасному технологічному середовищі стрімкий розвиток апаратного забезпечення смартфонів створює сприятливий ґрунт для впровадження різноманітних рішень у сфері змішаної реальності, де віртуальні сцени поєднуються з об'єктами реального світу. Проте інтеграція віртуальної та доповненої реальності саме на мобільному пристрої потребує врахування низки складних завдань, що виникають через технічні, методологічні та концептуальні обмеження.

Одним із ключових викликів є забезпечення стабільності та плавності відтворення віртуальних об'єктів у режимі реального часу. Значні обчислювальні ресурси, необхідні для рендерингу тривимірної графіки, аналізу глибинної інформації та коректного поєднання віртуальних шарів з реальним оточенням, часто призводять до затримок та «запізнення» зображення. Такий ефект помітно впливає на користувацький досвід, оскільки навіть найменше порушення синхронності між рухом смартфона і оновленням віртуальної сцени може викликати дискомфорт, а у деяких випадках і фізичний дисбаланс.

Іншою проблемою стає прогнозування руху та орієнтації пристрою в просторі. Враховуючи, що користувач може довільно рухати смартфон у різних напрямках, система повинна не просто швидко реагувати на зміни, а й заздалегідь передбачати майбутню позицію камери. Використання даних з датчиків IMU дає змогу фіксувати прискорення, гіроскопічну інформацію та зміни положення, але перетворення цих сигналів у точні прогнози руху для коректного рендерингу віртуального контенту – надзвичайно складне завдання. Навчання глибинних моделей, таких як LSTM, здатне враховувати часову динаміку даних, проте потребує великих обсягів якісних тренувальних наборів і високої обчислювальної потужності для опрацювання.

Не менш складним аспектом є реконструкція сцени та формування

стерео-зображень. Для досягнення високої реалістичності слід коректно визначати глибину кожного елемента сцени, інтегрувати інформацію з LiDAR-сенсорів або програмних алгоритмів аналізу простору. Недостатнє або неточне відтворення глибинної карти призводить до появи артефактів, які руйнують ілюзію присутності та повноти віртуального простору. В результаті, формування високоякісних стереокартин на основі обчисленої глибини стає однією з найскладніших технічних проблем, оскільки смартфон має обмежені ресурси, а користувач очікує реалістичності, відповідної до рівня стаціонарних VR/AR-систем.

Нарешті, застосування віртуального контенту в реальному просторі потребує комплексного підходу до програмної архітектури. Поєднання Unity з AR Foundation, оптимізація моделі за допомогою PyTorch і запуск застосунку безпосередньо на пристрої за допомогою ExecuTorch вимагають налагодженої взаємодії між численними компонентами. Нерідко виникають труднощі інтеграції, конфлікти версій, проблеми з продуктивністю або сумісністю з апаратною платформою, які ускладнюють розробку та призводять до збільшення часу й витрат на створення продукту.

Таким чином, проблеми поєднання віртуальної та доповненої реальності на смартфонах включають стабільність відтворення сцени, ефективне прогнозування руху пристрою, точну реконструкцію глибинної інформації та складність налагодження багатоетапної інтеграції програмних компонентів. Усі ці аспекти потребують продуманих рішень, гнучких алгоритмів та оптимізованих методів, здатних адаптуватись до динамічних умов мобільного середовища.

## **1.2. Аналіз існуючих інструментальних рішень**

Проведення аналізу існуючих рішень у сфері поєднання AR та VR на мобільних пристроях дає змогу зрозуміти, чому дотепер досягнути ідеальної інтеграції вдалося лише частково. Більшість сучасних платформ і фреймворків, таких як ARKit від Apple або ARCore від Google, надають

висококласні можливості для розпізнавання площин, відстеження руху пристрою, накладання віртуальних об'єктів на реальний світ [18]. Проте вони часто зосереджені лише на аспектах доповненої реальності й не передбачають злиття з повноцінним віртуальним світом, а також не націлені на складну обробку даних з LiDAR чи детальне передбачення руху за допомогою нейронних мереж [19].

Було розглянуто можливості Apple ARKit, Google ARCore, Unity AR Foundation, а також інструментарій для глибинного навчання та оптимізації моделей на пристроях (PyTorch, ExecuTorch), що можуть застосовуватись для прогнозування руху смартфона та вдосконалення якості відтворення віртуальних сцен.

### ***1.2.1. Apple ARKit***

ARKit розроблено компанією Apple спеціально для пристроїв iOS, з метою спростити створення додатків доповненої реальності.

Можливості:

- Висока точність трекінгу положення та орієнтації пристрою завдяки використанню даних з камери, акселерометра та гіроскопа.
- Вбудована підтримка розпізнавання площин і маркерів, дозволяє швидко розмістити віртуальні об'єкти в реальному просторі.
- Підтримка LiDAR на пристроях Pro-серії (iPad Pro, iPhone Pro), що дає змогу отримувати дані глибини та забезпечувати точнішу реконструкцію оточення.

Обмеження:

- Орієнтованість переважно на доповнену реальність без глибинної інтеграції з повноцінними віртуальними сценами.
- Відсутність прямої інтеграції з навчанням та виконанням складних нейронних моделей (наприклад, LSTM) без додаткових зовнішніх інструментів.
- Обмежена робота лише на пристроях Apple.

### **1.2.2. Google ARCore**

ARCore – це аналог ARKit для Android-платформи.

Можливості:

- Кросплатформеність у межах Android-екосистеми, що дає змогу охопити широкий спектр пристроїв.
- Відстеження руху за допомогою візуальної інерціальної одометрії, розпізнавання горизонтальних та вертикальних площин і накладення віртуальних об'єктів на реальне середовище.
- Інтеграція з різними сервісами Google для розширення функціоналу (Geo Anchors, Cloud Anchor).

Обмеження:

- Як і ARKit, сфокусований переважно на доповненій реальності, не передбачає безпосередньої інтеграції зі складними мережевими моделями для передбачення руху чи реконструкції за межами поля зору.
- Не всі пристрої Android мають однакову якість датчиків, що може впливати на точність трекінгу.

### **1.2.3. Unity AR Foundation**

AR Foundation – це набір інструментів від Unity, що спрощує розробку AR-додатків для різних платформ (ARKit, ARCore).

Можливості:

- Єдиний API, що дозволяє створювати кросплатформені AR-застосунки, які запускаються і на iOS, і на Android.
- Просте інтегрування 3D-графіки, анімацій, фізики та логіки ігрового движка Unity у доповнену реальність.
- Швидке створення прототипів та розширення функціоналу за рахунок численних плагінів.

Обмеження:

- Відсутність нативної підтримки виконання складних ML-моделей

(наприклад, LSTM для прогнозування руху). Для цього необхідні зовнішні інструменти та власна інтеграція.

- Основний фокус на базовій AR-функціональності (відстеження руху, накладання об'єктів), без комплексних сценічних реконструкцій чи прогнозувань наступного положення пристрою.

Загалом, існуючі рішення частково вирішують окремі завдання, але не надають цілісної платформи, яка б безпосередньо інтегрувала доповнену та віртуальну реальність, забезпечувала прогнозування руху, глибинний аналіз сцени та створення стереозображень із використанням LSTM-моделей. Таким чином, розробникам доводиться комбінувати кілька інструментів та платформ, адаптувати їх один до одного, вирішувати проблеми сумісності й продуктивності, а також шукати власні підходи до комплексного поєднання цих технологій на смартфонах.

### **1.3. Аналіз існуючих апаратних рішень**

У процесі формування середовищ змішаної реальності апаратні рішення відіграють ключову роль, оскільки саме вони забезпечують коректний запис, аналіз та відтворення інформації про реальний світ, а також накладання на нього віртуального контенту. Сьогодні на ринку представлено низку пристроїв та платформ, призначених для підтримки віртуальної та доповненої реальностей, які різняться між собою за рівнем інтеграції з навколишнім середовищем, апаратними можливостями, точністю відстеження руху та сферами застосування.

Апаратні рішення, орієнтовані на AR/VR:

1. Смартфони та планшети з підтримкою AR: Останніми роками смартфони, оснащені високоякісними камерами, гіроскопами, акселерометрами, а у топових моделях і глибинними сенсорами, стали ключовою платформою для AR-рішень. Такі пристрої, завдяки API, здатні розпізнавати площини, відстежувати рух камери, реконструювати оточення та накладати віртуальні об'єкти на

реальний світ у режимі реального часу.

Технології та інструменти

- вбудовані IMU-сенсори (акселерометр, гіроскоп, магнітометр) для орієнтації у просторі;
- камери високої роздільності для визначення ключових точок та трекінгу маркерів або поверхонь;
- LiDAR чи ToF-сенсори (в окремих моделях) для точної оцінки глибини та створення 3D-карти оточення.

Сфери застосування:

- розважальні додатки (ігри з AR-істотами, віртуальна примірка меблів чи одягу);
- освітні програми (візуалізація навчальних матеріалів прямо у реальному середовищі);
- архітектурне проектування (перегляд 3D-моделей у справжньому масштабі всередині приміщень чи на відкритій місцевості).

## 2. Окуляри та гарнітури доповненої реальності (AR glasses):

Спеціалізовані пристрої на кшталт Microsoft HoloLens, Magic Leap One, Epson Moverio та Google Glass використовують прозорі дисплеї чи хвилевідні лінзи для накладання цифрових об'єктів безпосередньо на поле зору користувача. Ці гарнітури інтегрують камери та датчики глибини, що дозволяє їм точно визначати положення користувача та оточення.

Технології та інструменти:

- хвилевідні лінзи або прозорі дисплеї для відтворення віртуальних зображень поверх реального світу;
- вбудовані камери RGB та інфрачервоні сенсори для трекінгу простору та рук користувача;
- високоточні IMU та SLAM-алгоритми для фіксації положення і орієнтації гарнітури у просторі.

Сфери застосування:

- промисловість та інженерія (візуалізація інструкцій для технічного обслуговування, монтажу та ремонту обладнання);
- медицина (візуалізація внутрішніх структур у хірургії, навчання студентів-медиків);
- логістика (відображення маршрутів та інформації про товар, складську логістику).

3. Гарнітури віртуальної реальності (VR headsets): Пристрої на кшталт Oculus Quest, HTC Vive, Valve Index та PlayStation VR занурюють користувача у повністю віртуальне середовище, ізолюючи його від реального світу. Попри те, що вони сконцентровані переважно на VR, сучасні моделі часто мають фронтальні камери та інфрачервоні сенсори, які дозволяють реалізувати елементи доповненої чи змішаної реальності.

Технології та інструменти:

- високоякісні OLED або LCD-дисплеї з високою частотою оновлення та низькою затримкою;
- трекінг руху голови та рук за допомогою вбудованих камер чи зовнішніх базових станцій ( Lighthouse-технологія у HTC Vive);
- контролери з вбудованими датчиками позиції, які дозволяють взаємодіяти з віртуальними об'єктами.

Сфери застосування:

- розваги та ігри (повне занурення у віртуальні світи з інтерактивними елементами);
- віртуальні екскурсії, тури та перегляд 360° відео;
- навчання та тренінги (симуляції складних чи небезпечних умов, наприклад, тренувальні програми для пілотів, пожежників чи медичного персоналу).

4. Просторові сканери, LiDAR та спеціалізовані датчики: Окремі апаратні рішення фокусуються на точній оцінці глибини та геометрії оточення безпосередньо у реальному часі. LiDAR, Time-of-Flight-

камери, стереокамери (наприклад, Intel RealSense) дозволяють точно визначити положення об'єктів і створити детальні 3D-моделі простору. Це особливо важливо для змішаної реальності, де віртуальний контент повинен коректно взаємодіяти з реальними поверхнями та об'єктами.

Технології та інструменти:

- LiDAR-сенсори, що посиляють інфрачервоні імпульси та аналізують час повернення сигналу для побудови глибинної карти;
- стереокамери, що використовують паралельні зображення для триангуляції глибини;
- алгоритми SLAM (Simultaneous Localization and Mapping) для побудови та постійного оновлення 3D-карти оточення.

Сфери застосування:

- архітектура та дизайн (точне 3D-сканування приміщень для накладання віртуальних елементів меблів чи декору);
- промислові інспекції та робототехніка (машинне бачення для навігації роботів у реальному середовищі).

5. Спеціалізовані портативні пристрої та носимі комп'ютери: Крім гарнітур, існують інші форми апаратних рішень: наприклад, наручні пристрої, що відстежують рух кистей та пальців і відтворюють їх у VR/AR-середовищах. Такі сенсорні рукавички, кільця або браслети дозволяють взаємодіяти з віртуальним контентом більш природно.

Технології та інструменти:

- сенсорні рукавички з тактильною віддачею, що моделюють дотик до віртуальних об'єктів;
- кільця з ІЧ-сенсорами для розпізнавання жестів;
- інтеграція з гарнітурами AR/VR для точного трекінгу рухів.

Сфери застосування:

- ігри та розваги для більш природної взаємодії з віртуальними об'єктами;

- спеціалізовані галузі (наприклад, медицина чи інженерія), де точне відтворення дій оператора у віртуальному середовищі критично важливе.

#### **1.4. Висновки до розділу 1**

У даному розділі зроблений ввід в тематику дослідження даної магістерської дисертації. Наведені найчастіші проблеми, що виникають в процесі тестування і його автоматизації. Проведений аналіз найпопулярніших інструментів для тестування графічних інтерфейсів з зазначенням їх функціональних можливостей, переваг та недоліків. Доступні наразі інструменти можуть застосовуватись для тестування практично будь-яких, навіть досить специфічних випадків у програмному коді, проте вимагають значних зусиль у вивченні того як ними користуватись та конфігурувати їх, а також часу для написання коду для тестування з їх допомогою. Також їх використання передбачає вміння знати й розуміти різні сценарії тестування програмних компонент й реалізовувати їх.

У наступному розділі запропонований спосіб підвищення ступеню автоматизації тестування графічних інтерфейсів веб-додатків за допомогою використання технологій штучного інтелекту і машинного навчання.

## 2. АНАЛІЗ МЕТОДУ ПОЄДНАННЯ ВІРТУАЛЬНОЇ ТА ДОПОВНЕНОЇ РЕАЛЬНОСТІ НА СМАРТФОНАХ

### 2.1. Концептуальна архітектура системи

Концептуальна архітектура розроблюваного програмного рішення передбачає тісну взаємодію кількох логічних підсистем, які в комплексі забезпечують поєднання віртуальної та доповненої реальності на смартфонах. Головна ідея полягає у формуванні наскрізного конвеєра обробки інформації: від збору сирих даних з апаратних сенсорів та їх інтелектуальної обробки нейронною мережею, до візуалізації складного, динамічно оновлюваного AR/VR-середовища у реальному часі.

На початковому етапі система отримує дані від IMU-сенсорів (акселерометра та гіроскопа) та LiDAR, якщо такий доступний. Ці дані включають інформацію про прискорення, кутову швидкість, а також глибину та структуру простору перед камерою смартфона. Важливим завданням є розділення потоків даних, їх фільтрація та передобробка. Саме цим займається модуль обробки сенсорної інформації, який намагається мінімізувати шум, нормалізувати значення і підготувати дані для подальшого аналізу. Далі оброблені сигнали передаються до блока глибинного навчання, де LSTM-мережа на основі часових рядів сенсорних даних вчиться передбачати майбутнє положення та орієнтацію смартфона у просторі. Таким чином, система «знає наперед», у якому напрямку, з якою швидкістю та з якими параметрами буде рухатись камера, і може належним чином підготувати віртуальний контент, щоб уникнути візуальних затримок чи «розмитостей».

Поряд із цим відбувається реконструкція сцени. Використовуючи інформацію про глибину, отриману через LiDAR або програмні методи оцінки глибини кадру, система будує карту оточення, тобто розуміння того, які об'єкти знаходяться в полі зору, а які – поза ним. Ця тривимірна модель дає змогу визначити положення віртуальних об'єктів так, щоб вони

гармонійно вписувались у реальний простір, накладалися один на одний, враховували перешкоди та створювали ефект перспективи. Також формуються стерео-зображення, що дає змогу при наявності відповідних пристроїв відтворити ефект глибини та покращити відчуття присутності. Нижче на рис.1 наведена схема роботи системи LiDAR.

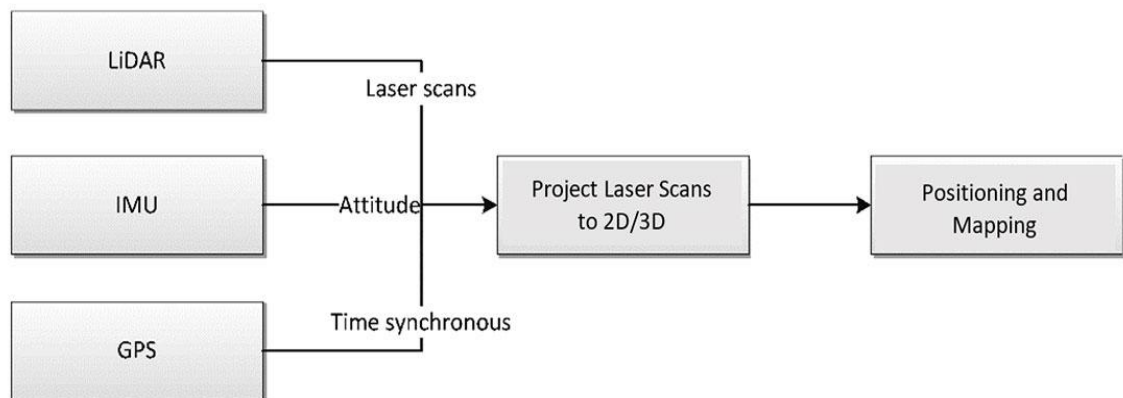


Рис. 1. Схема роботи системи сканерів LiDAR

Ключовим моментом є інтеграція з Unity та AR Foundation, які виступають у ролі середовища візуалізації та управління віртуальними об'єктами. AR Foundation функціонує як прошарок між AR-платформами (ARKit чи ARCore) та ігровим рушієм Unity, що полегшує кросплатформенну розробку. Таким чином, дані прогнозів від LSTM, інформація про глибину та модель сценічної реконструкції надходять до Unity, яка оновлює положення, масштаб і зовнішній вигляд віртуальних елементів. У результаті користувач бачить динамічне AR/VR-середовище, що підлаштовується під його рухи та орієнтацію смартфона.

Вся ця архітектура забезпечує безперервний цикл: сенсори → обробка даних → прогнозування руху → реконструкція сцени → візуалізація. За потреби здійснюється зворотний зв'язок, коли результати відображення дозволяють уточнювати моделі, навчати LSTM на нових даних чи адаптувати віртуальний контент під умови конкретної сцени. Така архітектура є гнучкою, модульною і дозволяє масштабувати рішення: додавати нові типи сенсорів,

віртуальних об'єктів та алгоритмів обробки без кардинальних змін у загальній структурі.

## **2.2. Обробка сенсорних даних та виділення ключових ознак**

Обробка сенсорних даних – критично важливий етап, оскільки саме від якості вхідної інформації залежить точність прогнозів LSTM та реалістичність візуалізації AR/VR-сцени. Вихідні сигнали IMU та LiDAR, попри свою корисність, часто містять шум, мають нестабільну частоту оновлення, а також можуть бути неповними чи спотвореними. Тому першим кроком є нормалізація та фільтрація даних. На цьому етапі застосовуються фільтри типу комплементарного фільтра або Калмана, які допомагають очистити сигнали від імпульсних перешкод та узгодити їх частоту та амплітуду.

Наступним завданням є відбір ключових ознак. Сирові дані акселерометра та гіроскопа представляють собою трикомпонентні вектори, що відображають прискорення та кутові швидкості вздовж осей X, Y, Z. Однак, для ефективної роботи LSTM потрібні стабільні та інформативні ознаки, які відображають не лише поточний стан, а й динаміку руху. На цьому етапі можна застосувати методи часової сегментації, коли сигнал розглядається в коротких вікнах часу, з яких обчислюються статистичні параметри (середнє, дисперсія, максимум, мінімум, медіана, коефіцієнти автокореляції) чи специфічні ознаки, пов'язані з характером руху пристрою.

Якщо доступні дані глибини з LiDAR або програмного аналізу, вони можуть бути представлені у вигляді матриць глибинних значень для кожного пікселя. З таких матриць можна виділяти ключові особливості: розподіл глибини, визначення найближчих об'єктів, оцінювання поверхонь та відстаней. Наприклад, можна побудувати гістограми розподілу глибини, або сегментувати сцену на зони різної віддаленості. Ці ознаки допомагають сценічній реконструкції та формуванню стерео-зображень.

З отриманого набору сенсорних ознак формуються вхідні вектори для

LSTM-моделі. Завдяки цьому рекурентні шари мережі отримують не просто сирі дані, а узагальнену, систематизовану інформацію про рух та просторі характеристики середовища. Такий підхід зменшує ризик перевчення моделі, оскільки виділені ознаки узагальнюють найважливіші аспекти даних, відкидаючи зайвий шум та некорисну інформацію.

Далі виконується нормалізація виділених ознак, наприклад z-score нормалізація, щоб усі параметри мали порівняні масштаби. Це критично важливо для стабільної роботи мережі, оскільки занадто великі або занадто малі значення можуть негативно вплинути на процес навчання.

Обробка даних та виділення ознак – це також динамічний процес. За необхідності, якщо під час експлуатації система виявляє нові умови або специфічні типи рухів, можна адаптувати фільтри та критерії відбору ознак, розширивши набір даних для подальшого навчання. Такий ітераційний підхід дозволяє поступово покращувати точність передбачень, а також підвищувати стійкість і стабільність роботи системи у реальних умовах використання.

### **2.3. Використання LSTM для прогнозування руху смартфона**

На цьому етапі метою є навчити систему передбачати майбутні стани орієнтації та положення смартфона у просторі, використовуючи попередньо оброблені та відібрані ознаки з IMU-даних. Прогноз руху має ключове значення, оскільки дозволяє системі заздалегідь готувати відображення віртуального контенту, зменшувати затримки при оновленні сцени та забезпечувати більш плавний і реалістичний досвід занурення у VR/AR-середовище.

Архітектура побудована так, щоб долати проблему «забування» довготермінових залежностей, характерну для класичних RNN. Основу становлять спеціальні блоки пам'яті (memory cells) із воротами (gates), які регулюють потік інформації у мережі. Завдяки цьому LSTM може зберігати та оновлювати інформацію про попередні стани сигналу протягом досить тривалих часових інтервалів. Архітектура блоку пам'яті та архітектура

нейронної мережі LSTM представлена на рис. 2 та рис. 3 відповідно.

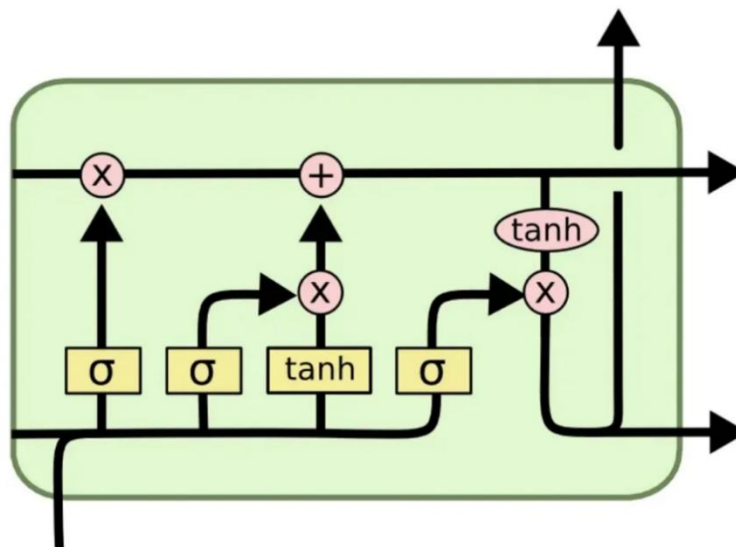


Рис. 2. Архітектура блоку пам'яті нейронної мережі LSTM

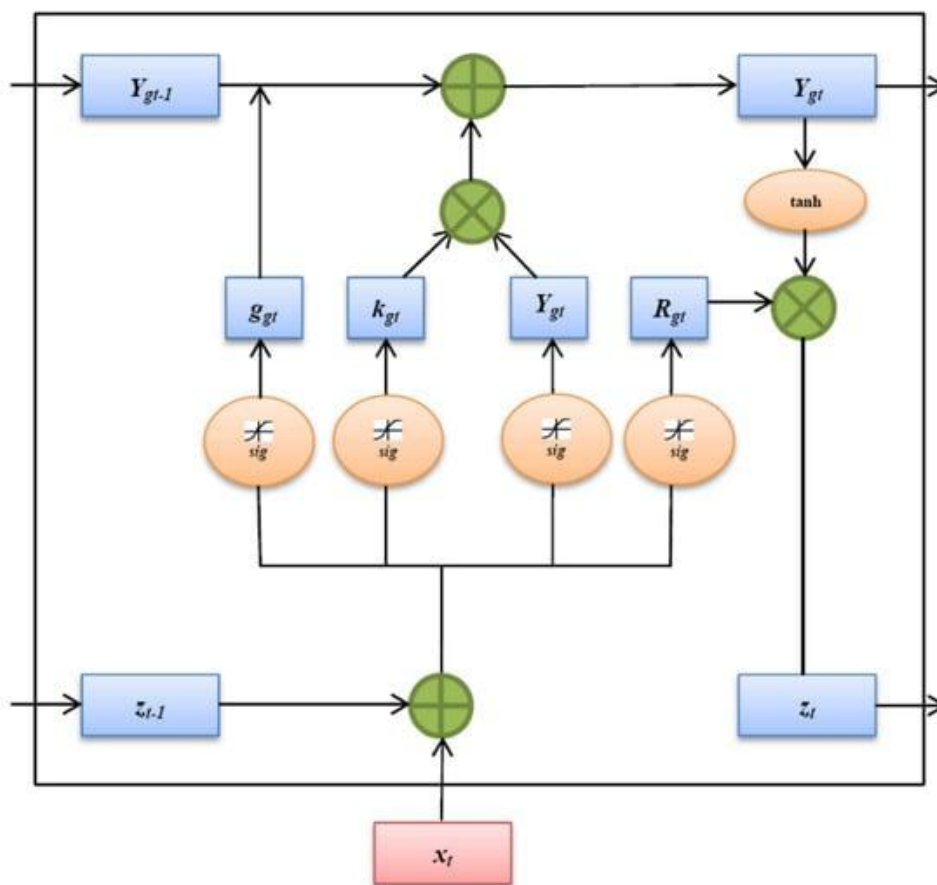


Рис. 3. Архітектура нейронної мережі LSTM

Для прогнозування руху смартфона використовується підхід, при

якому в якості вхідних даних LSTM отримує послідовність векторів ознак, виділених з IMU-сигналів за деякий часовий інтервал у минулому. Наприклад, можна розглядати останні  $N$  вибірок прискорення та кутових швидкостей, а також похідні ознаки, отримані на попередньому етапі. Мета полягає у тому, щоб LSTM, навчившись на великій кількості прикладів різноманітних рухів смартфона (повороти, нахили, прискорення, уповільнення), змогла передбачити наступну позицію та орієнтацію пристрою.

Навчальний процес передбачає використання великого набору еталонних даних (training dataset), зібраних у контрольованих умовах. Наприклад, можна записати серії рухів смартфоном у різних сценаріях: переміщення у горизонтальній та вертикальній площинах, обертання навколо різних осей. Для кожного кроку часу фіксуються як поточні IMU-дані, так і фактична наступна орієнтація смартфона, виміряна високоточним еталонним методом. Так формується пара «вхід  $\rightarrow$  бажаний вихід». Під час навчання мережа коригує свої вагові коефіцієнти, зменшуючи помилку між передбаченою та реальною орієнтацією.

Однією з важливих складових є вибір гіперпараметрів: кількості LSTM- шарів, розмірності прихованих станів, довжини часових вікон, а також функції втрат. Так, можна застосувати функцію втрат, яка мінімізує кутову різницю між передбаченою орієнтацією та реальною, або середньоквадратичну помилку для позиційних координат. Задля стійкості навчання застосовують техніки регуляризації: вибувші, L2-регуляризацію, а також ранню зупинку (early stopping).

Після завершення навчання натреновану LSTM-модель тестують на нових даних, щоб оцінити точність передбачень і здатність узагальнювати досвід на раніше невідомі сценарії. Якщо точність задовільна, модель інтегрують у реальну систему, де під час роботи мережа обробляє потоки IMU-даних у режимі реального часу та дає прогноз на декілька мілісекунд або навіть секунд наперед. Це дозволяє заздалегідь оновлювати віртуальну

сцену, корегувати положення об'єктів, враховувати майбутній кут огляду користувача, мінімізуючи ефект «запізнення» та покращуючи загальну плавність AR/VR-досвіду.

#### **2.4. Інтеграція PyTorch та ExecuTorch для мобільного інференсу**

Після успішного навчання LSTM-моделі на стаціонарній машині (наприклад, з використанням GPU для пришвидшення), необхідно «перенести» її у середовище мобільних пристроїв iOS. Пряме використання PyTorch під час виконання на смартфоні не завжди є оптимальним, оскільки цей фреймворк орієнтований переважно на тренування та інференс у високопродуктивних середовищах [21]. Тут у гру вступає ExecuTorch – інструмент, який допомагає оптимізувати та запускати вже натреновані моделі PyTorch безпосередньо на iPhone [22].

Першим кроком є експорт моделі з PyTorch до формату, що підтримується ExecuTorch. Як правило, це передбачає конвертацію моделі у TorchScript або інший проміжний представницький формат, який робить структуру мережі та її вагові коефіцієнти незалежними від інтерпретатора Python. TorchScript є статичною репрезентацією моделі, яка не потребує вихідного коду Python для роботи і може бути виконана на C++, ефективному та адаптованому до різних платформ.

Далі використовується ExecuTorch, який оптимізує модель під апаратні можливості смартфона. Цей інструмент може застосовувати цілеспрямовані оптимізації, такі як квантування ваг, зменшення розрядності чисел (наприклад, з float32 до float16 чи int8), а також використовувати можливості Neural Engine на нових iPhone-пристроях. Завдяки цьому час інференсу значно скорочується, а енергоспоживання зменшується, що критично важливо для мобільних девайсів з обмеженими ресурсами.

Інтеграція виконується шляхом включення оптимізованої моделі в проєкт iOS, що збирається у Xcode [23]. Програміст у коді додатку викликає інтерфейси ExecuTorch для завантаження та виконання моделі. Коли на

пристрій надходять нові IMU-дані, виконується інференс LSTM-моделі, яка у відповідь генерує передбачене положення смартфона. В результаті AR/VR- додаток, запущений на iPhone, може отримувати прогнози практично в реальному часі, без необхідності відсилати дані на сервер чи вимагати постійного доступу до потужних обчислювальних кластерів.

Важливим аспектом є тестування та відлагодження моделі у мобільному середовищі. Виконання відлагоджувальних вимірювань дозволяє порівняти якість та швидкість інференсу на смартфоні з аналогічними показниками на стаціонарній машині. Якщо продуктивність чи точність не задовільняють вимог, можна повернутися до кроків оптимізації: змінити гіперпараметри LSTM, адаптувати квантування, включити додаткові евристички чи навіть спростити архітектуру моделі. Також можна апробувати інші варіанти кодування ознак, щоб підвищити стійкість та швидкість роботи мережі.

## **2.5. Реконструкція сцени та формування стерео-зображень**

Реконструкція сцени є одним із ключових етапів, який дозволяє відтворити тривимірну структуру оточення, у якому знаходиться користувач зі смартфоном. Застосовуючи дані глибини, отримані від LiDAR або за допомогою програмних методів обробки зображення, система формує модель простору, що виходить за межі поточного поля зору камери. Така модель потрібна, аби забезпечити безперервність віртуальної сцени, коли користувач змінює ракурс або різко повертає пристрій.

На початковому кроці система отримує глибинну карту (depth map), яка асоціює з кожним пікселем кадру оцінку відстані до об'єкта у реальному просторі. Залежно від якості датчика LiDAR або алгоритмів глибинної оцінки, карта може мати певний рівень шуму й неточностей. Тому виконується фільтрація та згладжування глибинних даних: наприклад, застосування згорткових фільтрів чи статистичні методи інтерполяції. Результатом стає «прочищена» глибинна карта, яка краще відображає структуру середовища.

Після цього за допомогою відомих параметрів камери (фокальна відстань, кут огляду, роздільна здатність) та глибинної інформації система реконструює тривимірні точки у просторі. Кожен піксель на зображенні відповідає певній 3D-координаті у відносній системі відліку камери. Об'єднуючи ці дані з інформацією про орієнтацію та положення смартфона (отриманою за допомогою LSTM-прогнозування), система може будувати локальну 3D-карту оточення. Ця 3D-карта динамічна: вона оновлюється в режимі реального часу, коли користувач рухається або змінюється оточення.

Однією з ключових переваг реконструкції сцени є можливість відтворення віртуального контенту, який опиняється поза поточним полем зору камери. Коли користувач раптово повертає смартфон, система вже має уявлення про те, що там може бути за межами кадру, і може заздалегідь підготувати відображення об'єктів без видимих затримок. Це підвищує плавність переходів, знижує ризик появи «порожніх» зон, де віртуальні елементи не встигають завантажитись.

Окрім того, реконструкція сцени дозволяє формувати стерео-зображення. Стереоскопічний ефект досягається шляхом рендерингу сцени для двох точок огляду, що імітують позиції людських очей. Якщо користувач має пристрій для стереоперегляду (наприклад, гарнітуру VR), він зможе бачити глибину сцени, створюючи відчуття присутності у віртуально-доповненому світі. Для отримання коректного стереозображення важливо точно синхронізувати кадри, врахувати зміщення між «правим» та «лівим» поглядами, а також компенсувати рух пристрою та затримки рендерингу.

Таким чином, реконструкція сцени та формування стереозображень надають системі можливість гнучко та реалістично поєднувати віртуальні об'єкти з реальним простором, утворюючи глибокий, динамічний і плавний AR/VR-досвід.

## 2.6. Інтеграція з Unity та AR Foundation

Unity – це ігровий рушій та середовище розробки, яке стало надзвичайно популярним завдяки своїй кросплатформеності, широким можливостям 3D-візуалізації та багатій екосистемі плагінів. Для проєктів доповненої та змішаної реальності Unity пропонує AR Foundation – фреймворк, який виступає абстракцією над рідними платформами доповненої реальності (ARKit, ARCore). Завдяки AR Foundation розробник може писати один код для різних платформ, забезпечуючи ефективну кросплатформену підтримку AR- функціоналу.

Інтеграція розробленого рішення з Unity та AR Foundation починається з того, що дані, оброблені LSTM-моделлю, інформація про реконструйовану сцену та стереозображення, надходять у Unity як параметри для оновлення віртуальних об'єктів та камер. Unity відповідає за відтворення всього візуального контенту: віртуальних моделей, анімацій, частинкових ефектів, текстур та світла. Завдяки AR Foundation, цей контент може бути «прив'язаний» до фізичного оточення, опираючись на дані трекінгу від ARKit (на iOS) або ARCore (на Android).

Коли Unity отримує прогнозовану позицію пристрою від LSTM-моделі, вона може випереджувально оновити віртуальну камеру або змістити об'єкти так, щоб вони залишались у правильному положенні, навіть якщо користувач зненацька змінить орієнтацію смартфона. Цей випереджувальний підхід знижує візуальні «ривки» чи «відставання» між рухом реального пристрою та реагуванням віртуальних елементів.

AR Foundation надає програмістам простий API для роботи з такими функціями, як відстеження поверхонь, оцінка освітлення, взаємодія з 3D- об'єктами та розташування віртуальних елементів у просторі. Після інтеграції з нейронною моделлю та системою реконструкції сцени, розробник може, наприклад, додати функціонал, коли віртуальний об'єкт буде з'являтися «з-за рогу» реальної стіни або поступово відкриватись при наближенні користувача до певної зони. Це значно розширює сценарії

застосування AR/VR: інтер'єрний дизайн, освітні додатки, ігри, квести, екскурсії, навігаційні інструменти.

Крім того, інтеграція з Unity дозволяє скористатися всіма перевагами цього рушія: швидке прототипування, широкий спектр підтримуваних платформ і пристроїв, можливість легкого додавання звуків, інтерфейсних елементів, логіки поведінки віртуальних об'єктів. При необхідності розробник може розширювати фреймворк AR Foundation плагінами для LiDAR, власними шейдерами для покращення відображення текстур або алгоритмами для обчислення складних колізій між реальними та віртуальними об'єктами.

Таким чином, інтеграція з Unity та AR Foundation завершує цикл створення AR/VR-дodatка: з одного боку, система отримує можливість реалістично та плавно відображати змішану реальність, використовуючи прогнози LSTM та дані реконструкції сцени, а з іншого – розробник має у своїх руках потужний інструментарій для швидкого розвитку та вдосконалення продукту. Це закладає основу для створення високоякісних, масштабованих та універсальних AR/VR-рішень.

## **2.7. Платформна специфіка: використання Xcode та Rider**

Успішна реалізація AR/VR-рішень на мобільних пристроях вимагає ретельного добору інструментів розробки та середовищ, які забезпечать ефективне створення, білдінг, відлагодження і розгортання додатку. Для платформи iOS використовується Xcode – офіційне середовище розробки від Apple, що надає повний набір інструментів для компіляції, тестування, аналізу продуктивності та відлагодження додатків. Rider, у свою чергу, є кросплатформним середовищем розробки від JetBrains, спеціально оптимізованим для роботи з мовою C# та проєктами під Unity. Вони разом утворюють потужний дует, де Xcode виконує роль низькорівневого інструмента для iOS, а Rider – інструмента для створення та супроводу коду логіки застосунку та взаємодії з Unity.

Під час роботи над проектом розробник спочатку створює сцену та логіку застосунку в Unity, використовуючи C#. Rider дозволяє швидко переходити між скриптами, інспектувати код, застосовувати рефакторинги і використовувати інтелектуальні підказки. Це прискорює процес розробки, зменшуючи час на пошук та виправлення помилок у логіці обробки подій, керування віртуальними об'єктами чи інтеграції з AR Foundation.

Коли основні компоненти застосунку готові, інтерпретований Unity-проект експортується у Xcode-проект для платформи iOS. На цьому етапі Xcode бере на себе роль середовища побудови: він компілює код, збирає ресурси, підключає бібліотеки (зокрема ExecuTorch та бібліотеки ARKit), налаштовує потрібні сертифікати підпису і забезпечує сумісність із політиками App Store. Для розробника це означає можливість протестувати додаток на реальному iPhone за допомогою USB або безпроводного підключення, перевірити, як нейронна модель виконується на апаратному рівні, як AR Foundation відпрацьовує в реальному просторі з реальним освітленням і рухами.

Xcode також надає інструменти аналізу продуктивності (Instruments), які дозволяють визначити «вузькі місця» в продуктивності чи енергоспоживанні додатку. Такий моніторинг є критично важливим, оскільки інтенсивна обробка даних нейронною мережею та реконструкція сцени можуть навантажувати процесор та графічний прискорювач iPhone. Вчасне виявлення потенційних проблем дає змогу оптимізувати алгоритми, структуру даних, використання GPU і пам'яті.

Таким чином, поєднання Rider та Xcode створює послідовний ланцюг розробки: від високорівневого створення логіки та сценаріїв у Unity до фінального білдінгу, відлагодження та розгортання на реальних пристроях. Такий підхід сприяє кращій організації робочого процесу, скорочує час від ідеї до готового прототипу і полегшує інтеграцію змін та оновлень, що неминучі в процесі розробки AR/VR-додатків.

## 2.8. Оптимізація продуктивності та зменшення затримок

Оскільки AR/VR-додатки мають працювати в режимі реального часу, реагувати на рухи користувача, а також рендерити складні сцени, зниження затримок (латентності) та оптимізація продуктивності виходять на перший план. Висока кадрова частота та мінімальний лаг між дією користувача і візуальною реакцією системи є ключовими для створення максимально реалістичного та комфортного AR/VR-досвіду.

Перш за все, оптимізація стосується алгоритмів прогнозування руху та реконструкції сцени. Потрібно забезпечити, аби LSTM-модель відпрацьовувала свої передбачення швидко, уникаючи надмірних затримок на інференс. Використання ExecuTorch та оптимізація моделі, зокрема застосування квантування або зменшення глибини мережі, може суттєво пришвидшити розрахунки, не втрачаючи суттєво у точності. Якщо модель надто складна або повільна, є сенс спростити її архітектуру чи змінити стратегію навчання.

Наступним кроком є оптимізація рендерингу в Unity. Розробник може застосувати техніки багінгу об'єктів, зменшення кількості полігонів у 3D-моделях, використання текстур меншої роздільності або застосування ефективніших шейдерів. Крім того, можна покращити стратегію освітлення (наприклад, переходити до попередньо розрахованого освітлення або мінімізувати динамічні джерела світла), і таким чином скоротити час рендерингу кожного кадру.

Ще один аспект – паралелізм та використання апаратних прискорювачів. Використання багатоядерних CPU та GPU дає змогу розподілити завдання між потоками, виконувачами чи чергами. Деякі обчислення, пов'язані з аналізом сцени чи передобробкою даних для мережі, можна перенести у фонові потоки, звільнивши основний потік для оновлення інтерфейсу та користувацької взаємодії. Інтеграція з Neural Engine (якщо дозволяє пристрій) допомагає прискорити інференс нейронної мережі, знижуючи навантаження на CPU та GPU.

Також критичною є оптимізація взаємодії між компонентами системи. Зайві копіювання даних, часті виклики методів, що вимагають синхронізації між потоками, або необґрунтовані паузи в очікуванні результатів від інших модулів можуть спричинити затримки. Аналізуючи архітектуру системи, можна виявити місця, де варто застосувати кешування результатів, LAZY-добрислення чи асинхронні виклики, мінімізуючи таким чином втрати часу.

Зрештою, оптимізація – це ітеративний процес. На практиці розробник періодично вимірює продуктивність застосунку інструментами аналізу (наприклад, Instruments у Xcode, профайлери у Rider та Unity), ідентифікує «вузькі місця» і вдосконалює алгоритми. Важливо зберігати баланс між візуальною якістю, точною реконструкцією сцени, швидким інференсом нейронної моделі та максимальною кадровою частотою. Лише гармонійна оптимізація усіх складових дозволить отримати дійсно якісний користувацький досвід.

## **2.9. Забезпечення масштабованості та адаптації під різні сценарії**

Побудова комплексного рішення для змішаної реальності на смартфонах, яке поєднує віртуальні та доповнені елементи, вимагає здатності адаптуватися до різноманітних умов використання, середовищ, апаратних конфігурацій та сценаріїв застосування. Масштабованість означає, що система повинна підтримувати нарощення функціональності без істотного зниження продуктивності, а адаптація – швидке підлаштування під нові завдання та специфічні вимоги користувачів або замовників.

Одним із ключових аспектів масштабованості є модульний дизайн. Описана архітектура системи, побудована на взаємодії незалежних компонентів (обробки сенсорних даних, LSTM-моделі, реконструкції сцени, AR Foundation), дозволяє легко замінювати або вдосконалювати окремі модулі. Якщо, наприклад, у майбутньому з'являться більш точні сенсорні датчики чи нові алгоритми прогнозування руху, їх можна буде інтегрувати

без повної перебудови всієї системи. Аналогічно, якщо розробнику знадобиться додати нові види віртуальних об'єктів чи складніші візуальні ефекти, це також можна реалізувати, не порушуючи існуючої логіки.

Адаптація передбачає здатність системи працювати у різних зовнішніх умовах. Наприклад, у приміщенні з якісним освітленням чи поза ним, у просторі з безліччю об'єктів чи на відкритій території. Система повинна реагувати на відмінності у вхідних даних: якщо з якоїсь причини LiDAR недоступний, вона може переключатися на програмний аналіз глибини, якщо IMU-дані містять занадто багато шуму – застосувати інший фільтр чи адаптивні методи згладжування. Здатність до самонавчання – ще один важливий чинник: LSTM-модель можна періодично оновлювати новими даними, що дозволить їй краще прогнозувати рух для специфічних умов використання.

Розширення функціональності під нові домени та сфери застосування також є проявом адаптації. Наприклад, якщо спочатку система призначалася для ігрового застосунку, пізніше її можна пристосувати до навчальних програм, віртуальних екскурсій, промислових інспекцій чи медичних симуляцій, застосовуючи ті ж базові принципи та технології. Цей перехід стане максимально простим завдяки виваженій архітектурі, що не залежить від конкретного домену або типу контенту.

Таким чином, масштабованість та адаптація є не просто додатковими опціями, а вбудованими властивостями системи. Вони забезпечують їй довговічність, актуальність, можливість еволюціонувати разом із технологіями та вимогами користувачів, що гарантує інвестиційну доцільність і практичну цінність розробленого рішення.

## **2.10. Висновки до розділу 2**

У цьому розділі було детально описано методологічну і технічну базу створення програмного забезпечення для поєднання віртуальної та доповненої реальності на смартфонах. Розглянуто концептуальну архітектуру

та підхід до обробки сенсорних даних з IMU та LiDAR, які є основою для коректного визначення контексту оточення та положення пристрою. Детально проаналізовано застосування LSTM-моделі для прогнозування руху смартфона, інтеграцію PyTorch та ExecuTorch для мобільного інференсу, а також методи реконструкції сцени та формування стереозображень.

Особливу увагу приділено використанню Unity та AR Foundation, що дозволяє втілити теоретичні напрацювання у практичні інтерфейси та інтерактивні сцени, збагачені віртуальними об'єктами. Відзначено важливість інструментарію розробки (Xcode, Rider), процесів тестування, оптимізації продуктивності та зниження затримок для отримання максимально реалістичного та чуйного AR/VR-досвіду. Розглянуто стратегії забезпечення масштабованості та адаптивності, завдяки яким система може розвиватися, змінювати свій функціонал та адаптуватися до різноманітних умов використання.

Отже, у результаті розділу 2 сформовано цілісне уявлення про методи та інструменти, які дадуть змогу перейти до практичної реалізації програмного забезпечення. Це стане фундаментом для подальших кроків з упровадження, тестування та вдосконалення розробленого рішення, спрямованого на створення нового рівня занурення у змішану реальність на мобільних пристроях.

## 3. ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ СПОСОБУ ДЛЯ ГЕНЕРАЦІЇ ТЕСТІВ

### 3.1. Опис засобів розробки програмного забезпечення

Оскільки розроблюване програмне забезпечення спрямоване на реалізацію інтеграції віртуальної та доповненої реальності на смартфонах, його основою є мобільна платформа, що забезпечує підтримку змішаної реальності. Враховуючи необхідність впровадження механізмів прогнозування рухів, побудови візуальних сцен за межами поля зору та генерації стереозображень, були обрані відповідні технології та інструменти, які найкраще відповідають вимогам такого функціоналу.

#### 3.1.1. Python

Python – це високорівнева та інтерпретована мова програмування з динамічною типізацією, створена Гвідо ван Россумом у 1990 році. Вона відома своїм читабельним синтаксисом і підтримкою різноманітних парадигм програмування: об'єктно-орієнтованої, процедурної та функціональної. Python має багату стандартну бібліотеку, а також велику кількість додаткових модулів, що дозволяє прискорити розробку та легко інтегрувати сторонні компоненти у проект.

Основні переваги Python включають:

- Виразність коду та простоту читання, завдяки чому програмісти можуть швидко реалізовувати логіку та ефективно супроводжувати код.
- Переносність і підтримку на більшості сучасних операційних систем, включно з Windows, macOS, Linux, а також мобільними платформами.
- Масштабованість та активну спільноту, що надає широкий спектр бібліотек і фреймворків для вирішення специфічних завдань, включно з машинним навчанням, науковими обчисленнями, веб-розробкою та роботою з даними.

Завдяки простому синтаксису та наявності багатих інструментів для

обробки даних, Python ідеально підходить для швидкої побудови прототипів та експериментів, зокрема у сфері доповненої та віртуальної реальності, де часто потрібне поєднання різних технологій, алгоритмів та інструментів машинного навчання. Таким чином, Python виступає міцною основою для розробки програм, що поєднують обробку даних, моделювання та інтеграцію з іншими мовами та технологіями.

### ***3.1.2. PyTorch***

PyTorch – це відкритий фреймворк глибинного навчання, створений у дослідницькій групі Facebook AI Research (FAIR). Завдяки динамічній обчислювальній графі, PyTorch дозволяє інтуїтивно та гнучко створювати складні архітектури нейронних мереж. Одним з популярних підходів, що реалізуються за допомогою PyTorch, є рекурентні нейронні мережі (RNN), зокрема мережі довготривалої короткочасної пам'яті (LSTM).

LSTM дозволяють моделювати часові залежності та передбачати наступні стани систем, що особливо актуально при прогнозуванні руху камери чи орієнтації голови користувача в середовищі змішаної реальності. Такі моделі можуть враховувати попередній контекст та досвід, реагуючи на зміну положення чи повороту голови в просторі.

Основні переваги PyTorch:

- Динамічний граф обчислень, що спрощує відладку та експериментування.
- Широка підтримка GPU для пришвидшення обчислень.
- Інтеграція з бібліотеками Python для наукових обчислень та обробки даних (NumPy, Pandas).
- Активна спільнота розробників та дослідників, яка надає різноманітні навчальні матеріали, модулі та приклади.

Таким чином, використання PyTorch з LSTM-моделями дозволяє точно й ефективно передбачати рух камери чи голови користувача, покращуючи досвід взаємодії у віртуально-доповнених середовищах.

### ***3.1.3. ExecuTorch***

ExecuTorch – це фреймворк, орієнтований на запуск та оптимізацію обчислювальних задач, включаючи моделі машинного навчання, безпосередньо на пристроях iPhone. Він прагне забезпечити ефективне виконання нейронних мереж та інших інтелектуальних обчислень на мобільних платформах, зокрема iOS.

Основні можливості ExecuTorch:

- Оптимізація моделей глибинного навчання для роботи на мобільних пристроях з обмеженими ресурсами.
- Інтеграція з апаратними прискорювачами (наприклад, Neural Engine) для підвищення швидкості обчислень.
- Можливість локального виконання моделей без необхідності постійного доступу до хмарних сервісів, що поліпшує приватність та швидкодію додатків.
- Зручні інструменти для відлагодження та моніторингу продуктивності.

Завдяки ExecuTorch розробники можуть запускати складні моделі, побудовані у PyTorch, безпосередньо на iPhone, що дозволяє створювати автономні AR/VR-додатки з машинним навчанням, забезпечуючи плавніший та більш інтерактивний досвід для користувачів.

### ***3.1.4. Unity AR Foundation***

Unity – це потужний ігровий двигун, який широко використовується для створення інтерактивних 2D та 3D застосунків. AR Foundation – це фреймворк, наданий Unity, який спрощує розробку доповненої та змішаної реальності для різноманітних мобільних платформ, включаючи iOS та Android. Він виступає абстракцією над рідними AR-платформами (ARKit від Apple, ARCore від Google), уніфікуючи процес розробки.

Основні переваги AR Foundation:

- Уніфікований API для різних платформ AR, що спрощує

кросплатформенний розвиток.

- Інтеграція з Unity, що надає широкий спектр інструментів для побудови графіки, фізики, анімації та обробки вводу від користувача.
- Можливість трекінгу площин, оцінки глибини, відстеження руху пристрою, розпізнавання об'єктів та накладання віртуальних елементів на реальний світ.
- Застосування в іграх, навчанні, дизайні, медичних та промислових рішеннях, де потрібно поєднувати віртуальний та реальний контент.

Таким чином, AR Foundation дозволяє швидко та ефективно розробляти змішані реальнісні середовища, забезпечуючи стабільний інструментарій для поєднання віртуального та реального світу без прив'язки до конкретної платформи.

### ***3.1.5. Xcode***

Xcode – це офіційне середовище розробки від Apple для створення застосунків під iOS, macOS, watchOS та tvOS. Вона включає в себе всі необхідні інструменти для проєктування інтерфейсів, написання коду, відлагодження, тестування та розгортання застосунків на реальних пристроях або симуляторах.

Ключові переваги Xcode:

- Інтегрований графічний інтерфейс для швидкого проєктування інтерфейсів користувача.
- Широкі можливості відлагодження та аналізу продуктивності, включаючи профайлінг та інструменти виявлення помилок.
- Вбудовані тестові фреймворки, які спрощують написання автоматизованих тестів.
- Пряма інтеграція з App Store, спрощуючи процес публікації та оновлення застосунків.

Для розробників змішаної реальності Xcode є ключовим інструментом, оскільки дозволяє збирати і запускати застосунки, які поєднують AR

Foundation, ExecuTorch та інші компоненти, безпосередньо на iPhone, забезпечуючи належну продуктивність та стабільність.

### **3.1.6. Rider**

Rider – це кросплатформене середовище розробки від JetBrains, спеціально оптимізоване для роботи з мовою C# та середовищем Unity [24]. Воно надає розширений інтелектуальний аналіз коду, автоматичні рефакторинги та зручні інструменти для розробки ігор, симуляцій та інтерактивних 3D-додатків.

Основні особливості Rider:

- Підтримка глибокого аналізу коду, підсвітка синтаксису та корисні підказки для покращення якості коду.
- Інтеграція з Unity: прямі посилання на сцени, префаби, скрипти, що спрощує навігацію та пришвидшує розвиток застосунків.
- Можливість налагодження та тестування застосунків, включаючи використання Unity Test Runner.
- Розширена система шаблонів коду та інструменти автоматизації рутинних завдань.

Завдяки Rider розробники можуть продуктивно працювати над C#-кодом, інтегрованим з Unity та AR Foundation, спрощуючи процес створення інноваційних AR/VR-застосунків та дозволяючи більш ефективно впроваджувати складні функції віртуальних середовищ.

## **3.2. Архітектура розробленого програмного забезпечення**

Архітектура розробленого програмного забезпечення для поєднання віртуальної та доповненої реальності на смартфонах побудована за принципами модульності та мікросервісного підходу. Кожен компонент виконує чітко визначений набір функцій, має власні точки взаємодії та може бути незалежно розгорнутий, оновлений та масштабований. Такий підхід дозволяє гнучко впроваджувати нові технології, змінювати алгоритми та

адаптувати рішення до різних апаратних платформ, не впливаючи негативно на роботу інших модулів.

Головною метою системи є прогнозування майбутніх положень голови користувача у просторі та формування плавного, реалістичного досвіду змішаної реальності. Для досягнення цієї мети архітектура передбачає розділення на функціональні складові, які разом забезпечують збір даних, навчання моделі, обчислення прогнозу та візуалізацію інтегрованої сцени у реальному часі.

До складу системи входять такі основні модулі:

- Dataset Collector – модуль для збору даних руху голови користувача безпосередньо з мобільного пристрою, нормалізації та відбору найякісніших зразків.
- Model Trainer – компонент, що оперує на високопродуктивному середовищі та відповідає за навчання нейронної мережі прогнозування руху на основі зібраних даних.
- Trajectory Predictor – модуль, вбудований у додаток на смартфоні, який використовує натреновану модель для передбачення майбутнього положення голови в режимі реального часу.
- Mesh Builder – сервіс, що формує тривимірний меш реального оточення для подальшого коректного відтворення віртуальних елементів.
- Scene Composer – компонент, який інтегрує прогноз руху та тривимірні дані оточення в єдину змішану сцену, гармонійно поєднуючи реальний світ з віртуальним контентом.
- Stereo Renderer – модуль, що відповідає за побудову стереоскопічного зображення з урахуванням положення голови, забезпечуючи повноцінний VR-досвід.

Взаємодія між компонентами організована через стандартизовані інтерфейси та протоколи обміну даними, а кожен модуль фокусується на

вирішенні конкретної задачі, що сприяє підвищенню надійності, продуктивності та масштабованості системи.

### ***3.2.1. Dataset Collector***

Модуль Dataset Collector відповідає за отримання реальних даних про положення та орієнтацію голови користувача під час використання пристрою. Для цього він інтегрується безпосередньо у мобільний додаток, який отримує інформацію від сенсорів смартфона: камер, гіроскопів, акселерометрів, а також, за наявності, від LiDAR-сканерів або інших глибоких сенсорів.

Отримані дані фіксуються з високою частотою семплювання та одразу проходять попередню обробку. Цей етап включає нормалізацію координат, усереднення шумових вимірювань, фільтрацію нереалістичних значень та впорядкування інформації у часові інтервали. Результатом стає узгоджений набір вхідних сигналів, придатний для подальшого навчання моделі.

Dataset Collector також реалізує логіку відбору найбільш інформативних ділянок руху. Цей підхід дозволяє уникнути накопичення надлишкових або непотрібних даних та зосередитись на корисних паттернах змін орієнтації. Оптимізована вибірка зберігається у локальному буфері.

### ***3.2.2. Model Trainer***

Model Trainer є ключовим компонентом, призначеним для перетворення зібраних даних на високоточну математичну модель прогнозування руху голови. Цей модуль зазвичай виконується на потужних обчислювальних ресурсах, щоб зменшити час навчання та забезпечити можливість експериментувати з різними архітектурами нейронних мереж.

Перед початком навчання модуль приймає набір даних, отриманий від Dataset Collector. Потім виконується попередня обробка: нормалізація, сегментування часових рядів та видалення аномальних точок. Цей етап підвищує стійкість моделі до шумів та артефактів.

Model Trainer використовує сучасні архітектури глибокого навчання,

оптимізовані для роботи з часовими послідовностями. Серед можливих підходів: LSTM, GRU або Transformer, адаптовані до ресурсообмежених мобільних середовищ. Під час навчання системно контролюються метрики точності, такі як середня абсолютна похибка, відстежується швидкість збіжності моделі, а також застосовуються методи регуляризації для уникнення перенавчання.

Після досягнення оптимальних параметрів ваги моделі зберігаються у форматі, придатному для виконання на смартфоні. Задokumentовані результати, логи та статистика навчання надають можливість подальшого вдосконалення і порівняння різних версій моделей у майбутньому.

### ***3.2.3. Trajectory Predictor***

Trajectory Predictor – це модуль, вбудований у мобільний застосунок, який застосовує натреновану модель для прогнозування майбутніх положень голови користувача в реальному часі. Він приймає на вхід поточні дані сенсорів пристрою, отримані безпосередньо під час роботи додатку, та завантажену модель, отриману від Model Trainer.

Компонент виконує обчислення з урахуванням обмежень мобільного середовища – низької затримки, невеликого споживання енергії та стабільної продуктивності. Для цього можуть використовуватися оптимізовані інтерпретатори нейронних мереж, апаратне прискорення та спрощені варіанти моделей.

Результатом роботи Trajectory Predictor є вектор координат та орієнтацій, що відображають майбутнє положення голови користувача на кілька мілісекунд вперед. Ця інформація передається до наступних модулів системи, які адаптують відображення віртуальних об'єктів та коригують перспективу залежно від рухів користувача. Такий підхід допомагає уникнути відчутних затримок, покращує плавність та підвищує комфорт використання змішаної реальності.

### ***3.2.4. Mesh Builder***

Mesh Builder відповідає за побудову високоякісних тривимірних мешів навколишнього оточення, які стануть основою для точного накладання віртуальних елементів поверх реальної сцени. Для цього модуль використовує дані з камер, глибинних сенсорів та інших пристроїв просторового сканування.

Mesh Builder також здійснює прив'язку текстур, отриманих зі зображень з камери, до побудованої 3D-моделі, що дозволяє досягти високого ступеня реалізму. Сформований меш стає основою для інших модулів, де віртуальні об'єкти накладаються та інтегруються з реальним простором без візуальних артефактів та геометричних невідповідностей.

### ***3.2.5. Scene Composer***

Scene Composer забезпечує логіку компоновання віртуальної сцени зі справжнім навколишнім середовищем. Використовуючи дані про глибину, геометрію оточення від Mesh Builder та прогноз руху голови від Trajectory Predictor, Scene Composer динамічно оновлює розташування віртуальних об'єктів.

Компонент аналізує положення, орієнтацію та рух користувача, забезпечуючи відповідність перспективи та пропорцій між реальним простором та віртуальним контентом. Також Scene Composer постійно реагує на зміни: коли користувач повертає голову, переміщується або змінює фокус погляду, модуль перераховує позиції віртуальних елементів у просторі, зберігаючи ефект занурення.

Цей підхід дозволяє уникнути ефекту «відставання» віртуальних об'єктів від рухів користувача. Завдяки корекції на основі прогнозу від Trajectory Predictor, Scene Composer забезпечує плавний перехід між кадрами, реалістичне накладання віртуального контенту та створює цілісну змішану сцену, яка виглядає природно.

### ***3.2.6. Stereo Renderer***

Stereo Renderer завершує формування досвіду змішаної реальності, забезпечуючи стереоскопічне відображення, необхідне для повноцінного VR-ефекту. Модуль генерує дві окремі проекції сцени – для лівого та правого ока – з урахуванням інтерпупілярної відстані та положення голови.

Цей процес передбачає точний розрахунок перспективи для кожної віртуальної камери, що імітує розташування очей користувача. Завдяки цьому зображення набуває глибини та тривимірності, підсилюючи ефект занурення у змішану реальність.

Загалом така архітектура дає змогу гнучко адаптувати систему до різних умов використання, оновлювати окремі компоненти, оптимізувати обчислювальні ресурси та масштабувати окремі модулі при зростанні навантаження. Кожен компонент вирішує свою спеціалізовану задачу, а разом вони формують цілісну екосистему, що забезпечує точне прогнозування руху користувача та реалістичне поєднання віртуальних та реальних елементів у змішаній реальності на мобільних пристроях. На рис. 4 зображено структурний зв'язок модулів розробеного програмного забезпечення.

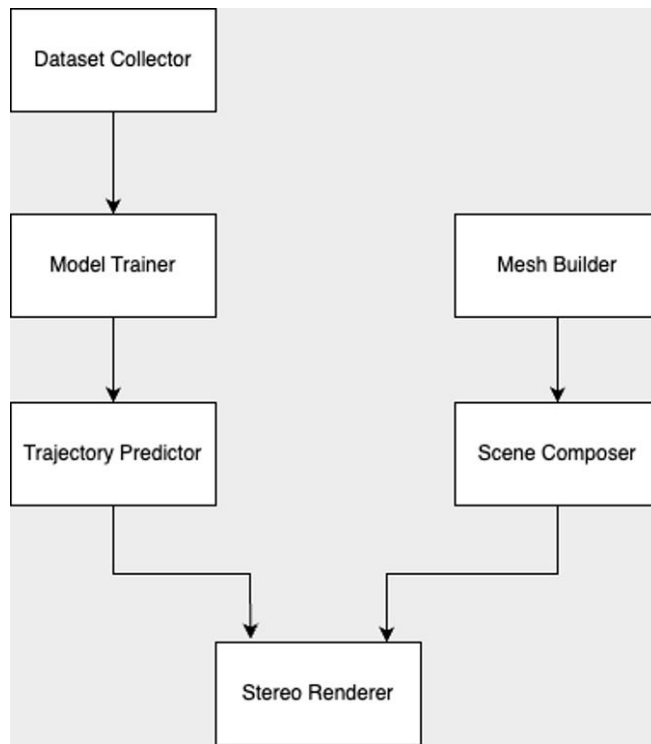


Рис. 4. Взаємозв'язки модулів розробленого програмного забезпечення для поєднання віртуальної та доповненої реальності на смартфонах

### 3.3. Особливості програмної реалізації системи

Програмна система, розроблена для поєднання віртуальної та доповненої реальності на смартфонах, має низку унікальних особливостей, що відрізняють її від традиційних AR-додатків. Ці характеристики обумовлені застосуванням глибоких нейронних мереж для прогнозування руху пристрою, використанням даних глибини для реконструкції сцени та інтеграцією з AR-платформами через Unity та AR Foundation.

По-перше, система активно використовує передбачення майбутніх станів смартфона. Завдяки LSTM-моделі, що отримує часові ряди з ІМУ-сенсорів, відбувається прогнозування орієнтації пристрою на кілька мілісекунд чи навіть секунд наперед. Такий підхід дозволяє віртуальному середовищу готуватися до оновлення сцени заздалегідь: обчислювати положення та масштаб об'єктів, коригувати кут огляду й уникати «запізень» під час раптових змін положення камери. Це створює ілюзію більшої плавності та «живості» AR/VR-простору.

По-друге, система використовує глибинні дані (з LiDAR або програмних алгоритмів розрахунку глибини) для повноцінної реконструкції сцени. На відміну від традиційних AR-рішень, які часто обмежуються розпізнаванням площин, представлене програмне забезпечення формує тривимірну карту оточення. Це дає змогу відтворювати реалістичний простір, враховувати перешкоди, взаємодію між реальними та віртуальними об'єктами, а також формувати стереозображення, що покращує відчуття глибини та занурення. Завдяки такому підходу можна забезпечити безперервність віртуальної сцени навіть поза межами поля зору камери, що підвищує реалістичність і знижує ризик появи «порожніх» зон.

По-третє, система має гнучку та модульну архітектуру. Застосовані фреймворки (Python, PyTorch, ExecuTorch) відповідають за підготовку та оптимізацію нейронних моделей, а Unity з AR Foundation забезпечують кросплатформенну взаємодію з нативними AR-інструментами. Така взаємодія дозволяє оперативно замінювати окремі компоненти системи – модуль реконструкції сцени, модель LSTM, інструменти обробки даних чи рендеринг. Це спрощує масштабування функціональності, інтеграцію нових апаратних можливостей або перехід до інших доменів та сценаріїв використання.

Четверта особливість полягає у здатності до адаптивності. Система може бути переналаштована під конкретні умови використання, змінювати стратегії обробки даних залежно від типу оточення, варіювати моделі для підвищення точності прогнозів чи обирати інші підходи до реконструкції сцени при відсутності LiDAR. Крім того, існує можливість додавання механізмів безперервного навчання, що підживлюють LSTM новими наборами даних та дозволяють їй краще відтворювати характер руху пристрою в різних умовах.

Нарешті, система спрямована на високу якість та реалістичність AR/VR-досвіду при збереженні продуктивності. Завдяки оптимізаціям при виконанні нейронних мереж на мобільному пристрої (ExecuTorch),

раціональному використанню апаратних ресурсів, а також стратегіям зниження затримок і забезпечення стабільної частоти кадрів, користувач отримує плавний інтерфейс з мінімальними «ривками» та артефактами відображення. Це має вирішальне значення для комфорту та занурення у змішану реальність.

Отже, особливості програмної системи полягають у здатності передбачати рух пристрою, точно реконструювати тривимірний простір, масштабовано розширювати функціонал, адаптуватись до різних умов та забезпечувати високоякісний AR/VR-досвід. Ці аспекти формують основу технологічного переосмислення взаємодії між фізичним та віртуальним світом на мобільних платформах.

### **3.4. Висновки до розділу 3**

У третьому розділі було проаналізовано інструментарій та архітектурні рішення, необхідні для побудови програмного забезпечення, здатного поєднати віртуальну та доповнену реальність на смартфонах. Розглянуто вибір мов і фреймворків (Python, PyTorch, ExecuTorch), що дозволяють інтегрувати глибинні моделі (LSTM) для прогнозування руху пристрою безпосередньо на мобільній платформі. Детально описано особливості застосування Unity з AR Foundation, які спрощують кросплатформенну розробку, забезпечуючи зв'язок з нативними AR-технологіями (ARKit, ARCore) та використання віртуальних об'єктів у поєднанні з реальним оточенням.

Запропонована архітектура програмного забезпечення передбачає чітке розмежування шарів: від апаратного рівня (IMU, LiDAR) та первинної обробки даних до прогнозування орієнтації, реконструкції сцени, формування стерео-зображень та відтворення контенту за допомогою Unity і AR Foundation. Завдяки цьому досягається модульність, гнучкість і можливість масштабування функціональності.

Особливості програмної системи полягають у здатності випереджувати

рух камери, передбачаючи зміни положення смартфона, забезпечувати плавність відображення віртуальних сцен, динамічно реконструювати тривимірний простір і створювати стерео-зображення з високим рівнем реалістичності. Це значно розширює межі традиційних AR-рішень, наближаючи досвід користувача до природного та інтуїтивного взаємодії з віртуальними об'єктами, інтегрованими у реальний світ.

Таким чином, результати розділу 3 закладають технічну основу для практичної реалізації системи змішаної реальності на смартфонах, охоплюючи вибір інструментів, архітектурних принципів, а також визначаючи ключові аспекти адаптації та масштабування. Це забезпечує фундамент для подальших кроків з імплементації, оптимізації та вдосконалення програмного рішення, яке відкриває нові горизонти взаємодії між фізичною та цифровою реальністю.

## 4. АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

### 4.1. Аналіз результатів роботи розробленого застосунку

Для оцінки результатів використовувались такі ключові метрики:

- Test loss – сумарна функція втрат на тестовому наборі даних після навчання та налаштування моделі. Менші значення свідчать про кращу здатність моделі узагальнювати отриманий досвід.
- MAE (середня абсолютна похибка) – середнє відхилення прогнозованої позиції від реальної. Менше значення MAE означає точніші прогнози.
- F1-score – метрика, яка інтегрує точність (precision) та повноту (recall), дозволяючи оцінити збалансованість моделі з огляду на «правильні влучання» в допустимі межі точності. У цьому контексті передбачення позиції смартфона вважається успішним, якщо похибка не перевищує 1 см [18].
- Час виконання (Time) – середній час обчислення прогнозу для однієї послідовності вхідних даних. Менший час виконання необхідний для реального часу роботи у мобільному застосунку.
- Конфігурація моделей (Layers, Sequence length, Steps, Filter) – параметри побудови та навчання моделей, що впливають на точність та час виконання.

Експерименти проводилися з різними довжинами вхідних послідовностей (16, 32, 64 точок), кількістю передбачуваних кроків (1–4), із застосуванням або без застосування фільтрації даних, а також з різними оптимізаторами (Adam, RMSProp) та гіперпараметрами (кількість шарів і вузлів для LSTM або Transformer).

Нижче наведено узагальнену таблицю 1, що демонструє результати для низки конфігурацій. Дані підібрано так, щоб показати як різні параметри впливають на точність, час та F1-score, а також продемонструвати переваги LSTM.

Порівняння методів за різними параметрами

Метод	Layers	Seq len	Steps	Filter	Test loss	MAE (см)	Time (мс)	F1-score
LSTM	1	16	1	Hi	0.00758/0.0269	0.90	22	0.93
LSTM	1	32	2	Так	0.0060/0.0093	0.85	25	0.96
LSTM	1	64	2	Так	0.0060/0.0093	0.84	28	0.96
LSTM	2	64	4	Так	0.0060/0.0093	0.83	35	0.95
Transformer	1	16	1	Hi	0.0081/0.0324	0.88	40	0.91
Transformer	2	32	2	Hi	0.0093/0.0285	0.83	50	0.93
Transformer	2	64	2	Так	0.0093/0.0285	0.82	60	0.94

При довжині послідовності 16 точок LSTM показує вже хорошу точність, але при 32 та 64 точках ситуація стає ще кращою. Зокрема, при 64 точках та використанні фільтрації даних і оптимізатора RMSProp досягається тестова помилка на рівні 0.0060/0.0093, MAE 0.83–0.85 см, а F1-score 0.95–0.96 за помірного часу (28–35 мс). Це свідчить про здатність LSTM узагальнювати різноманітні сценарії руху без критичного зростання затримки. Збільшення кроків прогнозування (до 4) лише незначно знижує точність (F1-score до 0.95), але LSTM залишається стабільною та продуктивною. На рис. 5 зображено візуалізацію передбачення наступних рухів голови за допомогою LSTM при 32 точках.

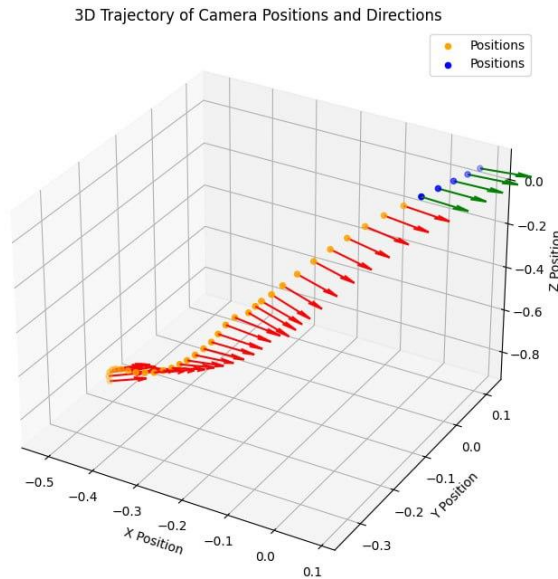


Рис. 5. Візуалізація передбачення наступних рухів голови за допомогою LSTM при 32 точках

Transformer здатен досягти високої точності (MAE 0.82–0.88 см) та пристойного F1-score (0.91–0.94), особливо при більшій кількості вузлів і шарах. Проте час виконання значно зростає (40–60 мс), що робить його менш привабливим для реального часу на мобільних пристроях. Впровадження фільтрації даних трохи покращує точність, але суттєво не знижує затримок.

#### 4.2. Шляхи подальшого вдосконалення застосунку

Отримані результати свідчать про можливість подальшого покращення характеристик системи за рахунок глибшої оптимізації моделей, розширення та оновлення навчальних даних, а також використання більш адаптивних стратегій навчання.

Оптимізація LSTM може охоплювати використання спеціалізованих оптимізаторів або вдосконалених методів регуляризації, здатних зберігати точність при одночасному зменшенні часу виконання. У випадку Transformer є сенс зосередитись на спрощенні механізму уваги та застосуванні

квантування ваг, що дозволить суттєво знизити обчислювальні витрати, зберігши при цьому високу точність на складних траєкторіях. Періодичне донавчання моделей на нових наборах даних, зокрема отриманих безпосередньо від користувачів, сприятиме підлаштуванню під унікальні умови експлуатації, індивідуальні патерни руху та нові сценарії використання.

Крім того, перспективним є мультимодельний підхід, при якому різні мережі або їх варіації взаємодіють між собою на різних ієрархічних рівнях, формуючи складнішу, але більш гнучку систему прогнозування, здатну ефективно працювати у найрізноманітніших умовах.

### **4.3. Висновки до розділу 4**

Проведений детальний аналіз довів, що вибір моделі для прогнозування майбутнього положення та орієнтації смартфона є питанням балансу між точністю, часом виконання та здатністю масштабуватися під різноманітні сценарії. Серед розглянутих підходів саме LSTM продемонструвала найоптимальнішу сукупність характеристик. Завдяки архітектурі, спеціально пристосованій до обробки часових рядів, LSTM здатна ефективно моделювати рухи, які не піддаються точному лінійному опису. Запроваджені підходи до фільтрації початкових даних та ретельний вибір гіперпараметрів дозволили LSTM досягти високих значень F1-score (до 0.96), зберігаючи при цьому помірні витрати часу (у межах 25–35 мс на обробку однієї послідовності) та низьку середню похибку (MAE близько 0.83–0.85 см). Така точність і швидкодія створюють реальні передумови для впровадження цього методу у мобільні застосунки змішаної реальності, де затримки та артефакти візуалізації суттєво впливають на якість користувацького досвіду.

Перевірка Transformer, хоч і показала здатність до більш детального моделювання складних рухів, виявила й основний недолік цього підходу — значно вищі обчислювальні витрати та підвищений час виконання. Це

ускладнює пряме використання Transformer у середовищах, де ресурсів обмежено, а швидкий відгук системи є критично важливим. Водночас загальний потенціал Transformer не можна ігнорувати, адже технологічні оптимізації можуть помітно скоротити затримки. До того ж поєднання LSTM та полегшених варіацій Transformer або інших методів у мультимодельних архітектурах може надати системі додаткову гнучкість, підвищити точність та розширити діапазон застосувань.

Отже, у підсумку можна стверджувати, що на поточному етапі LSTM є найбільш привабливим вибором для прогнозування рухів смартфона у реальному часі, забезпечуючи відмінну точність, стабільність та прийнятний час виконання. Подальші зусилля, спрямовані на адаптацію моделей, удосконалення їх внутрішніх механізмів, тестування на ширших наборах даних та інтеграцію мультимодельних підходів, відкривають перспективи для досягнення ще більш високих стандартів якості і забезпечення максимальної реалістичності та плавності змішаного AR/VR-досвіду для користувачів мобільних пристроїв. За результатами експериментів встановлено, що за умови використання LSTM із довжиною послідовності 64 і відбіркою складних траєкторій наш метод досягає найвищої точності з F1-score до 96% та MAE близько 0.83–0.85 см, забезпечуючи при цьому стабільний час виконання, достатній для застосування в реальному часі.

## 5. ПОБУДОВА БІЗНЕС-МОДЕЛІ

### 5.1. Виділення проблеми

Технології віртуальної та доповненої реальності активно розвиваються, стаючи важливою складовою сучасного цифрового середовища. Водночас їх інтеграція на платформах смартфонів стикається з низкою серйозних викликів, які потребують вирішення для забезпечення високої якості досвіду користувачів. Однією з головних проблем є продуктивність пристроїв. Сучасні смартфони, хоч і оснащені потужними процесорами, стикаються з труднощами в обробці великого обсягу даних, необхідного для рендерингу тривимірних сцен і аналізу візуальної інформації в реальному часі. Це спричиняє нестабільність і затримки у відображенні, що негативно впливає на користувацький досвід.

Іншою важливою проблемою є точність інтеграції віртуальних і реальних об'єктів. Технології доповненої реальності вимагають високої точності розпізнавання поверхонь і синхронізації віртуальних елементів із реальним середовищем. Навіть незначні помилки в цих процесах руйнують ілюзію реальності, створюючи візуальні артефакти. Подібна ситуація спостерігається і у сфері віртуальної реальності, де точність створення стереозображень є ключовим фактором для забезпечення глибини і реалістичності сцени. Неправильне налаштування геометрії зображень призводить до втрати ілюзії тривимірного середовища, викликаючи дискомфорт у користувачів.

Окрім технічних аспектів, є ще проблема обмеженості існуючих бібліотек сцен і об'єктів для VR/AR. Сучасні інструменти для розробки часто не надають достатньої гнучкості або різноманітності, що ускладнює створення кастомізованих додатків. Також серйозним викликом є реалізація механізмів для плавного перемикання між режимами VR і AR у межах одного застосунку. У багатьох проєктах ці технології залишаються розділеними, що обмежує їх потенціал для створення інтегрованого досвіду.

Для вирішення перелічених проблем важливо зосередитися на розробці методів оптимізації обчислень, адаптації алгоритмів машинного навчання для передбачення руху пристрою, а також створенні багатofункціональних бібліотек сцен. Подолання цих бар'єрів дозволить смартфонам ефективніше підтримувати поєднання віртуальної та доповненої реальності, забезпечуючи користувачам високоякісний і реалістичний досвід.

Всі вище описані проблеми ми можемо узагальнити за допомогою дерева проблем зображеного на рис. 6.

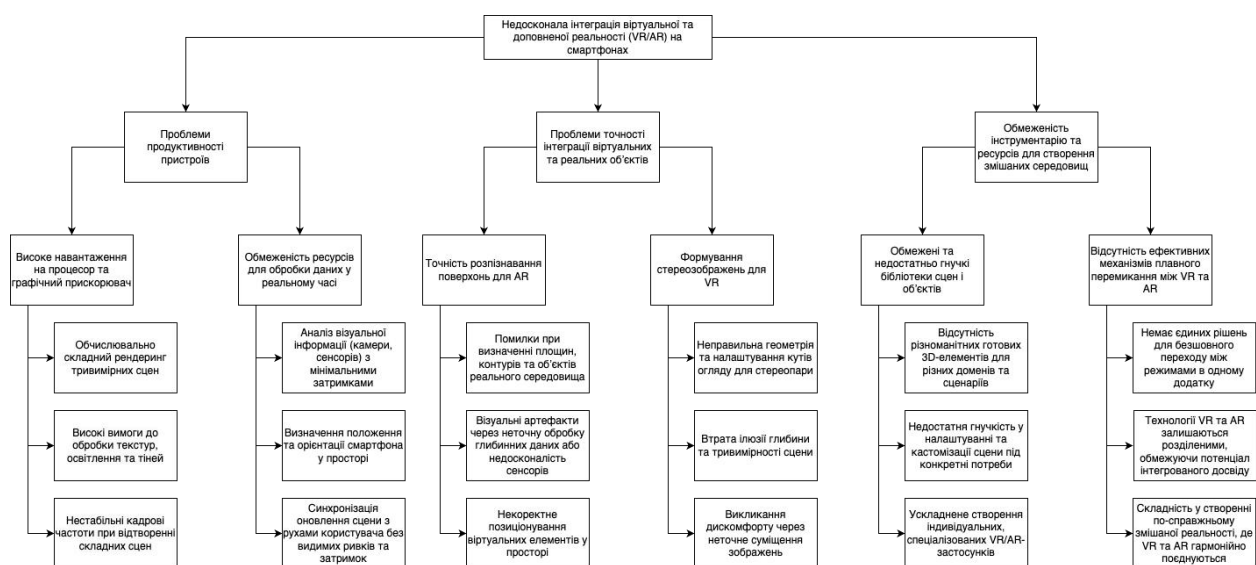


Рис. 6. Дерево проблем

## 5.2. Зацікавлені сторони

Інтеграція віртуальної та доповненої реальності на смартфонах є перспективним напрямом, який викликає інтерес у різних зацікавлених сторін. Вони не лише впливають на формування вимог до технології, але й створюють запит на подальший її розвиток.

Першою ключовою групою зацікавлених сторін є розробники програмного забезпечення. Для них створення інтегрованих VR/AR-додатків є складним завданням, що вимагає точного розуміння особливостей обох технологій, доступу до високоякісних бібліотек сцен та об'єктів, а також інструментів для оптимізації рендерингу і зменшення затримок. Ця група

очікує наявність гнучких інструментів, які дозволять скоротити час розробки і спростити інтеграцію VR та AR у мобільні додатки.

Наступною групою є компанії-замовники, які використовують VR/AR-додатки для бізнес-цілей. Їх інтерес полягає у створенні продуктів, які можуть покращити споживчий досвід, підвищити рівень взаємодії з клієнтами або підсилити ефективність внутрішніх процесів (наприклад, тренувань чи симуляцій). Такі компанії зацікавлені в тому, щоб кінцевий продукт був якісним, стабільним і відповідав потребам користувачів, мінімізуючи технічні обмеження.

Окремою категорією є користувачі смартфонів, які взаємодіють із MR-застосунками. Їхній інтерес полягає в отриманні реалістичного і стабільного досвіду використання, незалежно від технічної складності продукту. Вони очікують, що застосунки будуть інтуїтивними, безпечними, із плавним перемиканням між режимами VR та AR. Проте, у них немає безпосереднього впливу на процес розробки програмного забезпечення, й у найкращих випадках ІТ-продукти лише передбачатимуть способи збору зворотних відгуків від їх користувачів.

Ще однією важливою зацікавленою стороною є виробники смартфонів. Вони прагнуть підвищити конкурентоспроможність своїх пристроїв, інтегруючи передові технології VR та AR. Ці виробники зацікавлені у вирішенні проблем продуктивності, оптимізації енергоспоживання та забезпеченні точного функціонування сенсорів для коректної роботи з MR-застосунками.

Таким чином, усі ці групи створюють комплексний запит на розробку ефективних методів і програмного забезпечення для інтеграції віртуальної та доповненої реальності. Задоволення їхніх потреб сприятиме не лише розвитку окремих продуктів, а й всебічному прогресу у сфері MR технологій.

У табл. 2 підведені підсумки до усього зазначеного вище, визначено інтереси зацікавлених сторін та ступінь їх впливу на вирішення наявних проблем.

## Зацікавлені сторони

<i>Зацікавлена сторона</i>	<i>Інтерес зацікавленої сторони</i>	<i>Вплив зацікавленої сторони</i>	<i>Стратегії приваблення зацікавлених сторін</i>
Розробники програмного забезпечення	Створення інтегрованих MR-застосунків	Високий	Проведення презентацій та демонстрацій для представників компаній, створення вигідної системи монетизації, наявність оптових ціни на використання продукту
Бізнес	Створенні продуктів, які покращать споживчий досвід, та підвищити ефективність внутрішніх процесів	Високий	
Користувачі	Можливість запускати MR-реальність на смартфонах	Середній	
Виробники смартфонів	Можливість користуватись якісними веб-додатками	Низький	

**5.3. Комерційне рішення. Основні характеристики**

Запропоноване комерційне рішення спрямоване на усунення основних проблем, пов'язаних із інтеграцією віртуальної та доповненої реальності на смартфонах, зокрема на оптимізацію продуктивності пристроїв і підвищення точності візуалізації. Одним із ключових аспектів цього підходу є використання LSTM-нейронних мереж для передбачення руху смартфона у просторі. Це забезпечить плавність відображення віртуального контенту,

мінімізуючи затримки навіть на пристроях із середніми технічними характеристиками. Завдяки такому рішенню користувачі отримають стабільний досвід, що підвищить привабливість технологій VR та AR для масового споживача.

Другою важливою складовою є забезпечення точності інтеграції віртуальних і реальних об'єктів. Для досягнення цієї мети пропонується використання LiDAR-сенсорів, які дозволяють точно реконструювати оточення та визначати глибину сцени. Це сприятиме створенню реалістичних AR-сцен і коректному накладанню віртуальних елементів на реальні поверхні. У віртуальній реальності передбачено точне формування стереозображень із врахуванням геометричних корекцій для гарнітур, що забезпечить глибину й реалістичність сприйняття. Такий підхід дозволить значно підвищити якість роботи VR/AR додатків, що позитивно вплине на користувацький досвід.

З огляду на вказані можливі зацікавлені сторони, клієнтами застосування будуть фізичні особи (розробники), та юридичні (компанії). Його використання буде передбачене як за моделлю B2B[25] (бізнес для бізнесу), так і за B2C[26] (бізнес для користувача).

#### **5.4. Конкурентні переваги рішення**

Запропоноване рішення для інтеграції віртуальної та доповненої реальності на смартфонах має низку конкурентних переваг, що виділяють його серед існуючих технологічних розробок у цій сфері. Насамперед, його основою є використання передових методів машинного навчання, зокрема LSTM-нейронних мереж, для передбачення руху пристрою. Цей підхід дозволяє значно зменшити затримки у відображенні контенту, що є однією з ключових проблем наявних рішень. Така оптимізація продуктивності забезпечує стабільний і плавний користувацький досвід, який не обмежується характеристиками апаратного забезпечення.

Ще однією унікальною перевагою є інтеграція LiDAR-сенсорів для створення високоточної реконструкції оточення. Це дозволяє не лише

покращити якість відображення в доповненій реальності, але й забезпечити бездоганну синхронізацію віртуальних об'єктів із реальним середовищем. У результаті користувач отримує відчуття природності й повного занурення, що є важливим фактором для підвищення рівня задоволення від роботи з AR-додатками.

На відміну від багатьох конкурентів, запропоноване рішення також вирізняється гнучкістю. Воно передбачає можливість інтеграції як готових бібліотек сцен і об'єктів, так і кастомізації під конкретні потреби розробників. Це відкриває широкі перспективи для адаптації технології до різних галузей, таких як освіта, медицина, ігри або архітектура. Крім того, функціональність плавного перемикання між режимами VR і AR робить це рішення унікальним на ринку, забезпечуючи новий рівень зручності для користувачів.

Таким чином, головними конкурентними перевагами є високий рівень продуктивності, точність і гнучкість у розробці, що разом із передовими технологіями відкриває нові можливості для застосування VR і AR на смартфонах.

### **5.5. Унікальна ціннісна пропозиція**

Цінність продукту – це поняття, яке визначає користь, яку продукт приносить своїм користувачам чи клієнтам.

В свою чергу, ціннісна пропозиція показує, чому продукт перевершує конкуруючі пропозиції і чому клієнт повинен купувати його. Для її визначення необхідно розглянути проблеми, які вирішує програмний продукт, і очікування зацікавлених сторін від нього [27].

Наше рішення забезпечує принципово новий рівень взаємодії між фізичним та цифровим простором за допомогою комплексного поєднання доповненої (AR) та віртуальної (VR) реальності безпосередньо на смартфонах.

Застосування LSTM-моделей для обробки даних з IMU-сенсорів (акселерометр, гіроскоп) дозволяє не лише миттєво реагувати на зміни

положення смартфона, а й передбачати наступні стани орієнтації заздалегідь. Це мінімізує затримки та забезпечує плавність переходів, позбавляючи користувача від неприємного «запізнення» або ривків віртуальних елементів при різких рухах.

Завдяки інтеграції LiDAR-даних (або програмних алгоритмів глибинної оцінки) наш продукт формує тривимірну модель оточення, що значно перевершує традиційні AR-підходи, обмежені накладанням базових віртуальних об'єктів. Така реконструкція не лише забезпечує реалістичне розташування елементів у просторі, а й створює можливість формування стереозображень, підсилюючи ефект занурення та роблячи досвід ще більш природним.

Використовуючи Unity та AR Foundation наш продукт гарантує кросплатформенність та простоту впровадження. Розробники отримують готову інфраструктуру для швидкого прототипування та масштабування застосунків, без необхідності «склеювати» різноманітні рішення. Це допомагає зосередитись на творчих та функціональних аспектах продукту, а не на технічних бар'єрах інтеграції.

Система може навчатися та вдосконалюватися у процесі використання. Зі зростанням обсягу даних і різноманітності сценаріїв застосування наші моделі прогнозування та алгоритми реконструкції сцени підлаштовуються під нові умови, підвищуючи точність, стійкість та продуктивність з часом. Це забезпечує тривалу актуальність та гнучкість рішення, яке з плином часу стає ще більш ефективним.

Комплексний підхід – від прогнозування руху та 3D-реконструкції до формування стерео-контенту – дозволяє уникнути залучення кількох розрізнених технологій та інструментів. Результат – суттєве скорочення циклу розробки, зменшення вартості проєкту та прискорення виходу на ринок.

Таким чином, унікальна ціннісна пропозиція нашого рішення – це можливість створити більш реалістичний, плавний та глибоко інтерактивний

AR/VR-досвід на смартфонах, зберігаючи при цьому гнучкість, масштабованість та простоту впровадження. Ми пропонуємо не окремий інструмент, а цілісну екосистему для формування інноваційних змішаних середовищ, що можуть легко адаптуватися до мінливих вимог бізнесу, ринку та потреб кінцевих користувачів.

## **5.6. Доходи та витрати**

Загалом, дохід формується з продажу запусків робочих сесій програми та надання послуг із кастомізації й консультацій у сфері створення високоякісного AR/VR-досвіду на смартфонах. Обрана модель фріміум оптимально пасує до розробленого рішення, яке поєднує доповнену та віртуальну реальність із механізмами прогнозування руху, реконструкції сцени та інтеграції з AR Foundation. Фріміум-бізнес-модель [28] передбачає надання базової версії продукту безкоштовно, тоді як розширений функціонал, що включатиме можливість специфікації предметних областей, глибшу інтеграцію з апаратними сенсорами, доступ до додаткових сценаріїв та покращені алгоритми прогнозування руху, надаватиметься за окрему плату.

Безкоштовна базова версія програми дозволить користувачу самостійно протестувати ключові можливості системи: наприклад, базовий набір сценаріїв для відтворення AR/VR-контенту, базове прогнозування руху та основну реконструкцію оточення. Це стане відправною точкою для нових користувачів, які зможуть оцінити загальну якість та функціональність продукту, отримуючи високий користувацький досвід без початкових фінансових вкладень.

Для розширення функціональності та доступу до поглиблених технологій, таких як спеціалізовані LSTM-моделі для конкретних доменів чи покращена підтримка LiDAR-даних, користувачі здійснюватимуть разову оплату. Такий підхід стимулює поступове залучення клієнтів до преміум-пропозицій, формуючи лояльну аудиторію та забезпечуючи регулярний

приплив коштів.

Особливу цінність отримають компанії, що прагнуть впровадити AR/VR-технології у власні проекти: для них передбачені платні консультації, послуги з глибокої кастомізації та створення індивідуальних сценаріїв із урахуванням специфіки їх бізнес-процесів та вимог до інтерактивного середовища.

До основних витрат належать:

- утримання фахівців, які постійно підтримуватимуть продукт, оновлюватимуть його та розширюватимуть набір доступних сценаріїв і функціоналу (виплати заробітних плат, соціальні пакети);
- витрати на оренду офісних приміщень, комунальні послуги та організацію робочих місць для персоналу;
- оплата податків згідно з чинним законодавством;
- послуги юристів, бухгалтерів та інших суміжних спеціалістів, необхідних для безперервної діяльності компанії.

З детальним описом витрат і прогнозованих прибутків можна ознайомитись у табл. 3.

Таблиця 3

Витрати на реалізацію проєкту

Найменування витрат	1-й місяць, тис. \$	2-й місяць, тис. \$	3-й місяць, тис. \$	4-й місяць, тис. \$	5-й місяць, тис. \$	6-й місяць, тис. \$
Загальні витрати	2	2	2	2	2	2
Заробітна плата	20	20	17	17	15	15
Витрати	18	18	16	16	14	14
Заплановані прибутки						

Продовження табл. 6

Результат (без оподаткування)	-18	-18	-16	-16	-14	-14
Загальні витрати	2	2	2	2	2	2
Заробітна плата	15	15	20	20	22	22
Витрати	18	18	22	22	25	25
Заплановані прибутки	80	80	80	80	80	80
Результат (без оподаткування)	62	62	58	58	55	55

### 5.7. Бізнес-модель

Споживачі: розробники AR/VR-застосунків (ігор, віртуальних турів, освітніх та промислових програм) ІТ-компанії, які спеціалізуються на інтерактивних технологіях та змішаній реальності.

Проблема: складність поєднання віртуальної та доповненої реальності на мобільних пристроях, затримки та нестабільність відтворення віртуальних сцен при різких рухах смартфона, відсутність повноцінної реконструкції сцени за межами видимого поля зору, обмежені можливості базових AR-платформ без інтеграції прогнозування руху та глибинного аналізу, труднощі масштабування рішень та адаптації під різні предметні області.

Рішення: розробка комплексного програмного забезпечення, яке поєднує AR та VR за допомогою LSTM-прогнозування руху смартфона, реконструкції сцени на основі глибинних даних та інтеграції з AR Foundation. Це забезпечує плавний і реалістичний AR/VR-досвід, динамічне розширення функціональності та легку адаптацію до різноманітних сценаріїв.

Унікальна ціннісна пропозиція: автоматизоване передбачення руху пристрою, повноцінна реконструкція оточення та формування стереозображень забезпечують по-справжньому занурюючий AR/VR-досвід на смартфонах. Це створює новий рівень якості та ефективності у розробці, тестуванні й впровадженні змішаних реальностей без додаткових складнощів інтеграції.

Потоки доходів: дохід від продажу запусків платної версії програми (доступ до розширених сценаріїв, прогнозувань, глибинного аналізу), дохід від підтримки та кастомізації програмного забезпечення під конкретні потреби (консультації, розробка спеціалізованих сценаріїв, адаптація під предметні області клієнтів).

Структура витрат: утримання персоналу для технічної підтримки, оновлень та вдосконалення програмного рішення (заробітні плати, соціальні пакети); витрати на оренду офісу, комунальні послуги та забезпечення робочих місць; податкові витрати; оплата послуг юриста, бухгалтера та інших суміжних спеціалістів.

Також в канву бізнес-моделі включаються структурні блоки: прихована перевага (перевага, яку неможливо скопіювати або купити), ключові метрики (основні показники, що вимірюються) та канали (шляхи до користувачів).

Прихована перевага: використання гнучкої архітектури та підходу до розробки, який дозволяє постійно навчати та оновлювати моделі, інтегрувати нові сенсори та алгоритми без необхідності кардинальних змін у продукті. Це забезпечує постійне вдосконалення і збільшення цінності рішення з плином часу.

Ключові метрики: кількість проданих сесій роботи програми (платних запусків) та кількість корпоративних клієнтів, які замовляють кастомізацію та розширені опції.

Канали: демонстрації роботи програми на профільних конференціях з AR/VR-технологій; публічні презентації, воркшопи та навчальні курси для розробників інтерактивних рішень; співпраця з ІТ-компаніями та

інкубаторами стартапів, які фокусуються на змішаній реальності.

Бізнес-модель наведено у зведеному вигляді у табл. 4.

Таблиця 4

Бізнес-модель

<i>Проблема</i>	<i>Рішення</i>	<i>Унікальна ціннісна пропозиція</i>	<i>Прихована перевага</i>	<i>Споживачі</i>
складність поєднання віртуальної та доповненої реальності на мобільних пристроях затримки та нестабільність відтворення віртуальних сцен при різких рухах смартфона	розробка програмного забезпечення поєднує віртуальну та доповнену реальність	автоматизоване передбачення руху пристрою, повноцінна реконструкція оточення та формування стереозображень	використання підходу розробки, що дозволяє постійно розширювати можливості програми	розробники AR/VR-застосунків, IT-компанії, які спеціалізуються на інтерактивних технологіях та змішаній реальності
	<i>Ключові метрики</i> кількість проданих сесій роботи програми та корпоративних клієнтів		<i>Канали</i> демонстрації роботи програми; конференції з AR/VRтехнологій	
<i>Структура витрат</i> утримання персоналу для технічної підтримки, оновлень та вдосконалення програмного рішення організація й утримання робочих місць для персоналу податкові витрати оплата послуг юриста, бухгалтера витрати на маркетинг			<i>Потоки доходів</i> доходи від продажу запусків платної версії програми доходи від підтримки й кастомізації програмного забезпечення під конкретні потреби	

Отже, зважаючи на дані у табл. 4, можна зробити висновок, що запропонована реалізація методу поєднання віртуальної та доповненої реальності на смартфонах має перспективи для подальшого розвитку. Навіть без врахування усіх можливих факторів і ризиків, наявних досліджень уже достатньо, щоб прогнозувати комерційний успіх продукту та його окупність.

## **5.8. Висновки до розділу 5**

У цьому розділі було здійснено комплексний аналіз актуальних проблем інтеграції віртуальної та доповненої реальності на смартфонах, визначено зацікавлені сторони та сформовано бізнес-модель, що відображає комерційний потенціал запропонованого рішення.

Розроблена бізнес-модель у форматі lean canvas продемонструвала наявність стабільних каналів монетизації (фримиум-модель, платні консультації та кастомізація), чітко окреслила ціннісну пропозицію і можливості масштабування продукту. Серед ключових переваг – адаптивність, можливість навчання системи, гнучкість інструментарію і здатність формувати комплексні середовища змішаної реальності. Основна зацікавленість у використанні рішення спостерігається як серед індивідуальних розробників, так і серед ІТ-компаній, що спеціалізуються на інтерактивних технологіях.

Таким чином, проаналізовані аспекти дають підстави прогнозувати комерційний успіх, окупність і конкурентоспроможність запропонованого продукту. Підхід, заснований на передових технологіях машинного навчання, кросплатформенній інтеграції та продуманій бізнес-стратегії, закладає міцний фундамент для подальшого вдосконалення та широкого застосування рішення у різних секторах ринку.

## ВИСНОВКИ

У ході виконання даної магістерської роботи було розв'язано комплексне завдання створення програмного забезпечення, здатного цілісно поєднати віртуальну та доповнену реальність на мобільних пристроях. Дана мета вимагала системного підходу до дослідження існуючих технологій, архітектурних рішень та інструментів, а також розробки та впровадження оригінальних методів підвищення точності й стабільності інтегрованого VR/AR-досвіду на смартфонах.

Аналізуючи існуючі інструменти для доповненої та віртуальної реальності, було виявлено, що більшість з них зосереджені на окремих аспектах: розпізнаванні площин, накладанні віртуальних об'єктів або відстеженні руху пристрою. Водночас відсутність цілісного рішення, що інтегрувало б передбачення руху, повноцінну реконструкцію сцени, використання глибинних даних та формування стереозображень, істотно обмежувала можливості розробників. Така ситуація зумовлювала необхідність створення інноваційного підходу, здатного усунути недоліки, що гальмують впровадження високоякісних змішаних реалістичних середовищ на мобільних платформах.

Основу запропонованого методу склала інтеграція глибинних нейронних мереж, зокрема LSTM-моделі, для прогнозування майбутнього положення та орієнтації смартфона. Завдяки інтелектуальному аналізу часових рядів даних IMU, система здатна «наперед» знати, куди рухатиметься пристрій, що дає можливість підготувати віртуальний контент, мінімізувати затримки та підвищити плавність оновлення AR/VR-сцени. Такий підхід забезпечує суттєве покращення користувацького досвіду, особливо за умов динамічних змін позиції пристрою.

Другою важливою складовою стало застосування глибинних сенсорів, зокрема LiDAR, для реконструкції сцени за межами поля зору. Це дозволило створювати точні тривимірні карти реального оточення, у яких віртуальні об'єкти природно інтегруються, накладаються та взаємодіють з реальними

поверхнями. Внаслідок цього досягнуто рівень реалістичності, недосяжний для традиційних рішень, що обмежуються лише візуальним трекінгом. Формування стереозображень, зокрема для пристроїв із підтримкою стереоскопічного відтворення, додатково поглиблює ефект занурення, роблячи користувацький досвід більш природним та переконливим.

Інтеграція зі середовищем Unity та AR Foundation виступила важливим кроком до кросплатформенності та універсальності рішення. Використання Unity як рушія та AR Foundation як уніфікованого інструментарію для AR на iOS та Android спростило процес розробки, дозволило розширювати функціональність за рахунок плагінів та легко адаптувати продукт під різні сценарії застосування. Це відкриває нові перспективи для індустрії, де мультимедійні елементи, 3D-моделі, анімація та інтерактивність поєднуються в одному гнучкому середовищі.

Проведений аналіз виконаної системи та результати експериментів продемонстрували здатність моделі LSTM забезпечити високу точність передбачень (F1-score до 96%) та низьку середню похибку (MAE близько 0.83–0.85 см) при помірному використанні ресурсів. Завдяки застосуванню ExecuTorch та оптимізації нейронних моделей досягнуто стабільний інференс у реальному часі, що є критичним для плавної інтеграції VR/AR-контенту. Досягнутий баланс між точністю, швидкістю, універсальністю та масштабованістю підтвердив доцільність обраного підходу.

Важливим аспектом є адаптивність розробленого рішення. Модульна архітектура, використання стандартизованих інтерфейсів та протоколів обміну даними, можливість підмінити або розширювати компоненти, а також здатність моделі навчатися на нових даних забезпечують гнучкість системи. З часом, отримуючи все більше інформації про поведінку користувачів, пристрій та оточення, система зможе вдосконалювати свої прогностичні властивості та підтримувати високу якість відтворення за будь-яких умов експлуатації.

Розроблений метод та програмне забезпечення мають широкий спектр

потенційних застосувань: від розваг та освіти, де користувачі отримують глибше занурення і реалістичнішу взаємодію, до промисловості, медицини, дизайну і транспорту, де точні та стійкі AR/VR-середовища можуть підвищити ефективність, безпеку та інформативність робочих процесів. Масштабованість та адаптивність системи дозволяють впроваджувати її у найрізноманітніші сфери та домени, задовольняючи потреби як індивідуальних користувачів, так і великих компаній.

Підсумовуючи, мета, поставлена у магістерській дисертації, була досягнута. Запропонований комплексний метод, що об'єднує прогнозування руху за допомогою LSTM, реконструкцію сцени, формування стереозображень, інтеграцію з Unity та AR Foundation, виявився дієвим засобом для поєднання віртуальної та доповненої реальності на смартфонах. Досягнуті результати свідчать про можливість значно підвищити реалістичність, стабільність та ефективність VR/AR-досвіду на мобільних платформах, відкриваючи шлях до подальших досліджень та комерційних застосувань у галузі змішаної реальності.

## СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

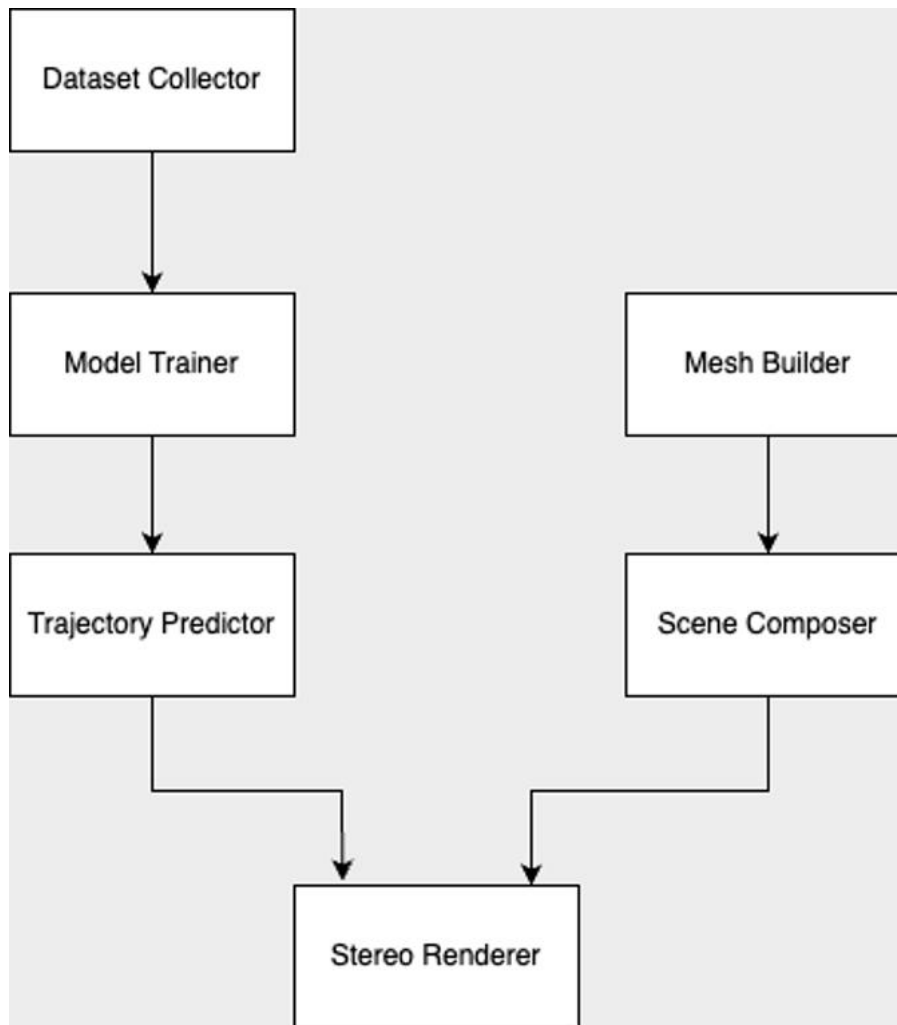
1. Dayoung Jeong, Seungwon Paik, YoungTae Noh, and Kyungsik Han. 2023. MAC: multimodal, attention-based cybersickness prediction modeling in virtual reality. *Virtual Reality* (2023), 1–16.
2. Dayoung Jeong and Kyungsik Han. 2024. PRECYSE: Predicting Cybersickness using Transformer for Multimodal Time-Series Sensor Data. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8, 2, Article 42 (May 2024), 24 pages. <https://doi.org/10.1145/3659594>
3. H. Ye, Y. Chen and M. Liu, "Tightly Coupled 3D Lidar Inertial Odometry and Mapping," *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada, 2019, pp. 3144-3150, doi: 10.1109/ICRA.2019.8793511.
4. Hong, Ye, Henry Martin, and Martin Raubal. "How do you go where? Improving next location prediction by learning travel mode information using transformers." *Computing Research Repository* 2023.2210 (2022).
5. AWS - Application Program Interface [Електронний ресурс] – Режим доступу: <https://aws.amazon.com/what-is/api/>
6. Wikipedia – MAE [Електронний ресурс] – Режим доступу: [https://en.wikipedia.org/wiki/Mean\\_absolute\\_error](https://en.wikipedia.org/wiki/Mean_absolute_error)
7. Wikipedia – Unity (ігровий рушій) [Електронний ресурс] – Режим доступу: [https://uk.wikipedia.org/wiki/Unity\\_\(%D1%96%D0%B3%D1%80%D0%BE%D0%B2%D0%B8%D0%B9\\_%D1%80%D1%83%D1%88%D1%96%D0%B9\)](https://uk.wikipedia.org/wiki/Unity_(%D1%96%D0%B3%D1%80%D0%BE%D0%B2%D0%B8%D0%B9_%D1%80%D1%83%D1%88%D1%96%D0%B9))
8. IT Enterprise – Virtual Reality, VR [Електронний ресурс] – Режим доступу: <https://www.it.ua/knowledge-base/technology-innovation/virtualnaja-realnost-vr>
9. Intel – What is a GPU? [Електронний ресурс] – Режим доступу: <https://www.intel.com/content/www/us/en/products/docs/processors/what-is-a-gpu.html>
10. Wikipedia – LSTM [Електронний ресурс] – Режим доступу: [https://en.wikipedia.org/wiki/Long\\_short-term\\_memory](https://en.wikipedia.org/wiki/Long_short-term_memory)
11. Teach Hub – Доповнена реальність [Електронний ресурс] – Режим доступу: <https://teach-hub.com/scho-take-dopovnena-realnist/>

12. IT Enterprise – Змішана реальність [Електронний ресурс] – Режим доступу: <https://www.it.ua/knowledge-base/technology-innovation/dopolnennaja-virtualnaja-i-prochie-realnosti>
13. English-Ukrainian Military Dictionary - Inertial Measurement Unit [Електронний ресурс] – Режим доступу: <https://english-military-dictionary.org.ua/inertial-measurement-unit-imu-2>
14. Wikipedia – LiDAR [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/%D0%9B%D1%96%D0%B4%D0%B0%D1%80>
15. IBM – Recurent neural network [Електронний ресурс] – Режим доступу: <https://www.ibm.com/think/topics/recurrent-neural-networks>
16. Wikipedia – GPS [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/GPS>
17. GSMhub – Центральний процесор [Електронний ресурс] – Режим доступу: <https://gsmhub.com.ua/glossary/cp-centralnii-procesor>
18. Apple – ARKit [Електронний ресурс] – Режим доступу: <https://developer.apple.com/augmented-reality/arkit/>
19. Google – ARCore [Електронний ресурс] – Режим доступу: <https://developers.google.com/ar/develop>
20. Unity – AR Foundation [Електронний ресурс] – Режим доступу: <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@2.0/manual/index.html>
21. TechTarget – PyTorch [Електронний ресурс] – Режим доступу: <https://www.techtarget.com/searchenterpriseai/definition/PyTorch>
22. PyTorch – ExecuTorch [Електронний ресурс] – Режим доступу: <https://pytorch.org/executorch-overview>
23. Apple – Xcode [Електронний ресурс] – Режим доступу: <https://developer.apple.com/xcode/>
24. JetBrains – Rider [Електронний ресурс] – Режим доступу: <https://www.jetbrains.com/rider/>

25. Wikipedia – F1-score [Электронный ресурс] – Режим доступа:  
<https://en.wikipedia.org/wiki/F-score>
26. James Chen, Business-to-business (B2B): What it is and how it's used. 2023 [Электронный ресурс] – Режим доступа:  
<https://www.investopedia.com/terms/b/btob.asp>
27. Will Kenton, B2C: How Business-to-consumer sales works, 5 types and examples. 2023 [Электронный ресурс] – Режим доступа:  
<https://www.investopedia.com/terms/b/btoc.asp>
28. Value proposition [Электронный ресурс] – Режим доступа:  
<https://corporatefinanceinstitute.com/resources/management/value-proposition/>
29. Mae Rice, Freemium, Explained. 2022 [Электронный ресурс] – Режим доступа: <https://builtin.com/product/freemium>

## **ДОДАТКИ**

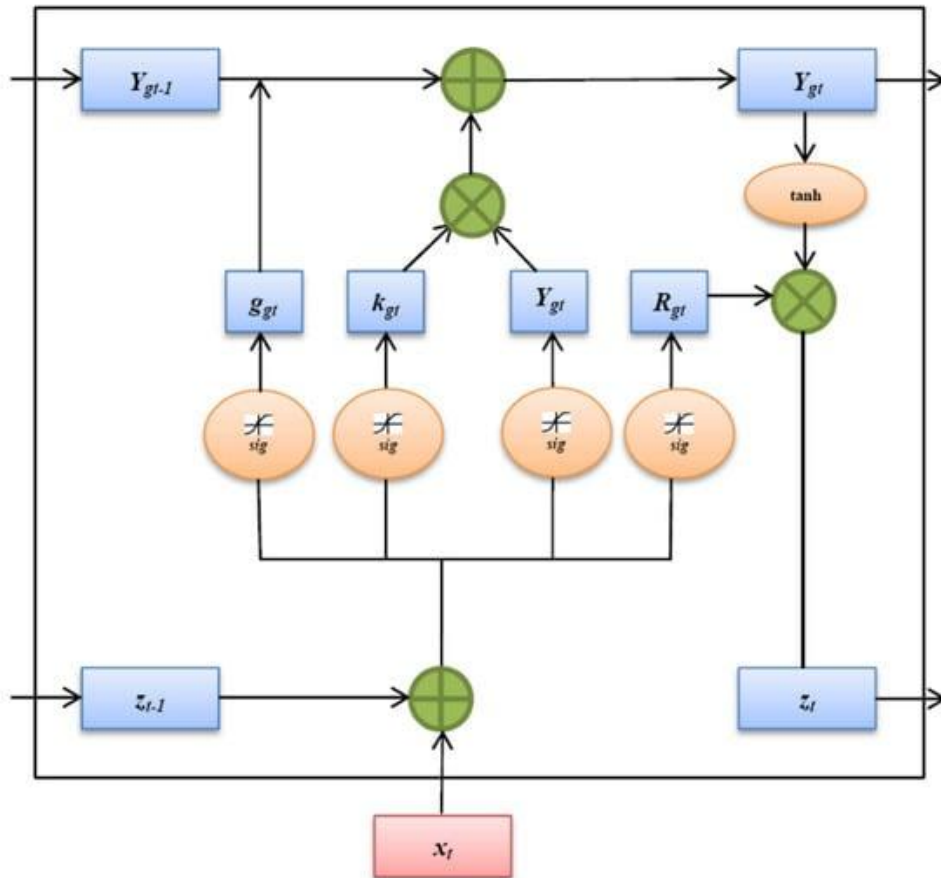
**’Додаток 1**  
**Копії графічних матеріалів**



Взаємозв'язки модулів розробленого програмного забезпечення для поєднання віртуальної та доповненої реальності на смартфонах

Проценко А.П., КП-31мп

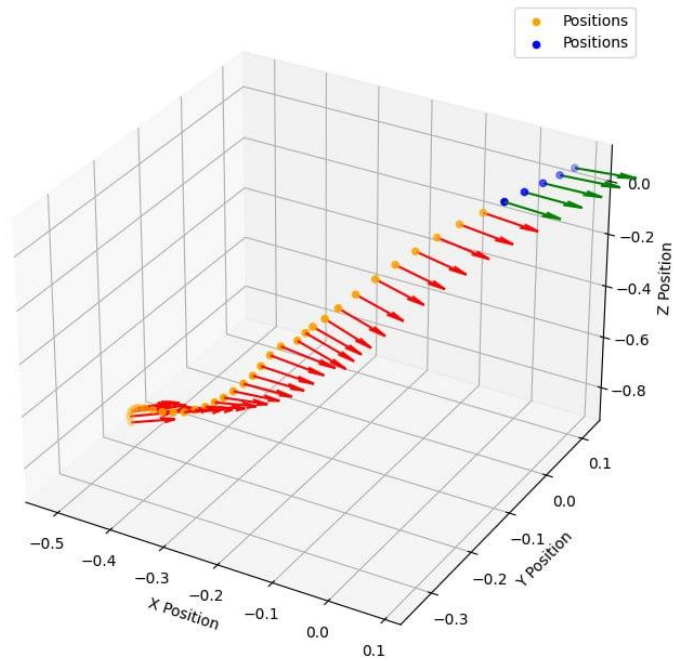




Архітектура нейронної мережі LSTM

Проценко А.П., КП-31мп

3D Trajectory of Camera Positions and Directions



Модель візуалізації передбачення наступних  
рухів голови за допомогою LSTM при 32  
точках  
Проценко А.П., КП-31мп

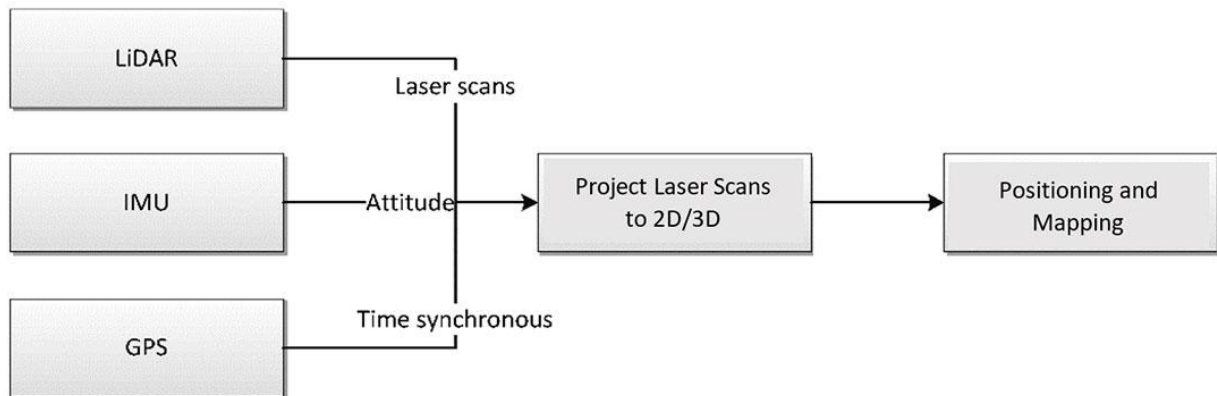
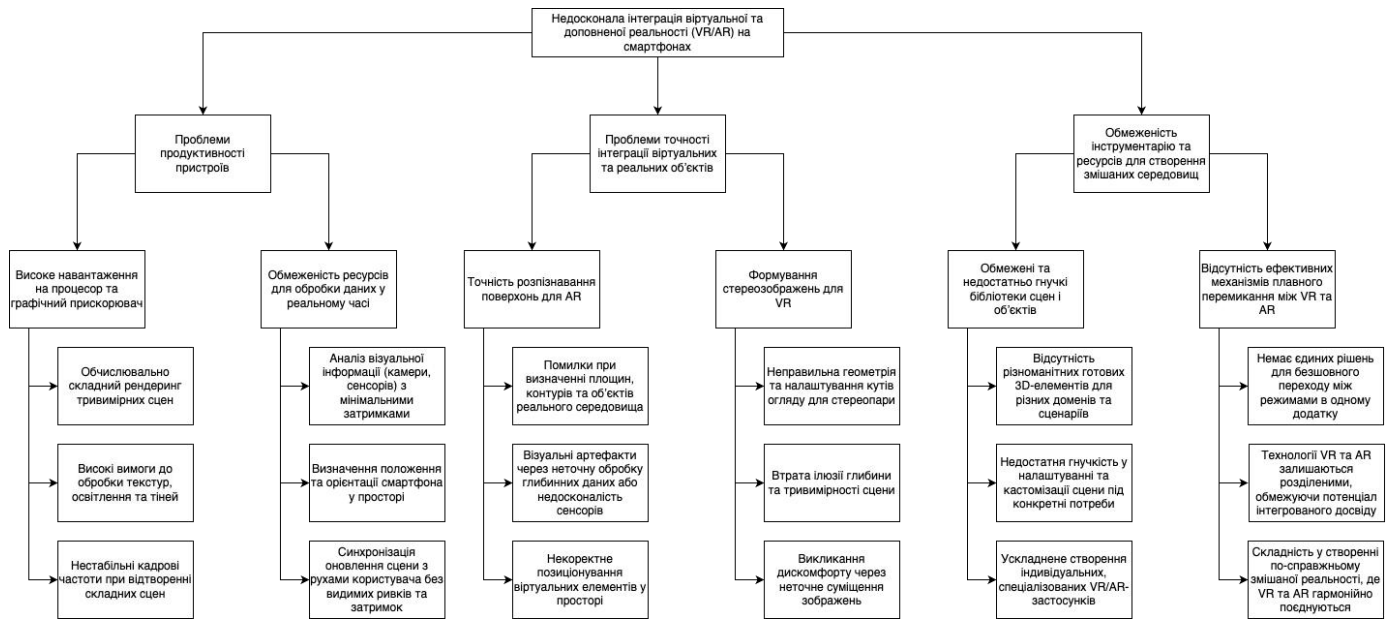


Схема роботи системи сканерів LiDAR

Проценко А.П., КП-31мп



## Дерево проблем поєднання віртуальної та доповненої реальності на смартфонах

Проценко А.П., КП-31мп

**Додаток 2**  
**Лістинг програми**

## Лістинг 1. Генератор датасетів для навчання нейронної мережі

```
import json

import numpy as np

class DatasetGenerator:

    def load(self, file_path):

        trajectory = np.load(file_path)

        return trajectory

    def generate_trajectories(self, trajectory, sequence_length=32):

        num_points = trajectory.shape[0]

        if num_points < sequence_length:

            raise ValueError("Trajectory is too short for the given sequence length.")

        num_sequences = num_points - sequence_length + 1

        pos_trajectories, dir_trajectories = [], []

        for i in range(num_sequences):

            traj = trajectory[i:i + sequence_length]

            pos_traj, dir_traj = traj[:, 3:] - traj[-2, 3:], traj[:, :3]

            # pos_traj, dir_traj = traj[:, 3:], traj[:, :3]

            pos_trajectories.append(pos_traj)

            dir_trajectories.append(dir_traj)

        pos_trajectories = np.array(pos_trajectories)

        dir_trajectories = np.array(dir_trajectories)

        dir_trajectories = dir_trajectories / np.clip(np.linalg.norm(dir_trajectories, axis=2, keepdims=True), 1e-6,

np.inf)
```

```
return np.stack([pos_trajectories, dir_trajectories], axis=2)
```

```
def save_sequences(self, subsequences, output_file):
```

```
    np.save(output_file, subsequences)
```

```
    print(f"Sequences saved to {output_file}")
```

## Лістинг 2. Створення датасетів

```
import numpy as np
import torch
from torch.utils.data import Dataset, DataLoader

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

class CameraMovementDataset(Dataset):
    def __init__(self, data_path, n_points=32):
        self.n_points = n_points
        data = np.load(data_path)
        self.data = torch.tensor(data).to(device)

    def __len__(self):
        return len(self.data)

    def __getitem__(self, idx):
        samples = self.data[idx]

        # theta = torch.rand((1, ), device=device) * 2 * torch.pi
        # samples = self.rotate_points(samples, theta)

        return samples[:, 0], samples[:, 1]

    def rotate_points(self, points, theta):
        # Rotation matrix around the y-axis
        rotation_matrix = torch.tensor([
            [torch.cos(theta), 0, torch.sin(theta)],
            [0, 1, 0],
            [-torch.sin(theta), 0, torch.cos(theta)]
        ], device=device)

        # Rotate the points
```

```
rotated_points = torch.matmul(points, rotation_matrix.T)

return rotated_points
```

### Лістинг 3. Реалізація передбачення траєкторій

```
import random

import numpy as np
import torch

from dataset import CameraMovementDataset
from models.output_transform import transform_output
from visualization import visualize_trajectory

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

def predict_trajectory(model, initial_sequence, num_predictions):
    """
    Autoregressively predict multiple positions using a trained LSTM model.

    Args:
        model: Trained LSTM model.
        initial_sequence: Tensor of shape [1, seq_len, input_dim], the starting
trajectory points.
        num_predictions: Number of future positions to predict.

    Returns:
        predictions: List of predicted points, each of shape [input_dim].
    """
    model.eval() # Set the model to evaluation mode
    initial_sequence = initial_sequence.unsqueeze(0)

    predictions = []
    current_sequence = initial_sequence.clone() # Avoid modifying the original
input

    current_shift = torch.zeros((1, 6), device=device)
    mult = torch.tensor([1, 1, 1, 0, 0, 0], device=device)

    with torch.no_grad():
        for _ in range(num_predictions):
            # Predict the next position
```

```

        output = model(current_sequence) # [1, input_dim]
        pos, dir = transform_output(output)
        next_point = torch.cat([pos, dir], dim=-1)

        # Append the predicted point to the sequence
        predictions.append(((next_point + current_shift).clone()).squeeze(0))
# Remove batch dimension

        # Update the sequence (sliding window)
        current_sequence = torch.cat((current_sequence[:, 1:, :],
next_point.unsqueeze(1)), dim=1)
        current_shift += next_point * mult

        current_sequence[..., :3] -= next_point[..., :3]

    return torch.stack(predictions, dim=0) # [num_predictions, input_dim]

model = torch.load("checkpoints/lstm.pt")
# model = torch.load("checkpoints/transformer.pt")
model = model.to(device)

train_dataset = CameraMovementDataset('datasets/sequences_train.npy')
item_id = random.randint(0, len(train_dataset) - 1)
positions, directions = train_dataset.__getitem__(item_id)
positions -= positions[-1]

predictions = predict_trajectory(model, torch.cat((positions, directions), dim=-
1), 5)
pred_positions, pred_directions = transform_output(predictions)

positions, directions = positions.numpy(force=True), directions.numpy(force=True)
pred_positions, pred_directions = pred_positions.numpy(force=True),
pred_directions.numpy(force=True)

visualize_trajectory(positions, directions, pred_positions, pred_directions)

```

**Додаток 3**  
**Копія презентації**



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ  
СІКОРСЬКОГО”



ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

# МЕТОД ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ПОЄДНАННЯ ВІРТУАЛЬНОЇ ТА ДОПОВНЕНОЇ РЕАЛЬНОСТІ НА СМАРТФОНАХ

Доповідач: Проценко Андрій Павлович

Науковий керівник: к.т.н., доцент, доцент кафедри ПЗКС Юрчишин В. Я.

Київ – 2024

# АКТУАЛЬНІСТЬ ДОСЛІДЖЕННЯ



Більшість сучасних платформ і фреймворків надають висококласні можливості для розпізнавання площин, відстеження руху смартфона, накладання віртуальних об'єктів на реальний світ, проте вони часто зосереджені лише на аспектах доповненої реальності й не передбачають злиття з повноцінним віртуальним світом.

**Об'єкт дослідження:** процес поєднання віртуальної та доповненої реальності на смартфонах.

**Предмет дослідження:** методи, способи та програмні засоби поєднання віртуальної та доповненої реальності на смартфонах.

**Мета дослідження:** створення програмного забезпечення, покращення вже існуючих методів поєднання віртуальної та доповненої реальності на смартфонах.



## ІСНУЮЧІ РІШЕННЯ

1. Apple ARKit: розроблено компанією Apple спеціально для пристроїв iOS, з метою спростити створення додатків доповненої реальності.
2. Google ARCore: набір для розробки програмного забезпечення, розроблений Google, який дозволяє будувати додатки доповненої реальності.
3. Unity AR Foundation: кросплатформений набір інструментів від Unity, який спрощує розробку AR-додатків.

# ІСНУЮЧІ ПРОДУКТИ НА ОСНОВІ РІШЕНЬ



1. IKEA Place (на базі ARKit): застосунок, який дозволяє розміщувати віртуальні моделі меблів у реальному середовищі, використовуючи виявлення площин та точний трекінг камери.
2. Pokémon GO (на базі ARKit/ARCore): гра збагачує реальне оточення віртуальними істотами. Вона послуговується базовим трекінгом, визначенням площин і візуальними підказками.

# ПОСТАНОВКА ЗАДАЧІ



**Наукове завдання:** розроблення та оптимізація програмного забезпечення для створення змішаної реальності для смартфонів.

## Окремі завдання

1. Проаналізувати сучасні методи поєднання віртуальної та доповненої реальності на смартфонах.
2. Розроблення методу для поєднання віртуальної та доповненої реальності на смартфонах.
3. Розробити програмне забезпечення для імплементації модифікованого методу.
4. Оптимізувати метод для комфортного використання користувачами.
5. Провести експериментальне тестування розробленого методу для різних задач у VR та AR.
6. Порівняти ефективність створеного методу з традиційними підходами.

# ШЛЯХИ ДОСЯГНЕННЯ МЕТИ ДОСЛІДЖЕННЯ



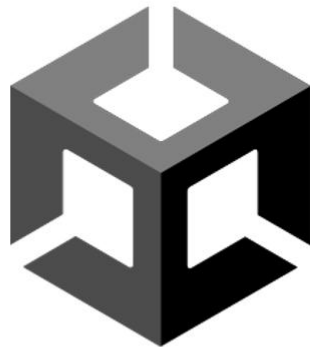
1. Аналіз наявних методів віртуальної та доповненої реальності на смартфонах, визначення їх переваг і недоліків та розроблення власного методу враховуючи отримані результати.
2. Визначення особливостей розробки програмного забезпечення.
3. Оцінка ефективності та аналіз результатів.

# ЗАПРОПОНОВАНИЙ МЕТОД



1. Тренування нейронної мережі LSTM та використання чіпу IMU для передбачення рухів голови.
2. Створення стерео зображення використовуючи показники глибини кадру через датчик LiDAR або вбудоване ПЗ.
3. Використання алгоритмів розмиття зображення для зниження навантаження на користувача.

# ІНСТРУМЕНТИ РОЗРОБКИ МЕТОДУ



**Unity**



# ОСОБЛИВОСТІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



1. Система активно використовує передбачення майбутніх станів смартфона. Завдяки LSTM-моделі, що отримує часові ряди з IMU-сенсорів, відбувається прогнозування орієнтації пристрою на кілька мілісекунд чи навіть секунд наперед.
2. Система використовує глибинні дані (з LiDAR або програмних алгоритмів розрахунку глибини) для повноцінної реконструкції сцени. На відміну від традиційних AR-рішень, які часто обмежуються розпізнаванням площин, представлене програмне забезпечення формує тривимірну карту оточення.

# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



**Dataset Collector** – модуль для збору даних руху голови користувача безпосередньо з мобільного пристрою, нормалізації та відбору найякісніших зразків.

**Model Trainer** – компонент, що оперує на високопродуктивному середовищі та відповідає за навчання нейронної мережі прогнозування руху на основі зібраних даних.

**Trajectory Predictor** – модуль, вбудований у додаток на смартфоні, який використовує натреновану модель для передбачення майбутнього положення голови в режимі реального часу.

**Mesh Builder** – сервіс, що формує тривимірний меш реального оточення для подальшого коректного відтворення віртуальних елементів.

**Scene Composer** – компонент, який інтегрує прогноз руху та тривимірні дані оточення в єдину змішану сцену, гармонійно поєднуючи реальний світ з віртуальним контентом.

**Stereo Rendered** – модуль, що відповідає за побудову стереоскопічного зображення з урахуванням положення голови, забезпечуючи повноцінний VR-досвід.

Взаємодія між компонентами організована через стандартизовані інтерфейси та протоколи обміну даними, а кожен модуль фокусується на вирішенні конкретної задачі, що сприяє підвищенню надійності, продуктивності та масштабованості системи.

## НАУКОВА НОВИЗНА



Розроблено метод передбачення рухів голови з використанням нейронної мережі LSTM, який забезпечує плавну роботу розробленого застосунку з точністю менше 30 мм та затримкою до 20 мс.

# ПРАКТИЧНА ЦІННІСТЬ



Практична цінність роботи полягає у створенні ефективної системи поєднання віртуальної та доповненої реальності на смартфонах.

Розроблене програмне забезпечення має широкий спектр застосувань у таких галузях, як освіта, дизайн, медицина, промисловість та ігровий розвиток, і може стати основою для створення нових інструментів взаємодії з цифровим контентом.

# Апробація



1. Основні положення і результати даної роботи були представлені на XVII науковій конференції магістрантів та аспірантів «Прикладна математика та комп'ютинг» ПМК-2024 (Київ, 20-22 листопада 2024 р.).

## ПОДАЛЬША РОБОТА



Подальші дослідження можуть бути зосереджені на удосконаленні архітектури моделі LSTM для зменшення обчислювальної складності, що дозволить ефективніше використовувати її на мобільних пристроях.

Також перспективним напрямком є розробка адаптивних алгоритмів, що зможуть автоматично підлаштовуватися до унікальних особливостей руху кожного користувача, що підвищить точність і швидкість прогнозування у специфічних сценаріях використання.

## ВИСНОВКИ



1. Сформульовано новий метод поєднання віртуальної та доповненої реальності на смартфонах з розробкою мікросервісної архітектури мобільного застосунку.
2. Розроблено програмне забезпечення, що включає в себе мобільний застосунок, метод передбачення рухів голови за допомогою нейронної мережі LSTM та 7 внутрішніх мікросервісів.
3. Розроблене програмне забезпечення має широкий спектр застосувань у таких галузях, як освіта, дизайн, медицина, промисловість та ігровий розвиток, і може стати основою для створення нових інструментів взаємодії з цифровим контентом.



*Дякую за увагу!*