

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Програмної інженерії
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

Дослідження методів інтеграції доповненої реальності з мультимедійними
засобами навчання на мобільній платформі iOS
(тема)

Виконав: студент 2 курсу, групи ППЗм-17-2
спеціальності 121- Інженерія програмного
забезпечення

(код і повна назва спеціальності)

освітньо-наукової програми Інженерія
програмного забезпечення

(повна назва освітньої програми)

Гончаров О.В.

(прізвище, ініціали)

Керівник к.т.н., доц. Лановий О.Ф.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри, проф. _____

З.В.Дудар

2019 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Програмної інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 121– Інженерія програмного забезпечення
(код і повна назва)

Освітньо-наукова програма Інженерія програмного забезпечення
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«___» _____ 20__ р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові Гончарову Олександрову Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів інтеграції доповненої реальності з мультимедійними засобами навчання на мобільній платформі iOS затверджена наказом по університету від “___” _____ 20__ р № _____
заповнюється вручну після отримання наказу
2. Термін подання студентом роботи до екзаменаційної комісії «___» _____ 2019 р.
3. Вихідні дані до роботи концепція доповненої реальності, компоненти, вимоги системи, що буде розроблено, середовище проектування Xcode, технологія ARKit, програмування Swift, фреймворк для роботи з базами даних Core Data.
4. Перелік питань, що потрібно опрацювати в роботі мета роботи, аналіз проблемної галузі та постановка задачі, дослідження засобів та технологій доповненої реальності на мобільній платформі iOS, аналіз комп'ютерного бачення в доповненій реальності, UML проектування системи, проектування архітектури мобільного додатку, вибір інструментів розробки.
5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Титульний слайд, мета роботи, постановка задачі, аналіз існуючих застосувань, класифікація доповненої реальності, формула зв'язку координат відповідних точок знімка камери і місцевості, UML-моделювання, Use case діаграма, діаграма послідовності, засоби реалізації, інтерфейс користувача, висновки.

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Спецчастина	к.т.н., доц. Лановий О.Ф.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз проблемної галузі		
2	Розробка моделі предметної галузі		
3	Розробка структури зберігання даних		
4	Створення коду програми		
5	Тестування і налагодження програми		
6	Підготовка пояснювальної записки		
7	Підготовка презентації та доповіді		
8	Попередній захист		
9	Нормоконтроль, рецензування		
10	Занесення диплома в електронний архів		
11	Допуск до захисту у зав. кафедри		

Дата видачі завдання _____ 2019 р.

Студент _____ Гончаров.О.В.
(підпис)

Керівник роботи _____ к.т.н., доц. Лановий О.Ф.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка до атестаційної роботи: 67 с., 28 рис., 18 джерел, 2 додатки.

ДОПОВНЕНА РЕАЛЬНІСТЬ, ОСВІТА, МУЛЬТИМЕДІА, ІНЕРЦІЙНА СИСТЕМА, iOS, SWIFT, AR, VR

Об'єктом дослідження є засоби та технології доповненої реальності в мультимедійних засобах навчання, алгоритми відстеження позиції у просторі.

Метою роботи є дослідження технології AR, проектування та розробка мобільного додатку з використанням технології доповненої реальності на основі маркерів.

У результаті роботи проведений аналіз предметної галузі, досліджені основні концепції та принципи роботи доповненої реальності в сфері навчання, проведений аналіз засобів доповненої реальності на мобільній платформі iOS. Також був написаний мобільний додаток з застосуванням знань, отриманих в результаті дослідження.

AUGMENTED REALITY, EDUCATION, MULTIMEDIA, INERTIAL SYSTEM, iOS, SWIFT, AR, VR

The object of the research is the means and technologies of the augmented reality in multimedia means of education, algorithms of position tracking in space.

The aim of the task is to study the technology of AR, design and development of a mobile application using the technology of augmented reality based on markers.

As a result of the work, the analysis of the subject area was carried out, the main concepts and principles of the work of the complementary reality in the field of education were explored, the analysis of the complemented reality on the mobile platform of iOS was conducted. A mobile application was also written using the knowledge gained through the study.

ЗМІСТ

Вступ.....	6
1 Аналіз предметної галузі та постановка задачі	8
1.1 Концепція доповненої реальності.....	8
1.2 Класифікація доповненої реальності.....	10
1.3 Аналіз існуючих застосувань доповненої реальності в освіті та суміжних галузях.....	14
1.4 Огляд принципу роботи доповненої реальності на мобільній платформі iOS.....	18
1.5 Постановка задачі	24
2 Комп'ютерне бачення в доповненій реальності.....	25
2.1 Поняття комп'ютерного бачення.....	25
2.2 Методи комп'ютерного бачення в доповненій реальності.....	28
3 Проектування програмної системи	33
3.1 UML проектування програмної системи.....	33
3.2 Архітектура мобільного додатку.....	37
4 Опис програмної реалізації.....	39
4.1 Вибір засобів розробки.....	39
4.2 Опис мобільного додатку.....	41
4.3 Тестування програмної системи.....	50
Висновки.....	52
Перелік джерел посилання.....	55
Додаток А.....	56
Додаток Б.....	64

ВСТУП

В контексті сучасного технологічного розвитку у світі, інновації є ключовим рушієм до позитивних змін в усіх сферах життєдіяльності. Усього за декілька десятиліть людина повністю змінила розуміння того, що таке інформація і як вона може використовуватися для поліпшення або спрощення деяких аспектів життя. Це призвело до зміни парадигм та шляхів виконання щоденних задач. Наприклад, якщо людина хоче отримати нову інформацію, вона відкриває пристрій з доступом до мережі Інтернет та за допомогою пошукових систем знаходить усю потрібну інформацію за декілька хвилин. Раніше, це можна було зробити тільки відвідавши бібліотеку. Насамперед, ці зміни вплинули на процеси та підходи отримання освіти. Більш за це, процес змін триває і на зараз та є викликом для дослідників, викладачів та інших працівників сфери освіти. Швидкість зміни технологій і підходів до навчання відрізняється. Якщо технології прогресують стабільно та без зупинок, підходи все ще потребують новаторства та інновацій. Прийшовши до поточного сценарію, персональні комп'ютери перетворили спосіб спілкування людей зі світом, відкрили можливість отримувати інформацію у новому вигляді, котрий був недосяжний до цього. Доповнена реальність (augmented reality, AR), інтегруючи краще з обох світів, може служити чудовим засобом зміни підходів до навчання та отримання інформації, бо відкриває можливість вводу інтерактивності у ці процеси.

Термін доповнена реальність є однією з мутацій віртуальної реальності (virtual reality, VR). Вона може бути сприйнята як ініціювання штучних стимулів над реальними за допомогою мультисенсорних технологій. Іншими словами, вона включає віртуальну інформацію про людські відчуття, покращуючи або модифікуючи сприйняття навколишнього середовища людиною. Основними характеристиками доповненої реальності є:

- наявність віртуальних елементів у реальному контексті;

- інтерактивність з реальністю;
- домінація віртуальних елементів в реальному контексті для зміни сприйняття інформації людськими відчуттями у 3D просторі.

Те, що робить технологія доповненої реальності – це створення добровільної ілюзії, яку ми використовуємо, щоб уточнити відчуття зору. Завдяки цьому ми можемо включати нову інформацію про оточення на нашому сайті або мапі городу, відкритій на смартфоні. Як результат, ця технологія легко реалізується в освіті.

Наприклад, студент може подивитися модель двигуна автомобіля в посібнику у вигляді кольорової картинки. Але якщо ми додамо елемент доповненої реальності, то інтерактивність інформації підвищиться у декілька разів. Крім картинки, студент буде мати можливість вивчити модель автомобіля у 3D просторі замість 2D зображення. Він запам'ятає місце розташування кожного елемента двигуна відносно реальних 3D координат з більшою вірогідністю. Також, доповнена реальність може надати можливість відокремити кожну деталь двигуна та роздивитися її незалежно, що також поліпшить процес запам'ятовування влаштування двигуна та розуміння принципу його роботи. Цей приклад є лише одним з багатьох, котрі демонструють яким чином доповнена реальність може поліпшити подання та сприйняття інформації в зрівнянні з традиційними методами.

Метою роботи є дослідження методів інтеграції доповненої реальності з мультимедійними засобами навчання на платформі iOS та створення стратегій їх ефективного використання, проектування та розробка мобільного додатку використовуючи технології доповненої реальності.

Було зроблено доповіді на Міжнародному молодіжному форумі. Тези доповідей наведено в додатку Б.

1 АНАЛІЗ ПРОБЛЕМНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Концепція доповненої реальності

Термін доповнена реальність був запропонований працюючим на корпорацію Boeing дослідником Томом Коделом в 1990 р. Термін позначає усі технології та проекти, спрямовані на доповнення реальності будь-якими віртуальними елементами [1]. Доповнена реальність є одним з окремих сценаріїв «змішаної реальності». В загальному сенсі доповнена реальність – це ситуація, в котрій в зображення реального світу додані віртуальні об'єкти, тобто основою для побудови зображення або відеоряду є реальний контекст котрий ми бачимо за допомогою зору. Змішана реальність також включає в себе доповнену віртуальність. Тобто, доповнена реальність є найпростішим випадком змішаної реальності. Схему змішаної реальності наведено на рисунку 1.1

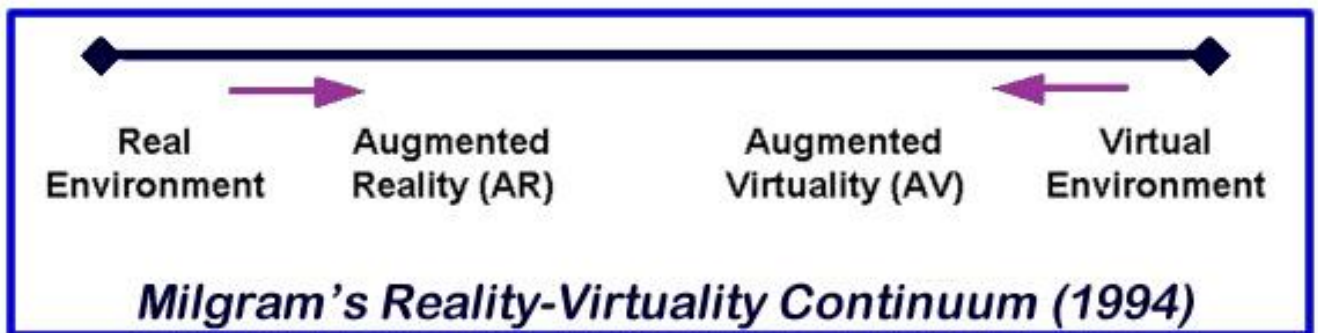


Рисунок 1.1 – Схема змішаної реальності

Суттєвою відмінністю доповненої реальності від віртуальної реальності є збереження фізичного стану навколишнього середовища як контексту, в якому представлені віртуальні об'єкти. В той час, віртуальна реальність повністю абстрагується від фізичного світу та поміщає користувача у віртуальний світ повністю. Щоб доповнена реальність працювала, користувач повинен мати можливість бачити навколишній світ таким, який він є зараз. AR маніпулює

реальним світовим простором, який бачить користувач, змінюючи сприйняття користувачем реальності.

Однією з форм AR, користувач спостерігає живий запис реального світу з віртуальними елементами, накладеними поверх нього. Багато спортивних заходів використовують цей тип AR: глядач може переглядати гру в прямому ефірі з власного телевізора, а також бачити результати, накладені на ігрове поле.

Інша форма AR дозволяє користувачеві оглядати навколишнє середовище нормально і в реальному часі, але за допомогою дисплея, який накладає інформацію для створення розширеного досвіду. Прикладом цього може служити Google Glass, які є схожими на звичайну пару окулярів, але містять невеликий екран, на якому користувач може бачити позицію за карті, перевіряти погоду, надсилати фотографії та багато інших функцій.

Коли віртуальний об'єкт розміщений між користувачем і реальним світом, розпізнавання об'єктів і комп'ютерний зір можуть бути використані для того, щоб дозволити об'єкту маніпулювати реальними фізичними об'єктами і дозволити користувачеві взаємодіяти з віртуальними елементами.

Наприклад, деякі мобільні додатки для роздрібної торгівлі дозволяють покупцям вибирати віртуальну версію того, що вони розглядають як купівлю, наприклад, предмет меблів і переглядати його в реальному просторі свого будинку через свій телефон. Наприклад, вони бачать свою власну вітальню, але віртуальний диван, який вони вибрали, тепер видно їм на екрані, дозволяючи їм вирішувати, чи зможе він вміститися в цій кімнаті та як він виглядає в умовах існуючого інтер'єру кімнати.

Інший приклад дозволяє клієнтам сканувати продукцію або спеціальні коди (наприклад, символи UPC), які використовують AR, щоб показати клієнтам більше інформації про фізичний продукт, перш ніж вони його придбають, наприклад, відгуки інших покупців або контент закритого пакету.

1.2 Класифікація доповненої реальності

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

- доповнена реальність на основі маркерів (Marker-based AR);
- доповнена реальність без маркерів (Marker-less AR);
- проекційна доповнена реальність (Projection AR);
- доповнена реальність на основі накладання (Superimposition based AR).

Доповнена реальність на основі маркерів також називається розпізнаванням зображень або розпізнаванням на основі AR. Приклад наведено на рисунку 1.2.



Рисунок 1.2 – Приклад AR технології, що базується на маркерах

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

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

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

- переводить слова, які фіксує камера, за допомогою технології оптичного розпізнавання символів і показує їх у перекладеній версії;

- технологія також корисна в освіті, оскільки вона може допомогти учням перетворити свою уяву в реальність;

- допомагає у створенні 3D моделей об'єктів або архітектури без побудови об'єкту у фізичній формі.

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

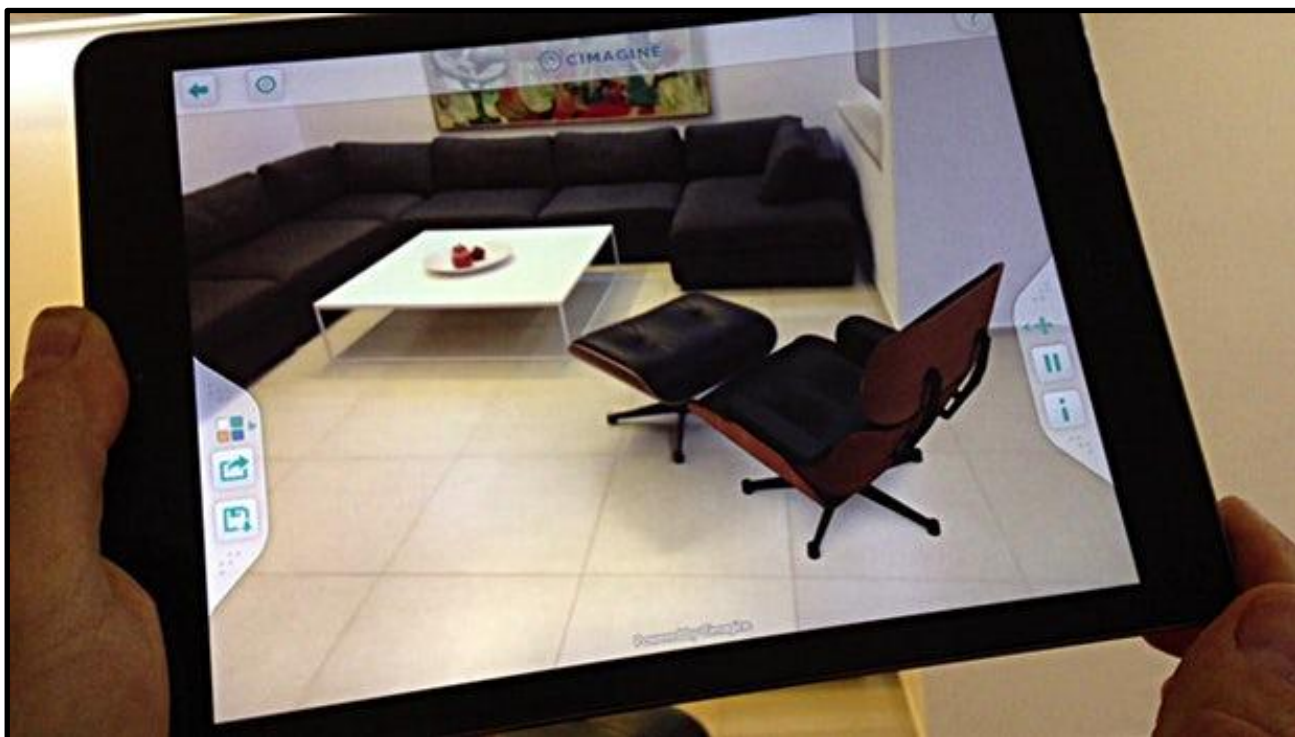


Рисунок 1.3 – Приклад доповненої реальності без маркерів

Це один з найбільш широко використовуваних і впроваджених типів технологій. Це єдиний тип, котрий не використовує ніякої системи розпізнавання існуючого об'єкту. Саме тому система не знає про об'єкт нічого, в той час як AR на

основі маркерів повинно знати. Щоб забезпечити ефективний результат, ця технологія використовує різні інструменти розміщення та допомагає обчислити розташування за допомогою сенсорів пристрою. Наприклад, ви можете помістити віртуальні меблі до будь-якої кімнати, бо системі все одно куди вони поставлені.

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

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



Рисунок 1.4 – Приклад проекційної доповненої реальності

Існують численні підходи, які роблять цю технологію більш цікавою. Один з підходів полягає в тому, щоб проектувати світло на поверхню реального

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

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

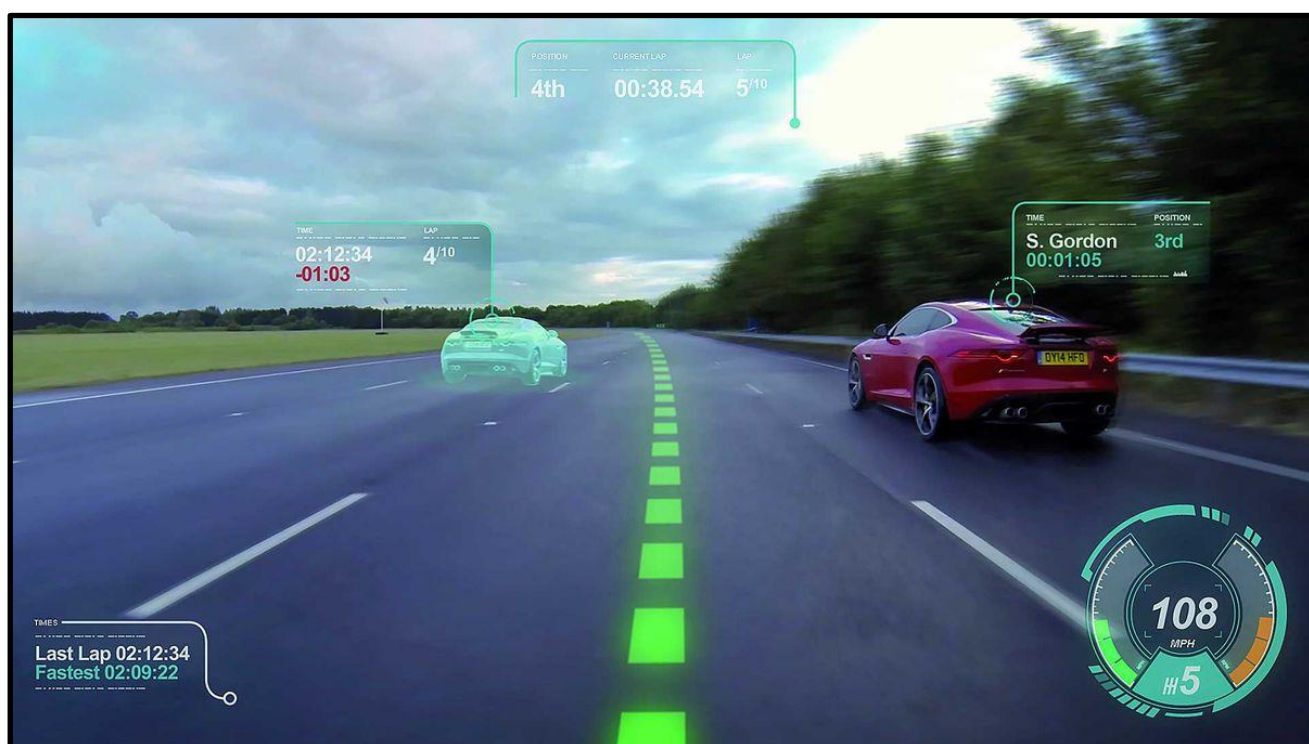


Рисунок 1.5. – Доповнена реальність на основі накладання

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

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

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

1.3 Аналіз існуючих застосувань доповненої реальності в освіті та суміжних галузях

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

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

можуть супроводжуватися звуком, що створює більш глибокий досвід занурення в при використанні додатку.



Рисунок 1.6. – Мобільний додаток Dinosaur 4D+

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

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

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

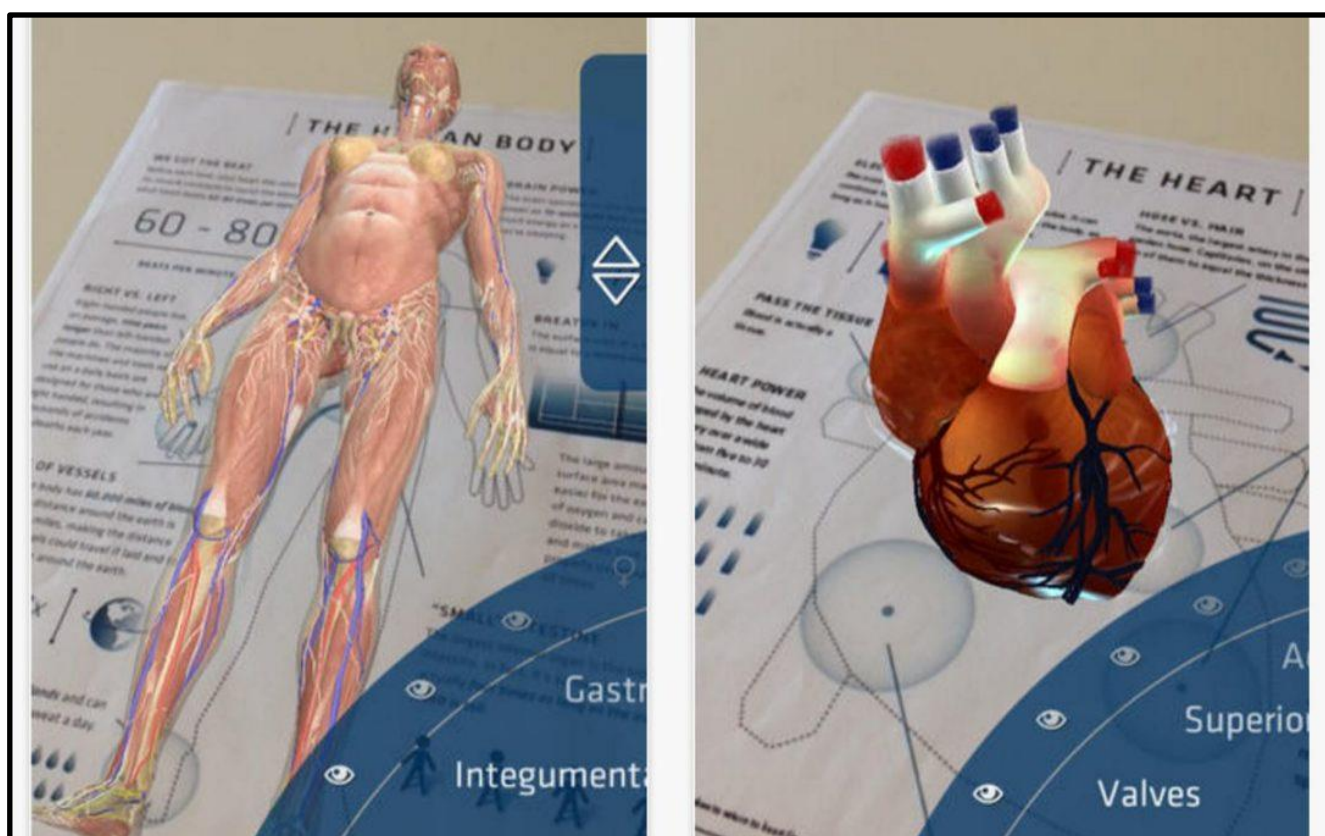


Рисунок 1.7 – Знімки екрану мобільного додатку Anatomy 4D

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

наводить фокус на текст, надрукований мовою оригіналу. Додаток замінює цей текст на англійський текст і створює ілюзію, нібито на поверхні цей текст надруковано англійською. Додаток використовує доповнену реальність на основі накладання. Приклад живого перекладу додатку Google Translate наведено на рисунку 1.8.



Рисунок 1.8 – Приклад живого перекладу за допомогою Google Translate

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

1.4 Огляд принципу роботи доповненої реальності на мобільній платформі iOS

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

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

Dead Reckoning використовується для вимірювання руху пристрою між показаннями IMU. Dead Reckoning – це припущення, на кшталт вимірювання дистанції за допомогою людського ока, але помилка є набагато меншою. Таким чином, візуальні та інерційні системи відстеження базуються на абсолютно різних системах вимірювань без взаємної залежності. Це означає, що фотокамера може бути закрита або може переглядати сцену з декількома оптичними функціями (наприклад, білою стіною), а інерційна система в цей час буде брати

«навантаження» на себе. В альтернативному сценарії пристрій може залишатися на місці і візуальна система буде отримувати більш стабільне розташування в просторі, ніж інерційна система. Фільтр Калмана постійно вибирає найкращу позицію з двох, тому результатом є стабільне відстеження. Приклад наведено на рисунку 1.9.

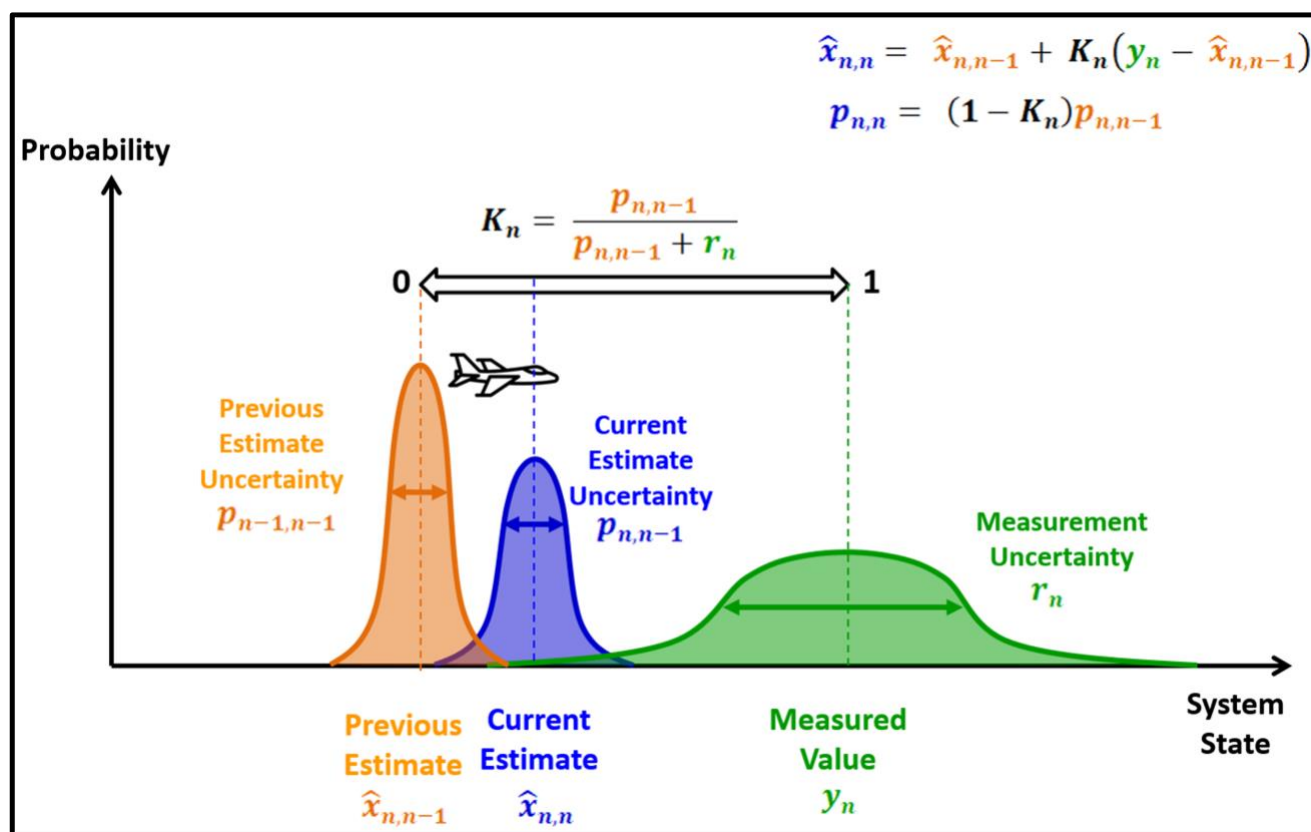


Рисунок 1.9 – Приклад роботи фільтра Калмана

В цілому, ARKit можна розділити на три компоненти. Перший – це відстеження, коли ARKit використовує трекінг реального світу, тобто технологія здатна знаходити положення користувача щодо навколишнього середовища. Апаратне забезпечення Apple використовує алгоритми комп'ютерного зору «Core Motion», що допомагає пристрою оцінювати відносну позицію за допомогою камери. У поєднанні з сенсорами та датчиками, наприклад акселерометром, гірометром і магнітометром, «Core Motion» здатний виконувати трекінг реального світу. Другий – розуміння сцени. Це компонент ARKit, який допомагає зрозуміти

компоненти навколишнього середовища, такі як горизонтальні поверхні та навколишнє освітлення. Третій – рендерінг. Це компонент у ARKit, який дозволяє відтворювати різні типи 2D/3D-контенту з відповідними текстурами. На рисунку 1.10 зображені напрями відстеження акселерометру та гіроскопу.

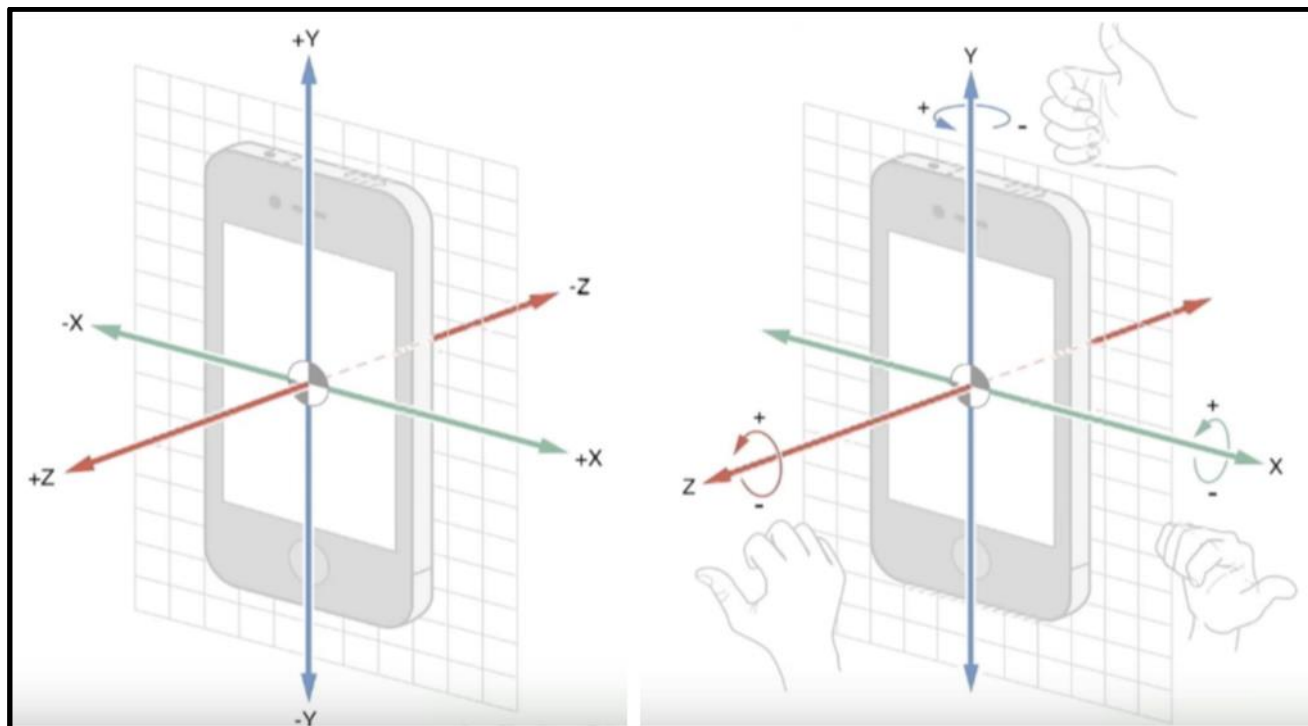


Рисунок 1.10 – Принцип роботи акселерометру та гіроскопу

Давайте розберемо ці компоненти, щоб зрозуміти, як вони працюють. Відстеження світу – це здатність ARKit зрозуміти середовище та інтерпретувати відносне розташування користувача. ARKit використовує спеціальний код для інтерпретації даних камери та даних з датчиків пристрою, таких як акселерометр, гіроскоп і магнітометр. На цьому етапі системна бібліотека «Core Motion» інтерпритує дані, отримані з сенсорів та датчиків. Дані з акселерометру дають можливість інтерпретації руху девайсу у тривимірному просторі за трьома осями, дані гіроскопу інтерпретуються для розуміння повороту девайсу щодо одної з трьох осей. Магнітометр, в свою чергу, надає можливість зрозуміти орієнтацію девайсу щодо універсального північного полюсу.

Наступний компонент – розуміння сцени. Цей компонент складається з декількох підсистем: виявлення площини, «Hit-testing» та оцінка освітлення. ARKit розпізнає площину на основі контрастної піксельної інформації в даній сцені. Технологія намагається інтерпретувати горизонтальну поверхню. Деякі інтелектуальні функції включають злиття площин, що означає, що вона може розуміти кілька поверхонь, але може об'єднати ці поверхні, якщо вони розміщені на одному і тому ж об'єкті. Приклад наведено на рисунку 1.11.

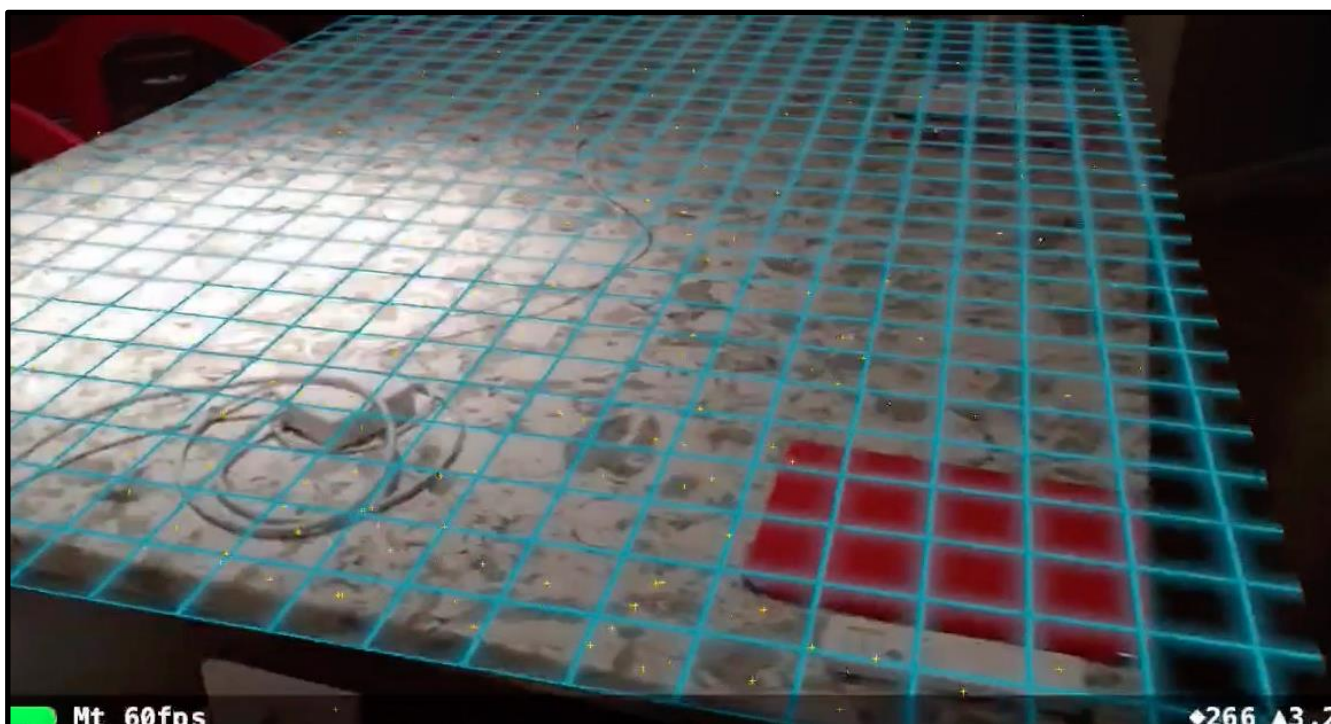


Рисунок 1.11 – Візуалізація роботи підсистеми виявлення площини

«Hit-testing» – це спосіб для ARKit зрозуміти середовище і підготуватися шляхом розуміння деяких опорних точок в просторі (див.рис. 1.12). Після цього він використовує ці точки прив'язки для позначення на них цифрових об'єктів. ARKit зазвичай рендерить опорні точки як жовті крапки. Коротко, «Hit-testing» є камерою пристрою, посилаючою промінь для виявлення фізичного світу і сортуючою кожену точку за відстанню у цьому світі. Це також спосіб для ARKit зрозуміти глибину середовища.

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

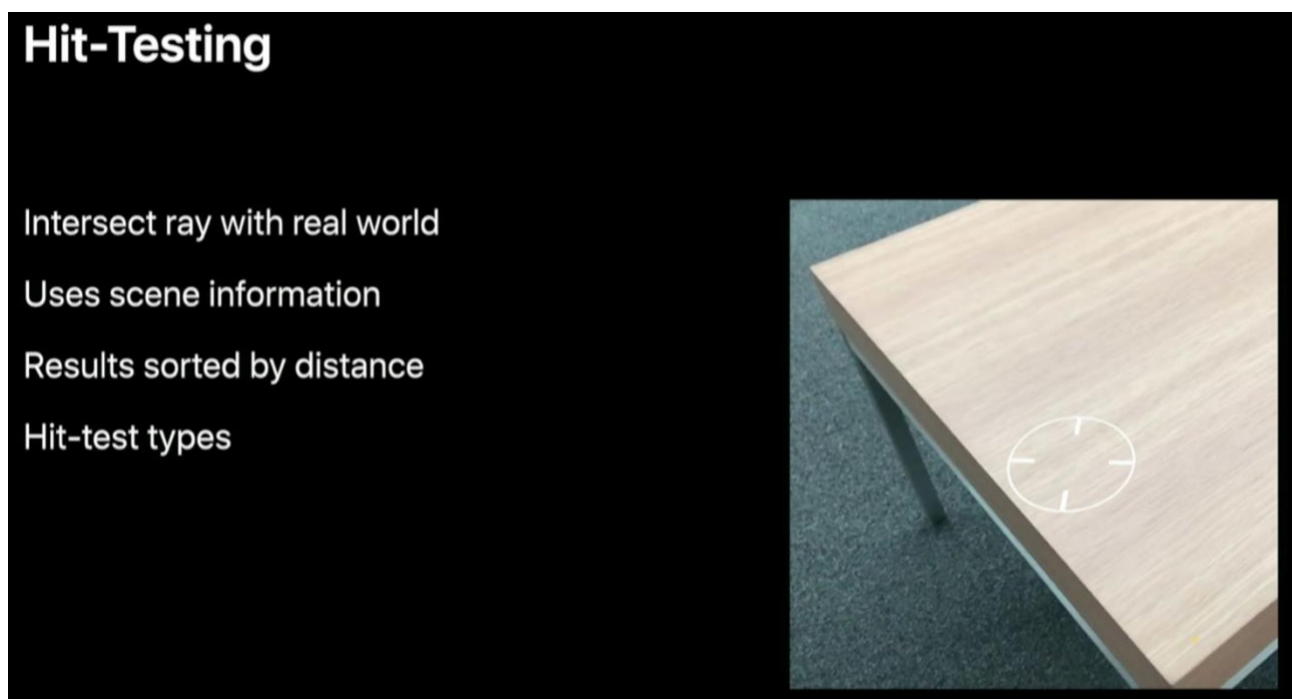


Рисунок 1.12 – Принцип роботи «Hit-testing»

Рендерінг – це останній значний компонент в технології доповненої реальності. ARKit використовує існуючі структури Apple для створення та надання активів. Розробники мобільних додатків можуть створювати контент, який ефективно використовує апаратне забезпечення Apple. Для рендерінгу можуть використовуватися декілька різних фреймворків: SpriteKit, SceneKit та Metal. В залежності від задачі, розробник може вибрати один з них.

SpriteKit є фреймворком Apple для створення 2D візуального контенту. У поєднанні з ARKit це призводить до створення 2D компонентів, які дотримуються

білбордингу. Білбординг – це, коли компоненти завжди орієнтуються на камеру. Можна створити глибину відображення з SpriteKit просто регулюючи масштаб.

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

Metal – це спеціальна бібліотека текстуровання, яку Apple пропонує розробникам. Те, що пропонує Metal – це покращений рендерінг для пристроїв Apple порівняно з OpenGL. За допомогою Metal можна створити текстури будь-якої складності, але він більш складний, ніж SpriteKit чи SceneKit. Більш того, SpriteKit та SceneKit побудовані на основі Metal. Саме тому він використовується для розробки складних мобільних додатків чи мобільних ігор. На рисунку 1.13 зображені логотипи перелічених фреймворків.



Рисунок 1.13 – Логотипи фреймворків SceneKit, SpriteKit та Metal

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

вирішувати складні завдання мотивації навчання та допомагати мільйонам студентів на всій планеті.

1.5 Постановка задачі

Метою атестаційної роботи є дослідження засобів та технологій доповненої реальності в мультимедійних засобах навчання на платформі iOS. Результатом дослідження повинна стати розробка мобільного додатку з використанням технології AR заснованої на маркерах та застосуванні знань, здобутих під час дослідницької роботи. Розроблена система повинна покращувати процес сприйняття інформації за допомогою додання віртуальних об'єктів до реального середовища.

Вимогою для забезпечення роботи мобільного додатку буде наявність у користувача мультимедійного пристрою з мобільною операційною системою iOS 11.3 або новішою. В подальшому, мобільний додаток планується для розміщення в App Store.

Функціонал додатку повинен бути наступний:

- авторизація користувача за допомогою паролю чи відбитку пальця, якщо ця процедура встановлена;
- додавання маркерів до каталогу розпізнавання;
- редагування каталогів розпізнавання;
- розпізнавання об'єктів за допомогою маркерів та виведення повної інформації на екран в режимі реального часу використовуючи функції AR;
- експорт та імпорт каталогів між сумісними девайсами.

Відповідно до аналізу предметної галузі у першому розділі, використання доповненої реальності в освіті є дуже пріоритетним та перспективним напрямом діяльності, саме тому аналіз технологій, методів доповненої реальності та створення мобільного додатку на мобільній платформі iOS є актуальним.

2 КОМП'ЮТЕРНЕ БАЧЕННЯ В ДОПОВНЕНІЙ РЕАЛЬНОСТІ

2.1 Поняття комп'ютерного бачення

Комп'ютерне бачення – це сфера комп'ютерних наук, яка працює над тим, щоб дозволити комп'ютерам бачити, ідентифікувати та обробляти зображення таким же чином, як і людське бачення, а потім забезпечувати відповідний вихід. Це подібно до передачі комп'ютеру людського інтелекту та інстинктів [8]. Насправді, це важке завдання, щоб дозволити комп'ютерам розпізнавати зображення різних об'єктів. Комп'ютерне бачення тісно пов'язане з штучним інтелектом, оскільки комп'ютер повинен інтерпретувати те, що він бачить, а потім виконувати відповідний аналіз або діяти відповідно.

У 1960-х роках проект мав на меті досягти комп'ютерного зору за допомогою камери. Ідея була проста: описати, що бачила камера. У 1970-х рр. було створено багато алгоритмів комп'ютерного зору, таких як вилучення країв, маркування ліній, небагатогранне і багатогранне моделювання, представлення об'єктів як взаємозв'язків менших структур, оптичний потік і оцінка руху. Сьогодні, з розвитком методів штучного інтелекту та машинного навчання, у багатьох додатках використовуються системи комп'ютерного зору. Особливо глибоке навчання, метод машинного навчання, прискорює поліпшення комп'ютерного зору. Нейронні мережі використовують підхід навчання, а не програмування. Цей підхід полегшує обробку величезної кількості зображень і відеоданих, доступних у сучасних системах.

Однак, для обробки складного обчислювального процесу, як на етапах навчання, так і на висновках, потрібне потужне обладнання. Наприклад, щоб підвищити точність від 75% до 80%, необхідні мільярди додаткових математичних операцій для застосування в комп'ютерному зорі. Розмір зображення є одним з найважливіших елементів комп'ютерного зору. Розмір безпосередньо впливає на здатність систем комп'ютерного зору виявляти та класифікувати об'єкт. У додатках, які вимагають виявлення та класифікації об'єктів на відстані, такі як камери

безпеки, системи спостереження, вирішальне значення має розмір зображення, а саме кількість пікселів по ширині та висоті.

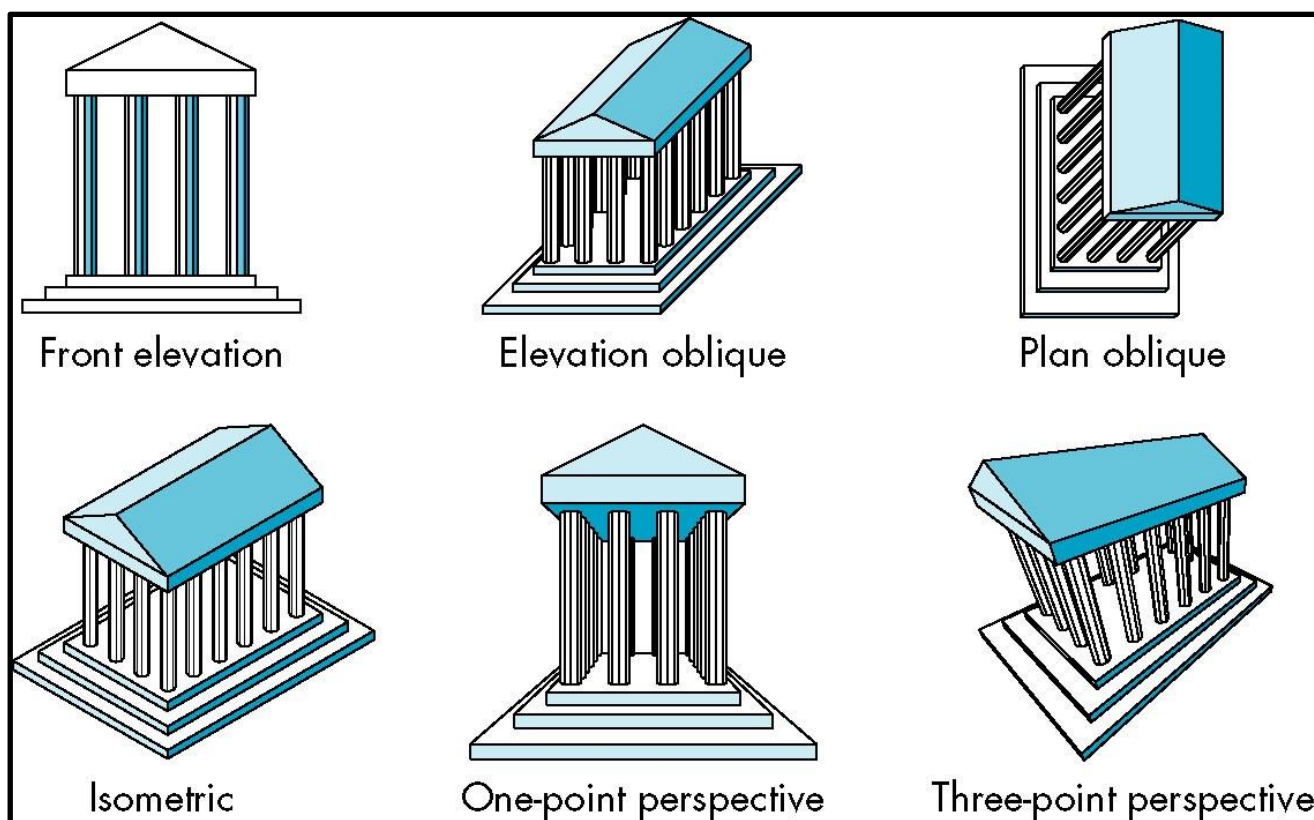


Рисунок 2.1 – Деякі шляхи проєкції 3D об'єкту в 2D

Важливість комп'ютерного бачення полягає в проблемах, які воно може вирішити. Це одна з основних технологій, яка дозволяє цифровому світу взаємодіяти з фізичним світом. Комп'ютерне бачення дає змогу самостійного управління автономним автомобілями, надаючи їм можливість зрозуміти їхнє оточення. Камери знімають відео з різних кутів навколо автомобіля і подають його до програмного забезпечення комп'ютерного зору, яке потім обробляє зображення в режимі реального часу, щоб знайти край дороги, прочитати дорожні знаки, виявити інші автомобілі, об'єкти і пішоходів. Самостійно керуючий автомобіль може тоді прокласти свій шлях на вулицях і автомагістралях, уникнути попадання перешкод і безпечно вести своїх пасажирів до місця призначення.

Комп'ютерне бачення також відіграє важливу роль у сфері розпізнавання обличь, технології, що дозволяє комп'ютерам поєднувати зображення обличь

людей з їх ідентичністю. Алгоритми комп'ютерного зору виявляють риси обличчя у зображеннях та порівнюють їх з базами даних профілів обличчя. Споживчі пристрої використовують розпізнавання обличчя для автентифікації ідентифікаторів своїх власників. Програми для соціальних медіа використовують розпізнавання обличчя для виявлення та позначення користувачів. Правоохоронні органи також покладаються на технологію розпізнавання осіб для виявлення злочинців у відео-каналах.

Комп'ютерне бачення також відіграє важливу роль у доповненій та змішаній реальності, технології, що дозволяють обчислювальним пристроям, таким як смартфони, планшети та смарт-окуляри, накладати та вставляти віртуальні об'єкти у зображення реального світу. Використовуючи комп'ютерне бачення, прилад AR виявляє об'єкти в реальному світі, щоб визначити місця на дисплеї пристрою та розмістити там віртуальний об'єкт. Наприклад, алгоритми комп'ютерного зору можуть допомогти AR додаткам виявити такі площини, як стільниці, стіни і підлоги, що є дуже важливою частиною розуміння глибини і розміру середовища та в подальшому розміщення віртуальних об'єктів у фізичному світі.

Комп'ютерне бачення в значній мірі спирається на діаграми Венна з цифровою обробкою зображень. Наступні задачі підпадають під сферу обробки цифрових зображень:

- розпізнавання образів: автоматизований пошук закономірностей в масиві даних зображення;
- вилучення ознак: розбиття зображення на різні ознаки;
- класифікація: віднесення певного кластеру ознак та артефактів зображення до об'єкту фізичного світу;
- багатоканальний аналіз сигналу: відповідає на питання, чи можна бачити зображення по іншому;
- графічна проекція: правила представлення 3D об'єкту у 2D проекції (див. рис. 2.1).

2.2 Методи комп'ютерного бачення в доповненій реальності

Комп'ютерне бачення перетворює 3D віртуальні об'єкти так само, як знімки реальної сцени приймаються камерами спостереження. Реєстрація зображень з доповненою реальністю використовує різні методи комп'ютерного зору, які в основному пов'язані з відстеженням відео. Ці методи зазвичай поділяються на два етапи: відстеження та реконструкція / розпізнавання. Під час відстеження в зображеннях камери виявляються довідкові маркери, оптичні зображення або точки інтересу з використанням різних способів обробки зображення, таких як виявлення ознак або визначення країв. Більшість методів відстеження комп'ютерного зору можна розділити на два класи: відстеження на основі ознак і відстеження на основі моделі. Методи, що базуються на особливостях, полягають у виявленні зв'язку між 2D-ознаками зображення та їх 3D-координатами світового кадру [9]. Методи, що базуються на моделях, використовують модель бажаних ознак відстежуваних об'єктів, таких як моделі САПР або 2D-шаблони елемента, використовуючи 10 відмінних ознак. Після з'єднання між 2D-зображенням і 3D-кадром світу, позицію камери можна розрахувати, проєціювавши 3D-координати об'єкта в спостережувані координати 2D-зображення і мінімізуючи відстань до відповідних 2D-ознак. Оцінка позиції камери найчастіше використовує точкові ознаки для визначення відповідних обмежень. Етап реконструкції / розпізнавання використовує дані, отримані на першому етапі, для реконструкції системи координат реального світу.

Якщо взяти до уваги калібровану камеру і модель перспективної проєкції, коли точка має координати $(x, y, z)^T$ в рамці координат камери, її проєкція на площину зображення дорівнює $(\frac{x}{z}, \frac{y}{z}, 1)^T$.

У точкових обмеженнях існують дві основні системи координат: система координат світу W і система координат 2D зображення (див. рис. 2.2). Нехай $p_i(x_i, y_i, z_i)^T$, де $i = 1, \dots, n$, $n \geq 3$, є сукупністю 3D неколінеарних опорних точок

у координаті світового кадру та $q_i(x'_i, y'_i, z'_i)^T$ відповідні координати простору камери, де p_i і q_i пов'язані наступним перетворенням (2.1):

$$q_i = Rp_i + T, \quad (2.1)$$

де $R = \begin{pmatrix} r_1^T \\ r_2^T \\ r_3^T \end{pmatrix}$ – матриця обертання;

$T = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$ – вектор трансляції.

Нехай точка зображення $h_i(u_i, v_i, 1)^T$ є проекцією p_i на нормовану площину зображення. Рівняння колінеарності, що встановлює взаємозв'язок між h_i та p_i за допомогою отвору камери, наводиться нижче (2.2):

$$h_i = \frac{1}{r_3^T p_i + t_z} (Rp_i + T) \quad (2.2)$$

Помилка простору зображення дає взаємозв'язок між 3D опорними точками, їх відповідними 2D точками вилучення зображення і параметрами позиції камери і відповідає обмеженням точки. Помилка простору зображення подається наступним чином (2.3):

$$E_i^p = \sqrt{\left(\hat{u}_i - \frac{r_1^T p_i + t_x}{r_3^T p_i + t_z}\right)^2 + \left(\hat{v}_i - \frac{r_2^T p_i + t_y}{r_3^T p_i + t_z}\right)^2}, \quad (2.3)$$

де $\hat{m}_i(\hat{u}_i, \hat{v}_i, 1)^T$ – точки спостереження зображення.

У доповненій реальності більшість методів комп'ютерного зору передбачають наявність довірених маркерів або об'єктів з відомою тривимірною геометрією в середовищі і використовують ці дані. Статичні системи відомих початкових позицій заздалегідь розраховують структуру 3D-сцени, а динамічні системи використовують методику одночасного локалізації та картографування

(SLAM) для відображення позицій довірчих маркерів або 3D-моделей. У рідкісних, але зростаючих випадках, коли неможливо зробити жодних припущень щодо тривимірної геометрії сцени, використовується метод Structure from Motion (SfM). Метод SfM розділений на дві частини: відстеження точок особливості та оцінювання параметрів камери.

Одночасна локалізація та відображення (SLAM) – це синхронне визначення місця розташування та запис середовища на карті комп'ютера, пристрою, робота, безпілотної або іншого автономного транспортного засобу [10]. SLAM є ключовим компонентом в автономних автомобілях, та інших автономних роботах, що дозволяє усвідомлювати, де вони знаходяться, та будувати більш актуальні маршрути до місця призначення.

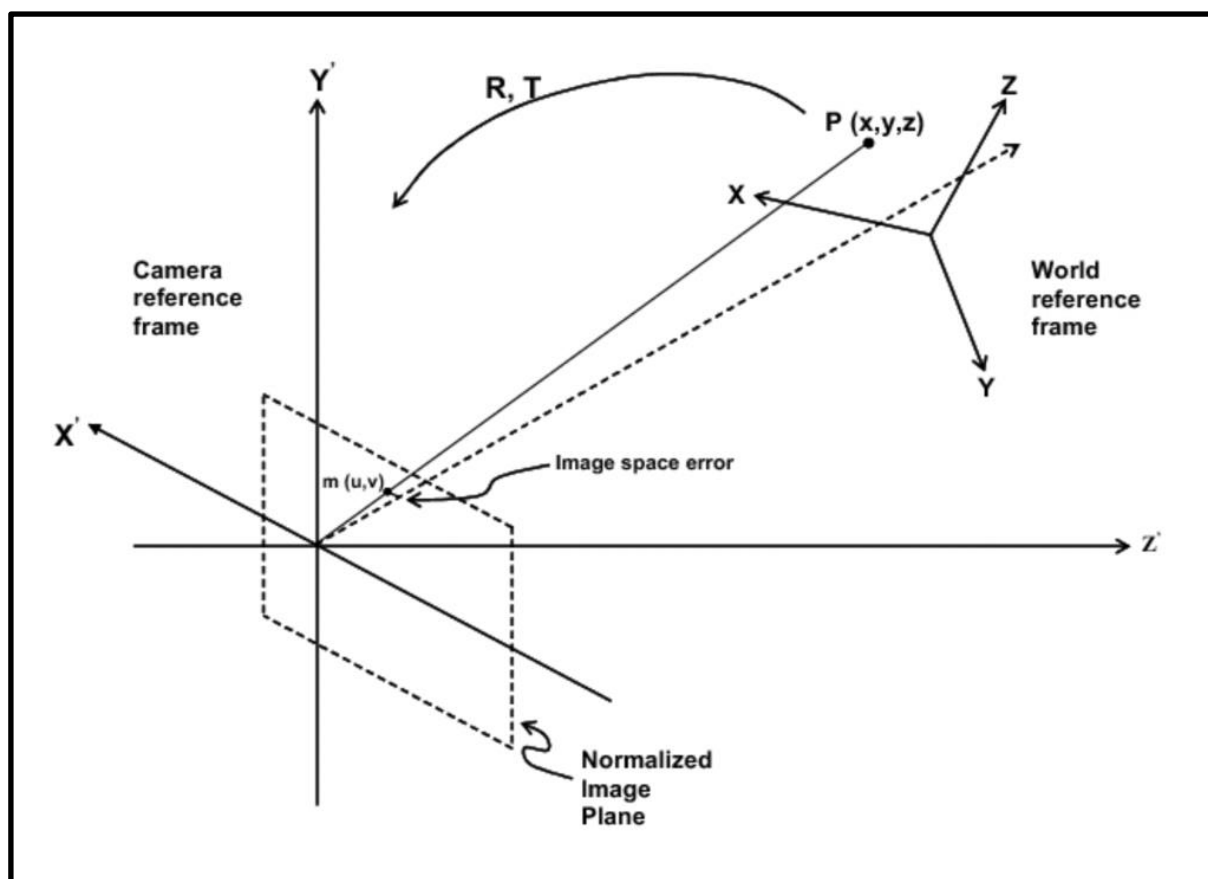


Рисунок 2.2 – Точкові обмеження для задачі розташування камери

Створюючи власні карти, SLAM забезпечує більш швидкий, більш автономний і адаптивний відгук, ніж попередньо запрограмовані маршрути. Ряд

випромінювачів і датчиків працюють разом з AI для однієї мети в SLAM. Робот, який використовує SLAM, наприклад, використовує різні типи камер і датчиків, таких як радар Lidar, ультразвукові та інші технології, щоб зрозуміти своє середовище. Краще зрозумівши навколишнє середовище, робот може більш ефективно відображати, переміщатися, уникати перешкод і пристосовуватися до змін. Високоточні модулі GPS зменшили потребу в SLAM в деяких додатках. Високоточний GPS може майже повністю замінити SLAM в деяких зовнішніх середовищах. Тим не менш, GPS може зазнати зниження продуктивності або відключень, і SLAM може заповнити прогалини в навігації, де потрібно більше деталей, а також узяти на себе відповідальність у разі труднощів. Принцип роботи SLAM для автономного автомобіля зображено на рисунку 2.3.

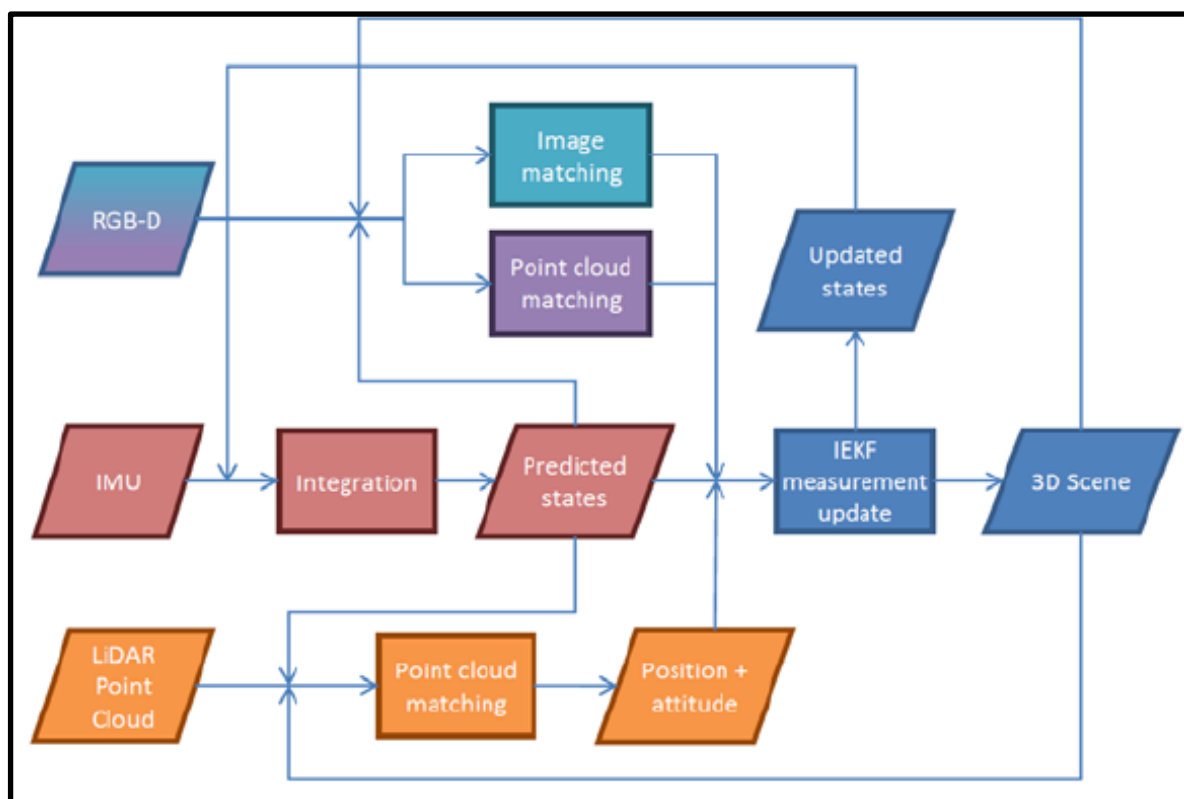


Рисунок 2.3 – Принцип роботи SLAM для автономного автомобіля з датчиком LIDAR

Методи відстеження доповненої реальності значною мірою залежать від типу середовища та типу побудованої системи AR. Знання того, що навколишнє

середовище може бути в приміщенні, на відкритому повітрі або в комбінації обох, завчасно диктуватиме можливість контролю системи блискавки, що є важливим фактором комп'ютерного зору. Аналогічним чином, чи буде побудована система AR статичною або динамічною, або ж середовище, в якому вона буде використана, буде відоме або невідоме, також будуть диктувати тип методів комп'ютерного зору для використання.

Хоча візуальне відстеження тепер має можливість розпізнавати і відстежувати багато речей, воно в основному спирається на інші методи, такі як GPS і акселерометри. Наприклад, для того, щоб комп'ютер виявляв і розпізнавав щось настільки знайоме і очевидне для людини, як автомобіль, для комп'ютеру це є досить складним завданням. Поверхня більшості автомобілів є одночасно і блискучою, і гладкою, і більшість точок особливості приходять з роздумів і, отже, не є актуальними для оцінки пози або навіть іноді розпізнавання об'єкта. Кілька стабільних ознак, які можна сподіватися розпізнати, такі як куточки вікон або колеса, надзвичайно важко підібрати через відбиття і прозорі деталі. Хоча цей приклад є трохи екстремальним, він показує труднощі та проблеми, з якими стикається комп'ютерний зір з більшістю об'єктів, що мають неправильну форму: їжа, квіти та більшість артефактів. Останнім часом новий підхід до вдосконалення у візуальному відстеженні полягав у вивченні того, як людський мозок розпізнає об'єкти, які також називають системою людського зору (HVS), оскільки люди можуть розпізнати нескінченну кількість об'єктів і осіб у частках секунд без будь-яких зусиль. Якщо зможемо змоделювати спосіб розпізнавання речей людським мозком, комп'ютерне бачення зможе впоратися з викликами, з якими зараз стикається, і продовжувати рухатися вперед.

Як результат, комп'ютерне бачення є невід'ємною частиною доповненої реальності, бо вона побудована на основі методів комп'ютерного бачення, тому розвиток останнього сильно впливає на покращення технологій AR.

3 ПРОЕКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

3.1 UML проектування програмної системи

UML – це аббревіатура, що означає «Єдина мова моделювання». Іншими словами, UML – це сучасний підхід до моделювання, візуалізації, опису та документування програмного забезпечення [11]. Він допомагає розкрити взаємозв'язок між діями користувача та відгуком програмної системи, взаємозв'язком між різними компонентами системи, тощо.

UML був створений в результаті хаосу, що обертався навколо розробки програмного забезпечення та документації. У 1990-х роках існувало кілька різних способів представлення та документування програмних систем. Виникла потреба більш уніфікованого способу візуального представлення цих систем, і в результаті в 1994-1996 роках UML розробляли три програмні інженери, що працювали в Rational Software. Пізніше він був прийнятий як стандарт у 1997 році і з тих пір залишається стандартом, отримуючи лише кілька оновлень.

Діаграми UML використовуються для передачі різних аспектів і характеристик системи. Проте вони використовуються для високорівневого опису системи і не можуть передати усі деталі, необхідні для повноцінної розробки програмного забезпечення [12].

В результаті проектування мобільного додатку було розроблено декілька UML діаграм для того, щоб описати його роботу на високому рівні. Дані діаграми буде розглянуто нижче.

Для побудови UML діаграм було використано онлайн ресурс Draw.io.

Use case діаграма відображає взаємодію користувача з системою та функції системи, доступні для того чи іншого актора. В розробленій системі є один актор, котрий може авторизуватися за допомогою використання Touch Id або Face Id щоб відкрити доступ до додатку. За замовчуванням, авторизація не встановлена, але може бути включена в налаштуваннях. Усі функції додатку працюють офлайн локально на девайсі користувача. Основними функціями є: розпізнавання об'єктів

за допомогою маркерів, створення та редагування каталогів маркерів, додання маркерів до них та експорт або імпорт каталогів з файлу в форматі .csv. Діаграма Use Case зображена на рисунку 3.1.

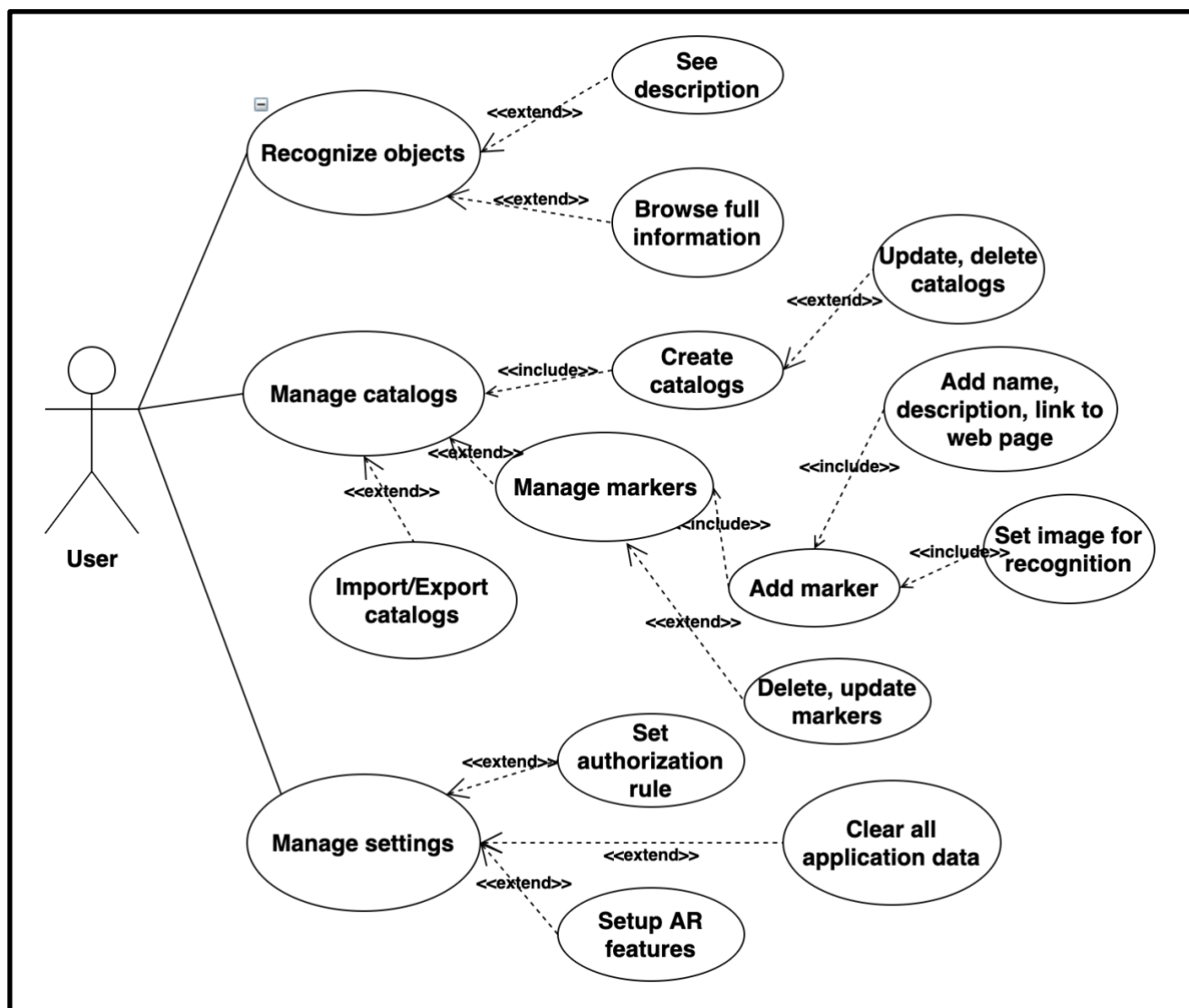


Рисунок 3.1 – Use case діаграма

На Use Case діаграмі зображено весь основний функціонал мобільного додатку та вказано, які функції розширюють інші. Наприклад, як видно на діаграмі, користувач може видаляти каталоги після їх додання. Але, він також може видалити усі дані мобільного додатку в налаштуваннях. В цьому випадку, усю локальну базу даних буде видалено, з усіма каталогами та маркерами, які в них були додані.

Діаграма послідовностей відображає те, що відбувається поза очима користувача, коли мобільний додаток намагається розпізнати об'єкт. Відразу після автентифікації, користувач бачить відеоряд з камери девайсу. Модель запускає Recognition Service, котрий використовує API ARKit, для того, щоб розпізнавати маркери в режимі реального часу. Це відбувається асинхронно та безперервно, тому ARKit намагається співвіднести завантажені маркери з відеорядом. На даному етапі працюють функції комп'ютерного бачення. Діаграму послідовності наведено на рисунку 3.2.

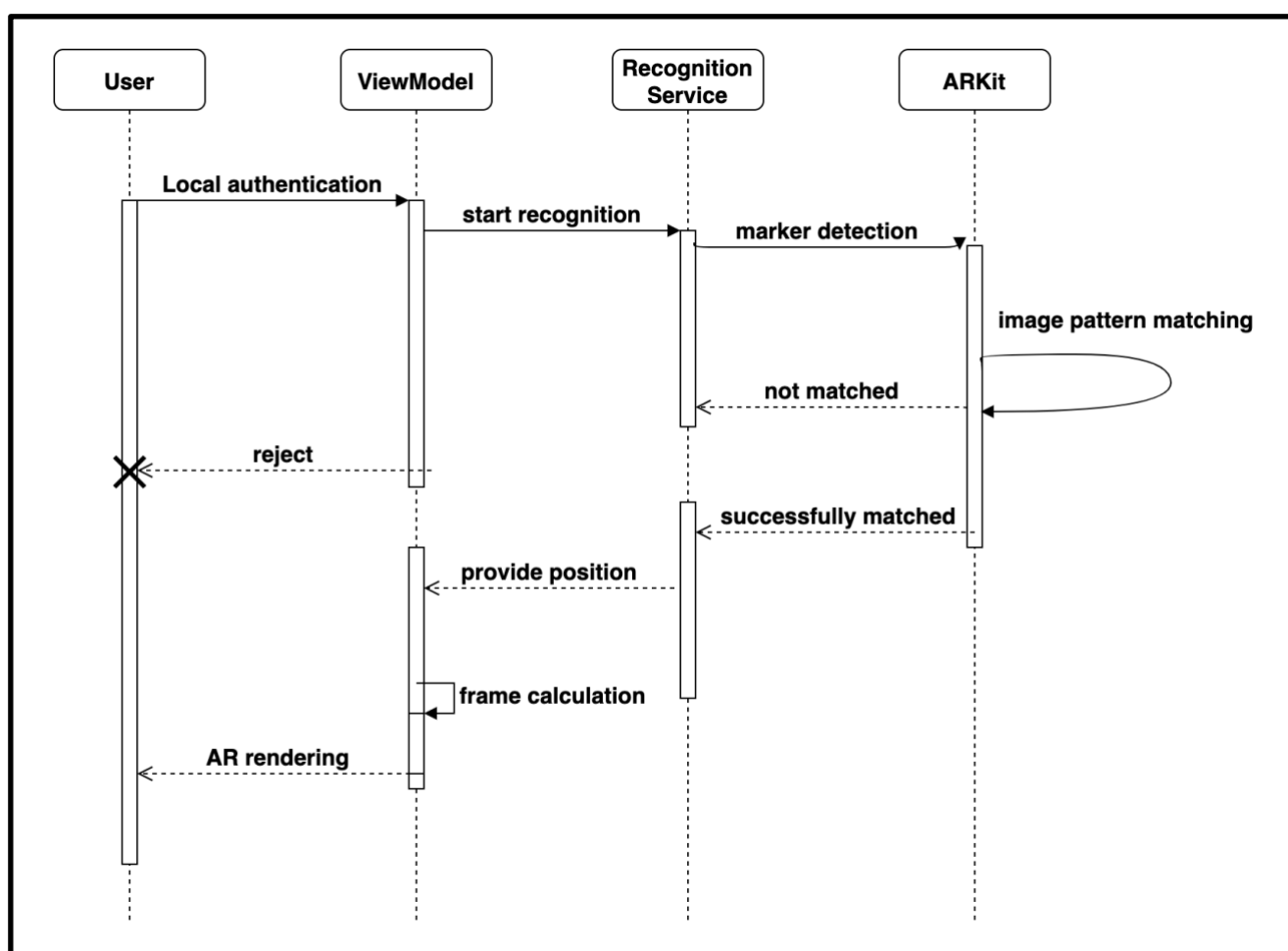


Рисунок 3.2 – Діаграма послідовності

Якщо маркер розпізнався, то дані про нього передаються до Recognition Service, котрий отримує відносну позицію щодо відеоряду. Далі він передає ці дані

до View Model, котра проводить необхідні обчислення для того, щоб накласти AR об'єкти на екран поверх відеоряду.

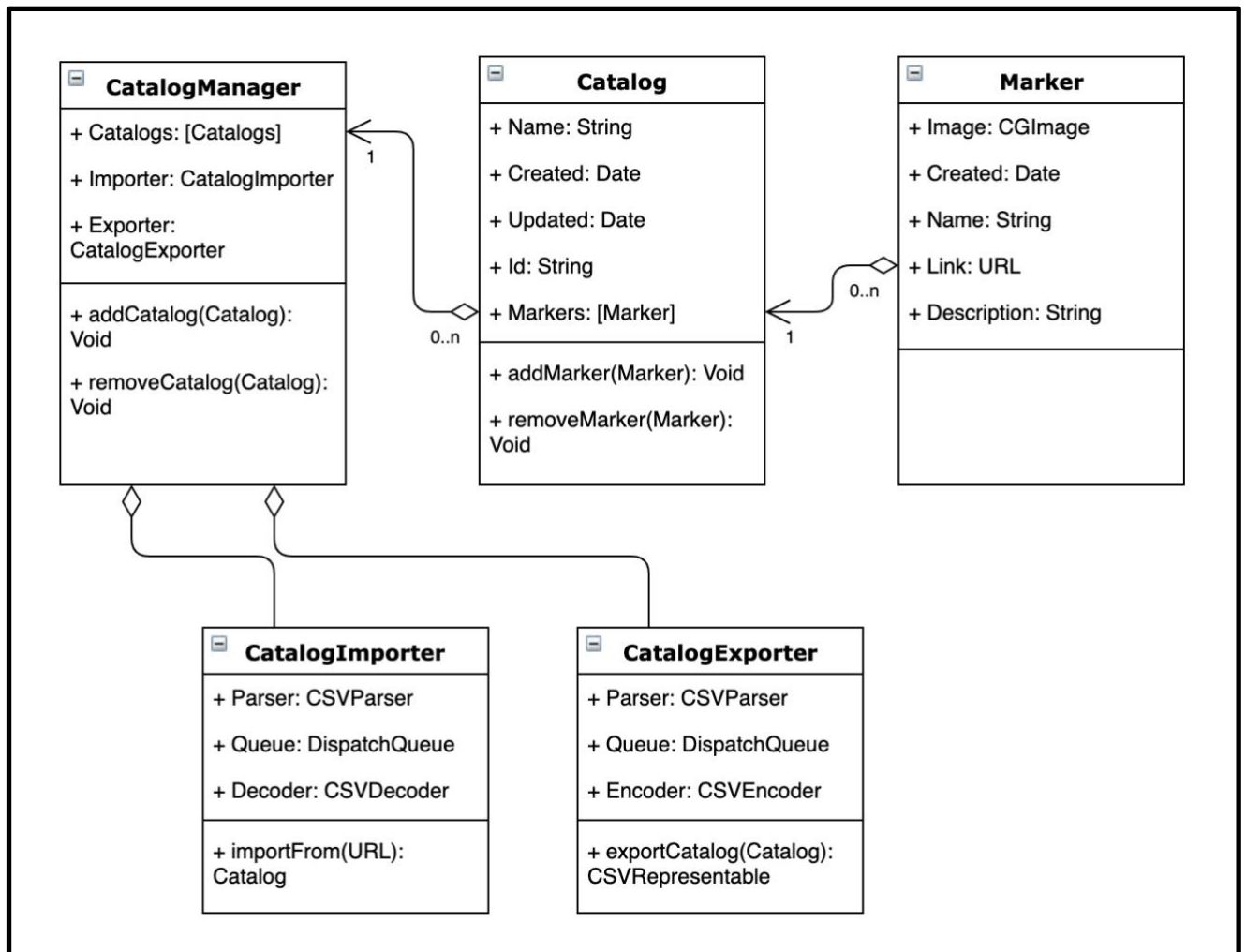


Рисунок 3.3 – Діаграма класів

Наступним кроком було створення діаграми класів (див. рис. 3.3). Ця діаграма відображає основні моделі системи та залежності між ними. Більшість службових класів та класів роботи з базою даних не були включені до діаграми, бо вони мають службове значення та не допомагають зрозуміти систему на високому рівні. Діаграма включає в себе клас **CatalogManager**, котрий дозволяє маніпулювати каталогами та також робити експорт та імпорт з .csv файлів.

Клас **Catalog** дозволяє додавати маркери та категорусє їх щодо певної предметної області знань, наприклад хижі звірі чи автомобільні деталі. Клас **Marker**

відповідає безпосередньо за той об'єкт, котрий буде розпізнаватися та включає в себе усю інформацію, котру треба буде додавати на екран після успішного розпізнавання.

В результаті UML-моделювання було описано та спроектовано основні компоненти програмної системи. Це поліпшило розуміння функціоналу на високому рівні та сприяло вирішенню недоліків системи ще на етапі проектування.

3.2 Архітектура мобільного додатку

Оскільки програмна система є офлайн мобільним додатком з локальною базою даних, було вирішено обрати архітектурний патерн Model-View-ViewModel (MVVM). MVVM полегшує розподіл розробки графічного інтерфейсу користувача – будь то через мову розмітки або код UI – від розробки бізнес-логіки та взаємодії з локальною базою даних [13].

Модель перегляду MVVM є патерном для конвертації даних моделей, що означає, що ViewModel відповідає за усі перетворення об'єктів даних з моделі таким чином, що вони легко керуються і подаються до контролеру UI. У цьому відношенні ViewModel є більш моделлю обробки даних предметної області та створення даних, якими легко оперувати контролеру інтерфейсу. ViewModel може реалізовувати шаблон медіатора, організовуючи доступ до локальної логіки навколо набору випадків використання, підтримуваних інтерфейсом користувача [14].

Одною з переваг MVVM є покращене перевикористання коду та поліпшення тестування моделей і контролерів, бо вони менш залежать від один одного, ніж в класичному MVC.

Шаблон MVVM намагається отримати обидві переваги розділення функціоналу, що надаються MVC, при цьому використовуючи переваги прив'язки даних і фреймворку за допомогою зв'язування даних, наскільки це можливо,

наближених до чистої прикладної моделі. Для перевірки вхідних даних використовуються функції перевірки даних, моделі перегляду та перевірки даних будь-яких бізнес-шарів. Результатом є те, що модель і структура обробляють якомога більше операцій, усуваючи або зводячи до мінімуму логіку програми, яка безпосередньо керує переглядом (наприклад, за кодом). Схема роботи MVVM в зрівнянні з MVC наведена на рисунку 3.4.

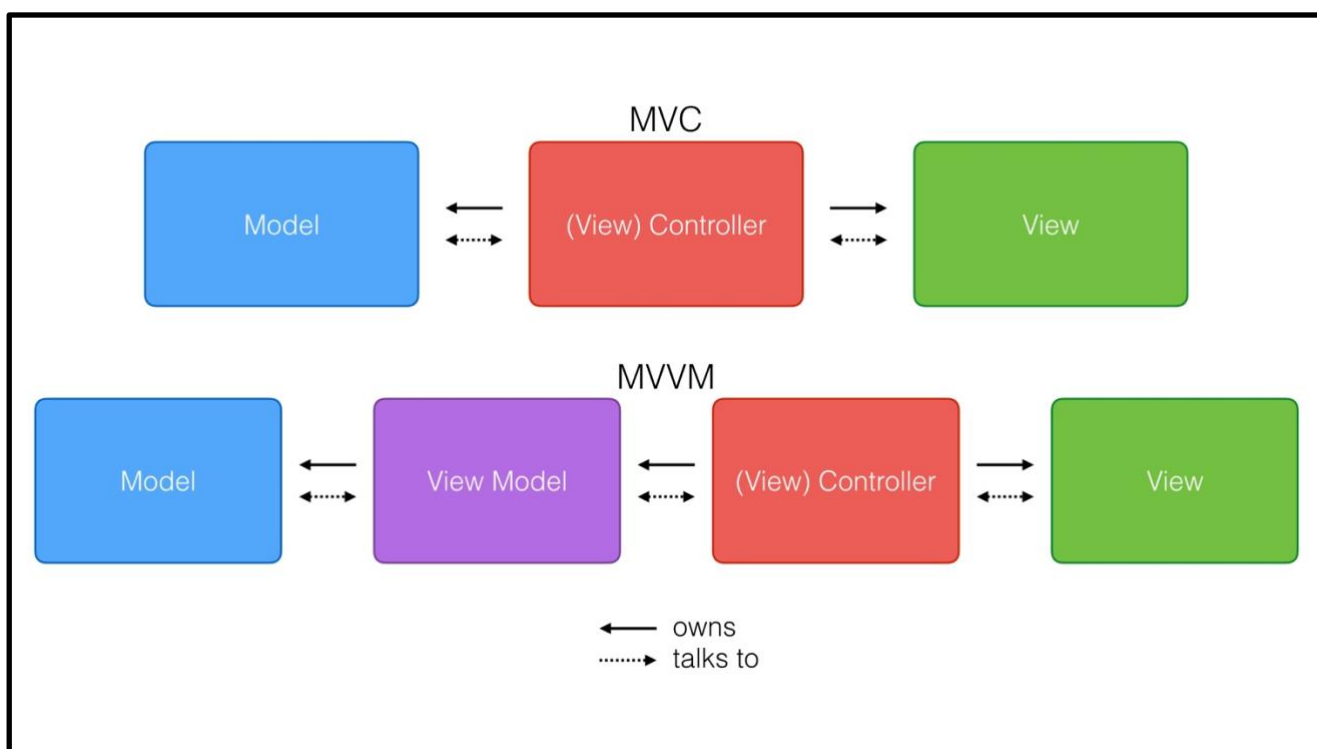


Рисунок 3.4 – Схеми роботи MVC та MVVM

MVVM як шаблон дизайну в iOS є корисним і дає багато переваг. Однак, як і у випадку з будь-яким архітектурним дизайном, необхідно подбати про те, щоб зрозуміти обмеження та вибрати відповідну реалізацію шаблону, яка більш усього підійде для проекту. В результаті аналізу було визначено, що для запланованої функціональності MVVM підходить більше, ніж MVC, та буде більш ефективним з точки зору програмної реалізації.

4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

4.1 Вибір засобів розробки

В результаті проектування та моделювання програмної системи було визначено набір задач для реалізації. З цього випливає вибір технологій. Оскільки мобільний додаток буде працювати на платформі iOS, було вирішено використовувати усі нативні фреймворки та SDK, доступні розробникам зі сторони Apple.

Мовою реалізації було обрано Swift версії 5.0. Swift 5 є важливою віхою в еволюції мови. Завдяки стабільності ABI [15], Swift runtime тепер включено до поточних і майбутніх версій операційних систем платформи Apple: macOS, iOS, tvOS і watchOS. Swift 5 також вводить нові можливості, які є будівельними блоками для майбутніх версій, включаючи реалізацію String, забезпечення ексклюзивного доступу до пам'яті під час виконання, нові типи даних і підтримку dynamically callable типів.

Для реалізації інтерфейсу користувача було обрано стандартні бібліотеки: UIKit та SceneKit. В той час, як UIKit дозволяє будувати стандартні елементи інтерфейсу, до котрого звикли усі користувачі платформи iOS, SceneKit надає можливість працювати з 3D моделями та відображенням. Це саме те, що потрібно для реалізації додатку з AR функціями розпізнавання. Перевагою використання цих фреймворків є те, що вони створені для роботи один з одним і тому не буде ніяких складностей їх комбінації.

Середовищем для розробки було обрано Xcode, котрий створений компанією Apple для того, щоб розробляти програмні продукти для операційних систем macOS, iOS, tvOS, watchOS. Він надає багато переваг в зрівнянні з іншими продуктами та зменшує час, витрачений на написання програмного коду.

Для локальної бази даних було обрано технологію Core data [16]. Core Data – це фреймворк, який розробники використовують для керування об'єктами модельного шару у мобільному додатку. Фреймворк забезпечує узагальнені та

автоматизовані рішення загальних завдань, пов'язаних з життєвим циклом об'єктів та управлінням графіками об'єктів, включаючи стійкість. Його побудовано на основі бази даних SQLite з великою кількістю покращень.

Core Data зазвичай зменшує кількість коду, який ви пишете для підтримки шару моделей на 50-70 відсотків. Це, перш за все, пов'язано з такими вбудованими функціями, які вам не потрібно впроваджувати, тестувати або оптимізувати:

- відстеження змін та вбудоване керування скасуванням та повторенням, за винятком основного редагування тексту;
- збереження розповсюдження змін, включаючи збереження послідовності відносин між об'єктами;
- lazy завантаження об'єктів, частково матеріалізовані ф'ючерси (faulting) і обмін даними для копіювання на запис для зменшення накладних витрат;
- автоматична перевірка значень властивостей. Керовані об'єкти розширюють стандартні методи валідації кодування значень ключів, щоб гарантувати, що окремі значення лежать в межах допустимих діапазонів, так що комбінації значень мають сенс;
- інструменти міграції схем, які спрощують зміни схеми та дозволяють здійснювати ефективну міграцію на місці;
- додаткова інтеграція з шаром контролера програми для підтримки синхронізації інтерфейсу користувача;
- групування, фільтрація та організація даних в пам'яті та в інтерфейсі користувача;
- автоматична підтримка зберігання об'єктів у зовнішніх сховищах даних;
- складна компіляція запитів. Замість того, щоб писати SQL, можна створювати складні запити, пов'язуючи об'єкт NSPredicate із запитом вибірки;
- відстеження версій і оптимістичне блокування для підтримки автоматичного вирішення конфліктів з декількома авторами.

Для реалізації AR задач було обрано технологію ARKit, розроблену Apple. Вона є найбільш популярним інструментом для розробки мобільних додатків з функціями доповненої реальності та розпізнавання. ARKit, технологія доповненої

реальності, надає користувачам захоплюючий досвід, в якому плавно змішуються віртуальні об'єкти з реальним світом. У AR-додатках камера пристрою представляє живий перегляд фізичного світу через екран девайсу. Тривимірні віртуальні об'єкти накладаються на відеоряд, створюючи ілюзію, що віртуальні об'єкти насправді існують. Користувач може переорієнтувати свій пристрій, щоб досліджувати об'єкти з різних кутів і, якщо це доречно для досвіду, взаємодіяти з об'єктами за допомогою жестів і рухів.

Для експорту та імпорту каталогів було вирішено інтегрувати функціонал iCloud. iCloud Drive безпечно зберігає будь-які файли, тому їх можна завжди зчитати в програмних продуктах, які підтримуються iCloud на iPhone, iPad, Mac або PC. Створення папки для свого мобільного додатка в iCloud Drive надає надійне сховище для будь-яких файлів. Більш того, усі редагування автоматично синхронізуються з усіма пристроями.

При реалізації мобільного додатку було вирішено використовувати принципи проектування SOLID [17]. У об'єктно-орієнтованому комп'ютерному програмуванні, SOLID є мнемонічною абревіатурою для п'яти принципів дизайну, призначених для того, щоб зробити розробку програмного забезпечення більш зрозумілою, гнучкою і підтримуваною.

Обрані технології є сучасними, актуальними та перспективними, тому повністю відповідають критеріям реалізації поставленої задачі.

4.2 Опис мобільного додатку

Мобільний додаток призначений для використання в сфері освіти, а саме для створення каталогів об'єктів, які мають відношення до конкретних областей знань, та збагачення процесу сприйняття інформації за допомогою додання віртуальних інтерактивних елементів після їх розпізнавання через мультимедійний пристрій. Об'єкти області знань виступають маркерами для системи розпізнавання. Після

встановлення мобільного додатку на мультимедійний девайс, наприклад iPad чи iPhone, застосування готове до використання. Якщо користувач відкриває мобільний додаток в перший раз, то він побачить екран для додання каталогів. Цей екран є другим у таб барі, але якщо жодного каталогу за маркерами ще не додано, то додаток ще не готовий нічого розпізнавати. Інтерфейс користувача для екрану каталогів зображено на рисунку 4.1.

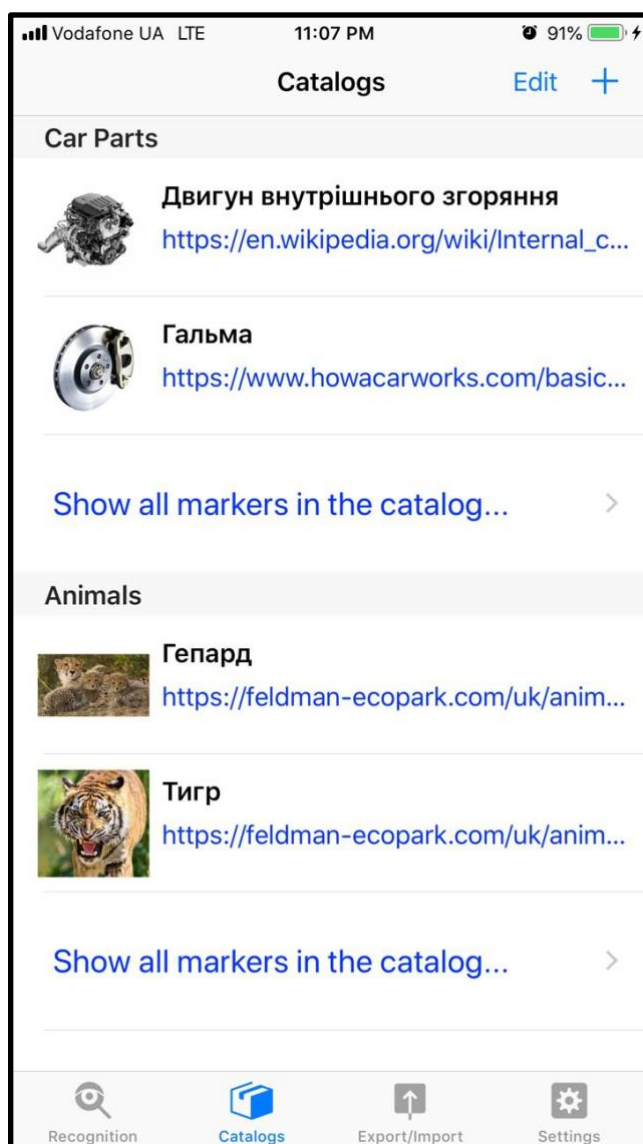


Рисунок 4.1 – Сторінка управління каталогами

Перебуваючи на даній сторінці, користувач може управляти каталогами, а саме додавати нові, чи редагувати або видаляти вже існуючі. При додаванні нового каталогу, інтерфейс дуже простий. Це звичайний алерт з текстовим полем, в котрий

треба ввести назву нового каталогу. В подальшому, кожен каталог можна редагувати та змінювати існуючу назву. Мобільний додаток немає обмежень на кількість каталогів. Наступним екраном є відображення сторінки для одного каталогу з усіма доданими маркерами. На цій сторінці користувач може додавати нові маркери, або редагувати чи видаляти існуючі маркери. Сторінку управління каталогом наведено на рисунку 4.2.

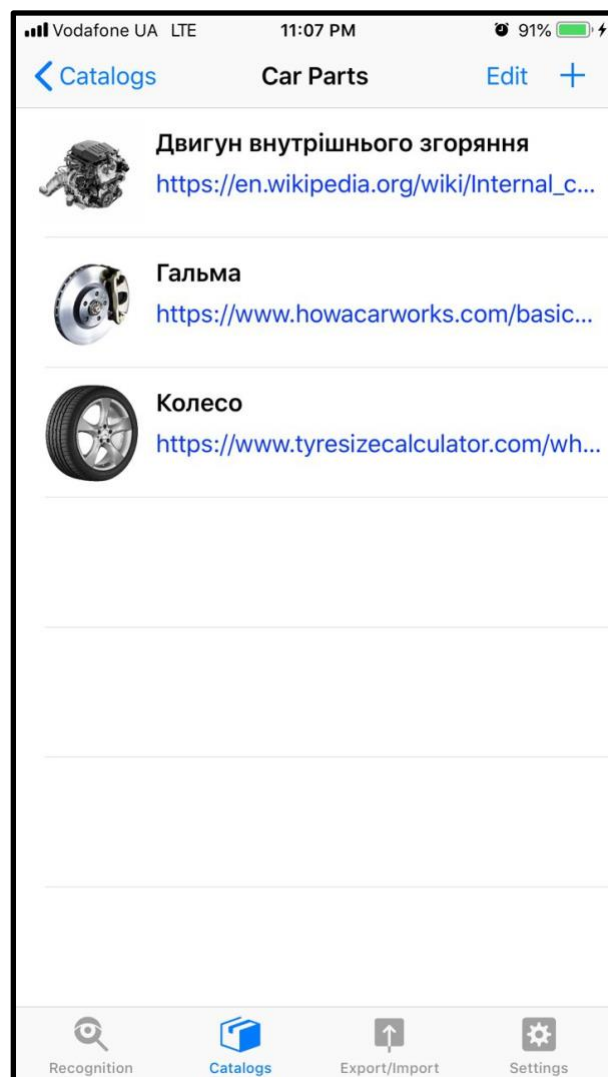


Рисунок 4.2 – Сторінка управління конкретним каталогом

Так як було використано патерн MVVM, для додавання та редагування маркеру використовується той самий контролер, але різні ViewModel. Це є гарним прикладом використання MVVM замість MVC. Суть в тому, що контролер не знає нічого про конкретний клас моделі, а лише про її відкритий інтерфейс. Це надає

можливість використовувати той самий контролер інтерфейсу користувача, але іншу бізнес логіку.

Наступним кроком користувача є заповнення даних на сторінці створення нового маркеру. Цю сторінку наведено на рисунку 4.3.

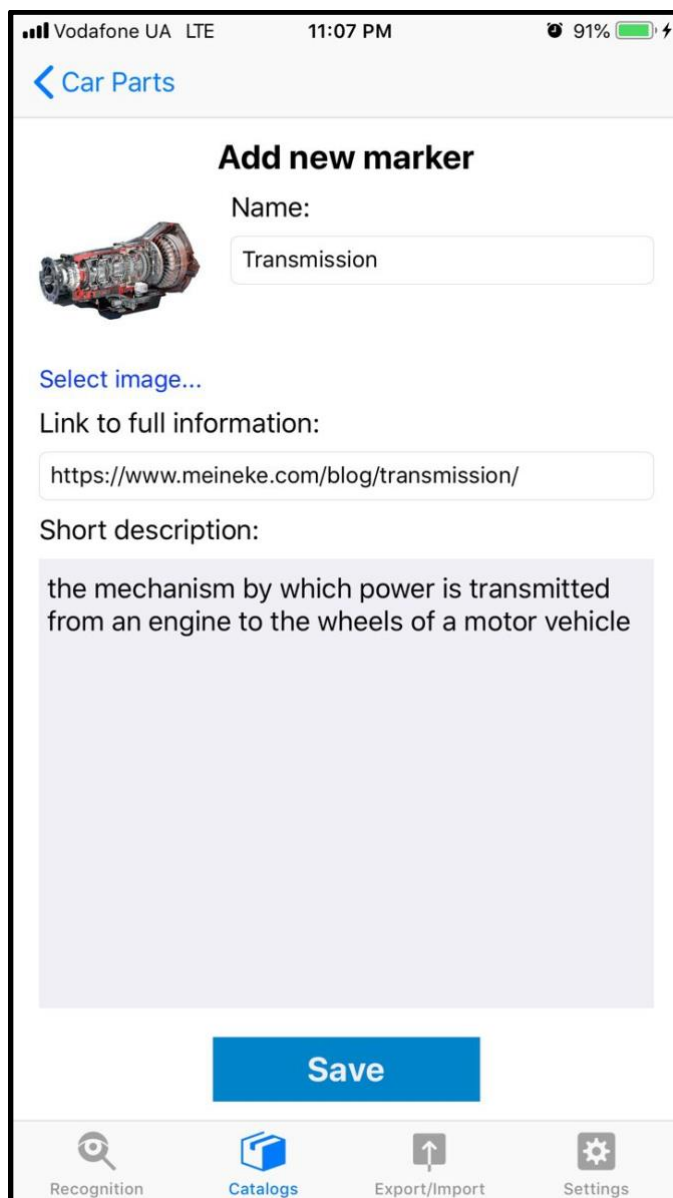


Рисунок 4.3 – Сторінка створення нового маркеру для розпізнавання

Як можна побачити на рисунку 4.3, користувач має заповнити декілька текстових полей та вибрати картинку для розпізнавання. Є два способи зробити це, вибрати фото з галереї, або створити фото одразу в додатку. Для цього треба підтвердити запит на використання камери. Добрим правилом роботи з функціями

AR є використання чітких маркерів. Тобто, якщо користувач робить фотографію якогось приладу з підручнику, то йому треба переконатися, що фотографія не розмита та освітлення гарне. Якщо якість фотографії недостатньо добра, то можуть бути проблеми з розпізнаванням маркеру у подальшому.

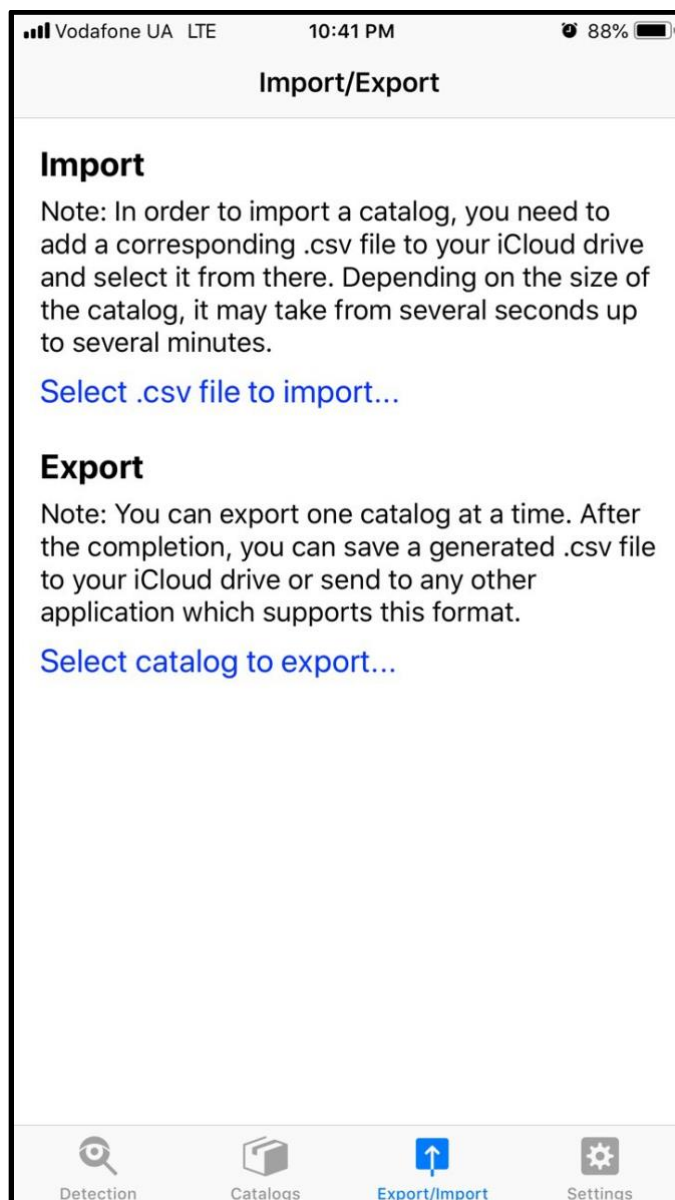


Рисунок 4.4 – Сторінка експорту та імпорту каталогів

Ще однією важливою функцією мобільного додатку є можливість експортувати та імпортувати каталоги. Це дозволяє зберігати каталоги у виді бекапу або переносити їх на інші девайси. Для реалізації цієї функції було підключено інтеграцію з iCloud Drive. Формат, в котрому зберігається дані є .csv.

Для збереження картинок в .csv використовується Base64 енкодинг, котрий дозволяє записувати картинки як набір символів у строку. На сторінці можна вибрати .csv файл для імпорту каталогу, або вибрати каталог для експорту в .csv файл (див. рис. 4.4).

На рисунку 4.5 зображено сторінку налаштувань.

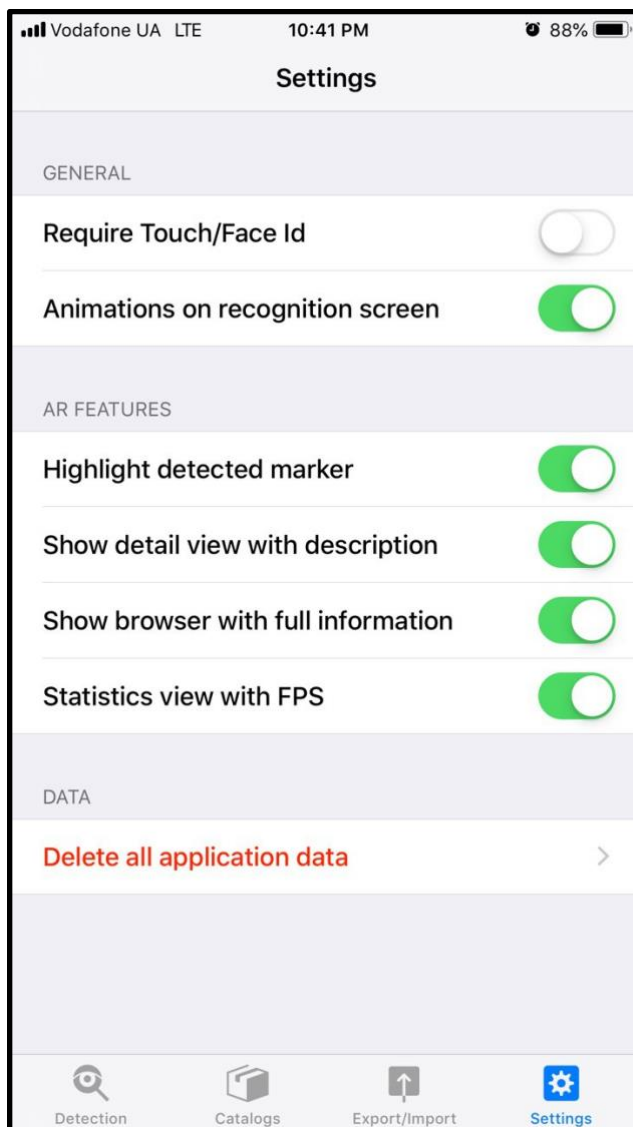


Рисунок 4.5 – Сторінка налаштувань мобільного додатку

Сторінка складається з трьох секцій: загальні налаштування, налаштування функцій AR та секція очистки усіх даних додатку. В першій секції користувач має можливість налаштувати авторизацію за допомогою Touch/Face Id та включити або виключити анімації під час розпізнавання маркерів. Наступна секція відповідає за

налаштування AR функцій на екрані розпізнавання. Тут можна налаштувати ті об'єкти, котрі будуть відображатися біля маркера після розпізнавання та включити або виключити відображення статистики FPS.

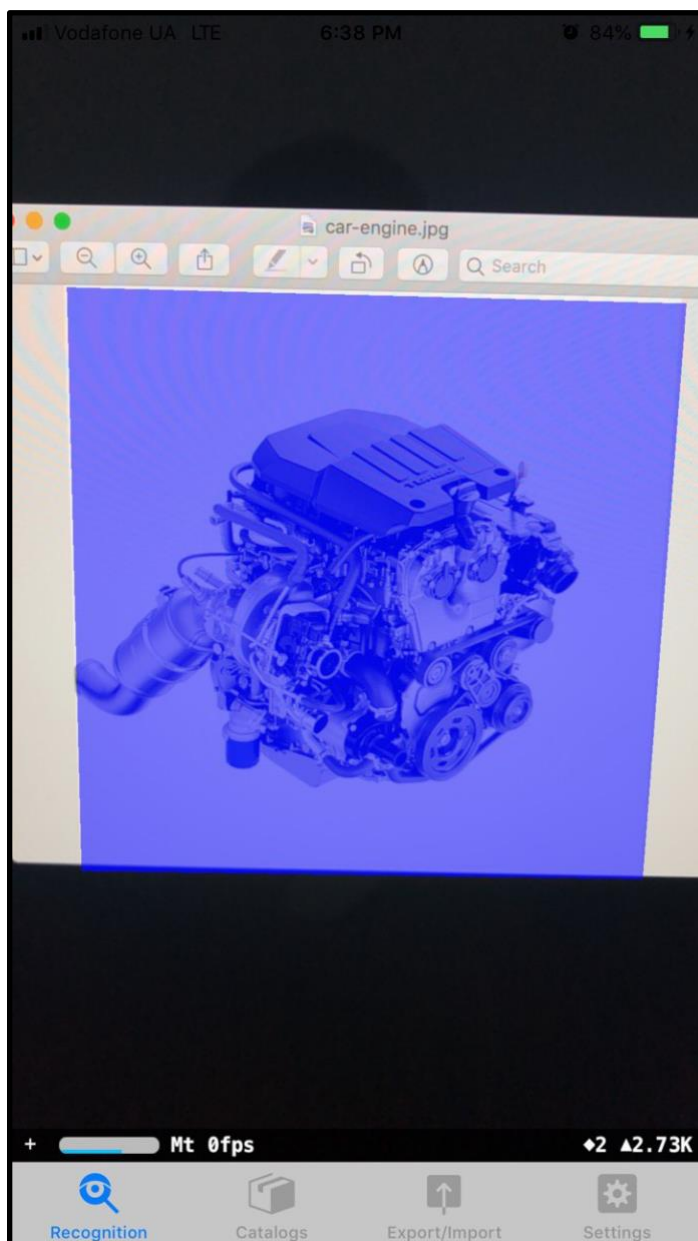


Рисунок 4.6 – Сторінка розпізнавання маркеру

На рисунку 4.6 зображено сторінку розпізнавання маркерів, а саме підкреслення зони розпізнаного маркера (рисунку). Це відбувається автоматично, мобільний додаток аналізує відеоряд та завжди намагається розпізнати один з наявних маркерів. Коли маркер розпізнано, його зона на екрані підсвічується синім

кольором та блимає на протязі короткого часу. Після цього спрацьовують функції накладення віртуальних об'єктів поверх відеоряду, як відображено на рисунку 4.7. Зліва виїзжає панель, яка відображає ім'я та опис розпізнаного об'єкту. З правої сторони відображається інтерактивний об'єкт браузеру з відкритою сторінкою, яка задається в об'єкті маркеру.

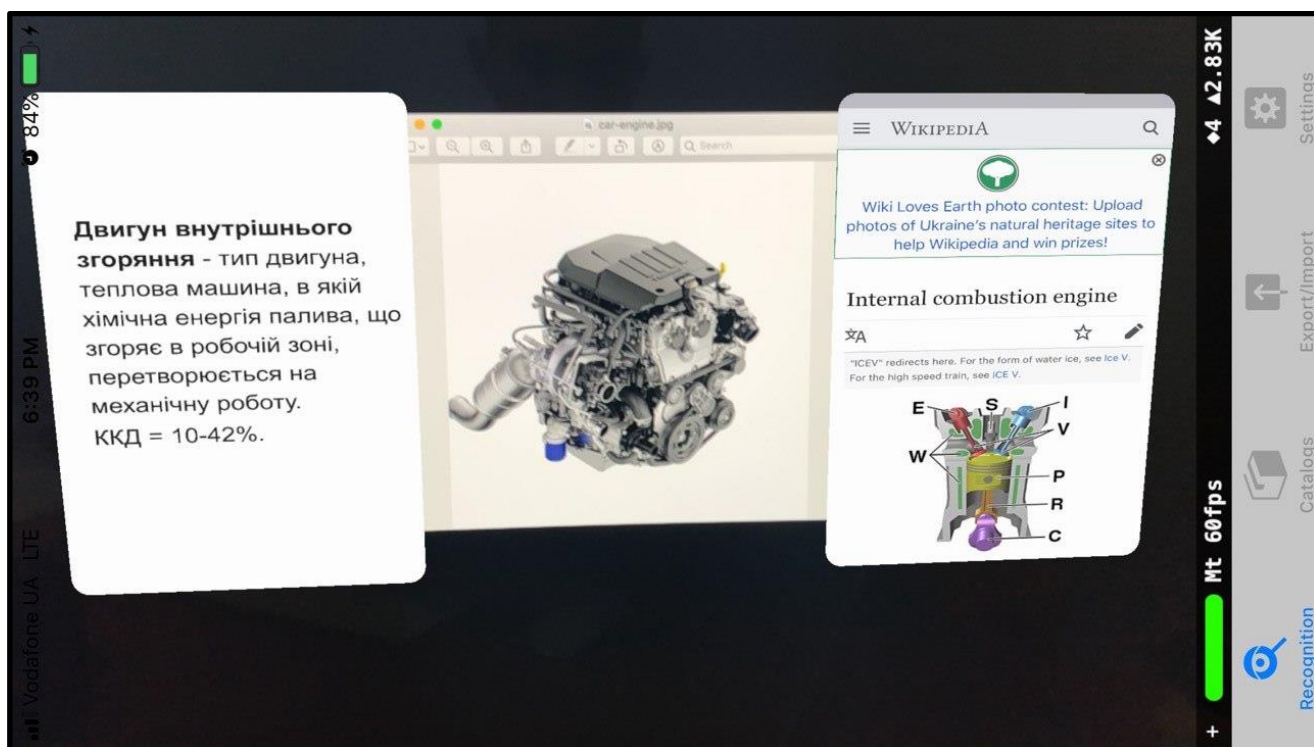


Рисунок 4.7 – Додання віртуальних об'єктів до відеоряду девайсу

Як видно на скриншоті сторінки, біля розпізнаного маркеру відображено 2 віртуальні об'єкти, додані за допомогою функціоналу технології ARKit та SceneKit. Мобільний додаток обчислює позицію та реальний розмір маркеру відносно відеоряду та додає віртуальні об'єкти біля нього. Більш цього, якщо рухати маркер, віртуальні об'єкти будуть рухатися відносно нього зберігаючи відстань та бути пропорційними за розміром. Скриншоти було зроблено посередені екрану з чорним фоном. Якщо перемістити зображення маркеру у край екрану, то можна побачити, що додаток коректно опрацьовує подібний сценарій і правильно відображає віртуальний об'єкт з правого боку. Також, правий віртуальний елемент є веб браузером в котрому відкрита сторінка з детальною інформацією про розпізнаний

об'єкт. Браузер є повністю функціонуючим та інтерактивним, як видно на рисунку 4.8. За допомогою свайпів на екрані девайсу ми можемо скролити сторінку вниз чи вгору.

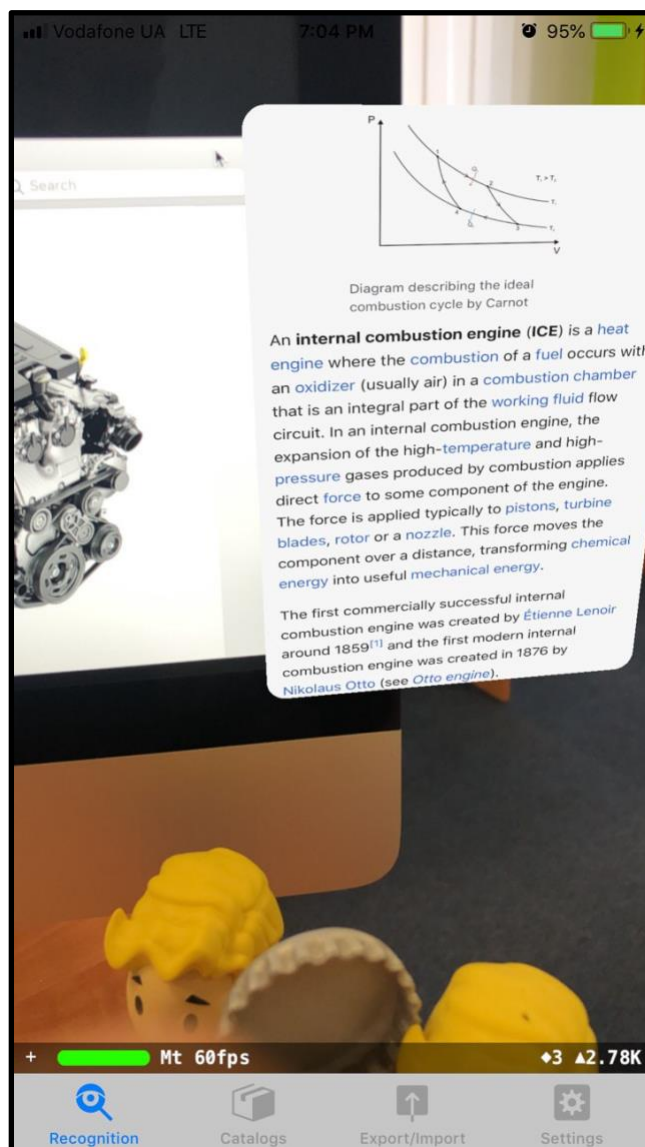


Рисунок 4.8 – Інтерактивний віртуальний об'єкт браузеру

В цілому, сторінка розпізнавання маркерів та додання об'єктів доповненої реальності до відеоряду є найбільш функціонально насиченою та складною з точки зору бізнес логіки додатку, бо в кожен момент часу виконується багато взаємопов'язаних операцій, котрі в результаті дозволяють розпізнати маркер, знайти інформацію щодо нього та відобразити відповідні AR об'єкти.

4.3 Тестування програмної системи

Тестування мобільних додатків – це процес, за допомогою якого прикладне програмне забезпечення, розроблене для портативних мобільних пристроїв, перевіряється на його функціональність, зручність, послідовність та відсутність помилок. Тестування мобільних додатків може бути як автоматизованим, так і ручним та включає багато різних типів. Під час процесу розробки було проведено наступні види тестування:

– димове тестування (Smoke testing), також відоме як «Build Verification Testing». Цей тип тестування програмного забезпечення складається з малого набору тестів, спрямованих на перевірку роботи найважливіших функцій системи. Після кожного етапу розробки було проведено цей тип тестування, що допомогло знайти близько 20% критичних помилок;

– функціональне тестування (Functional testing) є типом тестування програмного забезпечення, при якому система перевіряється на відповідність функціональним вимогам / специфікаціям. Функціональне тестування гарантує, що вимоги належним чином задовольняються додатком [18]. Цей тип тестування не стосується того, як відбувається обробка, а, скоріше, результатів обробки. Він імітує фактичне використання системи, але не робить жодних припущень до структури системи. Було проведено після реалізації усіх функцій мобільного додатку;

– регресійне та повторне тестування (Regression testing and Retesting). Виконується після внесення змін до програмної системи або її середовища (у новій версії програми), яка використовується, щоб переконатися, що функції, які працювали у попередній версії системи працюють коректно. Воно проводилося з використанням ручного тестування. Регресійне тестування виявило кілька помилок у системі. Помилки отримали пріоритет в залежності від впливу на основну функціональність програми. Як результат, всі критичні та пріоритетні помилки були виправлені;

– тестування графічного інтерфейсу (User Interface testing). Виконано після завершення усіх фаз розробки програмної системи. Було знайдено декілька косметичних помилок, які були виправлені.

– тестування продуктивності (Performance testing) проводиться для підтвердження швидкості роботи програмної системи. Було виконано за допомогою вбудованих інструментів середовища Xcode. Також, на екрані розпізнавання є статистичні дані щодо продуктивності додатку в реальному часі, що спростило процес тестування на різних пристроях.

В результаті використання різних типів тестування під час та після процесу розробки було виявлено критичні та пріоритетні помилки, які були виправлені. Завдяки ретельному процесу було досягнуто високої якості та надійності мобільного додатку та графічного інтерфейсу користувача. Також, було додано додаткові налаштування до системи для поліпшення досвіду користувача, наприклад, налаштування вибору віртуальних об'єктів для відображення, включення / виключення анімацій та можливість видалення усіх даних в системі.

ВИСНОВКИ

У ході атестаційної роботи був проведений аналіз предметної галузі технологій доповненої реальності та її використання в сфері освіти на мобільній платформі iOS. Були досліджені основні типи доповненої реальності, а саме:

- доповнена реальність на основі маркерів (Marker-based AR);
- доповнена реальність без маркерів (Marker-less AR);
- проекційна доповнена реальність (Projection AR);
- доповнена реальність на основі накладання (Superimposition based AR).

На основі аналізу предметної галузі було виявлено найбільш ефективні сценарії використання різних типів технології доповненої реальності для вирішення специфічних завдань та реалізації мобільних додатків з певним функціоналом. Також було виявлено, що в багатьох випадках в мобільних додатках можуть використовуватися одразу декілька різних типів AR для комбінації можливостей кожної з технологій.

Під час аналізу було розглянуто методи комп'ютерного бачення в доповненій реальності та зроблено висновок про їх вплив на загальні принципи та алгоритми роботи AR в бібліотеці ARKit зокрема.

Під час проектування було створено та детально розглянуто UML діаграми, а саме діаграми прецедентів, послідовності та класів. Діаграма прецедентів, або Use case, відображає функціональні можливості користувачів на високому рівні. Діаграма послідовності ілюструє виконання системою процесів, котрі скриті від очей користувача та допомагає зрозуміти взаємозв'язок та передачу даних між різними підсистемами. Діаграма класів описує декілька основних класів системи, котрі допомагають зрозуміти представлення предметної області в рамках комп'ютеру.

Як результат, було реалізовано мобільний додаток з використанням доповненої реальності на основі маркерів, котрий може використовуватися для освітніх цілей використовуючи мультимедіа девайси на платформі iOS. Його

розроблено за допомогою мови програмування Swift 5 та технологій ARKit та SceneKit. Для локальної бази даних використано Core Data.

Як результат, перша версія мобільного додатку демонструє можливості доповненої реальності в мультимедійних засобах навчання. Однак в подальшому планується доробити можливості системи для впровадження додатку в процеси навчання учбових закладів. По-перше, планується створення серверу для синхронізації даних між декількома пристроями та створення акаунтів користувачів. Також, планується можливість додання віртуальних інтерактивних 3D моделей об'єктів після розпізнавання маркерів.

В цілому, дослідження є важливим для вибору та використання ефективних методів та засобів реалізації додатків на основі технології доповненої реальності. Доповнена реальність в сфері освіти надає інформації можливість бути інтерактивною та більш придатною для сприйняття і запам'ятовування учнями та студентами, що підвищує ефективність навчання та мотивує їх на більші досягнення.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Доповнена реальність [Електронний ресурс] / Доповнена реальність, або AR-технології. Як це працює? – Режим доступу: <http://thefuture.news/page1837780.html> – 11.04.2019 р. – Загол. з екрану.
2. Steve Aukstakalnis. Practical Augmented Reality [Текст]: A Guide to the Technologies, Applications, and Human Factors for AR and VR/ Steve Aukstakalnis. — Addison-Wesley Professional, Sep 8, 2016. - 448с.
3. Helen Papagiannis. Augmented Human [Текст]: How Technology Is Shaping the New Reality/ Helen Papagiannis. — O'Reilly Media, Aug 31, 2017. - 156с.
4. 4D Interactive Anatomy [Електронний ресурс] / Real human specimens online – Режим доступу: <https://www.4danatomy.com> – 12.04.2019 р. – Загол. з екрану.
5. ARKit 2 [Електронний ресурс] / Apple Developer Portal. – Режим доступу: <https://developer.apple.com/arkit/> – 7.04.2019 р. – Загол. з екрану.
6. Dan Simon. Optimal State Estimation [Текст]: Kalman, H Infinity, and Nonlinear Approaches/ Dan Simon. — Wiley-Interscience, 2006. - 552с.
7. Apple's A12 Bionic is the first 7-nanometer smartphone chip [Електронний ресурс] / It's the first to ship. – Режим доступу: <https://www.engadget.com/2018/09/12/apple-a12-bionic-7-nanometer-chip/> – 15.04.2019 р. – Загол. з екрану.
8. Computer Vision [Текст]: Algorithms and Applications/ Richard Szeliski. — Springer London, 2010. - 852с.
9. Computer Vision: Models, Learning, and Inference [Текст]: 1st Edition/ Simon J. D. Prince. — Cambridge University Press, June 18, 2012. - 598с.
10. An Introduction to Simultaneous Localisation and Mapping [Електронний ресурс] / Портал kudan.io – Режим доступу: <https://www.kudan.io/post/an-introduction-to-simultaneous-localisation-and-mapping> – 19.04.2019 р. – Загол. з екрану.

11. Фаулер М. UML Основы / М. Фаулер, К. Скотт – СПб: Символ-Плюс, 2002. – 192 с.
12. UML 2 Sequence Diagrams: An Agile Introduction [Электронный ресурс] / Портал Agile Modeling – Режим доступа: <http://www.agilemodeling.com/artifacts/sequenceDiagram.htm> – 20.04.2019 г. – Загол. з экрана.
13. Model-View-ViewModel (MVVM) [Электронный ресурс] / Definition from WhatIs.com – Режим доступа: <https://whatis.techtarget.com/definition/Model-View-ViewModel> – 15.04.2019 г. – Загол. з экрана.
14. Mediator Design Pattern [Электронный ресурс] / Портал SourceMaking – Режим доступа: https://sourcemaking.com/design_patterns/mediator – 27.04.2018 г. – Загол. з экрана.
15. Swift 5 ABI Stability [Электронный ресурс] / Learn why ABI Stability is important? – Режим доступа: <https://medium.com/swift-india/swift-5-abi-stability-769ccb986d79> – 03.05.2018 г. – Загол. з экрана.
16. What Is Core Data? [Электронный ресурс] / Core Data Programming Guide – Режим доступа: <https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/CoreData/index.html> – 29.04.2018 г. – Загол. з экрана.
17. SOLID (object-oriented computer programming) [Электронный ресурс] / Портал wikipedia.org – Режим доступа: <https://en.wikipedia.org/wiki/SOLID> – 03.05.2018 г. – Загол. з экрана.
18. Functional Testing [Электронный ресурс] / Software Testing Fundamentals – Режим доступа: <http://softwaretestingfundamentals.com/functional-testing/> – 12.05.2018 г. – Загол. з экрана.