

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Програмної інженерії
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти – другий (магістерський)

Дослідження методів управління складними об'єктами шляхом
реалізації контролю за поглядом людини
(тема)

Виконав: студент 2 курсу, групи ІПЗм-18-4
Радченко Г.С.
(прізвище, ініціали)
спеціальності 121- Інженерія програмного забезпечення
(код і повна назва спеціальності)
Освітньо-наукової програми
(тип програми)
Інженерія програмного забезпечення
(повна назва освітньої програми)

Керівник доц. Лановий О.Ф.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри, проф. _____

З.В.Дудар

2020 р.

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет Комп'ютерних наукКафедра Програмної інженерії

Рівень вищої освіти – другий (магістерський)

Спеціальність 121-Інженерія програмного забезпечення

(код і повна назва)

Тип програми освітньо-наукова програмаОсвітня програма Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

« ____ » _____ 20 ____ р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові Радченко Глібу Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів управління складними об'єктами шляхом реалізації контролю за поглядом людини.

затверджена наказом по університету від « ____ » _____ 20 ____ р. № _____

2. Термін подання студентом роботи до екзаменаційної комісії 05 травня 2020 р.

3. Вихідні дані до роботи моделі та алгоритми управління складними об'єктами, методи розпізнання складних об'єктів, пояснювальна записка. Використовувати ОС macOS, середовище об'єктно-орієнтованого проектування Xcode.

4. Перелік питань, що потрібно опрацювати в роботі мета роботи, аналіз проблемної галузі і постановка задачі, огляд існуючих методів управління складними об'єктами, етапи аналізу, застосування методів управління складними об'єктами шляхом реалізації контролю за поглядом.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка *
1.	Аналіз предметної галузі	27 січня 2020 р.	
2.	Огляд існуючих методів	10 лютого 2020 р.	
3.	Методи управління складними об'єктами. Розпізнавання обличчя та його властивостей.	09 березня 2020 р.	
4.	Підготовка пояснювальної записки	23 березня 2020 р.	
5.	Спецчастина	06 квітня 2020 р.	
6.	Підготовка презентації та доповіді	14 квітня 2020 р.	
7.	Попередній захист	20 квітня 2020 р.	
8.	Нормоконтроль, рецензування	05 травня 2020 р.	
9.	Занесення диплома в електронний архів	09 травня 2020 р.	
10.	Допуск до захисту у зав. кафедри	10 травня 2020 р.	
* заповнюється вручну після виконання чергового пункту			

Дата видачі завдання _____ 2020 р.

Студент _____
(підпис)

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

РЕФЕРАТ / ABSTRACT

Атестаційна робота магістра: 82 с., 10 рис., 1 табл., 4 фор., 31 джерел.

IOS, SWIFT, C, C++, OBJECTIVE-C, AR, ARKIT, SCENEKIT, XCODE, ДОПОВНЕНА РЕАЛЬНІСТЬ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, МАШИННИЙ ЗІР.

Об'єкт – дослідження методів управління складними об'єктами шляхом реалізації контролю за поглядом людини.

Мета роботи – дослідити методи управління складними об'єктами, основні методи, моделі, алгоритми контролю за поглядом людини та побудувати зручний механізм управління графічним інтерфейсом.

Методи дослідження. У роботі використовувалися різні методи машинного зору та доповненої реальності, моделювання для розробки математичної моделі вибору найкращого методу для обробки вибірок на підставі характеристик вхідних даних.

IOS, SWIFT, C, C++, OBJECTIVE-C, AR, ARKIT, SCENEKIT, XCODE, AUGMENTED REALITY, SOFTWARE, COMPUTER VISION.

Object - Analyzing methods of controlling sophisticated objects using gaze tracking approach.

The purpose of the work is to investigate the concept of controlling sophisticated objects, general methods, models, algorithms of controlling and tracking user gaze to build a convenient mechanism to control user interface.

Research methods. The paper used different computer vision and augmented reality, modeling to develop a mathematical model for selecting the best method for processing the samples based on the characteristics of the input data.

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ	8
1.1 Дослідження і аналіз методів доповненої реальності та машинного зору	8
1.2 Виявлення проблем та актуалізація рішень.....	15
1.3 Можливі шляхи вирішення задачі за рахунок створення доповненої реальності на мобільному пристрої	17
1.4 Постановка задачі дослідження.....	22
2 ДОСЛІДЖЕННЯ АЛГОРИТМУ ВІДСТЕЖЕННЯ ЗОРУ ЛЮДИНИ	25
2.1 Опис проблеми відстеження очей	25
2.2 Огляд аналогів.....	27
2.3 Аналіз і опис основних етапів застосування алгоритмів доповненої реальності для відстеження зору людини.....	29
2.4 Уточнена постановка задачі дослідження.....	34
3 РОЗРОБКА МЕТОДІВ ТА АЛГОРИТМІВ УПРАВЛІННЯ	37
3.1 Аналіз алгоритмів доповненої реальності та способів їх використання	37
3.2 Вибір алгоритму машинного навчання.....	39
3.3 Штучні нейронні мережі	41
3.4 Огляд бібліотек для глибинного навчання	45
4 РЕАЛІЗАЦІЯ ПРОГРАМНОЇ МОДЕЛІ УПРАВЛІННЯ ОБ'ЄКТОМ	48
4.1 Опис алгоритму програмної реалізації системи машинного навчання	48
4.2 Опис діаграми класів програмного додатку	49
4.3 Опис діаграми діяльності системи	53
4.4 Опис діаграми послідовності системи	55
4.5 Результати експериментів	57
ВИСНОВКИ	60
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	61
ДОДАТОК А Фрагменти коду програми	64
ДОДАТОК Б Слайди презентації	71
ДОДАТОК В Апробація результатів роботи. Наукова стаття.....	79
ДОДАТОК Г Акт прийому наукової статті	82

ВСТУП

Взаємодія людини та комп'ютера увійшли у життя людини досить давно та дуже міцно. Ця взаємодія є найшвидшою у розвитку сьогодні. Багато наукових аспектів, галузей та досліджень залежить від комп'ютерів та їх потужності. Можна сказати, що сьогодні увесь світ залежить від комп'ютерів та технологій, пов'язаних з ними.

Регулярне та систематичне вдосконалення технологій привело до появи абсолютно нових парадигм інтерфейсів, призначених для користувача. Сьогодні можна спостерігати дуже багато цікавих та різноманітних інтерфейсів, наприклад звичайних графічний, слуховий, інтерфейс жестів, інтерфейс який використовує сигнали нервових клітин, інтерфейси доповненої та віртуальної реальності. Усі вони знаходяться на етапі потужної розробки, особливо останні. Глобалізація в сучасному суспільстві стала тією самою тенденцією, яка поширилася і на комп'ютерні науки.

Віртуальна реальність може допомогти покращити існуючі інтерфейси та технології. Наприклад, використовуючи доповнену реальність, можна вдосконалити взаємодію людини з електронними пристроями, а також надати змогу користуватися цими пристроями людям, які не мають змоги користуватися ними класичним шляхом.

Доповнена реальність – результат додавання у поле сприйняття будь яких сенсорних даних з метою доповнення відомостей про навколишнє середовище та покращення сприйняття інформації. Різниця віртуальної та доповненої реальності полягає у тому, що користувач бачить оточуючий світ який доповнено додатковою інформацією на екрані пристрою, а у віртуальній реальності користувач бачить тільки реальність яку було створено комп'ютером. Основна цінність доповненої реальності полягає в тому, що вона привносить компоненти цифрового світу у сприйняття людини реальним світом і робить це не як просте відображення даних, а через інтеграцію іммерсних відчуттів, які сприймаються як природні частини

навколишнього середовища. Перші функціональні системи доповненої реальності, які надавали користувачам поглиблену змішану реальність, були винайдені на початку 1990-х років. Існує декілька видів доповненої реальності, наприклад:

- доповнена реальність на основі маркерів;
- безмаркерна доповнена реальність;
- доповнена реальність з проектуванням;
- накладання на основі розширеної реальності.

Більшість розроблених технологій зосереджена на взаємодії людини і використовується на пристроях (стаціонарний комп'ютер, ноутбук, смартфон та інші). Доповнена реальність, в свою чергу, на підставі використання деяких конкретних комп'ютерних технологій, робить можливим поліпшення інтерфейсу людини і його взаємодії з реальним навколишнім світом.

При створенні мобільних додатків, які базуються на технології доповненої реальності є певні складнощі. Основною є ефективне використання ресурсів пристрою. Також доповнена реальність потребує великих обчислювальних потужностей від пристрою.

Метою цієї роботи є дослідження методів управління складними об'єктами шляхом реалізації контролю за поглядом людини та побудова алгоритму, який надасть змогу керувати графічним інтерфейсом за допомогою контролю погляду користувача.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Дослідження і аналіз методів доповненої реальності та машинного зору

Сфера інформаційних технологій є одною з найпередовіших галузей у нашому столітті. Такий швидкий розвиток технологій обумовлено тим, що нові технології спрощують життя у багатьох сферах життєдіяльності.

Наука, бізнес, розваги – усе це залежить насамперед від інформаційних технологій. За допомогою них ми можемо прискорювати багато процесів які потребують складних обчислювань.

Сьогодні існує багато проблем при взаємодії людини та пристрою або комп'ютеру. По перше це відволікання людини на цей пристрій та втрата уваги у реальному світі. Також усі пристрої досить сильно зав'язані на енергії що їх живить та акумуляторах. Ще одним недоліком є те, що деякі системи або пристрої є незручними для використання у повсякденному житті, адже вони можуть потребувати додаткового обладнання, чи є дуже великими для пересування [13]. Наступним фактором незручності та проблеми використання є фізичні обмеження або недоліки людини. Не усі пристрої проектуються з урахуванням цих факторів, що призводить до проблем та навіть конфліктів.

Якщо взяти технології навігаторів, як приклад системи, з якою взаємодіє дуже багато людей, можна знайти певні недоліки:

- постійне відволікання на пристрій-навігатор, як вже зазначалося вище та як наслідок втрата уваги;
- значна похибка у точності навігаторів та відображення інформації;
- помилкові або застарілі дані, які можуть не оновлюватися досить довго;
- високе енергоспоживання таких пристроїв, адже йде прямий зв'язок із системами супутників та інтернету;
- досить важкий для розуміння інтерфейс програми навігатора, який може відштовхнути потенційних користувачів від програмного забезпечення або продукту.

Усі вищезазначені проблеми може бути вирішено за допомогою використання технології доповненої реальності. Перш за все доповнена реальність має інтуїтивно зрозумілий інтерфейс для роботи з нею, а також вона не відволікає користувача від дійсної реальності. Великі компанії вже почали розробку окулярів доповненої реальності, що у подальшому дасть змогу мігрувати усі поточні рішення, які застосовуються на смартфонах на ці пристрої. Це вирішить багато проблем, пов'язаних із користуванням тих або інших додатків та представить новий інтерфейс взаємодії комп'ютера та людини у повсякденному житті.

Впровадження інформаційних технологій у усі сфери життя дали потужний поштовх у створенні нових комп'ютерів та інтерфейсів взаємодії з нами. Це дивовижно, адже менш ніж п'ятдесят років тому люди навіть уявити не могли, як комп'ютер та обчислення тісно увійдуть у повсякденне життя звичайного користувача. Взаємодія комп'ютера та людини сьогодні є невід'ємною частиною кожного з нас, саме тому ця галузь є популярною та обговорюваною.

Взаємодія людини та комп'ютера зараз швидко розвивається. Регулярне і систематичне вдосконалення технологій призвело до появи абсолютно нових парадигм призначених для користувача інтерфейсу комп'ютеру. Глобалізація в сучасному суспільстві стала тією самою тенденцією, яка поширилася і на комп'ютерні науки. Віртуальну реальність торкнулася та сама глобалізація, що призвело до появи нового терміну – «доповнена реальність». Основна маса розроблених технологій зосереджена в більшості випадків на взаємодії людини і використанні ним пристроїв (стаціонарний комп'ютер, ноутбук, смартфон тощо). Доповнена реальність, в свою чергу, на підставі використання деяких конкретних комп'ютерних технологій, робить можливим поліпшення інтерфейсу людини і його взаємодії з реальним навколишнім світом.

Актуальний етап досліджень доповненої реальності стартував вже давно. За кордоном видавалося дуже багато робіт на дану тему, хто вивчає її актуальність і неймовірний потенціал. Дуже довго подібні технології здавалися важко реалізованими і залишалися більш гіпотетичними, ніж реальними. Коли ж розвиток споживчої електроніки досяг рівня, здатного забезпечити масове впровадження

даної технології, доповнена реальність стала однією з найшвидших у розвитку галузей інформаційних технологій.

На теперішній час ця технологія є однією з найбільш цікавих тем для досліджень. Доказом цього є неймовірна хвиля популярності однієї з ігор для смартфонів – «Pokémon Go» від компанії Nintendo в 2016 році.

Доповнена реальність (від англ. Augmented Reality, AR – «розширена реальність») – це реальність із вбудованими (розширеними) в неї можливостями, яку сприймає людина. Тобто це те, що виходить в результаті введення в поле сприйняття будь-яких сенсорних даних, з метою доповнення відомостей про оточення [1].

Однак, якщо звернутися до витоків, то інструменти доповненої реальності пробував створити ще в п'ятнадцятому столітті архітектор Філіппо Брунеллески. Він малював об'єкт, який доповнював інший, існуючий в реальному світі і пропонував дивитися на нього через дзеркало з отвором. А в 1994 році Пол Милграм створив спектр реальностей. В цей спектр входить: фізичне навколишнє середовище (всі об'єкти реальні, їх можна відчувати), доповнена реальність, доповнена віртуальність і, за підсумком, повністю віртуальна реальність (всі об'єкти і середовище згенеровано комп'ютером).

Том Коделл, дослідник в корпорації побудови авіаційної техніки Boeing, ввів термін «доповнена реальність» в 1990. Він застосував термін до приладу з цифровим дисплеєм, який «звертав» робітників під час збирання електричних проводів в літаках. У той час «доповнена реальність» визначалася як схрещування віртуальної і фізичної реальності, де цифровий візуальний ефект додавали до реального світу, щоб поліпшити його сприйняття людиною.

Існують також синоніми терміну «доповнена реальність»: «розширена реальність», «поліпшена реальність», «збагачена реальність», «збільшена реальність». Але таке використання термінів в глобальному сенсі не є правильним. Наприклад, терміни «розширена реальність», «збільшена реальність», «збагачена реальність» застосовні лише для позначення певних форм реалізації та аспектів практичного застосування доповненої реальності.

На сьогоднішній день доповнена реальність має дуже великий спектр застосування. Її використовують, перш за все, у військовій області, області медицини, авіації і мобільних технологій.

Доповнену реальність широко використовують в медицині, так як це значно спрощує процес проведення складних операцій і допомагає пацієнтам краще розуміти, що відбувається з їх організмом. Технологію використовують також для наочного навчання студентів-медиків. Технологія застосовується у військовій справі: на бойовому транспорті (літаки, машини, танки) і в обмундируванні самих військових (шоломи, зброя). Один із прикладів цього застосування доповненої реальності – проектування на лобовому склі кабіни літака інформації для пілота (швидкість, погодні зміни і інші показники).

Доповнену реальність активно використовують у інформаційному просторі. Поширення так званих браузерів доповненої реальності – зокрема, Wikitude, Layar, blippAR і інших, виводить технологію в масове використання. У газети, буклети, рекламні проспекти, журнали, карти поміщаються зображення-мітки для подальшої візуалізації цифрових об'єктів. У ролі додаткової інформації виступають всі види цифрових даних: текст, зображення, відео, звук або тривимірні об'єкти, статичні або анімовані. Реалізація відбувається за допомогою спеціальних додатків для смартфонів.

Існує досить багато додатків, які дозволяють за допомогою доповненої реальності отримати розширені відомості про оточення: браузери доповненої реальності, ігри, інформаційні програми.

Зростаюча популярність технології серед споживачів пов'язана з тим, що розвиток технологій дозволив ці функції реалізувати. Наприклад, обчислювальна потужність і набір датчиків в смартфонах дозволяють виробляти накладення будь-яких цифрових даних на отриманий в реальному часі звук або зображення. В іграх на все ті ж смартфони вже давно використовуються методи доповненої реальності. Наприклад, ще в 2004 році була випущена гра «Mosquitos», в якій зображення накладалося на відео, яке надходить до екрану з камери пристрою. Як уже було згадано вище, вибуховий успіх мала гра «Pokémon Go», що привело до перенесення

на додану реальність таких ігор як «Minecraft», «Super Mario Bros» та інших, менш популярних ігрових додатків.

Великі компанії вже давно почали працювати над доповненою реальністю. Деякі з цих проектів вже встигли зачинитись із-за складності реалізації, або неготовності користувачів до таких технологій. Корпорація Google працювала над окулярами «Project Glass» – одна з перших спроб вивести доповнену реальність в маси в 2013 році, на даний момент розробка припинена.

Одночасно з окулярами розроблялася платформа для доповненої реальності «Project Tango», яка була випущена у 2016 році та показала потужних потенціал цієї технології. Проект «Project Tango» – це технологія побудови трьох вимірної моделі навколишнього середовища за допомогою смартфона або планшета. Група АТАР створила 5-дюймовий смартфон, оснащений стереокамерой, сенсорами і програмним забезпеченням, які відстежують стан смартфона в 3D-просторі, а також сканують навколишній світ в реальному часі зі швидкістю 250 тисяч вимірів в секунду. Все це об'єднується в єдину 3D-модель за допомогою унікального процесора Myriad 1, розробленого стартапом Movidius. Коли модель готова, пристрій може перманентно визначати своє місце розташування всередині неї.

Microsoft в 2016 році випустила «Hololens», а влітку 2017 року Apple випустила платформу «ARKit» і відразу за нею компанія Google випустила свій аналог – «ARCore». Аналогічні розробки ведуть інші великі компанії, а також багато початківці компанії, так як ця тема дуже перспективна.

Ринок напрямків доповненої та віртуальної реальності знаходиться на дуже ранній стадії розвитку, але при цьому є вкрай інвестиційно привабливим, тому що він потенційно може стати дуже розвиненим та навіть перевершити ринок мобільних пристроїв. Загалом, в індустрію за минулий 2016 рік було інвестовано понад двох мільярдів доларів. Це величезна можливість для стартапів та інвесторів. Зараз основна виручка генерується шоломами віртуальної реальності і контентом, який для них створюється. Але картина буде змінюватися – велика ставка буде зроблена на доповнену реальність, так як на її розвиток виділяються величезні ресурси передовими компаніями в світі електроніки та інформаційних технологій.

Навіть Ілон Маск згадував у своїх інтерв'ю про перспективи і можливості, які нам може відкрити доповнена реальність. За його словами, доповнена реальність та пристрої, пов'язані з нею, можуть повністю замінити ринок смартфонів.

Прогнозовані гігантські розміри ринку породжують величезні можливості для створення власного бізнесу. Згадаємо Apple і їх мобільні додатки: iPhone зробив революцію на ринку смартфонів. Хто заробив на цьому більше всіх? Розробники, які робили додатки під операційну систему даних смартфонів. Так як на той момент ринок додатків був дуже маленький, попит на них був величезний за рахунок дистрибуції цих пристроїв. Люди купували і качали практично все, що потрапляло на очі – популярними ставали навіть ігри, які раніше здавалися приреченими на провал.

Така поведінка ринку може скластися і на ринку технологій доповненої реальності. Це великі можливості для користувачів, компаній та власне фахівців інформаційних технологій усіх галузей.

У основі доповненої реальності лежить технологія комп'ютерного зору, адже пристрій повинен аналізувати навколишнє середовище і доповнювати його. Комп'ютерний зір – це набір методів, які займаються вилученням інформації з зображень, і важливо відзначити, що зображення можуть бути самих різних типів. Це можуть бути фотографії, потокове або статичне відео, набори фото або медичний томографічний знімок.

Останнім часом дуже популярними алгоритмами є ті, що працюють із зображеннями і поєднують колірну інформацію про точки і їх положення в просторі. Так сталося, тому що сенсори, які отримують подібну інформацію, стали доступнішими, а розуміти фізичну реальність за допомогою таких сенсорів роботам або комп'ютерами стало набагато легше і швидше. Саме приріст потужності комп'ютерів надав таку змогу.

Інформація, яку різні алгоритми витягають із зображень в комп'ютерному зорі, також може мати різну природу, тобто різний тип. Деякі алгоритми просто розбивають зображення на частини, які відповідають окремим об'єктам або частин об'єктів.

Другий важливий розділ комп'ютерного зору – це побудова тривимірних моделей із зображень, наприклад з набору фотографій або з відео. Для наочності візьмемо приклад із тисячі фотографій, зроблених різними людьми в різний час, що відобразили Таймс Сквер в Нью Йорку. З даних фотографій буде будуватися тривимірна модель площі. Щоб це зробити комп'ютеру потрібно зіставити частини сцени, тобто зрозуміти, що на різних зображеннях різні фрагменти відповідають одним і тим же об'єктам в тривимірному світі. Це робиться за принципом відбору унікальних фрагментів і поширення сформованого знання на менш унікальні фрагменти. Комп'ютер для цього використовує алгоритми оптимізації, а ось що для цього використовують люди або тварини, коли будують в мозку тривимірні карти, ще до кінця не відомо. В результаті такого моделювання комп'ютер отримує фактично тривимірний зліпок сцени, і розуміє становище камери в момент, коли відбувалася фото та відео зйомка. Надалі ця інформація може використовуватися різними способами. Наприклад, для вимірювань, так званої фотограмметрії. Або отриману тривимірну модель можна використовувати в додатку комп'ютерної графіки в процесі створення фільмів (спецефекти: поєднання реальних і віртуальних об'єктів). Крім цього тривимірна модель активно використовується в комп'ютерних іграх, додатках реальної і доповненої реальності.

Комп'ютерний зір сьогодні займається рішенням проблеми розуміння зображень та що на них відбувається. Це називається високорівневим комп'ютерним зором. Цей розділ займається завданнями та інструкціями до комп'ютеру, що дуже легко сформулювати. Наприклад, «виписати всі об'єкти з даного фото, визначити їх клас і місце розташування на фото». Або дано відео, на якому потрібно визначити, яким видом діяльності зайняті об'єкти [24].

Саме завдяки вищесказаному, є можливість створювати нові ідеї, програми та продукти, які дадуть змогу користувачеві відчути та отримати нові враження від взаємодії з його пристроєм. А швидкість створення нових технологій та конкуренція між компаніями-гігантами сприяє гарному розвитку цієї галузі, що дає змогу багатій кількості людей займатися технологіями, пов'язаними з доповненою та віртуальною реальністю.

Отже, можна зробити висновок про доцільність використання продукту який підтримує доповнену реальність. Ця технологія дозволяє створити новий досвід використання мобільних додатків, покращити взаємодію людини та комп'ютера, покращити існуючі системи та технології.

1.2 Виявлення проблем та актуалізація рішень.

Доповнена реальність потребує багато ресурсів від комп'ютеру, для того щоб аналізувати зображення, обчислювати необхідні дані та відображати інформацію користувачеві у реальному часі.

Продуктивність сучасних смартфонів вже зараз можна порівняти зі стаціонарними комп'ютерами минулого десятиліття. Розробники нарощують кількість ядер, обчислювальних блоків графічних процесорів, підвищують швидкість роботи контролера пам'яті. Творці програмного забезпечення не відстають, і створюють ігри і програми, які навантажують навіть сучасні значні потужності.

За останні роки усі мобільні процесори створюються використовуючи багатоядерну архітектуру. На сьогоднішній день існують процесори, які мають в своєму складі від десяти обчислювальних ядер. Але далеко не завжди збільшена кількість ядер стає явною перевагою в питанні продуктивності мобільного пристрою. Більша кількість ядер в основному справляється із завданням збільшення кількості обчислювальних потоків, що рівносильно збільшенню кількості одночасно виконуваних завдань на пристрої.

Мобільні додатки повинні бути оптимізовані під роботу одночасно з декількома обчислювальними ядрами з представлених в процесорі. А це не завжди враховується розробниками програмного забезпечення, тому вище згадувалося, що більша кількість ядер не завжди і є перевагою. Наприклад, процесори Apple тільки в 2015 відійшли від двоядерних процесорів і зараз перейшли до 6-ти ядерних. Але

це не заважало їм протягом більше п'яти років залишатися одними з кращих на світовому ринку. Це є можливим завдяки оптимізації програмного забезпечення і використання комплектуючих пристроїв, спеціально зроблених для роботи з даними чіпами. Процесор Apple A11 був представлений 12 вересня 2017 року разом з iPhone 8, iPhone 8 Plus і iPhone X. Він містить 6 обчислювальних ядер, з яких 2 ядра є продуктивними, а інші 4 – енергоефективними. У ядрах використовується чотири з половиною мільярдів транзисторів, які виготовлені за передовими технологіями щільного розміщення. Також, у нових пристроях компанії Apple вперше використовується вбудована графіка власного виробництва. Це дає змогу максимально оптимізувати роботу з графікою та створити нові, потужні інструменти та інтерфейси для роботи з цими чіпами.

Зараз на ринку мобільних процесорів найбільш крупними та впливовими є наступні компанії:

- Qualcomm – створює чипсети серії Snapdragon;
- Samsung – створює чипсети серії Exynos, які використовуються у своїх продуктах, а також продаються;
- MediaTek – створює флагманські процесори під торговою маркою Helio;
- Huawei – чипсети під суббрендом HiSilicon, які вбудовуються у власні смартфони;
- Apple, як вже зазначалося вище, також використовують процесори тільки у своїй продукції.

При цьому неможливо точно сказати, які чіпи потужніші, а які слабкіше. Авжеж, існують всілякі тести і бенчмарки, але їх результат варто називати умовним та гіпотетичним. Так як на практиці кожен процесор працює у власному режимі і дуже рідко доводить тактові частоти до максимально можливих [5].

Фактично, зараз, в процесі розробки нового мікропроцесора, головне обмеження – це не те, якими можливостями він повинен володіти або яка повинна бути його продуктивність. Головне обмеження – це те, скільки він повинен коштувати. Звідси і випливають всі наслідки.

Саме розвиток процесорів та їх доступність дає змогу нам проектувати потужні обчислювальні системи навіть на мобільних пристроях таких як смартфони, планшети та навіть смарт-часи. У подальшому це дозволить використовувати ці пристрої, як стаціонарні комп'ютери. Деякі компанії вже намагаються це зробити, впроваджуючи нові рішення у розробці програмного забезпечення.

Наступною проблемою є точність даних отриманих із камер пристрою. У системах, які використовують камери є досить велика похибка, адже існує багато факторів, які впливають на доцільність фінального зображення.

Доповнена реальність дозволить вирішити ці проблеми шляхом аналізу даних отриманих із сенсорів, які надають більше інформації ніж звичайна камера пристрою. Для вирішення цієї проблеми на мобільному пристрої з операційною системою iOS можна використати зручний інструмент для розробки мобільних додатків, які використовують доповнену реальність – ARKit.

1.3 Можливі шляхи вирішення задачі за рахунок створення доповненої реальності на мобільному пристрої

У травні 2017 року компанія Apple презентувала світу ARKit – набір інструментів для розробки додатків віртуальної і доповненої реальності. Компанія зробила розрахунок на те, що розробники почнуть з ентузіазмом освоювати новий інструмент і тим самим дадуть вектор, в якому напрямку компанії далі буде розвивати технологію впровадження віртуальних об'єктів в реальний світ. Така стратегія себе виправдала і вже на сьогоднішній день є значна кількість додатків і доповнень до існуючих з використанням ARKit [17].

Активні експерименти розробників дозволили зовсім скоро отримати доступ до використання технології простим користувачам. Це такі програми як: проста віртуальна рулетка, ресторанні додатки, яке зображують на вашій порожній тарілці

віртуальне блюдо в момент замовлення. Швидке розповсюдження і популяризація цієї технології дає потужний поштовх у розробці програмного забезпечення з використанням доповненої або змішаної реальності. У минулому ці технології використовувалися лише вузько направленими сферами, такими як інженерія, військова справа чи медицина. Сьогодні ж, ця технологія дозволяє вийти розробникам програмного забезпечення на новий рівень, що у подальшому сприяє тому, що вже усі сфери життєдіяльності, які так чи інакше пов'язані з інформаційними технологіями, будуть використовувати змішану, доповнену та віртуальну реальності у повсякденному житті. Тенденція впровадження цих технологій вже спостерігається і сьогодні.

Одним з перших додатків, що використовують цю технологію, став додаток Джеймі Шорда. Крихітна компанія «Shord», що має в штаті всього чотирьох осіб, розробила додаток «Pigeon Panic». «Це абсолютно дурна гра, яка дозволить вам виконати мрію про безтурботне розполохування великих зграй віртуальних голубів», – описує свій додаток Джеймі Шорд.

Якщо компанія Apple планує почати революцію в сфері доповненої реальності, то перед нею полягає завдання про подолання декількох серйозних перешкод. В першу чергу – це наявність значної кількості конкурентів.

Google кілька років займався розвитком власної платформи доповненої реальності під назвою «Tango», яку поки можливо запустити тільки на декількох смартфонах, а з виходом ARKit компанія була змушена змінити курс. Так Google відмовилася від бренду «Tango» і запустила проект, який буде працювати на мільйонах пристроїв працюють під операційною системою Android, а також закликала розробників взяти участь у розвитку технології проекту. Розробники самої корпорації вже встигли продемонструвати перші плоди своїх експериментів – нове доповнення до Google Streetview, яке дозволяє виявитися біля входу в Британський музей і простеньке навчальне додаток про те, як користуватися кавомашиною. Цей інструмент має назву ARCore.

Система досить чітко і доцільно відображає тіні, враховує масштаб та підлогу. Також технології доповненої реальності, які були впроваджені у

інструмент ARCore дозволяють не тільки зображати статичний, нерухомий об'єкт, а також анімувати його, та робити так, щоб він аналізував оточуюче середовище та змінював свою поведінку відповідно до нього.

Microsoft – це ще один серйозний гравець, що виходить на ринок доповненої реальності, просуваючи власну розробку – шолом HoloLens. Це окремий пристрій у вигляді шолома, який дозволяє користувачеві повністю зануритись у віртуальну або доповнену реальність, та навіть відчувати та взаємодіяти з цією системою.

З його допомогою компанія має намір створювати в школі і на робочому місці, так звану, змішану реальність. З аналогією з тим, як свого часу для навчання в університетах була абсолютно безкоштовно надана операційна система Windows. Цей шолом дає можливість більш глибокого занурення, ніж програма доповненої реальності на екрані телефону, але на даний момент це недешево задоволення. І тому на актуальному етапі розробки продукт ще не націлений на споживчий ринок.

Інструмент ARKit – революція в світі доповненої реальності, яка може змінити світ. На стороні нової технології маса переваг:

- мільйони готових пристроїв і зацікавлених користувачів у всьому світі. Для використання доповненої реальності, користувачам продукції Apple не потрібно докуповувати спеціальне обладнання, окуляри, гарнітури, кріплення і інші аксесуари. Використовувати додатки можна на будь-якому iPhone, iPad і iPod Touch, які вже багато років б'ють всі рекорди з продажу і утримують найвищі позиції рейтингів;
- у технології все влаштовано таким чином, що працює без додаткових позначок, маркерів та інших «помічників». Можна легко розмістити потрібні об'єкти прямо зі смартфона, і вони відразу ж будуть доступні іншим користувачам мережі. Система самостійно працює з тінями, перспективою і масштабом, що значно полегшує завдання розробників;
- ринок доповненої реальності на сьогодні ще пустує. Більшість додатків, які зараз можна знайти в App Store під тегом AR, побудовані на старому принципі міток і маркерів;

- віртуальна реальність, на жаль, до сих пір знаходиться в зародковому вигляді. Не перший рік на прилавках магазинів електроніки продається значна кількість аксесуарів для занурення у віртуальну реальність, але це досі не штовхнуло мобільних розробників на випуск ігор і додатків. Користувачі не захотіли масово зануритися у віртуальну реальність, але все ж інтерес до технології проявили. Відповідно вони готові для прийняття подібного феномену як доповнена реальність;
- Apple допоможе з розкручуванням та популяцією. На World Wide Developer Conference 2019 року розробники отримали серйозні інструменти для створення гідного контенту та вже продемонстрували наробітки у цій сфері. Адже з економічної точки зору дітище такого гіганта як Apple – це апріорі вигідне вкладення для інвесторів;
- доповнена реальність – новий рекламний майданчик. Користувачі можуть отримати не тільки чергову порцію розважального контенту, але і корисні програми, які будуть встановлювати із задоволенням.

Інтерактивні об'єкти в доповненій реальності можуть з'явитися для допомоги відвідувачам в банках, супермаркетах, аптеках, вірогідна поява спеціальних стендів в музеях, пам'ятках і парках розваг. Представиться можливість вибирати меблі або предмети інтер'єру, розміщуючи їх віртуальні копії прямо у себе вдома. Це можна використовувати для продажу косметики та одягу, а також багато чого іншого.

ARKit – це набір інструментів, який є новітнім і неупередженим поглядом на поняття доповненої реальності. Розробка стала можливою завдяки співробітникам Apple та сучасному обладнанню.

Робота технології забезпечується за принципом використання даних з камери та вбудованих сенсорів. За допомогою технології ARKit камера пристрою веде спостереження і допомагає визначити поверхні, потім проводяться всі розрахунки, які пов'язані з детальною побудовою всіх тіней і змін положення при русі камери. ARKit працює під операційною системою iOS 11 або новіше.

Також, розробники вищезазначеного програмного продукту забезпечили підтримку ARKit на платформі Unity [14]. Тепер розробник програмного

забезпечення не повинен регулярно продумувати нові способи вирішення завдань. Існує готовий фреймворк, який можна використовувати в розробці програми. Це значно скорочує час і полегшує завдання розробника, а також показує певну етику і стандарти існуючі між розробниками додатків на платформу iOS. Продукти, які написані з використанням інструменту доповненої реальності, зможуть бути використані на апаратах, оснащених комплексом процесорів груп A9, A10, A11 і вище.

Розробники відреагували на новий продукт ARKit дуже позитивно. Начальник маркетингового відділу продуктів компанії Apple Грег Джозвіак навіть стверджував, що реакція була неймовірною.

Набір інструментів тепер дозволяє розробляти програми для виконання таких завдань:

- додатки для проведення лінійних вимірювань;
- малювання в тривимірному просторі;
- різноманітні ігри;
- інтерактивні додатки;
- розміщення моделей об'єктів на будь-якій місцевості.

Спектр застосування цього інструменту дуже широкий. Всі сфери життя людини, де може бути задіяна камера, датчик руху, освітлення, можуть бути використані при розробці програм.

Всесвітньо відома компанія IKEA вельми зацікавилася цим продуктом, адже він дозволить створити віртуальний каталог продукції, в який може увійти більше 2 500 одиниць товару. Це дозволить компанії значно зменшити об'єми своїх приміщень для продажу продукції, а також значно швидше впроваджувати новий товар до електронного каталогу.

Для звичайного користувача все буде вкрай просто: завантажувати додатки і безпосередньо використовувати можливості технології.

Усі вищезазначені технології використовують принцип візуальної одометрії. Візуальна одометрія – метод оцінки стану і орієнтації пристрою або іншого обладнання, наприклад робота, за допомогою аналізу послідовності зображень,

знятих встановленої на ньому камерою (або камерами). Методи візуальної одометрії використовуються, наприклад, в оптичних мишах.

Звичайна одометрія не може бути застосована до приладів із нестандартним типом руху (наприклад – крокуючі роботи). Також достовірно відомо, що звичайна одометрія страждає втратами точності, які посилюються при русі пристрою по нерівній поверхні, простіше кажучи, зі стабілізацією одержуваного зображення.

Візуальна одометрія – це процес отримання одометричної інформації, використовуючи послідовність зображень з камер пристрою. З неї виходить інформація про пройдено відстань та вектор переміщення. Візуальна одометрія дозволяє поліпшити навігаційну точність пристроїв або роботів, будь-якого типу пересування і на будь-якій поверхні [24].

Ця технологія і алгоритм її роботи буде розглянуто у наступних розділах, при аналізі проектування та розробці програмного продукту.

1.4 Постановка задачі дослідження

Відстеження напрямку зору має багато можливих застосувань – від засобу спілкування людини з комп'ютером до допомоги людям з обмеженими можливостями. Розпізнавання напрямку зору неодноразово використовувалось для визначення поведінки користувачів в мережі Інтернет [1]. Розуміння того, на що звертають увагу користувачі під час перегляду веб-сторінок є важливим для просування реклами, розробки більш зручного веб-інтерфейсу, оптимізації веб-сайтів з метою покращення сумісності та полегшення використання комп'ютера в цілому. Найпростіший варіант відстеження погляду – це безпосередня взаємодія людини з комп'ютером з метою підвищення здатності людини до взаємодії з комп'ютером. Використання зору може прискорити та зробити взаємодію більш інтуїтивно зрозумілою та природньою у порівнянні із звичайними засобами управління – клавіатурою та комп'ютерною мишкою. Це також може допомогти

користувачам з обмеженими можливостями отримати доступ до управління іншими пристроями взаємодії.

Існуючі на теперішній час системи для виявлення та стеження за поглядом досить часто використовують інфрачервоне світло, стереокамери та спеціальне обладнання, що призводить до погіршення зручності використання цих систем. Крім того, інфрачервоне випромінювання не є повністю безпечним – воно впливає на людське око, яке не може повністю відобразити інфрачервоне випромінювання від сітківки через природу людського ока. [2, 4, 5].

На підставі проведеного аналізу предметної галузі було прийнято рішення щодо проведення дослідження методів управління складними об'єктами за допомогою зору людини.

Основною відмінністю складних об'єктів є наявність у них значної кількості властивостей на відміну від простих об'єктів. Під управлінням складними об'єктами розуміють таку зміну їх властивостей, яка викликає зміни як у самому об'єкті, так і у варіантах його використання.

Основним засобом впливу на властивості об'єкта в комп'ютерних системах є курсор мишки та пункти меню, що дозволяють змінювати ці властивості.

Існує декілька методологій розробки постановки задачі програмного забезпечення: «SMART» та «TOTE». «SMART» – це абревіатура від Specific, Measurable, Attainable, Relevant, Time-bound. Іншими словами, задача повинна бути конкретизованою, задачу можна оцінити, вона повинна бути досяжною, доречною та її можна укласти у деякі часові межі. Цей метод постановки задачі використовується перш за все менеджерами та є повним і доречним при розробці програмного забезпечення. Другий метод – «TOTE» – це абревіатура від Test, Operation, Test, Exit [18]. Вона визначає методологію, яка найбільш підходить для розробників програмного забезпечення, та передбачає наступне:

- чітко сформулювати бажаний стан, який розробник повинен досягти;
- які дії розробник повинен виконати, щоб досягти бажаного результату;
- за якими ознаками розробник зрозуміє, що він просунувся у бік отримання бажаного результату;

– за якими критеріями розробник зрозуміє, що результат остаточно досягнуто.

Ця методологія є чіткою та детермінованою та найбільш вдало підходить для розробки програмного забезпечення у команді, яка оперує малою кількістю людино-ресурсів. Надалі постановка задачі буде виконуватись у відповідності до цієї методології [13].

Підводячи підсумки можна сформулювати наступне. Метою роботи є дослідження методів управління об'єктами за рахунок відстеження погляду користувача за допомогою вбудованої камери та сенсорів пристрою.

Доступ до властивостей об'єкту буде здійснюватися за допомогою позиціювання курсору та вибору відповідних пунктів меню. Ці дії будуть виконуватись виключно за допомогою відстеження напрямку зору людини та не будуть використовувати додаткове обладнання.

Технології, які будуть використовуватися під час розробки програмної моделі системи, будуть розглянуті більш детально у наступних розділах.

2 ДОСЛІДЖЕННЯ АЛГОРИТМУ ВІДСТЕЖЕННЯ ЗОРУ ЛЮДИНИ

2.1 Опис проблеми відстеження очей

Погляд – це напрямок, в якому дивиться людина. Під розпізнаванням погляду будемо розуміти перевірку та відстеження цього напрямку. Завданням цієї роботи є дослідження методів відстеження погляду та розробка алгоритму відстеження для його подальшої програмної реалізації.

У процесі будь-якого перегляду, наприклад, зображення чи піктограм на комп'ютерному моніторі або екрані смартфона, очі вказують напрямок, на якому вони зосереджені. Через фізіологічні обмеження люди можуть спостерігати лише невеликий регіон навколо місця, на якому зосереджені їх очі. Це властивість людського ока може бути використана для створення системи розпізнавання погляду. Відстеження очей можна використовувати для моніторингу та реагування на основі точки зору користувача [4]. Під час звичайного використання комп'ютера користувач часто робить рухи головою у всіх напрямках. Таким чином, точка, на якій користувач фокусується, залежить не тільки від напрямку зору очей користувача, а також від нахилу його обличчя, відстані між обличчям та монітором, а також багатьох інших параметрів [4]. Через те, що є багато факторів, які впливають на розпізнавання зовнішнього вигляду, а також на наявність шумів на зображеннях, використання штучних нейронних мереж для цього завдання буде найбільш підходящим. Існують певні обмеження щодо швидкодії роботи системи, адже для простоти управління курсором миші необхідна мінімальна затримка, що вимагає значної кількості обчислювальних потужностей, оскільки обробка зображень є ресурсомістким процесом. В ідеальному сценарії користувачеві потрібно буде лише встановити програмне забезпечення та використовувати камеру для взаємодії з комп'ютером або смартфоном.

Існує значна кількість досліджень, присвячених проблемам відстеження очей. У своєму дослідженні 1996 року Ток намагався відстежувати положення повік людини, поки вона знаходилась в транспортному засобі. Ця система може

визначити, коли водій стає сонливим і тим самим зменшити кількість аварій, викликаних втомою водія [5].

Стеження за очами під час використання тесту має ключове значення для вивчення веб-сайтів та програмних додатків. Завдяки цьому ми можемо перевірити:

- траєкторію руху очного яблука та час фокусування на конкретному об'єкті;
- у якій послідовності об'єкти привертають увагу респондентів;
- дослідити відволікаючі елементи;
- визначити елементи, які часто розглядаються та на які більше звертають уваги;
- правильність відображення об'єктів на веб-сайті або їх послідовність.

Використання спостереження за очами для вивчення ергономіки. З точки зору замовника, ергономіка є ключовим при виборі товару. Тут вимірюється, як простий і зрозумілий інакше товар використовується. Естетичні дослідження на тему ергономіки використовуються при тестуванні мобільних телефонів, аудіо- та відео- пристроїв, домашніх приладів. Завдяки цьому дослідженню ми можемо перевірити:

- точність елементів;
- відволікаючі фактори;
- які елементи не були знайдені і чому;
- чи було передбачуваним їх використання та поведінка;
- чи були елементи саме в тих місцях, де респондент на них очікував.

Використання відстеження очей у рекламі. Перші секунди, протягом яких клієнти контактують з рекламою, дуже важливі. Відстеження напрямку зору дозволяє з'ясувати, які елементи є більш привабливими та привертають увагу, а які залишилися невидимими. Використовуючи відстеження очей з'являється можливість дізнатися про ефективність наступних типів реклами:

- реклама в пресі;
- відеореклама;
- інтернет-реклама;

- зовнішня реклама;
- електронна розсилка.

Використання спостереження за очима під час розташування товарів на полицях магазинів. Відстеження очей використовується в торгівлі, коли сучасні методи та підходи до розміщення товарів не дають задовільних результатів. Це стосується як макрорівня, так і мікрорівня їх розташування. Макрорівень навігації відноситься до місця розташування окремих категорій товарів у всьому магазині. Тут спостереження за очами спрямоване на те, щоб клієнти могли не тільки вільно пересуватися магазином, а й одночасно бачити всі інформаційні елементи, які допомагають досягти місця їх розташування. Навігація на мікрорівні – це спроба створити оптимальне розташування окремих товарів на полиці, зробити розміщення продуктів максимально ефективним.

Дослідження відстеження очей, спрямоване на тестування публічних інформаційних систем. Завдяки їм наше середовище стає все більш зрозумілим, тобто ми можемо з легкістю знайти шлях до станції метро, каси чи певного магазину в торговельному центрі. Стеження за очами також використовується, коли необхідно оцінити евакуаційні та попереджувальні знаки [7].

2.2 Огляд аналогів

На теперішній час існує значна кількість програмних систем для стеження за очима. Вони надають нам такі можливості: контроль за рухами мишки, отримання інформації про рух очей користувача та аналіз даних тощо.

Німецький програміст Маттіас Хеммінгссон (Mattias Hemmingsson) розробив розширення для Google Chrome, яке ставить на паузу відео на YouTube, коли користувач відвертається від екрану [25]. Для відстеження напрямку погляду система використовує вбудований FaceDetector API браузера та веб-камеру пристрою. В якості переваг можна вказати простоту установки та використання,

можливість використання на різних платформах. До недоліків слід віднести те, що система працює лише з веб-камерами, відсутні прості способи отримання даних та в системі не передбачено функції аналізу даних. Також система немає підтримки.

GazePointer – ще одна програма, яка призначена для роботи в середовищі Windows та не вимагає додаткового обладнання. До переваг цієї програми слід віднести простоту установки та використання, але присутній і ряд недоліків: вона працює лише з веб-камерами, з боку розробника відсутня підтримка, не передбачено функції аналізу даних.

Ogama – програмне забезпечення з відкритим кодом, яке було розроблено у Вільному університеті Берліна, однак протягом останніх 5 років проект було залишено без розвитку. Головною перевагою програми є забезпечення основних параметрів аналізу даних. Недоліки: відсутня підтримка та оновлення.

OpenEyes – це програмне забезпечення з відкритим кодом дозволяє відслідковувати очі з використанням інфрачервоного та видимого освітлення. Обробка отриманих даних здійснюється за допомогою пакету Matlab. До переваг програми слід віднести більш високу точність отримання даних за рахунок використання веб-камер та інфрачервоних камер. Але є і суттєві недоліки: необхідність придбання та встановлення Matlab (комерційна програма), навички роботи в Matlab, відсутність підтримки, необхідність самостійно розробляти функції аналізу даних.

PyGaze – програмне забезпечення розроблено з використанням Python у 2014 р. групою ентузіастів з Оксфордського, Екс-Марсельського та Утрехтського університетів. У якості головної переваги слід відзначити наявність потужної системи аналізу даних, однак для її використання необхідно мати знання з програмування на Python.

OpenGazer орієнтований на підвищення доступності використання комп'ютера та підтримується Samsung та благодійним фондом Gatsby. До переваг слід віднести потенційну сумісність з ОС Apple (вимагає знання програмування). Недоліки: вимагає знань щодо використання Linux, відсутність підтримки.

TurkerGaze – програма, розроблена дослідниками з Принстона. Система працює на Linux, однак її функціонування також залежить від декількох інших програм Linux. Переваги: Забезпечує основні функції аналізу даних. Недоліки: повноцінне використання системи вимагає застосування підголівника (додаткове обладнання), вимагає знань щодо використання Linux.

GazeParser – це програмне забезпечення складається з двох компонентів: GazeParser (конвертація та аналіз даних) та SimpleGazeTracker (використовується для запису руху очей), створених з використанням Python. До переваг відноситься вміння системи аналізувати дані. Недоліки: необхідна окрема камера для відстеження руху або камера машинного зору, спеціальний підголівник для обмеження руху голови, знання Python.

ITU Gaze Tracker - програма, створена в Копенгагенському університеті. Це платформа з відкритим кодом, яка використовується для спрощення доступу до технологій. Переваги: простота установки. Недоліки: необхідність створення власного інфрачервоного трекеру очей (якщо веб-камера не підтримує інфрачервоне бачення), відсутність підтримки.

Під час аналізу програмного забезпечення для розпізнавання напрямку зору користувача були розглянуті принципи роботи таких додатків, їх переваги та визначені недоліки. На підсумок слід зазначити, що всі існуючі системи мають певні обмеження – такі як необхідність в знаннях додаткових технологій або використання сторонніх пристроїв. Ще один суттєвий недолік – ці аналоги були розроблені досить давно, і багато з них вже не підтримується розробниками.

2.3 Аналіз і опис основних етапів застосування алгоритмів доповненої реальності для відстеження зору людини

Основна ідея полягає у створенні такого алгоритму, який дозволить відстежувати погляд та увагу користувачів на конкретному пристрої на базі

операційної системи iOS. Існує багато бібліотек, які забезпечують деяку необхідну базову функціональність, але на сьогоднішній день не існує жодного відомого алгоритму, який би надав можливість відстежувати та аналізувати увагу користувачів під час користування своїми пристроями.

Перш за все, потрібно проаналізувати та відстежити обличчя користувача за допомогою фронтальної камери. Тоді треба виявити очі користувача, щоб знайти його зіниці. Маючи наявну інформацію про зіниці користувача, обчислимо вектори зору користувача. Зрештою, щоб досягти найкращої точності, треба виміряти приблизну відстань від пристрою до голови та очей користувача. Беручи до уваги все вище, треба проаналізувати положення пристрою в реальному світі та створити площину пристрою, створити рівняння з існуючими векторами зору користувача для виявлення точок перетину зору користувача на площині пристрою. Зрештою, ми можемо перетворити обчислені точки перетину в точки, пов'язані з екраном пристрою (внутрішній простір координат пристрою). Маючи інформацію про погляд користувача, можна проаналізувати інформацію, зробити її гладкою та точною, використовуючи існуючі математичні підходи.

У 2016 році компанія під назвою Apple представила нову революційну технологію ARKit. Цей інструмент дозволяє програмістам створювати досвід користування додатковою реальністю за допомогою пристроїв Apple. Пізніше компанія Apple представила розширення до цієї технології для аналізу точок обличчя та обличчя користувачів за допомогою сучасних камер True Depth, встановлених на останніх пристроях. Основна ідея цієї технології – аналіз реального середовища, відстеження руху пристрою, камери та реальних об'єктів.

За допомогою сучасних камер та сенсорів пристрої можуть точно вимірювати відстань до реальних об'єктів. Потрібно використовувати функцію відстеження обличчя в цих рамках, щоб отримати інформацію про місцезнаходження голови користувача в просторі координат розширеної реальності пристрою. Приклад інформації щодо положення голови можна побачити на рисунку 2.1.

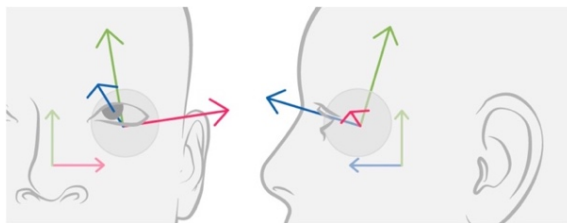


Рисунок 2.1 – Модель положення голови користувача в просторі координат доповненої реальності

Інформація про розташування обличчя надається у вигляді матриці перетворення з чотирма рядками та чотирма стовпцями. Ця матриця визначає відстань (перенос), обертання, масштаб обличчя користувача в реальному світі відповідно до простору координат доповненої реальності. Інформація про особливості обличчя (рот, ніс, брови тощо) надається у вигляді словника з ключами та значеннями. Кожен ключ – це ключ до певної частини обличчя. Кожне значення цього словника – це значення від 0 до 1, що визначає, ступінь зміни частини обличчя. Вся ця інформація у поєднанні, представлена у вигляді сітки 3D-обличчя (див. рис. 2.2).



Рисунок 2.2 – Представлення сітки 3D-обличчя

Маючи наявну інформацію про місцезнаходження та трансформацію обличчя користувача, можна розрахувати приблизне розташування очей. Наступний крок – пошук зірок. Цю проблему можна вирішити за допомогою інструментів Computer Vision. Apple пропонує використовувати фреймворк Vision для розв’язання таких завдань. Він застосовує алгоритми комп’ютерного зору для

виконання різноманітних завдань щодо вхідних зображень та відеозаписів. Для досягнення мети знайти зірок очей існує багато методів. Найбільш влучним для нашої мети алгоритмом є СНТ (Circle Hough Transform). Цей алгоритм забезпечує можливість виявлення кругових об'єктів у цифровому зображенні. Перетворення Хафа – це алгоритм для виявлення кіл на зображенні. Мета методики – знайти кола в недосконалих даних зображень.

Людські очі мають радіус 12 мм, тому цей алгоритм може легко передбачити розташування очей, маючи сітку обличчя та його положення. Спроекуємо сфери очей та зіниць на сцену доповненої реальності (рис. 2.3).

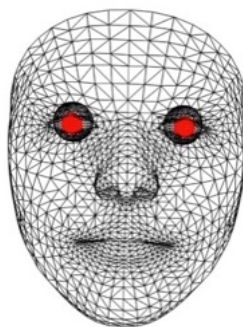


Рисунок 2.3 – Спроековані очі та зіниці на сітці 3D-обличчя

Маючи всю необхідну початкову інформацію, можна вирахувати вектори погляду, які починаються із центру сфери ока та проходять через центр сфери зіниці. Ці вектори знадобляться для вирахування погляду користувача, та побудови рівнянь, для знаходження точок перетину площини пристрою з цими векторами (рис. 2.4).

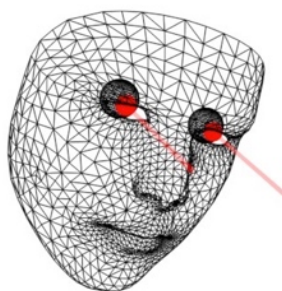


Рисунок 2.4 – Спроековані вектори погляду на сітці 3D-обличчя

Наступним кроком є підрахунок відстані між обличчям користувача та реальним пристроєм (смартфоном, планшетом чи іншим). Візуальна одометрія та інтерфейс використання True Depth камери, які вбудовані в фреймворк Apple ARKit, можуть надавати інформацію про цю відстань. Щоб створити відповідність між реальним та віртуальним просторами, ARKit використовує техніку, яку називають візуально-інерціальною одометрією. Цей процес поєднує інформацію з обладнання пристрою iOS, що визначає рух, з комп'ютерним зором аналізу сцени, видимої в камері пристрою. ARKit розпізнає помітні властивості в зображенні сцени, відстежує відмінності в положеннях цих властивостей у відеокадрах і порівнює цю інформацію з даними зондування руху. Результатом цього є високоточна модель положення та руху пристрою.

Камери Apple базуються на технології SLAM (одночасна локалізація та картографування). Камера iPhone робить знімок, який створює зображення, і ARKit визначає властивості(мітки) цього зображення. Потім камера робить інший кадр і знову визначає мітки. Надалі порівнюються перше зображення з другим і робиться ряд обчислень, щоб визначити зміну співвідношень між ознаками. Роблячи це знову і знову, коли камера продовжує створювати кадри – iPhone може «відстежувати» збережені мітки.

Коли ми бачимо речі очима, наш мозок робить якийсь миттєвий розрахунок різниці поглядів від правого до лівого ока та визначає глибину – відстань до різних предметів у вашому полі зору. Тепер є об'єкти, для яких ви знаєте приблизний розмір, і ця інформація використовується також для судження про приблизний масштаб речей навколо вас – це «відомі мітки», які відстежуються. Важливо, щоб мозок робив цей розрахунок за різницею двох зображень. По суті це тріангуляція – праве око бачить щось по-одному, ліве око бачить це дещо по-іншому, мозок знає відстань між очима, приблизний кут фокусу та різницю зображень, відстежує деякі відомі особливості, а тому може обчислити приблизні відстані та зрозуміти тривимірний простір. У камери пристрою немає двох різних зображень, за допомогою яких можна обчислити відстань. Однак через мить після першого зображення у нього є друге зображення. Завдяки даним з датчиків акселерометра

пристрій може також оцінити різницю – від першого зображення до другого – положення 3D-камери iPhone та її напрямку. Для кожного зображення пристрій знаходить якомога більше міток. Окрім проведення тріангуляції щодо кожної з міток зображень, алгоритм також проводить порівняння відмінностей у взаємозв'язку кожної мітки з іншими мітками зображення. Тож тепер пристрій має два різні погляди на щось, знає приблизні кути фокусування, знає відстань між положенням об'єктива, відстежує відомі мітки та їх відношення один до одного. Виходячи з цього, алгоритм може отримати дуже гарне наближення до того, як кожна мітка розміщена в просторі по відношенню до інших міток, по суті створюючи 3D-відображення простору.

Маючи інформацію отриману за допомогою використання методів доповненої реальності та машинного зору, можна створити алгоритм, який буде проектувати погляд людини на екран пристрою

2.4 Уточнена постановка задачі дослідження

В роботі досліджуються методи управління складними об'єктами шляхом реалізації контролю за поглядом людини. Проведений аналіз предметної галузі та дослідження аналогів дозволяє деталізувати постановку задачі та визначити вимоги до об'єкту дослідження.

Загальну задачу дослідження можна поділити на дві частини: аналіз зображення та машинне навчання.

Аналіз зображень включає в себе виявлення та ізоляцію функцій, які є достатніми для визначення напрямку зору. Машинне навчання – це створення та навчання нейромережі за допомогою цих функцій та визначення розташування курсора мишки на пристрої виводу.

В роботі розглядається алгоритм створення системи управління поглядом, яка використовує стандартне апаратне забезпечення. Апаратні засоби, які будуть

необхідні для реалізації цієї системи – вбудована або веб-камера. Кінцевою метою цього проекту є програмна реалізація моделі керування об'єктом з використанням управління поглядом, яка базується на використанні системи меню для зміни властивостей об'єкта.

Програмна система, що буде створена, повинна надавати користувачеві можливість керувати графічним інтерфейсом за допомогою елементів доповненої реальності. Реалізація програмного додатку буде використовувати веб-камеру для розпізнавання напряму зору користувача для управління курсором мишки та вибору властивостей об'єкту. Додаток повинен використовувати лише веб-камеру та не вимагати додаткових пристроїв, що робить систему більш зручною та доступною для користувачів.

Головною проблемою та завданням побудови цієї системи є аналіз очей користувача та його обличчя, що існує у реальному світі, та виведення напряму погляду на екран пристрою з подальшим перетворенням в команди управління об'єктом.

У ході розробки програмного забезпечення буде розроблено алгоритм перетворення даних з камери та сенсорів пристрою а також доповненої реальності у систему координат екрану пристрою. Цей алгоритм планується оптимізувати в рамках цього дослідження для використання на мобільних пристроях. Основою цього алгоритму є азимутальні та матричні перетворення [8], які нададуть змогу пристрою орієнтуватися у реальному просторі та аналізувати положення очей користувача.

Технології, які будуть використані при розробці програмного забезпечення:

- мова програмування Swift, C, C++ та Objective-C;
- iOS, як платформа під яку буде створюватися мобільний додаток;
- інструменти для роботи з доповненою реальністю ARKit та SceneKit;
- інструменти для розробки мобільного додатку UIKit та Foundation;
- інструмент розробки мобільних додатків iOS SDK;
- середовище розробки XCode 11.

Основною перевагою використання саме цього стеку технологій є той факт, що вони не потребують додаткового обладнання, як, наприклад HoloLens (потребує додатковий шолом) та Tango (потребує окремий пристрій). Виходячи з того, що ARKit є єдиним масовим продуктом та не потребує додаткового обладнання, було прийнято рішення використовувати його для подальшої розробки програмного забезпечення [20].

3 РОЗРОБКА МЕТОДІВ ТА АЛГОРИТМІВ УПРАВЛІННЯ

3.1 Аналіз алгоритмів доповненої реальності та способів їх використання

Щоб визначити положення погляду користувача, потрібно створити віртуальну площину, яка буде повторювати та відображати положення пристрою у сцені доповненої реальності. Фреймворк ARKit разом із вбудованим CoreMotion надає змогу отримати інформацію щодо положення пристрою. Положення пристрою визначається матрицею переміщення(вектору) відносно початку координат та значенням повороту пристрою по трьох осях координат.

Маючи дані о площині пристрою, позиції обличчя користувача та векторів поглядів, можна вирахувати перетини цих векторів із віртуальною площиною пристрою та проаналізувати їх.

Для початку створимо рівняння прямої, яка проходить через дві точки у просторі. Маємо формулу 3.1, на якій наведено рівняння прямої.

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} = \frac{z - z_1}{z_2 - z_1}, \quad (3.1)$$

де $A(x_1, y_1, z_1)$ та $B(x_2, y_2, z_2)$ – точки у просторі.

Тепер створимо рівняння площини. Віртуальна площина пристрою має матрицю трансформації з інформацією щодо куту нахилу, тому можна створити вектор нормалі, шляхом повертання базового вектору $\underline{n}(a, b, c) = (1, 1, 1)$ на кут трансформації площини. Щоб створити рівняння площини, потрібно мати точку на цій площині. Беручи до уваги те, що камера пристрою має нульову позицію, а дані щодо положення пристрою отримані відносно його центру, можна взяти цю точку, як точку на площині пристрою. Отримана формула 3.2:

$$a(x - x_1) + b(y - y_1) + c(z - z_1) = 0, \quad (3.2)$$

де x_1, y_1, z_1 – координати точки, що належить площині.

Адже точка, що належить площині нульова, можна спростити це рівняння. Отримана формула 3.3:

$$ax + by + cz = 0, \quad (3.3)$$

де a, b, c – компоненти вектору нормалі площини.

Щоб знайти перетин, потрібно щоб точки задовольняли системі рівнянь. Маємо формулу 3.4:

$$\begin{cases} \frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} = \frac{z - z_1}{z_2 - z_1} \\ ax + by + cz = 0 \end{cases} \quad (3.4)$$

Отримана система рівнянь є системою з трьох рівнянь з трьома невідомими, тобто вона має рішення. Рішення може бути знайдено за допомогою різних методів (таких як метод Гаусса або Крамера). Рішенням цієї системи (якщо воно існує) є точка $M(x_3, y_3, z_3)$, яка зображує точку перетину між прямою та віртуальною площиною пристрою.

Завдяки цьому методу можна знайти дві точки перетину між площиною пристрою та векторами погляду правого та лівого ока. Знайшовши середину отриманого відрізка отримаємо точку погляду користувача, спроектовану на площину пристрою.

Останнім кроком є перетворення середньої точки перетину, представлені на сцені доповненої реальності, в локальні координати екрану пристрою. Для цього можна застосувати нативний підхід, використовуючи хіт-тестування.

Після всіх обчислень, отримані точки аналізуються та обробляються, щоб відобразити маркер погляду користувача на екрані пристрою.

Цей метод може надати розробникам та користувачам різні можливості. Одним із прикладів є створення таких інструментів, які дозволять керувати інтерфейсом програми, використовуючи лише погляд користувача. Такий підхід дозволить інвалідам користуватися своїми пристроями без проблем. Можна

створити новий революційний досвід користувача в іграх. Наприклад, люди можуть грати в локальні або багатокористувацькі ігри, використовуючи лише очі. Ще один варіант використання методу – відстеження погляду користувачів для збору та аналізу даних для створення теплових карт додатків або для поліпшення користувацького інтерфейсу та користувацького досвіду.

3.2 Вибір алгоритму машинного навчання

Під машинним навчанням в термінах постановки задачі дослідження будемо розуміти процес створення та навчання системи, за допомогою якої в моделі буде реалізовано функцію визначення розташування вказівника мишки. Розглянемо існуючі алгоритми.

Дерево прийняття рішень. Перевагами цього методу вважається той факт, що він стабільно функціонує у разі втрат даних та не вимагає від даних нормалізації. Недоліком є те, що він чутливий до шуму і змін в даних – незначна зміна даних може суттєво змінити дерево. Існує можливість повторного навчання, що вирішується за допомогою обрізки і деревовинних композицій. Ще одним недоліком є той факт, що цей алгоритм відноситься до «жадібних», тому що в кожній вершині вибирається предикат, який найкраще локалізовано.

Метод найближчих сусідів. Перевагою цього алгоритму є простота, тому що основним параметром для його оптимізації є лише кількість сусідів. Він є найбільш підходящим, коли інформативний показник відстані легше знайти між двома символами, ніж визначити відстань для послідовності символів. Також до недоліків слід віднести той факт, що він ресурсномістким, оскільки основні розрахунки проводяться саме під час оцінки, а не шляхом навчання [17].

Лінійна регресія. Лінійна регресія має багато переваг, одна з яких швидке навчання через те, що кількість параметрів досить мала – кількість параметрів дорівнює кількості ознак. Цей алгоритм можна використовувати для великих даних

через оптимізацію стохастичного градієнтного спуску, який приймає об'єкти по черзі. Ще одна перевага полягає в тому, що цей метод легко інтерпретувати – чим більша вага, тим важливішим є ця ознака. Разом з тим алгоритм має певні обмеження: лінійна регресія не може досягти високої якості, наприклад, у випадкових лісах, які вимагають обов'язкову попередню обробку даних. Збільшення масштабів, шум та викиди повинні бути усунені, що, у свою чергу, потребує додаткових ресурсів та знижує швидкодію методу.

Метод опорних векторів. У цьому методі функція завжди має єдиний мінімум, який є глобальним, тобто немає проблемі із застряганням у локальних мінімумах. Корисним наслідком цього алгоритму є виділення допоміжних об'єктів. Зазвичай він добре працює на невеликих і чистих даних. Недоліком методу є те, що опорними об'єктами можуть бути обрані викиди або шум, тому дані повинні бути очищені перед підготовкою класифікатора. Крім того, він не дуже добре працює з великими даними, тому що при великій кількості даних навчання може зайняти багато часу [18].

Випадковий ліс. Цей алгоритм відновлює складні залежності і не допускає перетренувань, оскільки додавання нових дерев до композиції тільки покращує якість. Це не вимагає попередньої обробки даних, тому що це логічний класифікатор. Одним з недоліків є те, що його тренування займає довгий час і буде зберігати далекоглядність протягом тривалого часу через глибину дерев. Крім того, випадковий ліс не дуже добре працює з розрідженими даними у зв'язку з тим, що дерева не пристосовані для цього [19-25].

Підвищення градієнта. Це машинний метод навчання застосовується для дослідження проблем регресу. Цей алгоритм відновлює складні залежності і не вимагає попередньої обробки даних. Одним з недоліків є те, що під час тренування моделі можуть бути перепрограмовані; також він не достатньо добре працює з розрідженими даними [26-27].

Нейронні мережі. Нейронна мережа є комплексом лінійних класифікаторів. Перевагами її застосування є те, що градієнтний спуск використовується в нейронних мережах для оптимізації моделі, яка може ефективно працювати з

великими даними. Також нейронні мережі можуть відновлювати складні залежності, оскільки вони є сховищем алгоритмів та можуть мати багато нейронів. Особливо ефективними є так звані глибокі нейронні мережі, які останнім часом майже витіснили всі інші методи машинного навчання під час розв'язання складних завдань, таких як розпізнавання зображень, тексту тощо. Що стосується недоліків цього підходу, то одним з головних недоліків є те, що під час навчання модель зростає, збільшується складність її оптимізації і збільшується час на отримання результатів розрахунків. Також цей метод має схильність до перепрограмування моделі, для запобігання якого застосовуються різні методи регуляризації [17, 27].

На підставі проведеного аналізу алгоритмів машинного навчання, в якості алгоритму пошуку напрямку зору користувача було обрано штучну нейронну мережу, оскільки проблема має нелінійний характер та досить складна для наближення [27].

3.3 Штучні нейронні мережі

Штучна нейронна мережа є концептуальною моделлю біологічної нейронної мережі і складається з пов'язаних між собою різним чином шарів штучних нейронів, які організовують загальну активну структуру і функціонально впливають на роботу один одного. У більшості архітектур штучних нейромереж активність нейрона визначається перетворенням зовнішнього сумарного впливу інших нейронів на даний нейрон. Застосування нейронних мереж для дослідження методів управління складними об'єктами за допомогою напрямку зору людини обумовлено тим фактом, що на початку роботи система не готова до використання – спочатку її необхідно навчити. Навчання здійснюється шляхом оновлення параметрів, що покращують поведінку системи. У процесі навчання відбувається моніторинг навчального процесу з використанням визначених прикладів введення-виведення [27].

У більшості архітектур штучних нейронних мереж функції активації нейронів фіксовані, а ваги синапсів є параметрами мережі. Деякі входи нейронів є зовнішніми входами сукупної мережі, а деякі виходи нейронів – виходами сукупної мережі. Завдання нейромережі полягає в перетворенні вхідного вектора у вихідний вектор, що здійснюється вагою і топологією мережі.

Штучний нейрон характеризується своїм поточним станом. Тут можна провести аналогію з нервовими клітинами головного мозку, які можуть бути пошкоджені або неправильно працювати. Він володіє групою синапсів – односпрямованих вхідних зв'язків, з'єднаних з виходами інших нейронів, присутня така частина, як аксон – вихідний зв'язок штучного нейрона, з якого сигнал (збудження або гальмування) надходить на синапси наступних нейронів. Загальний вигляд штучного нейрона наведено на рисунку 3.2. Штучний нейрон спочатку імітує властивості, що дані біологічному нейрону. Тут безліч вхідних сигналів, позначених $x_1, x_2, \dots, x_n$, надходить на штучний нейрон. Ці вхідні сигнали, в сукупності позначаються вектором X , приходять до синапсів біологічного нейрона. Кожен синапс характеризується величиною синапсичного зв'язку або його вагою w_i [3].

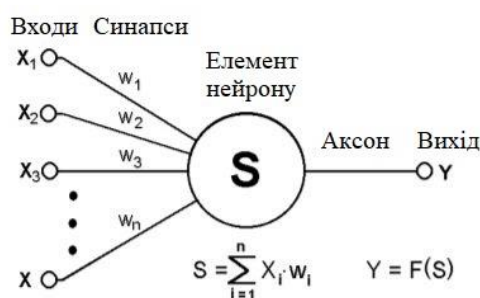


Рис. 3.1 – Модель штучного нейрона

Для реалізації нелінійності при активації нейрона, його активність, крім різних видів суматорів і систем ваг на входах, визначається функцією одного аргументу - функцією активації. Нейрон в цілому реалізує скалярну функцію векторного аргументу, а вихідний сигнал нейрона визначається видом функції активації і може бути дійсним або цілим. Функція активації застосовується до

зваженої суми постсинаптичних сигналів на вході нейрона. Таким чином, активність нейрона повністю визначається його параметрами - вагами і його функцією активації.

Відмінності обчислювальних процесів в нейронних мережах часто обумовлені способом взаємозв'язків нейронів. За сукупністю критеріїв на сьогоднішній день багат шарові архітектури можна розділити на статичні і динамічні. Кожен з класів архітектур нейронних мереж може включати безліч підкласів, реалізуючи різні підходи.

До статичних архітектур відносять мережі прямого поширення, в яких реалізована однонаправлений зв'язок між шарами, відсутні динамічні елементи і зворотній зв'язок, а вихід навченої нейромережі однозначно визначається входом і не залежить від попередніх станів мережі.

Статичні штучні нейронні мережі прямого поширення:

- перцептрон;
- нейронна мережа Кохонена;
- когнітрон та неокогнітрон;
- сучасна згорткова нейронна мережа.

На противагу статичним архітектурам, існують динамічні архітектури штучних нейромереж, що реалізують рекурентну структуру з використанням зворотніх зв'язків, завдяки чому стан мережі в кожний момент часу залежить від попереднього стану. Рекурентні нейромережі як правило базуються на багат шаровому перцептроні.

Перцептрон. Моделлю нейрона в штучних нейронних мережах є перцептрон. Перцептрон (англ. perceptron від лат. perceptio — сприйняття; нім. perceptron) — математична або комп'ютерна модель сприйняття інформації мозком (кібернетична модель мозку), запропонована Френком Розенблатом в 1957 році. Незважаючи на свою простоту, перцептрон здатен навчатися і розв'язувати досить складні завдання — наприклад, лінійне розділення довільних нелінійних множин, так зване забезпечення лінійної сепарабельності.

Персептрон складається з трьох типів елементів, а саме: сигнали, що надходять від давачів, передаються до асоціативних елементів, а відтак до реагуючих. Таким чином, персептрони дозволяють створити набір «асоціацій» між вхідними стимулами та необхідною реакцією на виході. В біологічному плані це відповідає перетворенню, наприклад, зорової інформації у фізіологічну відповідь рухових нейронів. Відповідно до сучасної термінології, персептрони можна класифікувати як штучні нейронні мережі:

- з одним прихованим шаром;
- з пороговою передавальною функцією;
- з прямим розповсюдженням сигналу.

Процес навчання штучний нейронних мереж розглядається як налаштування архітектури і ваг зв'язків між нейронами (параметрів) для ефективного виконання поставлених перед мережею завдань. Існує два великих класи навчання: клас детерменірованих методів і клас стохастичних методів.

До класу детерменірованих методи входять методи, в основі яких лежить ітеративна корекція параметрів мережі, в ході поточної ітерації яка ґрунтується на поточних параметрах. Основним детерменірованим методом і найпоширенішим методом навчання мереж сьогодні є метод зворотнього поширення помилки.

До класу стохастичних методів входять методи, що змінюють параметри мережі випадковим чином і зберігають тільки ті зміни параметрів, які призвели до поліпшення результатів роботи. Стохастичні алгоритми навчання реалізуються за допомогою порівняння помилок.

Класичний метод навчання персептрону — це метод корекції помилки [26]. Він являє собою такий вид навчання з учителем, при якому вага зв'язку не змінюється до тих пір, поки поточна реакція персептрона залишається правильною. При появі неправильної реакції вага змінюється на одиницю, а знак (+/-) визначається протилежним від знаку помилки.

Метод зворотного поширення помилки. Це ітеративний градієнтний алгоритм, який використовується з метою мінімізації помилки роботи багат шарового персептрону та отримання бажаного виходу. Основна ідея цього

методу полягає в поширенні сигналів помилки від виходів мережі до її входів, в напрямку, зворотному прямому поширенню сигналів у звичайному режимі роботи. Барц і Охонін запропонували відразу загальний метод («принцип подвійності»), який можна застосувати до ширшого класу систем, включаючи системи з запізненням, розподілені системи, тощо. Для можливості застосування методу зворотного поширення помилки функція активації нейронів повинна бути диференційованою. [11].

Навчання з учителем. Під час навчання мережі з учителем кожному прикладу з навчальної вибірки відповідає вектор, що характеризує однозначну правильну відповідь, що подається відразу на вихід мережі в обхід всієї її архітектури. Після отримання власного результату мережі, алгоритм порівнює результуючий вектор з правильною відповіддю, на основі чого відбувається корекція подальших помилок. Правило корекції помилки, на якому заснований метод зворотного поширення помилки, є класичним прикладом навчання з учителем.

3.4 Огляд бібліотек для глибинного навчання

На сьогоднішній день у реалізації нейронних мереж важливу роль відіграють, як правило, спеціалізовані пакети, що надають вже повністю реалізовані базові структури та алгоритми для зручної розробки. Найбільш широко використовуються такі популярні бібліотеки, як Theano, Tensorflow та Torch та ін. Дані пакети реалізують програмний інтерфейс для реалізації низькорівневих розрахунків, тому прийняття рішень, в першу чергу, повинно базуватися на суперечностях про продуктивність та зручність розробки.

Варто відзначити, що Torch реалізує інтерфейс для мови Lua, не дуже популярної серед програмістів, що істотно знижує зручність і швидкість розробки.

Згідно дослідженням Theano and Torch показують приблизно однакову продуктивність на одну й тому ж наборі даних.

Окрім описаної вище незручності використання бібліотеки Torch, існує ще одна, також пов'язана зі зручністю розробки. Інші два згаданих вище інструменти здатні працювати з бібліотекою більш високого рівня Keras [14], що надає загальні, для Tensorflow [15] і Theano, заздалегідь визначені алгоритми та структури. За результатами порівняння та аналізу документації бібліотек було прийнято рішення використовувати для програмної реалізації згорткової нейронної мережі зв'язок бібліотек Tensorflow і Keras.

Бібліотека Keras [14] - програмний продукт, написаний мовою python, що надає високорівневе API для роботи зі штучними нейронними мережами. Даний продукт створювався з метою запобігання труднощів в розробці нейронних мереж, пов'язаних із синтаксичними особливостями конкретних пакетів. В січні 2016 року даний інструмент було придбано компанією Google і адаптовано як основне високорівневе надлаштування над бібліотекою Tensorflow.

Бібліотека Theano була написана в першу чергу як розширення мови Python, яка дозволяє ефективно розрахувати математичні вирази, що містять багатомірні масиви. В бібліотеці реалізується базовий набір інструментів для побудови нейронних мереж. Процес створення моделі та визначення її параметрів вимагає написання об'ємного коду, що включає реалізацію класу моделі, самостійного визначення її параметрів, реалізації методів, що визначають функцію помилок, правило обрахування градієнтів, способи зміни ваги нейронів.

Бібліотека Caffe реалізована на мові програмування C++. Топологія нейромереж, вихідні дані та спосіб навчання задаються за допомогою конфігураційних протоколів (застосовується технологія і протокол передачі даних) у прототиповому форматі. Побудова структури мережі виконується з простотою та зручністю. Бібліотека Caffe підтримується досить великою спільнотою розробників та користувачів і на сьогоднішній день є самою розповсюдженою бібліотекою глибинних навчальних програм широкого кола застосування.

Огляд методів вирішення задач контролю за напрямком зору людини обґрунтував переваги нейромережевого методу. Принцип роботи нейронів штучної мережі дозволяє виявити його активність, що визначається його параметрами – вагами і його функцією активації. Ефективність навчання управління складними об'єктами з використанням управління за напрямком зору на початковому етапі досліджень можна досягти саме методом навчання з учителем. Перейдемо до програмної реалізації моделі системи.

4 РЕАЛІЗАЦІЯ ПРОГРАМНОЇ МОДЕЛІ УПРАВЛІННЯ ОБ'ЄКТОМ

4.1 Опис алгоритму програмної реалізації системи машинного навчання

Програмна модель для управління об'єктами за допомогою напрямку зору людини буде складатися з двох компонентів:

А) ViewpointData – дані, що містять інформацію про розпізнавання положення обличчя і зберігати ці дані. Він також буде мати можливість тренувати нейронні мережі за цими наборами даними.

Б) Viewpoint – система розпізнавання напрямку зору, яка використовує навчені нейронні мережі.

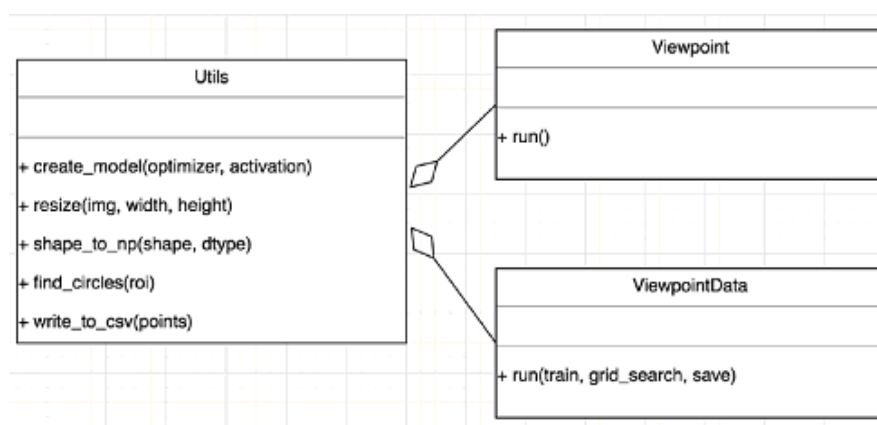


Рис. 4.1 – Діаграма класів програми – тренера

Задача визначення напрямку погляду. Задачу можна поділити на дві частини:

- аналіз зображень для ідентифікації особливостей обличчя, яких буде достатньо, щоб розпізнати напрямок погляду, використовуючи методи аналізу зображень.
- машинне навчання – це розробка алгоритму машинного навчання з використанням ознак, виявлених під час аналізу зображення.

Загальний опис алгоритму. Перший крок заснований на отриманні одного кадру з камери. Після того, як кадр був отриманий, починається розпізнавання обличчя. З цією метою виконується знаходження частин обличчя – окремих регіонів, де на зображенні розташовані очі. Один із способів їх знаходження –

винайти коло, але для цього необхідно спочатку очистити зображення, для чого доцільно використовувати порогову технологію зменшення шуму на зображенні. Виявлення контурів зображення також підвищує точність пошуку потрібного об'єкту. Після пошуку у регіонах, де розташовані очі, можуть бути застосовані фільтри. Після знаходження координат зіниць та інших частин обличчя, починається навчання нейронної мережі з використанням цих координат.

Надалі буде детально розглянуто програмний додаток який використовує натреновану модель яка розпізнає обличчя та його основні характеристики.

4.2 Опис діаграми класів програмного додатку

Основним елементом проектування системи є діаграма класів. Ця діаграма дозволяє спроектувати конкретну систему, та отримати уявлення про внутрішній стан системи в цілому.

Діаграма класів – діаграма, що демонструє класи системи, їх атрибути, методи і взаємозв'язок між ними. Діаграма створюється за допомогою мови моделювання UML.

Діаграма класів UML дозволяє позначати відносини між класами і їх екземплярами. Вони потрібні, наприклад, для моделювання прикладної області.

Асоціації мають навігацію: двосторонню або односпрямовану, що вказує на напрям зв'язку. Тобто у кожного виду асоціації ще є два підвиди.

Існує декілька типів зв'язку у діаграмі класів: асоціація, агрегація, композиція, узагальнення, реалізація, залежність.

Клас – це основний будівельний блок. Це поняття є і в об'єктно орієнтованих мовах програмування, тобто між класами UML і програмними класами є відповідність, що є основою для автоматичної генерації програмних кодів або для виконання реінжинірингу. Кожен клас має назву, атрибути і операції. Клас на ді показується у вигляді прямокутника, розділеного на 3 області. У верхній міститься

назва класу, в середній – опис атрибутів (властивостей), в нижній – назви операцій - послуг, що надаються об'єктами цього класу.

Мова моделювання UML має набір відносин для побудови моделі класів, але навіть така розвинена ООП мова, як Swfit має тільки дві явні конструкції для відображення зв'язків: *extends* (розширення) і *interface/implements* (реалізація).

При проектуванні програмного забезпечення, а саме діаграми класів, було вирішено використовувати гнучку архітектуру мобільної розробки VIPER. Суть цього патерну полягає у слідуванні принципу чистої архітектури та принципів SOLID.

Clean Architecture ділить логічну структуру програми на різні рівні обов'язків. Це спрощує ізолювання залежності (наприклад, ваша база даних) і тестування взаємодії на кордонах між рівнями.

VIPER – модульна архітектура, тобто кожна окрема ламка системи має свій модуль, кожна частина якого виконує єдину окрему роль. У цій архітектурі є п'ять основних компонентів:

- *view* – відображає що повідомив *Presenter*, та повідомляє його про дії користувача;
- *interactor* – об'єкт, який відповідає за бізнес логіку модуля, та діє за сценарієм;
- *presenter* – об'єкт, який відповідає за підготовку даних для відображення у *View*, а також за обробку даних, які було введено користувачем програмного забезпечення, повідомляє *Router*, коли необхідно виконати перехід у наступний модуль, питає у *Interactor* дані, необхідні для відображення;
- *entity* – об'єкт даних системи, це може бути гео-дані, чи об'єкти які зберігаються у базі даних тощо;
- *router* – об'єкт, який відповідає за навігацію у системі, слухає *Presenter*.

Всі ці компоненти дозволяють дотримуватися принципу єдиної відповідальності, який описано у SOLID підході.

SOLID – мнемонічний акронім, введений Майклом Фезерсом для перших п'яти принципів, названих Робертом Мартіном на початку 2000-х, які означали п'ять основних принципів об'єктно-орієнтованого програмування і проектування, які вказують, як писати код, щоб його легко було супроводжувати та доповнювати.

SOLID складається із таких принципів:

- The Single Responsibility Principle – принцип єдиної відповідальності, полягає у тому що клас може змінюватися тільки за одною причиною. Конкретний клас повинен вирішувати конкретну задачу, яка повинна бути унікальною і єдиною;
- The Open Closed Principle – усі програмні сутності повинні бути відкриті для розширення та закриті для модифікації. Всі класи, функції та інші компоненти системи повинні проектуватися так, щоб для зміни їх поведінки, нам не потрібно було змінювати їх вихідний код;
- The Liskov Substitution Principle – клас який наслідує інших повинен розширювати функціонал а не змінювати його. При використанні спадкоємця класу результат виконання коду повинен бути передбачуваний і не змінювати властивостей методу;
- The Interface Segregation Principle – багато інтерфейсів, спеціально назначених для клієнтів краще ніж один багатофункціональний інтерфейс;
- The Dependency Inversion Principle – залежності всередині системи будуються на основі абстракцій. Модулі верхнього рівня не залежать від модулів нижнього рівня. Абстракції не повинні залежати від деталей. Деталі повинні залежати від абстракцій.

Ці принципи, коли застосовуються разом, призначені для підвищення ймовірності того, що програміст створить систему, яку буде легко підтримувати і розширювати протягом довгого часу. Принципи SOLID – це керівництва, які можуть застосовуватися під час роботи над програмним забезпеченням для видалення поганого коду, наказуючи програмісту виконувати рефакторинг вихідного коду, поки той не стане розбірливо написаним і розширюваним. Це частина загальної стратегії гнучкої і адаптивної розробки. У подальшому система,

яка написана за цим принципом, дуже легко тестується та перевіряється на доцільність та відповідність вимогам до програмного забезпечення.

Саме цей підхід є запорукою того, що система буде адекватною, та буде відповідати усім вимогам до цієї системи.

На основі вимог до реалізації програмного продукту було створено діаграму класів системи, яку можна побачити на рисунку 4.2.

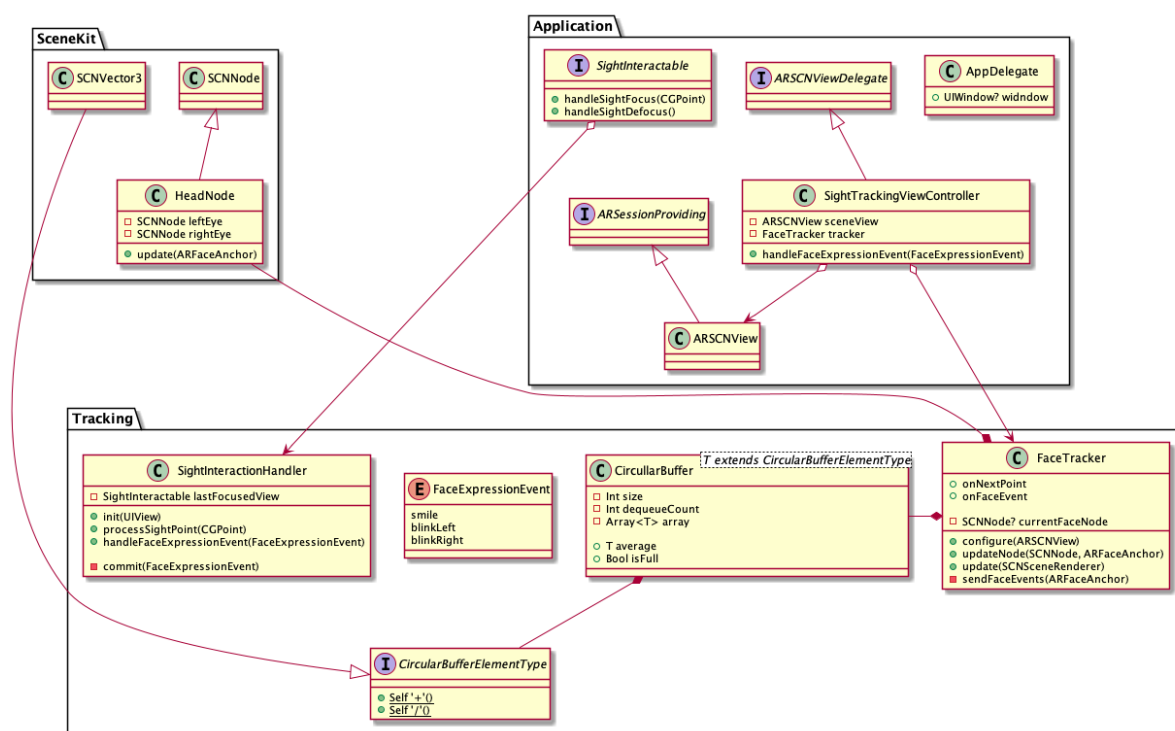


Рис. 4.2 – Діаграма класів програмного додатку

Побудована діаграма класів описує базову ієрархію та взаємодію між класами у програмі. Як можна зазначити вище, у програмі є основний клас системи – `AppDelegate`, який реалізує інтерфейс `UIApplicationMain`, що означає те що операційна система буде сприймати програмне забезпечення як мобільний додаток. У цього об'єкта є зв'язок з головним вікном додатку `UIWindow`, та головним контролером системи `UIViewController`. Це основні об'єкти для старту мобільного додатку та відображення контенту для користувача.

Система поділена на три основні компоненти:

- SceneKit – основний компонент для роботи із сценою доповненої реальності;
- Tracking – компонент для роботи із розпізнаванням обличчя та наданням інформації щодо поточного положення обличчя та його основних параметрів;
- Application – компонент роботи із операційною системою та графічним інтерфейсом користувача.

Ці модулі пов'язані та спілкуються між собою завдяки інтерфейсам, що приховують реалізацію та надають змогу змінювати залежності системи, що робить її легкою для поширення та тестування. Спроектowana система повністю задовольняє усім принципам SOLID.

Надалі буде розглянуто алгоритм роботи програми та діаграму діяльності системи.

4.3 Опис діаграми діяльності системи

Діаграма діяльності системи потрібна для того щоб визначити стани скінченного автомату системи. Цей тип відображення поведінки системи надає змогу уявити її в форматі графу, який описує базову поведінку системи та її зміни. У системі існує два основні потоки виконання. Перший відповідає за розпізнання обличчя та надає нормалізовану інформацію щодо поточної точки фокусу користувача. Другий потік виконання отримує інформацію від компоненту, який відповідає за розпізнання обличчя, та відображає курсор і графічні елементи системи, які реагують на цей курсор. Ці основні потоки діяльності системи, а також опис того як працює додаток можна побачити на рисунку 4.3.

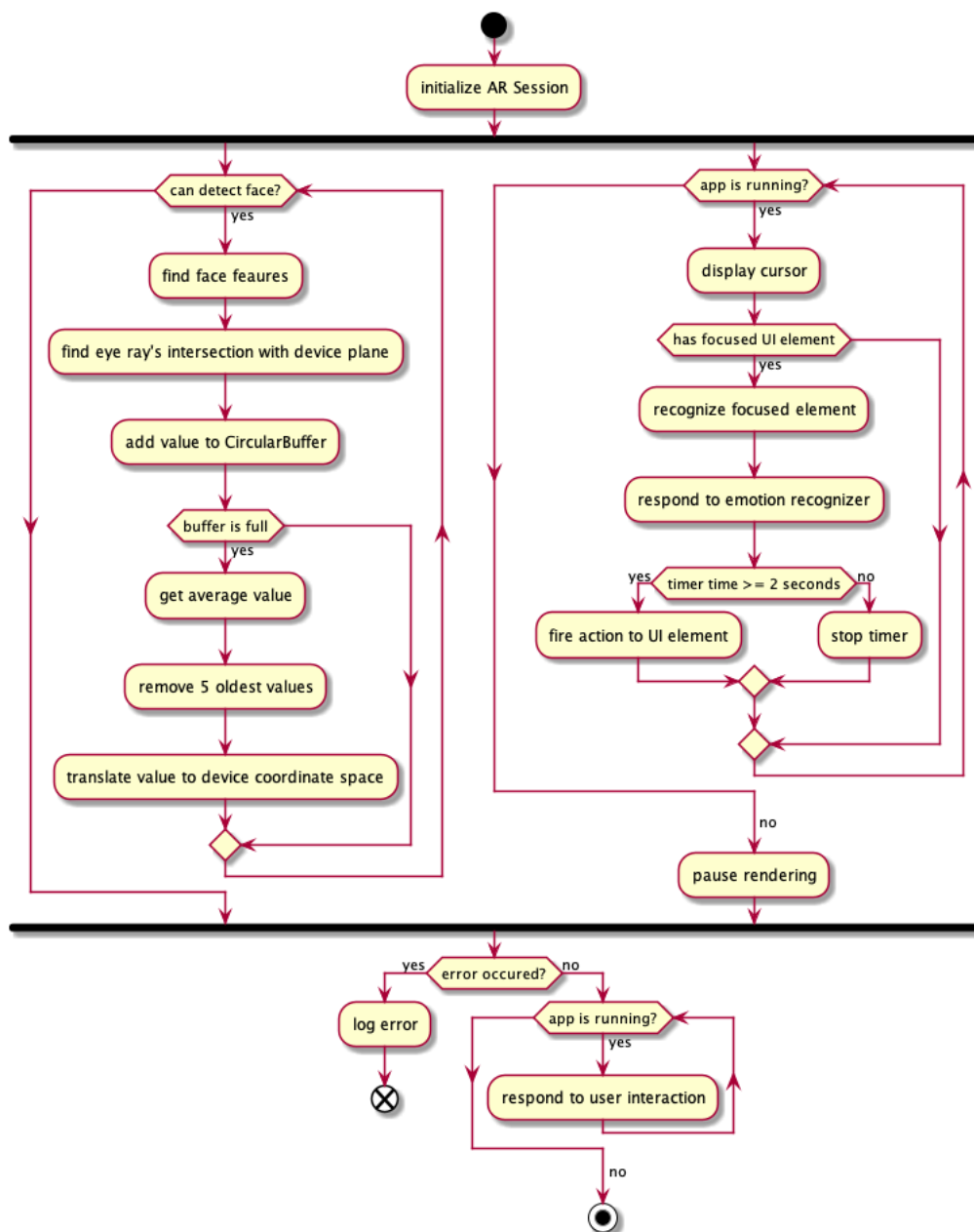


Рис. 4.3 – Діаграма діяльності системи

Як можна зазначити із діаграми, система створює сцену поширеної реальності на початку роботи. Після цього починає працювати розпізнавання обличчя із даних, отриманих із камери пристрою. Коли обличчя та усі його характеристики знайдено система буде два промені які виходять із координат розпізнаних очей, яким відповідають об'єкти у доповненій реальності. Далі система знаходить точки перетину цих проміней із штучною площиною пристрою та зберігає отримані данні для подальшої нормалізації. Коли данні нормалізовані та зведені до координатної системи екрану пристрою система сповіщує інші

компоненти про ці данні. Також система починає аналіз поточного відео потоку, який отримано із камери пристрою. У процесі цього аналізу система виконує декілька складних операцій. По перше, вона аналізує середовище на наявність горизонтальних та вертикальних поверхонь, це дозволяє дуже точно визначити переміщення пристрою у реальному просторі за допомогою методу візуальної одометрії. По друге, система аналізує попередні дані, та прораховує різницю між минулими даними та поточними. Це надає змогу відстежити куди саме я на скільки пристрій змінив своє положення у процесі виконання програми. Після чого система вираховує координати пристрою у просторі сцени доповненої реальності, а саме три координати евклідового простору, визначає висоту пристрою у реальному світі відносно об'єкту який відповідає сітці обличчя та запам'ятовує отримані дані.

Паралельно з цим працює підсистема, яка відповідає за отримку нормалізованих даних погляду людини. Цей алгоритм відображає курсор на екрані пристрою та оновлює графічні елементи, які взаємодіють із цим курсором. В залежності від типу об'єкту система пропонує ряд дій які можна виконати із цим об'єктом.

На кожному циклі роботи цих двох підсистем, програма перевіряє наявність необроблених помилок та припиняє роботу системи, якщо такі присутні. Якщо помилок не знайдено – система починає роботу алгоритму спочатку.

4.4 Опис діаграми послідовності системи

Діаграма послідовності системи потрібна для візуального уявлення поведінки та зміни станів системи за часом. Цей тип діаграми дозволяє швидко охопити поведінки усієї системи у часі.

Для того щоб описати взаємодії об'єктів впорядкованих за часом, було побудовано діаграму послідовності, яку можна побачити на рисунку 4.4.

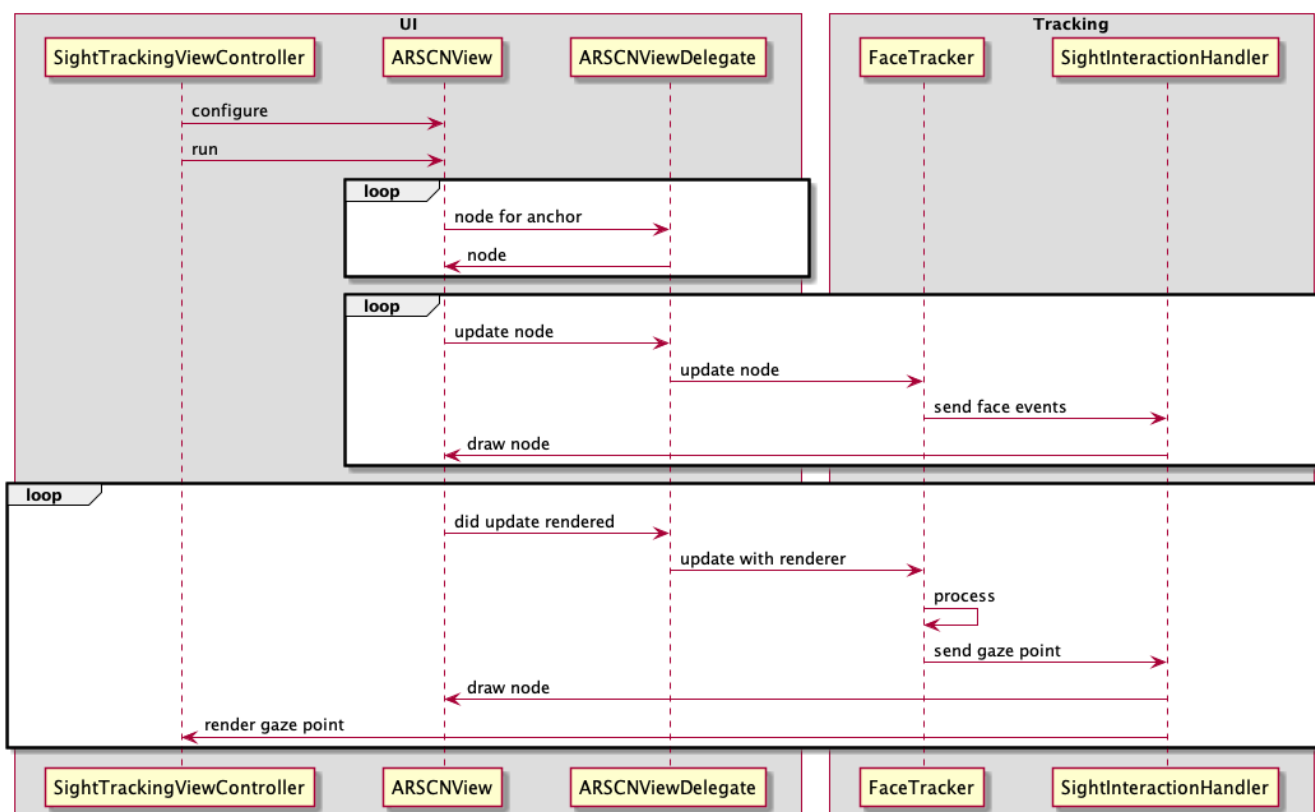


Рис. 4.4 – Діаграма діяльності системи

Як можна зазначити із рисунку, на діаграмі зображено схематичну поведінку системи, її об'єктів та повідомлень які ці об'єкти відправляють одне одному. На початку роботи системи об'єкт SightTrackingViewController, який відповідає за відображення графічного інтерфейсу, конфігурує та запускає ARSCNView, яка відповідає за відображення сцени доповненої реальності на екрані пристрою.

Після запуску ARSCNView починається нескінченний пошук вузлів сцени доповненої реальності які належать до відповідних якорей цієї сцени. Ці якорі характеризують особливі точки у просторі, які можуть бути розпізнані у різних кадрах за допомогою методів візуальної одометрії. Також ARSCNView починає рендерінг об'єктів у сцені доповненої реальності, а разом із FaceTracker, який відповідає за розпізнавання обличчя та надання інформації щодо його положення відображає сітку віртуального обличчя користувача, об'єктів що відповідають очам, зіницям, проміням які виходять із очей та віртуальній площині пристрою.

Паралельно з цим FaceTracker аналізує кожен кадр отриманий із камери смартфона та намагається знайти і розпізнати обличчя за допомогою обученої

перед цим нейромережі. Якщо обличчя знайдене, дані щодо його положення зберігаються та нормалізуються. Після нормалізації даних, FaceTracker відправляє повідомлення до SightTrackingViewController для того щоб оновити графічний інтерфейс. На кожному кроці цього алгоритму FaceTracker спілкується з SightInteractionHandler, який оброблює данні та робить їх зручними для використання у SightTrackingViewController, а також надає інформацію щодо ймовірності розпізнаних емоцій на обличчі та змін характеристик обличчя.

4.5 Результати експериментів

У ході виконання роботи було проведено ряд експериментів, щоб перевірити працездатність та доцільність системи.

Першим експериментом було перевірено працездатність навченої нейронної мережі, де вхідні данні являють собою кадри об'єктів, які нейронна мережа ніколи не бачила, але навчалася на схожих з цими об'єктами даних. Результати експерименту наведено у таблиці 4.1.

Таблиця 4.1 – Результати експерименту навчання нейронної мережі

Об'єкт	Правильне розпізнавання	Неправильне розпізнавання	Відсутність розпізнавання
Rectangle 1	89%	7%	4%
Rectangle 2	92%	6%	2%
Square	69%	25%	6%
Ellipse	95%	4%	1%

Як можна зазначити із таблиці, системі надавалися кадри різних геометричних фігур, таких як прямокутники під різними кутами, квадрат та еліпс. Усі ці фігури надавалися нейромережі із різним ступенем шуму, освітлення,

кольору тощо. Виходячи із отриманих даних, можна зробити висновок що система дає найбільшу похибку при розпізнанні квадратних фігур, адже ця фігура дуже схожа із прямокутником. Цю похибку можна зменшити шляхом збільшення вибірки даних, або якщо ми дозволимо системі сприймати квадрати, як прямокутники.

Також до цього експерименту було надано перевірку роботи нейромережі із елементами графічного інтерфейсу, який зображено на рисунку 4.5.

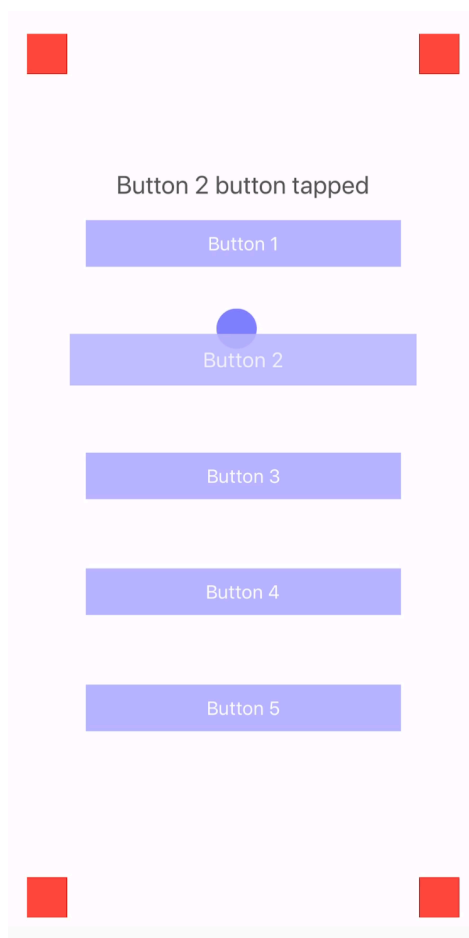


Рисунок 4.5 Розпізнавання елементу графічного інтерфейсу нейромережою

На рисунку наведено приклад роботи нейромережі, яка відстежує зону на яку сфокусовано погляд людини, та при розпізнанні об'єкту класу Rectangle відбувається виділення цього елементу. Якщо елемент був у фокусі людини більше ніж 3 секунди – відбувається натиск цього елементу. При роботі системи на реальному пристрої підтримується постійна частота кадрів – 60 кадрів у секунду.

На основі проведених експериментів, які показали що розроблена система працює із незначною похибкою, що присутня нейромережам та алгоритмам машинного зору із використанням методів візуальної одометрії, можна зробити висновок щодо придатності розробленого методу до використання. Нейромережа швидко та точно розпізнає нові об'єкти при різних зовнішніх умовах. Система у поєднанні з алгоритмами машинного навчання, надає змогу користувачам керувати поглядом складними об'єктами, що і є метою цієї роботи.

ВИСНОВКИ

В результаті проведеної роботи було проведено дослідження методів управління складними об'єктами шляхом реалізації контролю за поглядом людини.

Було приведено вирішення завдання відстеження погляду користувача, а також емоцій, завдяки яким можна керувати складним графічним інтерфейсом. Порівняно із першими комерційними досвідами з поширеною реальністю у якості розважальних і ігрових продуктів цей метод теоретично може полягти в основу більш серйозних продуктів. Зараз інші галузі також зацікавлені в можливостях доповненої реальності, наприклад, в обміні знаннями, освіті, управління інформаційними потоками та організації далеких зустрічей.

Під час виконання роботи вдалося поєднати графічний інтерфейс користувача із штучною доповненою реальністю, дані для якої в свою чергу беруться із датчиків пристрою (акселерометру, гіроскопу, компасу та камер). Таке поєднання джерел інформації дало змогу отримати найкращі результати та мінімізувати похибки.

У дослідженому методі було реалізовано ряд загально важливих моментів, серед яких швидкий алгоритм аналізу погляду користувача, застосування нової технології доповненої реальності, машинного зору, синхронізація та калібрування обох технологій. Це надало можливість досягти найбільш точного результату визначення точки фокусу погляду користувача та його емоцій.

Отже, у процесі виконання роботи було спроектовано програмне забезпечення, яке базувалося на вимогах користувача та загальних вимогах до системи. Після проектування програмного продукту, створення інфологічної моделі та створення схематичного графічного інтерфейсу, було розроблено програмне забезпечення, яке повністю задовольняє вимогам. Цей програмний продукт легко підтримувати, тестувати та доповнювати, адже його було створено використовуючи принципи чистої архітектури та коду.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Adler I. Augmented Reality in modern life. Tutorial – Williams, 2016. – 359 p.
2. Berd R. Pearls of Functional Algorithm Design. Tutorial – DMK-Press, 2013. – 454 p.
3. Cardeli L. Typeful programming. Tutorial – Surveys, 1991.- 670 p.
4. Graham I. Object-Oriented: Principles & Practice. Tutorial – Williams, 2004. – 880 p.
5. Klements P., Norsrop L. Software Product Lines: Practices and Patterns. Tutorial – Addison-Wesley, 2002. – 608 p.
6. Larman K. Applying UML and Patterns : An Introduction to Object-Oriented Analysis and Design and Iterative Development. Tutorial – Williams, 2006. – 736 p.
7. Ахо А., Хопкрофт В., Ульман Д. Структуры данных и алгоритмы. Пособие – Вильямс, 2000. – 384 с.
8. Берг М., Чеонг О., Кревельд М. Вычислительная геометрия. Алгоритмы и приложения. Пособие – ДМК-Пресс, 2016. – 438 с.
9. Гамма Э., Хелм Р., Джонсон Р. Приемы объектно-ориентированного проектирования. Паттерны проектирования. Пособие – Питер, 2007. – 366 с.
10. Гранд М. Шаблоны проектирования в JAVA. Каталог популярных шаблонов проектирования, проиллюстрированных при помощи UML. Пособие – Новое издание, 2004. – 560 с.
11. Кнут Д. Искусство программирования, том 4, А. Комбинаторные алгоритмы. Пособие – Вильямс, 2013. – 960 с.
12. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ, 3-е издание. Пособие – Вильямс, 2013. – 1328 с.
13. Макконнелл С. Совершенный код. Пособие – Питер, 2005. – 896 с.
14. Марк Д. Swift. Разработка приложений в среде Xcode для iPhone и iPad. Пособие – Вильямс, 2016. – 816 с.
15. Мартин Р. Принципы OOD. SOLID. Пособие – Вильямс, 2010. – 690 с.

16. Мухортов В., Рылов В. Объектно-ориентированное программирование, анализ и дизайн. Пособие – Новосибирск, 2010. – 211 с.
17. Нойбург М. Objective-C Основы. Пособие – Вильямс, 2014. – 416 с.
18. Орлов С. Технології розробки програмного забезпечення. Пособие – Питер, 2008. – 464 с.
19. Препарата Ф., Шеймос М. Вычислительная геометрия: Введение. Пособие – Мир, 1989. – 478 с.
20. Приват М. Программирование на Objective-C в Xcode. Пособие – Вильямс, 2012. – 384 с.
21. Романовский К. Метод разработки документации семейств программных продуктов. Пособие – СПбГУ, 2007. – 112 с.
22. Седжвик Р. Фундаментальные алгоритмы на С. Алгоритмы на графах. Пособие – ДияСофтЮП, 2003. – 480 с.
23. Фаулер М. Шаблоны корпоративных приложений (Signature Series). Пособие – Вильямс, 2012. – 544 с.
24. Форсайт Д.А., Понс Дж. Компьютерное зрение. Современный подход. Пособие – Вильямс, 2018. – 960 с.
25. NEURAL NETWORK APPROACH FOR EMOTIONAL RECOGNITION IN TEXT стаття Журнал Біоніка інтелекту, Харків: ХНУРЕ, 2019. – 1(92). – С. 9-14 I.V. Afanasieva, D.S. Nazarenko, N.V. Golian.
26. Yerokhin, A.L., Babii, A.S., Nechyporenko, A.S., Turuta, O.P. / A Lars-Based Method of the Construction of a Fuzzy Regression Model for the Selection of Significant Features // Cybernetics and Systems Analysis. №4, 2016. - P. 167–173.
27. Andriy Yerokhin, Valerii Semenets , Alina Nechyporenko, Andrii Babii, Oleksii Turuta. F-transform 3D Point Cloud Filtering Algorithm // Proc. of the 2th IEEE International Conference on Data Stream Mining & Processing. 21-25 August 2018, Lviv, Ukraine. - P.524-527. DOI: 10.1109/DSMP.2018.8478581.
28. Andriy Yerokhin, Alina Nechyporenko, Andrii Babii, Oleksii Turuta, Ihor Mahdalina. Usage of Phase Space Diagram to Finding Significant Features of Rhinomanometric Signals // Computer Science & Information Technologies

(CSIT'2016), 14-17 September 2016, Lviv, Ukraine. – P. 70 - 72. DOI: 10.1109/STC-CSIT.2016.7589871.

29. This Creepy Chrome Extension Pauses YouTube for You When You Look Away <https://gizmodo.com/this-creepy-chrome-extension-pauses-youtube-for-you-whe-1826257465>

30. Rosenblatt, Frank (1958), The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Cornell Aeronautical Laboratory, Psychological Review, v65, No. 6, pp. 386—408.

31. M. Abadi, A. Agarwal et al., “Tensorflow: Large-scale machine learning on heterogeneous distributed systems,” arXiv preprint arXiv:1603.04467, Mar. 2016. [Online]. Available: <https://arxiv.org/abs/1603.04467>