

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Центр післядипломної освіти
Кафедра Програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

_____ другий (магістерський) _____

(рівень вищої освіти)

_____ Дослідження методів безконтактного керування віртуальними об'єктами _____

Виконав:

студент 2 курсу групи ПЗмзд-21-1

_____ Чуйков О.С. _____

(прізвище, ініціали)

Спеціальність

121 – Інженерія програмного
забезпечення

Тип програми

Освітньо-наукова

Керівник

доц. Лановий О. Ф.

_____ (посада, прізвище, ініціали)

Допускається до захисту

Зав. Кафедри

_____ З.В. Дудар

2023

Харківський національний університет радіоелектроніки

Факультет _____ Центр післядипломної освіти _____

Кафедра _____ Програмної інженерії _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 121 – Інженерія програмного забезпечення _____

(код і повна назва)

Тип програми _____ освітньо-наукова програма _____

Освітня програма _____ Інженерія програмного забезпечення _____

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«__» _____ 2023 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студента _____ Чуйкова Олександра Сергійовича _____

(прізвище, ім'я, по батькові)

1. Тема роботи: «Дослідження методів безконтактного керування віртуальними об'єктами», затверджена наказом університету від «03» квітня 2023 р. № 83 Стз.

1. Термін подання студентом роботи до екзаменаційної комісії: «15» травня 2023 р.

2. Вихідні дані до роботи електронні ресурси за обраною тематикою, методичні вказівки до виконання кваліфікаційної роботи магістра, пояснювальна записка.

3. Перелік питань, що потрібно опрацювати в роботі аналіз предметної області, постановка задачі, огляд існуючих методів та патентів, опис розробленої системи, математичне моделювання, опис проектних рішень, опис та аналіз експериментів.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз предметної галузі	26.01.2023	виконано
2	Виявлення проблематики галузі	28.01.2023	виконано
4	Аналіз алгоритмів управління віртуальними об'єктами	25.02.2023	виконано
5	Дослідження можливостей програмної реалізації обраних моделей	28.02.2023	виконано
6	Розробка плану проведення експериментальних досліджень	06.04.2023	виконано
7	Проведення експериментальних досліджень	12.04.2023	виконано
8	Аналіз отриманих результатів	16.04.2023	виконано
9	Підготовка пояснювальної записки	06.05.2023	виконано
10	Підготовка презентації та доповіді	11.05.2023	виконано
11	Перевірка на плагіат	11.05.2023	виконано
12	Нормоконтроль	11.05.2023	виконано
13	Архівування	16.05.2023	виконано
14	Попередній захист	18.05.2023	виконано
15	Допуск до захисту у зав. кафедри	18.05.2023	виконано

Дата видачі завдання 26 січня 2023 р.

Студент _____

(підпис)

Керівник роботи _____

(підпис)

доц. Лановий О.Ф.

(посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Кваліфікаційна робота магістра: 80 с., 24 рис., 22 джерел.

**ВІРТУАЛЬНА РЕАЛЬНІСТЬ, ДОПОВНЕНА РЕАЛЬНІСТЬ,
МЕТАПРОСТІР, МЕТОДИ УПРАВЛІННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ,
МАШИННИЙ ЗІР**

Об'єкт дослідження – методи управління віртуальними об'єктами шляхом реалізації контролю за напрямком погляду людини.

Мета роботи – дослідити методи управління складними об'єктами, основні методи, моделі та алгоритми відстеження напрямку погляду людини та можливості використання веб-камери для управління віртуальними об'єктами.

Методи дослідження. В роботі використано системний підхід до аналізу системи, методи машинного зору та доповненої реальності, математичне моделювання для моделі вибору кращого методу при обробці вибірок вхідних даних.

**VIRTUAL REALITY, AUGMENTED REALITY, METAVERSE, CONTROL
METHODS, SOFTWARE, MACHINE VISION**

Research subject: Control methods for virtual objects through the implementation of gaze direction monitoring.

Objective: To investigate control methods for complex objects, including the main methods, models, and algorithms for tracking gaze direction and the possibilities of using a webcam to control virtual objects.

Research methods: The study employs a systemic approach to analyze the system, machine vision and augmented reality methods, as well as mathematical modeling to determine the optimal method for processing input data samples.

Умови публікації пояснювальної записки

Я, Чуйков Олександр Сергійович
(прізвище, ім'я, по батькові)
студент(ка) групи ІІЗм-21-1 здобувач вищої освіти на другому (магістерському)
рівні кафедра програмної інженерії,
(повна назва кафедри)

заявляю: моя кваліфікаційна робота на тему

Дослідження методів безконтактного керування віртуальними об'єктами,
(назва роботи)

що буде представлена до ЕК для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений (а) з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

Вступ.....	7
1 Опис проблемної галузі	9
1.1 Аналіз предметної області.....	9
1.2 Актуальність теми дослідження	13
1.3 Постановка задачі.....	16
2 Віртуальні об’єкти та віртуальна реальність.....	18
2.1 Метапростір	18
2.2 Віртуальна реальність.....	21
2.3 Віртуальні об’єкти та засоби керування	23
2.4 VR та технології комп’ютерного зору	26
2.5 VR та технології Eye-tracking	29
3 Принципи керування поглядом	31
3.1. Опис проблеми	31
3.2 Аналіз алгоритмів відстеження напрямку зору	33
3.3 Геометрична модель	37
3.4 Методи безконтактного керування з використанням веб-камери	40
3.5 Існуючі програмні рішення	42
3.6 Загальні умови проведення досліджень.....	45
4 Аналіз результатів дослідження	48
4.1 Алгоритм відстеження очей і калібрування	48
4.1.1 Калібрування в GazeRecorderAPI	49
4.1.2 Калібрування в WebGazer	52
4.2 Порівняння результатів калібрування.....	55
4.3 Проведення експериментів та порівняння результатів	58
Висновки	65
Перелік джерел посилання	66
Додаток А	70
Додаток Б.....	71
Додаток В	80

ВСТУП

Достатньо уповільнений розвиток економіки України був зафіксований протягом останніх років ще до початку повномасштабного вторгнення РФ, але найбільших втрат він зазнав після 24 лютого 2022 року. Вимушене переміщення науково-дослідних та виробничих центрів на захід України та за її межі суттєвим чином вплинув і на індустрію розробки програмного забезпечення. Однак це також значно прискорило розробку та впровадження технологічних рішень, пов'язаних з реалізацією віддаленої взаємодії між людиною та апаратно-програмними комплексами, які знаходять своє застосування в різних сферах. Одним з таких напрямків є взаємодія з віртуальними об'єктами.

Зазвичай управління віртуальними об'єктами здійснюється з використанням звичайних засобів управління – клавіатури, миші, джойстика, керма або їх аналогів. Дуже прогресивними технологіями вважаються системи, що орієнтовані на розпізнавання руху людини, однак ще одним елементом, який можна ефективно використовувати для керування такими об'єктами, є напрям погляду людини.

Основну частину інформації про навколишній світ людина сприймає за допомогою зору. Елементи зорової системи – око, нерв і зоровий аналізатор головного мозку – тісно пов'язані між собою, тому вивчення траєкторії руху очного яблука дозволяє робити висновки про процес розпізнавання окремих образів та розумових процесах в цілому. Відстеження напрямку погляду також дозволяє будувати принципово нові інтерфейси взаємодії між людиною та технічними засобами. У зв'язку з цим розвиток систем, що використовують напрям погляду для управління складними об'єктами, є актуальною науково-технічною проблемою.

Технології відстеження напрямку людського погляду (eye-tracking systems) використовують у різних системах проектування комп'ютерних інтерфейсів. Час фіксації ока і щільність траєкторії дозволяють робити висновки про те, яким елементам об'єкта, що розглядається, приділяється найбільша увага.

Сучасні напрями застосування eye-tracking систем в інженерії програмного забезпечення можна розділити на дві основні частини. Це системи, які здійснюють пасивний збір та аналіз інформації про рух ока протягом експерименту. До цієї галузі можна віднести основне на даний момент комерційне використання eye-tracking систем – оцінку візуальної зручності інтерфейсів додатків та веб-сторінок. Це надає можливість врахувати під час розробки програмної системи об'єктивну інформацію про те, на які елементи інтерфейсу найчастіше звертає увагу користувач під час роботи з програмою. В медицині eye-tracking системи використовуються для діагностики захворювань нервової системи – при деяких із них рухи очей хворої та здорової людини суттєво різняться. З іншого боку eye-tracking системи можна буде застосовуватися для ідентифікації особистості, оскільки характер рухів очей суто індивідуальні, як почерк, але ці рухи майже неможливі для імітації.

Сучасні операційні системи (наприклад, Windows 11) вже адаптовані до використання окремих видів eye-tracking систем, які передбачають наявність додаткового апаратного забезпечення. До таких систем, наприклад, відноситься Tobii Eye Tracker (<https://www.amazon.com/Tobii-Eye-Tracker-5/dp/B0897GCBWW>). Разом з тим окремими видами таких систем слід вважати такі, які дозволяють використовувати звичні засоби комунікації для реалізації eye-tracking систем – смартфони та веб-камери.

Метою дослідження є аналіз методів використання eye-tracking систем для управління віртуальними об'єктами, до яких можна віднести систему меню програмної системи.

Предметною галуззю дослідження, з огляду на окреслену проблематику, є методи управління складними об'єктами, що дозволяє акцентувати увагу на доцільність використання результатів проведеної роботи в різних технічних системах.

1 ОПИС ПРОБЛЕМНОЇ ГАЛУЗИ

1.1 Аналіз предметної області

З моменту появи обчислювальної техніки інтерфейси користувальницької взаємодії з нею еволюціонували від опосередкованих (клавіатура, миша) до природних, які базуються на фізіологічних здібностях людини без участі додаткових гаджетів.

Людино-машинна взаємодія (ЛМВ) є міждисциплінарною галуззю, яка розвинулася на рубежі 70-х і 80-х років ХХ століття у зв'язку з розвитком комп'ютерів і комп'ютерних мереж. ЛМВ досліджує питання взаємодії та спілкування між комп'ютерами (комп'ютерними системами) і людьми (користувачами), як окремими особами, так і групою. ЛМВ включає в себе багато різних галузей навчання, які не мають багато спільного, але всі ці галузі вважаються ключовими дисциплінами: інформатика, дизайн, ергономіка, психологія, когнітивні науки, штучний інтелект та деякі інші.

Взаємодія відбувається у формі відкритого діалогу між користувачем і комп'ютером. Простір, у якому відбувається цей діалог і забезпечує взаємодію, зазвичай називають інтерфейсом користувача. Він включає як апаратне, так і програмне забезпечення. У більшості випадків інтерфейс не складається лише з однієї частини (апаратне забезпечення або програмне забезпечення), а складається з багатьох компонентів, які утворюють повну систему. Наприклад, користувач пише простий текст за допомогою персонального комп'ютера, для цього йому потрібні клавіатура, додаток і монітор. Людино-машинна взаємодія визначається як дисципліна, що займається розробкою, оцінкою та впровадженням інтерактивних комунікацій.

Природні інтерфейси, у свою чергу, поділяються на голосові (сервіси голосових помічників), сенсорні (тач-панелі, екрани) та безконтактні (керування жестами рук, рухом тіла, мімікою, рухами очей). Перспективні, але поки недоступні для широкої аудиторії та перебувають найчастіше в стадії тестування інтерфейси «мозок-комп'ютер», що пов'язані з використанням ЕЕГ (неінвазивні методи зчитування активності кори мозку).

У сучасних системах управління робототехнічними пристроями використовуються різноманітні засоби введення інформації, що надають дані у певних форматах на окремих каналах. До найпоширеніших способів, крім традиційних систем введення інформації з клавіатури, миші та маніпуляторів, належать безконтактні системи голосового управління та управління за допомогою жестів. Речовий спосіб подання команд особливо зручний, коли руки людини зайняті. Жести є більш зручними, коли людина обмежена у звичних методах керування або обмежена у речових командах. Але ще одним методом управління, який має високий ступінь застосування в комп'ютерних системах, є відстеження руху очей людини.

Дослідження рухів ока проводилися вже наприкінці ХІХ – на початку ХХ століть, зокрема, офтальмологом Луї Емілем Жавалем, психологами Едмундом Х'ю та Гаєм Баксвеллом [1-2]. Дослідження, що проводились багатьма вченими щодо вивчення характеру рухів ока при читанні текстів і розгляданні об'єктів і зображень встановили, що очі рухаються не безперервно, а різкими стрибками (саккадами) з деяким часом фіксації між ними, протягом якого відбуваються значно менш амплітудні мікросаккадичні рухи [3]. Також було встановлено досить очевидний зв'язок між когнітивною діяльністю людини та траєкторією ока. Залежно від того, що саме було важливим для людини, що розглядалось на зображенні, змінювалася щільність та форма траєкторії в різних її ділянках.

Після Другої світової війни одне з перших досліджень щодо напрямку погляду було здійснено в 1947 році групою Фітса, Джонса та Мілтона [9]. Наприкінці 1940-х років вони опублікували технічні звіти, які вважаються основоположними дослідженнями візуальної вибірки і представляють собою найбільшу колекцію даних про рух очей, зібраних під час візуального моніторингу. Ці дані охоплюють понад 500 000 кадрів відеозаписів понад 40 пілотів, зроблених за різних умов польоту. Загальні висновки наступні:

«Доцільно припустити, що частота фіксацій очей є показником відносної важливості цього інструменту. Тривалість фіксацій, навпаки, можна розглядати як показник відносної складності перевірки та інтерпретації певних приладів. [...]

Якщо ми знаємо, куди дивиться пілот, ми не обов'язково знаємо, що він бачить але ми знаємо дещо з того, про що він думає» [4].

В результаті цієї роботи автори змогли запропонувати більш ефективне розташування приладів і визначити ті з них, які важко зчитуються, для можливого перепроєктування власне приладу. Це було перше дослідження, яке дозволило взаємодіяти між додатком (кабіною літака) і системою відстеження погляду людини, коли для проведення вимірювань було використано відео. Власне, вони проводилися переважно за допомогою медичної техніки, яка дозволяла реєструвати рухи очних яблук за допомогою низки електродів (рис.1.1).



Рисунок 1.1 – Система для реєстрації руху очей

Більшість описаних методів вимагали, щоб голова глядача була нерухомою під час відстеження очей, і використовували різноманітні інвазивні пристрої.

Головним нововведенням у відстеженні погляду став винахід налобного трекера, який все ще широко використовується (див. рис. 1.2). З цього періоду часу роботи з відстеження руху очей людини тісно пов'язані з продуктивністю комп'ютерів. Чим більше вони прогресують, тим більше вимагають необхідних ресурсів для роботи в реальному часі та для складних додатків.

Протягом 80-х років минулого сторіччя інтерес до відстеження погляду не згасав, неймовірний бум персональних комп'ютерів дозволив розробити нові інтерфейси та способи взаємодії з комп'ютером.

Одним з провідних дослідників у цій області є доктор Левін, який визнав потенціал відстеження погляду в інтерактивних додатках ще на початковому етапі розвитку [5].



Рисунок 1.2 – Система SR Research EyeLink II для відстеження рухів очей

За останні роки відстеження очей і погляду стало важливою галуззю промисловості, і багато виробників розробили продукти, спрямовані саме на цю область. Наприклад, найбільш доступний на сьогоднішній день стаціонарний айтрекінг-пристрій (вартість близько \$150) – Tobii eye-tracking 4с. Цей пристрій (як і IR-датчик глибини Kinect, що отримав досить велику популярність) – ігровий пристрій, який вийшов ще у 2017 р. та дозволяє відстежувати не тільки рухи очей, а й повороти голови.

Зараз ці пристрої можна використовувати і як заміну мишки для керування Windows, і як ігровий інтерфейс. Особливо важливим є той факт, що розробники випустили SDK для найбільш популярних на сьогоднішній день ігрових движків Unreal Engine та Unity, що дозволяє збирати будь-які об'ємні інтерактивні сцени. Завдяки нодовій системі програмування Unreal Engine blueprint не обов'язково розумітися на кодї – формування такого контенту стає доступним великому колу розробників. Створені в Unreal Engine сцени можна імпортувати майже куди завгодно – відкривати у вигляді програм Windows, у вигляді додатків на мобільному телефоні, через сайт в браузері. Індивідуальне калібрування зведене

до мінімуму і займає близько хвилини. Підтвердження потрібної дії може бути реалізовано декількома способами: і поєднанням з натисканням клавіш, і затримкою погляду на підтверджувальному елементі (як зараз – джойстиком у VR-окулярах), або морганням чи закриттям ока. SDK дозволяє активізувати об'єкти при погляді на них: у деяких іграх, наприклад, вони можуть руйнуватися, запускати анімацію, а неігрові персонажі можуть буквально ловити на собі погляд і проявляти реакцію у відповідь.

Підсумовуючи можна сказати, що хоча системи відстеження погляду існують вже давно, донедавна відстеження та вимірювання поведінки очей і реакції погляду було дуже складним і дорогим завданням, яке належало до компетенції дослідницьких або військових лабораторій. Пристрої стеження за очима були незручними системами, що кріпилися на голові, тому вони в основному використовувалися як вказівні пристрої для дуже вузького кола застосувань (в основному військових). Однак швидкий технологічний прогрес (збільшення швидкості процесорів, вдосконалення цифрової обробки відео) знизив вартість і значно підвищив ефективність обладнання для відстеження очей і погляду.

1.2 Актуальність теми дослідження

На сьогоднішній день більшість комерційних трекерів для очей покладаються на активне освітлення, наприклад, трекери на основі відбиття зіниці в центрі рогівки, що використовують інфрачервону підсвітку. Хоча такі трекери очей забезпечують високу точність, вони дорогі, зазвичай використовуються в закритих лабораторних умовах і сильно обмежують рухи голови користувача. В своїй роботі я досліджую спрощення процесу відстеження очей за допомогою звичайної веб-камери.

Актуальність теми «Дослідження методів безконтактного керування віртуальними об'єктами» може бути обґрунтована наступними аргументами:

- розширення застосування віртуальної реальності: Віртуальна реальність широко використовується в різних галузях, таких як ігрова індустрія,

медицина, освіта, тренінги, дизайн, архітектура та багато інших. Дослідження методів безконтактного керування віртуальними об'єктами може допомогти розширити можливості використання віртуальної реальності, забезпечуючи більш зручні та ефективні способи взаємодії з віртуальними об'єктами;

- поліпшення користувацького досвіду: Керування віртуальними об'єктами може бути складним та вимагати використання різноманітних контролерів або клавіш на клавіатурі. Використання безконтактних методів, таких як eye-tracking, може поліпшити користувацький досвід, забезпечуючи більш природні, зручні та інтуїтивно зрозумілі способи керування віртуальними об'єктами;
- розвиток технологій eye-tracking: Eye-tracking є динамічно розвиваючоюся технологією, яка дозволяє відстежувати рухи очей користувача та використовувати цю інформацію для керування віртуальними об'єктами. Завдяки постійному розвитку технологій eye-tracking, відкриваються нові можливості для її застосування в різних галузях, включаючи віртуальну реальність.

Дослідження методів безконтактного керування віртуальними об'єктами з використанням eye-tracking також може допомогти в розробці більш ергономічних рішень, які дозволять користувачам взаємодіяти з віртуальними об'єктами більш природним та комфортним способом. Це може включати контроль рухів очей, міміки обличчя, а також комбінації цих факторів, що може допомогти створити інтуїтивні та зручні інтерфейси взаємодії з віртуальними об'єктами.

Крім того, безконтактне керування віртуальними об'єктами може допомогти зробити віртуальну реальність більш доступною для різних категорій користувачів, таких як люди з фізичними обмеженнями, люди з руховими порушеннями або з обмеженими можливостями використання традиційних контролерів. Eye-tracking може стати альтернативним рішенням для керування

віртуальними об'єктами, що відкриє можливість взаємодії з віртуальним середовищем для більш широких верст користувачів комп'ютерних систем.

Методи безконтактного керування віртуальними об'єктами може мати широкий спектр застосувань в різних галузях, таких як медицина, освіта, наука, мистецтво, розваги та інше. Наприклад, в медицині він може бути використаний для реабілітації пацієнтів з руховими порушеннями, в освіті – для створення взаємодії зі симуляційними середовищами, у науці – для проведення досліджень у віртуальних лабораторіях, а в мистецтві – для проведення інтерактивних вистав та віртуальних музейних експозицій.

Слід також зазначити, що дослідження методів безконтактного керування віртуальними об'єктами має інноваційний характер, оскільки використання eye-tracking або інших безконтактних технологій є новаторським підходом до взаємодії з віртуальною реальністю. Це може сприяти розвитку нових технологій та створенню нових рішень, що покращать користувацький досвід віртуальної реальності та має значні перспективи майбутнього розвитку. Безконтактне керування віртуальними об'єктами може мати великий потенціал, особливо з урахуванням швидкого технологічного розвитку та зростання інтересу до віртуальної реальності. Дослідження методів безконтактного керування може відкрити нові можливості для створення інноваційних продуктів та сервісів, що задовольняють зростаючі потреби користувачів.

Методи безконтактного керування віртуальними об'єктами може сприяти вдосконаленню та оптимізації методів управління, забезпечуючи кращу точність, швидкість та надійність керування. Результати таких досліджень можуть допомогти в розробці нових технологій та алгоритмів, які спрямовані на покращення ефективності та зручності безконтактного керування віртуальними об'єктами.

Методи безконтактного керування віртуальними об'єктами можуть мати широкий спектр застосувань. Наприклад, вони можуть бути використані в розвагах, іграх, віртуальному навчанні, медичних симуляціях, дизайні та моделюванні, архітектурному проектуванні, тренажерах для пілотів, військовому

тренінгу та інших сферах. Такі методи можуть мати значний практичний вплив на ряд галузей, дозволяючи розширювати можливості віртуальної реальності та поліпшувати досвід користувачів.

Розробка ефективних та зручних методів безконтактного керування може мати високу комерційну цінність. Розробка нових методів безконтактного керування віртуальними об'єктами може допомогти покращити користувацький досвід, забезпечити більш ефективну та зручну взаємодію з віртуальними об'єктами, що може привести до збільшення популярності та використання віртуальної реальності в різних галузях. Це може створити нові можливості для розробників програмного забезпечення, виробників обладнання та інших стейкхолдерів на ринку розробників програмного забезпечення.

Не треба забувати і про важливість соціального аспекту використання безконтактного керування, забезпечуючи доступність технологій віртуальної реальності для різних груп користувачів. Наприклад, це може бути особливо корисно для людей з фізичними обмеженнями, які можуть знайти в безконтактному керуванні віртуальними об'єктами зручний спосіб взаємодії як з віртуальним, так і реальним середовищем. Це також може сприяти поширенню використання віртуальної реальності в навчанні та освіті, де можливості безконтактного керування можуть забезпечувати більш інтерактивний та залучаючий досвід для учнів.

Отже, з огляду на широкий спектр можливого застосування віртуальної та доповненої реальності, потенційні переваги безконтактного керування, соціальний вплив та технічний прогрес в цій галузі, тема кваліфікаційної роботи магістра за темою «Дослідження методів безконтактного керування віртуальними об'єктами» є актуальною.

1.3 Постановка задачі

З огляду на проведений аналіз першоджерел можна наступним чином сформулювати постановку задачі – проведення досліджень, які дозволяють

оцінити ефективність застосування алгоритмів відстеження руху очей для безконтактного керування об'єктами.

Мета роботи – дослідити різні методи безконтактного керування віртуальними об'єктами з використанням eye-tracking технології.

Для досягнення цієї мети в роботі необхідно розв'язати наступні завдання:

- провести аналіз першоджерел щодо віртуальної реальності, безконтактного керування, eye-tracking технології та їх застосування в різних галузях;
- дослідити різні методи використання технології відстеження напрямку зору за допомогою веб-камери у якості засобу реєстрації, включаючи розпізнавання рухів очей, жестів, міміки обличчя та інших вхідних сигналів;
- впорядкувати та проаналізувати дані для порівняння ефективності різних методів безконтактного керування віртуальними об'єктами;
- визначити критерії оцінювання ефективності різних методів безконтактного керування віртуальними об'єктами на підставі відстеження напрямку зору;
- провести апробацію розробленого методу безконтактного керування віртуальними об'єктами з метою оцінки їх ефективності, зручності та користувацького досвіду;
- виконати аналіз отриманих результатів на основі зібраних даних та встановлених критеріїв оцінки;
- сформулювати висновки щодо ефективності використання безконтактного керування віртуальними об'єктами за допомогою веб-камери з використанням eye-tracking технологій щодо їх застосування в практичних додатках.

2 ВІРТУАЛЬНІ ОБ'ЄКТИ ТА ВІРТУАЛЬНА РЕАЛЬНІСТЬ

2.1 Метапростір

Віртуальна реальність (VR) – це імітаційне середовище, яке відтворює реальний світ або створює абсолютно нове уявлення про навколишній світ, дозволяючи користувачам взаємодіяти з ним у віртуальному просторі. Завдяки використанню спеціального обладнання, такого як VR-шоломи, контролери руху та сенсори, користувачі можуть переживати віртуальну реальність відчуттями зору, слуху, дотику та навіть руху.

Концепція віртуальних реальностей виникла з метою створення іммерсивних, інтерактивних та уявних досвідів для користувачів. Концепція віртуальних реальностей існує вже досить давно, а перші дослідження відносяться до 60-х років ХХ сторіччя. З розвитком інформаційно-комунікаційних технологій віртуальні реальності привертають все більше уваги, а їхні потенційні сфери застосування стають дедалі ширшими. Під віртуальною реальністю розуміють надання користувачам доступу до віртуальних середовищ, які або нагадують реальний світ, або створюють уявне оточення відповідно до власних уподобань за допомогою технологій [6].

Донедавна віртуальна реальність використовувалася переважно в таких галузях, як військова справа та охорона здоров'я – VR використовується для створення симуляційних середовищ, які можуть бути використані для тренування та навчання. Ігрова індустрія також побачила великий потенціал в адаптації віртуальної реальності для її застосування у своїх продуктах, оскільки це дозволяє сприяти більш глибокому емоційному зв'язку для гравців. Зараз застосування VR розширюється і створює нові можливості для бізнесу.

Метапростір (Metaverse) – це ідея віртуальної реальності, яка була започаткована Нілом Стівенсоном у 1992 році, коли люди створюють аватари, які існують у тривимірному віртуальному світі [7]. Вважається також, що метавіртуальність перебуває в процесі еволюції, і, оскільки обсяги віддаленої роботи тільки зростають, такі компанії, як Facebook і Microsoft бачать все більше можливостей для її застосування з метою покращення досвіду віддаленої роботи

для співробітників. Хоча використання віртуальної реальності розширюється на нові способи роботи, важливо враховувати ставлення користувачів до цих технологічних розробок, оскільки це може або збільшити переваги роботи з дому, або ще більше підкреслити її негативні аспекти.

Причиною зростання популярності метаспростору серед широкої громадськості стало оголошення Facebook у жовтні 2021 року про проведення компанією ребрендингу на Meta в результаті переорієнтації своєї діяльності на розвиток Metaverse. Хоча стверджується, що до повноцінного впровадження метаспростору ще достатньо велика відстань, досягнення у сфері VR доводять, що ця технологія має великий потенціал стати майбутнім Інтернету, відкриваючи світ нових цифрових можливостей (див.рис.2.1).

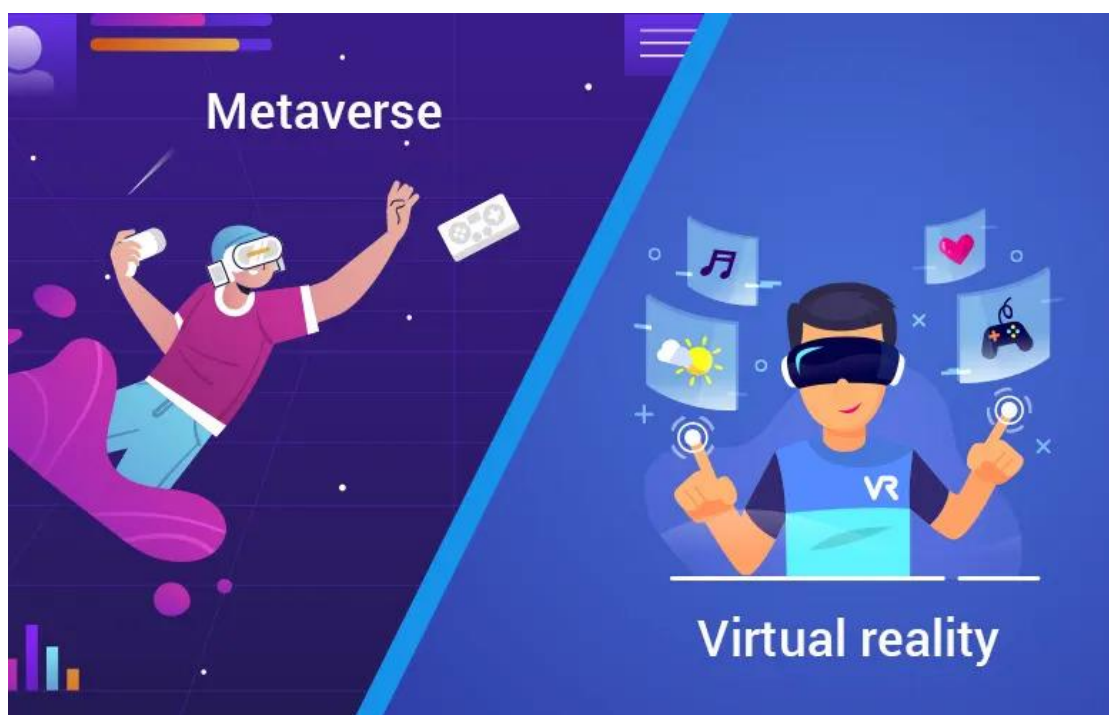


Рисунок 2.1 – Метаспростір та віртуальні реальність

Глобальна пандемія COVID-19 сприяла реалізації можливостей для використання метаспростору на віддаленому робочому місці, змінивши напрямок його використання з розважальної сфери на професійну [8].

Основна ідея метаспростору полягає у створенні цифрового простору, який інтенсифікує соціальну взаємодію, що нагадує взаємодію у фізичному контексті. Звичайні відеоконференції, тобто віддалена робота в тому вигляді, в якому ми її

бачимо сьогодні, не здатні охопити людську взаємодію під час особистих зустрічей, що, у свою чергу, впливає на рівень співпраці між працівниками. Вважається, що метaprостір допомагає віддаленій роботі повністю розкрити свій потенціал, відтворюючи фізичні робочі контексти для заохочення товариських стосунків, які надихають на творчість і підвищують продуктивність [8].

На сьогодні не існує єдиного визначення того, що саме таке метaprостір. Metaverse – це віртуальний простір, доступ до якого здійснюється через дисплей або засіб VR, через який користувачі можуть взаємодіяти з іншими так, ніби вони присутні в одній кімнаті, що робить можливим також взаємодію за допомогою невербальних жестів. Ідея метaprостору полягає в тому, щоб створити повністю імерсивний світ, де люди створюють аватари, які взаємодіють з іншими людьми через їх аватари і навколишнім середовищем у віртуальному просторі з метою відтворити людську взаємодію в реальному світі (рис.2.2).

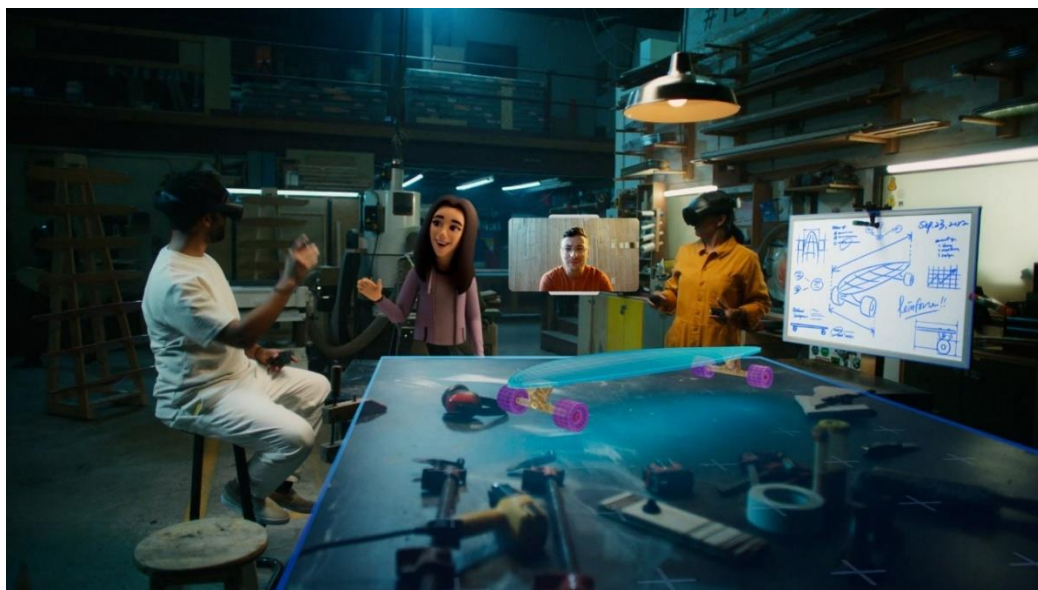


Рисунок 2.2 – Метaprостір як соціальне середовище

Донедавна такий підхід до віртуальної реальності здебільшого асоціювалася з ігровою індустрією, де гравці можуть створювати власні віртуальні світи, такі як Fortnite [9]. На теперішній час Fortnite також визнала потенціал метaprостору і змінила свою концепцію використання віртуальної реальності, перетворивши її з гри на соціальний простір для своїх користувачів.

2.2 Віртуальна реальність

Як і Інтернет, VR зараз стрімко розвивається і впроваджується у різних сферах. VR відкриває нові способи взаємодії з комп'ютерними системами та середовищами. Вона дозволяє користувачам взаємодіяти з віртуальними об'єктами та світом шляхом рухів, жестів і голосових команд, що створює більш природні та інтуїтивні взаємодії.

Розвиток віртуальних середовищ пройшов довгий шлях, хоча бар'єр між фізичними та віртуальними середовищами не є чітким, а цифровий перехід все ще триває. VR дозволяє користувачам зануритися у повністю іммерсивне середовище, відчуваючи себе частинкою віртуального світу. Відчуття присутності і взаємодії з навколишнім середовищем робить досвід більш реалістичним та захоплюючим.

VR, у свою чергу, є лише одним з трьох елементів концепції розширеної реальності. На рис.2.3 зображено всі реальні та віртуальні комбіновані середовища, які підпадають під загальну термінологію розширеної реальності.

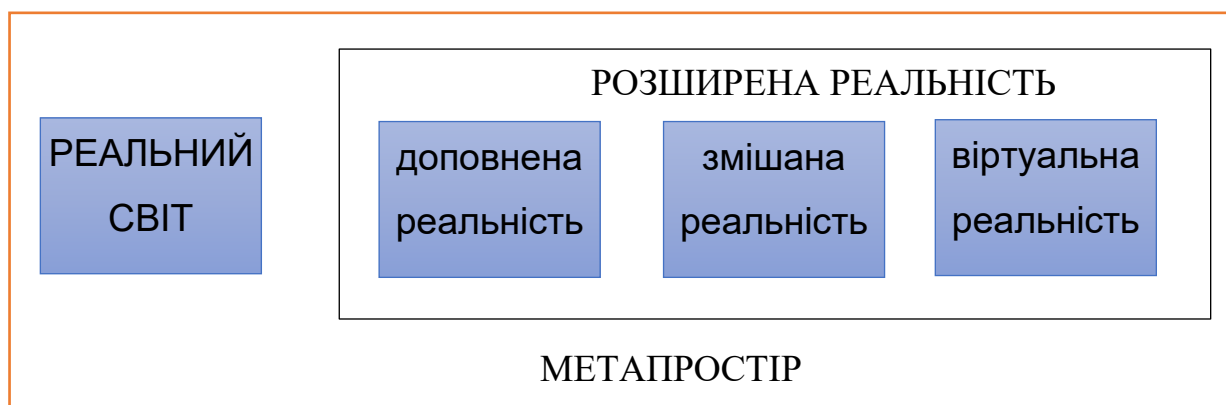


Рисунок 2.3 – Спектр метапростору

Окрім VR розширена реальність охоплює також доповнену та змішану реальність. Їх розташовують на спектрі, посилаючись на рівень взаємодії з реальним або віртуальним світом. Доповнена реальність має найменшу взаємодію з віртуальним світом, оскільки вона функціонує, нашаровуючи реальний світ на віртуальні об'єкти, щоб змінити те, що ми бачимо, за допомогою технологічного пристрою. Змішана реальність, з іншого боку, є вдосконаленням доповненої

реальності, оскільки вона полегшує інтерактивність з віртуальним світом, дозволяючи більш тісне співіснування двох світів в межах одного дисплею.

На початку свого існування VR була не більше ніж ідеєю та баченням, які не були достатньо переконливими через недостатній розвиток технологій на той час. Зі стрімким розвитком технологій і впровадженням численних різноманітних інструментів VR стала якісним продуктом інформаційно-комунікаційних технологій, доступним широкому загалу. VR не тільки існує, але й стає все більш досконалою, стає симуляцією реальності.

Доповнену реальність активно використовують у інформаційному просторі. Поширення браузерів доповненої реальності (Wikitude, Layar, blippAR та інших) виводить технологію у масове використання – в газети, буклети, рекламні проспекти, журнали, карти поміщаються зображення-мітки для подальшої візуалізації цифрових об'єктів. У ролі додаткової інформації можуть виступати всі види цифрових даних: текст, зображення, відео, звук або тривимірні об'єкти – статичні або анімовані.

VR може бути використана в медицині для реабілітації пацієнтів, що відновлюються після травм, інсультів або інших неврологічних уражень. За допомогою VR можна створювати інтерактивні середовища, які допомагають пацієнтам відновити рухові навички, координацію та розвивати моторні функції. Наприклад, пацієнти можуть виконувати вправи рухової терапії або ігри у віртуальному середовищі, що сприяє поліпшенню їх фізичного стану. Крім того, вона дозволяє створювати контрольовані середовища для закріплення практичних навичок, зменшення болю та зниження стресу (див.рис.2.4).

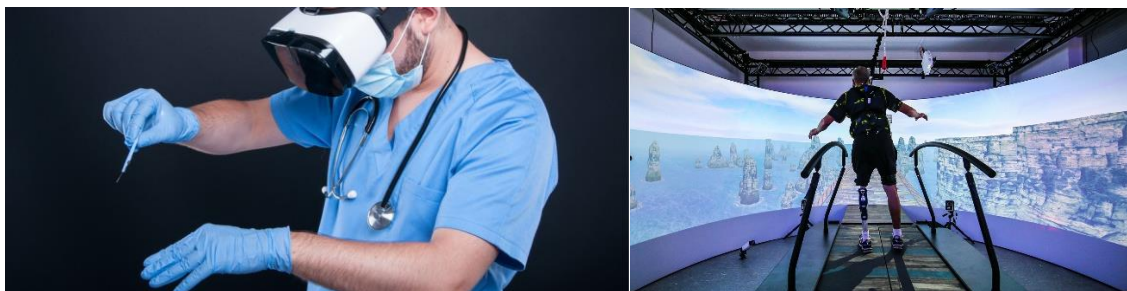


Рисунок 2.4 – VR в медицині

VR виявляється корисною у численних аспектах військової діяльності. VR може бути використана для симуляційного навчання військових особового складу. За допомогою VR створюються віртуальні середовища, які дозволяють солдатам тренувати різні сценарії бойових дій, такі як ведення бойових дій у міському середовищі або бойові маневри (рис.2.5).



Рисунок 2.5 – VR в підготовці військових

Об'єкти, які існують у віртуальному світі VR, можуть бути створені комп'ютерними програмами або моделюватися з використанням реальних об'єктів. Віртуальні об'єкти можуть бути різних типів і форм, таких як будівлі, пейзажі, тварини, люди, транспортні засоби тощо, але все одно вони тим чи іншим способом пов'язані з реальними фізичними об'єктами. Тому необхідно розглянути особливості побудови та взаємодії з віртуальними об'єктами.

2.3 Віртуальні об'єкти та засоби керування

Віртуальний об'єкт – це графічний елемент або об'єкт, який відображається на екрані додатку з прив'язкою до VR. Віртуальні об'єкти можуть бути різних типів і форм, статичні та динамічні, від чого залежить і ступінь взаємодії з ними.

Взаємодія з віртуальними об'єктами залежить від можливостей технології VR та використаного обладнання. Зазвичай користувачі можуть пересуватися по віртуальному простору, взаємодіяти з об'єктами за допомогою контролерів руху або жестів.

Взаємодія зі статичними об'єктами з боку користувача може включати роздивлення деталей, обертання або переміщення об'єкту, активацію або

взаємодію з кнопками або лаштунками. Відчуття дотику або силової взаємодії зі статичними об'єктами може бути обмежене, оскільки вони не реагують на дії користувача та залежать від контролерів зворотного зв'язку, яким обладнані користувальницькі засоби VR (рис.2.6). Зазвичай взаємодія з нерухомими статичними об'єктами (стіна або важкий предмет) імітується фрізінгом руху, щоб запобігти проникненню користувача всередину нерухомого об'єкту.



Рисунок 2.6 – Статичні об'єкти у VR

Динамічні об'єкти у VR – це ті, які можуть рухатися або змінювати своє становище під час взаємодії. Наприклад, це можуть бути рухомі перешкоди, рухомі об'єкти, тварини чи інші суб'єкти, які впливають на взаємодію віртуального світу. Взаємодія з динамічними об'єктами може включати уникання перешкод, вплив на їх рух або поведінку, керування ними. Динамічні об'єкти можуть реагувати на дії користувача, що створює відчуття взаємодії та реалізму.

Статичні об'єкти зазвичай не реагують на дії користувача, бо вони залишаються незмінними незалежно від цих дій. Динамічні об'єкти, натомість, можуть реагувати на дії або на зміну умов віртуального світу. Наприклад, можна впливати на рух або поведінку динамічних об'єктів і вони можуть взаємодіяти з оточуючими елементами або з іншими об'єктами дотримуючись фізичних особливостей побудови віртуального середовища.

Керування віртуальними об'єктами за допомогою системи меню є одним зі способів взаємодії користувача з об'єктами VR. Система меню надає зручний спосіб доступу до різних опцій та дій, пов'язаних з об'єктами, безпосередньо з віртуального середовища (рис.2.7) за допомогою контролерів руху.



Рисунок 2.7 – Система меню для керування об'єктами у VR

Контролери руху зазвичай реалізуються як фізичні пристрої з датчиками, такими як акселерометр для вимірювання прискорення пристрою, гіроскоп для вимірювання обертання пристрою або магнітометр для вимірювання сили та напрямку магнітного поля в точці простору. Широко відомими прикладами таких контролерів є контролери Nintendo Wii Remote Plus5, Sony PlayStation, Kinetics або система STEM від Sixense (рис.2.8).

Хоча ці контролери надають досить точні сенсорні дані, вони вимагають, щоб користувач тримав фізичний об'єкт у руках.



Рисунок 2.8 – Приклади контролерів фізичного руху

Керування рухом на основі зору, яке іноді також називають розпізнаванням жестів на основі комп'ютерного зору, є потужним інструментом для підтримки користувача у взаємодії та маніпуляціях у 3D-просторі без необхідності тримати фізичні сенсорні пристрої.

2.4 VR та технології комп'ютерного зору

В основі VR лежить технологія комп'ютерного зору. Комп'ютерний зір – це набір методів, які займаються вилученням інформації з зображень, і важливо відзначити, що зображення можуть бути самих різних типів [10]. 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. 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 (рис.2.9) [11].

Комп'ютерний зір на сьогодні займається розв'язанням проблеми розуміння зображень та того, що на них відбувається. Це явище має назву високорівневого комп'ютерного зору [17]. Ця галузь займається формуванням завдань та інструкцій для комп'ютерів, які досить легко сформувавати. Наприклад, «виписати всі об'єкти з даного фото, визначити їх клас і місце розташування на фото». Або необхідно визначити за відео яким видом діяльності зайняті об'єкти.

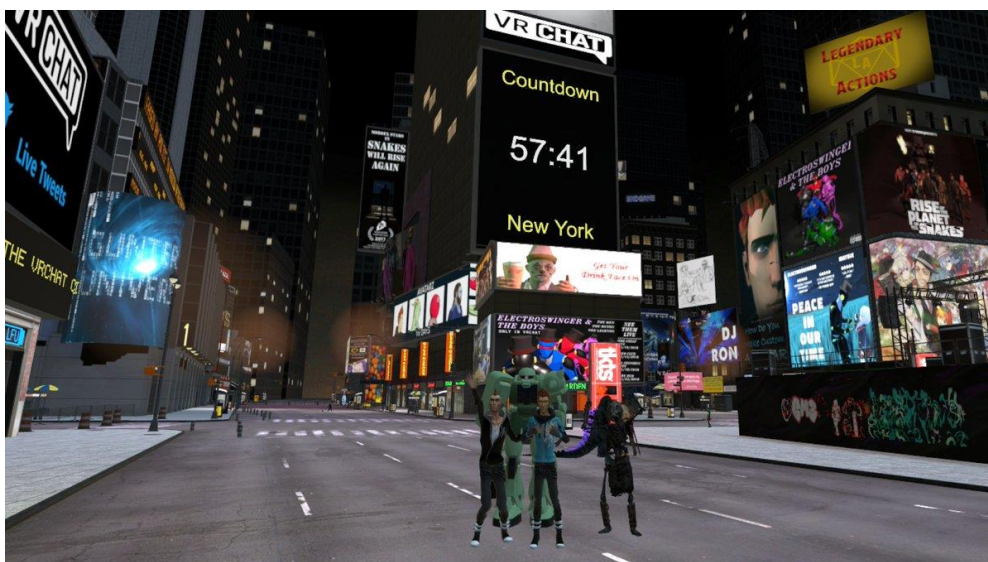


Рисунок 2.9 – Тривимірна модель Times Square

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

Для моделювання об'єктів у VR найчастіше використовується покроковий процес. Першим кроком є створення віртуального середовища, яке складається з віртуального оточення та набору віртуальних об'єктів, якими реальні люди можуть маніпулювати в реальному часі. Наступним кроком є організація мультимодальної взаємодії між віртуальними об'єктами і користувачами, що може включати розмову, зоровий контакт, посилення на віртуальні об'єкти, погляд на них та пересування віртуальних об'єктів [19].

Сприйняття реального об'єкту та його віртуального образу як єдиного об'єкту здійснюється на підставі співставлення даних, які описують віртуальний об'єкт, з тими, які відповідають очікуванням людини, тобто відповідають їх параметрам, і, якщо сукупність усіх оброблених даних дозволяє мозку створити когерентну модель світу, людина сприймає результат як реальний (рис.2.10).

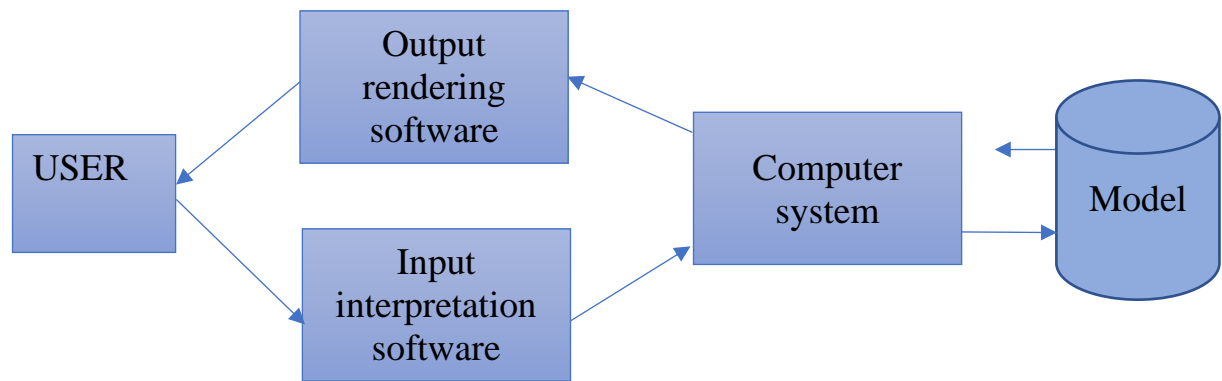


Рисунок 2.10 – Ланцюг взаємодії людини та віртуальних об’єктів

Бажання позбутися фізичних засобів керування віртуальними об’єктами напряду пов’язано з розв’язанням задачі розпізнавання рухів. Таке розпізнавання жестів відбувається на основі використання технологій комп’ютерного зору та порівняння цих дій із заздалегідь навченими рухами з використанням методів штучного інтелекту. Цей підхід є потужним інструментом для підтримки користувача у взаємодії та маніпуляціях об’єктами у 3D-просторі без необхідності тримати фізичні сенсорні пристрої. Однак ще однією технологією управління віртуальними об’єктами є Eye-tracking системи, які є одним з інноваційних способів взаємодії користувача з віртуальною реальністю на підставі вимірювання рухів очей користувача і використовують цю інформацію для визначення точки фокусу погляду.

2.5 VR та технології Eye-tracking

Використання Eye-tracking технології у VR відкривають нові можливості для покращення взаємодії користувачів з віртуальними середовищами. Eye-tracking може використовуватись для визначення точки фокусування користувача у віртуальному середовищі. Наприклад, якщо користувач дивиться на певний об’єкт або місце, система може реагувати на це і взаємодіяти з цим об’єктом, відображати додаткову інформацію або виконувати певні дії. Це значно покращує взаємодію користувача з віртуальним середовищем, забезпечує більш реалістичний підхід та зручність взаємодії.

Також технологія Eye-tracking знаходить своє використання при забезпеченні взаємодії користувача з віртуальними персонажами або іншими користувачами у віртуальному середовищі. Система, що відстежує напрям погляду користувача, коли він спілкується з віртуальним персонажем, дозволяє персонажу реагувати на погляд користувача, наприклад, рухатися в напрямку його погляду або взаємодіяти з об'єктами, на які він дивиться.

Eye-tracking також може знайти використання для навігації та управління віртуальним середовищем. Наприклад, користувач може обирати напрямок руху, дивлячись на певні об'єкти або місця, або взаємодіяти з інтерфейсом користувацького середовища шляхом руху очей, вибираючи меню, або виконуючи певні команди за допомогою погляду. Це дозволяє забезпечити більш інтуїтивний спосіб навігації та управління віртуальним середовищем, особливо у тих випадках, коли використання рук або контролерів може бути обмеженим.

Також Eye-tracking може допомогти взаємодіяти з докільцями віртуального середовища. Наприклад, система може відстежувати погляд користувача на об'єкти докільця, такі як предмети або двері, і реагувати на це, наприклад, відкриваючи двері, коли користувач дивиться на них, або взаємодіючи із предметами, на які користувач спрямовує свій погляд.

Зараз технологія відстеження напрямку зору використовується для аналізу поведінки користувачів у віртуальному середовищі – розробники використовують результати трекінгу очей для вивчення уваги користувачів, оцінки ефективності інтерфейсу користувацького досвіду, виявлення патернів взаємодії або для досліджень у галузі психології та когнітивних наук.

Ці застосування Eye-tracking технології для віртуальної реальності можуть покращити взаємодію користувачів з віртуальними середовищами, забезпечити більш реалістичний та інтуїтивний досвід, а також відкрити нові можливості для дослідження та аналітики. Саме тому в рамках написання кваліфікаційної роботи магістра увагу приділено використанню Eye-tracking технології для управління віртуальними об'єктами.

3 ПРИНЦИПИ КЕРУВАННЯ ПОГЛЯДОМ

3.1. Опис проблеми

Сучасні системи для Eye-tracking'a використовують різні методи отримання даних про рухи очей. Перший метод – це застосування контактних систем, які потребують фіксації датчиків безпосередньо на органах людини. Зазвичай це досягається за допомогою присоски, яка прикріплюється до ока і з'єднується із пристроєм, що реєструє рухи. Зараз контактні системи найчастіше використовуються в медичних дослідженнях; існують два типи таких систем: з присоскою та із лінзою. Лінза, яку розміщують безпосередньо на оці, може мати дзеркальну поверхню, яка дозволяє визначати рухи очного яблука за відбитим від неї світлом. Іноді до лінзи додають мініатюрну котушку індуктивності, а випробувана особа поміщається у високочастотне магнітне поле, що дозволяє отримати більш точні дані про положення ока.

Другий метод полягає в застосуванні безконтактних оптичних систем. Ця технологія передбачає використання цифрового відеозапису руху одного обраного ока. Зазвичай такий запис ведеться за допомогою спеціального обладнання в інфрачервоному діапазоні, що дозволяє освітлювати око досліджуваної особи без засліплення його. Для визначення напрямку погляду використовується аналіз взаємного розташування променів інфрачервоного світла, відбитих від різних частин очного яблука. Цей метод називається методом зображень Пуркіньє. За допомогою процедур калібрування, виконаних за допомогою пристроїв для eye-tracking, можна вимірювати точку, на яку спрямована зірка глядача на площині, на якій відображаються калібрувальні точки, розташовані перпендикулярно до площини.

Для розрізнення руху очей від рухів голови необхідно використовувати дві точки. Коли очі рухаються звичайним чином, то позиційна різниця між центром зіниці та відбиттям рогівки змінюється, але залишається майже незмінною при незначних рухах голови. На рисунку 3.1 показані відносні положення відбитків зіниці та першого Пуркіньє. Цей рисунок відображає око, яке по черзі спрямовується на дев'ять калібрувальних точок, розташованих у різних частинах

поля зору. На цьому рисунку відбиток Пуркінє зображено як маленьке біле коло поруч зі зіницею, яка позначена чорним колом. Оскільки джерело світла, яке використовується під час дослідження, зазвичай залишається у фіксованому положенні щодо розташування ока, то зображення Пуркінє залишається відносно стабільним, тоді як саме око, а отже і зіниця, обертаються навколо орбіти [2].

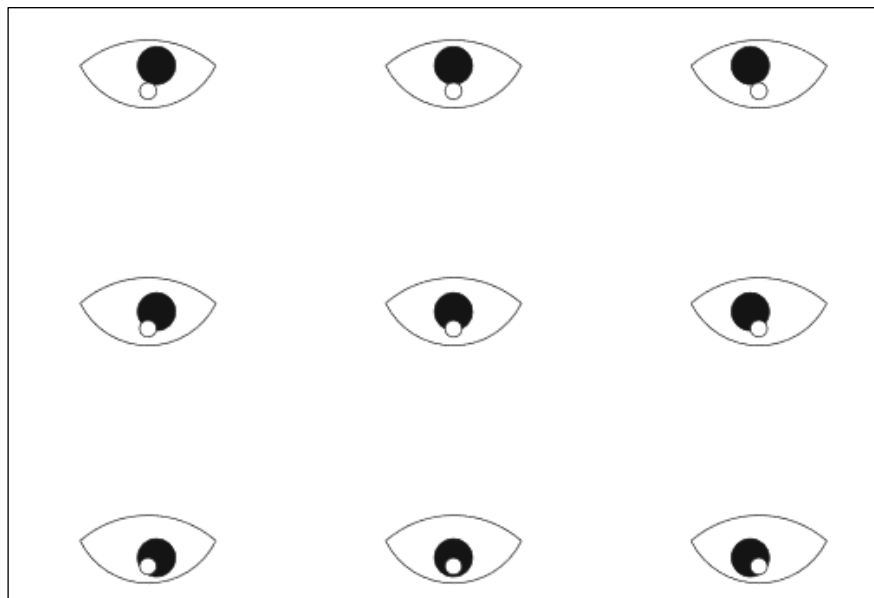


Рисунок 3.1 – Відносне розташування зображень зіниці та першого Пуркінє

Однак для реалізації перерахованих вище технічних рішень необхідне спеціальне обладнання та лабораторні умови. У зв'язку з цим більш актуальним є задача розробки такої системи керування за допомогою відстеження напряму погляду, яка б їх не вимагала. Цифрові відеокамери та мобільні телефони широко доступні на поточний момент, крім того розробники апаратного забезпечення досить часто використовують бібліотеки доповненої реальності, що дозволяє використовувати їх у якості пристроїв для реєстрації напряму зору. В якості об'єкта управління, що керується шляхом фокусування людського зору, під час досліджень було обрано спеціальні тестові об'єкти.

3.2 Аналіз алгоритмів відстеження напрямку зору

За допомогою сучасних камер та сенсорів пристрої можуть досить 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 віртуальної реальності пристрою (рис. 3.2).

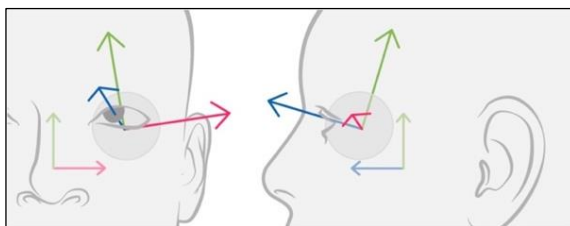


Рисунок 3.2 – Модель голови користувача у просторі координат віртуальної реальності

Одним із способів реалізації eye-tracking технології є використання веб-камери, яка може бути підключеної до комп'ютеру або вбудованою у ноутбук, смартфон або інший пристрій. Веб-камера може фіксувати рухи очей користувача та аналізувати їх, щоб визначити точку фокусування, напрямок погляду та інші параметри взаємодії з екраном. Використання веб-камери для eye-tracking має кілька переваг, таких як низька вартість, широка доступність вбудованих веб-камер у багатьох пристроях та зручність їх використання.

Керування віртуальними об'єктами за допомогою eye-tracking з використанням веб-камери базується на спостереженні рухів очей користувача та визначенні напрямку його погляду. Для цього можуть бути використані різні алгоритми та методи обробки відеоінформації з веб-камери. Основні принципи керування віртуальними об'єктами з використанням веб-камери включають:

- визначення позиції погляду: з використанням веб-камери можна визначити позицію погляду користувача на екрані. Алгоритми комп'ютерного зору можуть аналізувати відео з веб-камери та виявляти місце, на яке дивиться користувач. Це може бути використано для

визначення точного місця на екрані, на яке спрямований погляд користувача;

- визначення дій користувача: відстеження рухів очей може допомогти визначити дії користувача. Наприклад, можна визначити, коли користувач фіксує погляд на певному об'єкті, відводить погляд від екрана, робить рухи головою або моргає. Ці дії можуть бути використані для взаємодії з віртуальними об'єктами, такими як вибір, пересування, збільшення/зменшення об'єктів та інші дії;
- відслідковування руху очей: з використанням веб-камери можна відслідковувати рухи очей користувача, такі як рухи зліва направо, зверху вниз, кругові рухи та інші. Ці рухи можуть бути використані для керування рухом віртуальних об'єктів;
- інтерпретація результатів: після отримання даних про рухи очей користувача, необхідно їх інтерпретувати та визначити дії, які користувач хоче виконати. Наприклад, якщо користувач фіксує погляд на певному об'єкті на екрані, це може означати, що він хоче вибрати цей об'єкт. Якщо користувач робить рух вгору, це може вказувати на бажання прокрутити екран вгору. Визначення дій користувача на основі рухів очей може бути різним в залежності від конкретної системи та її налаштувань;
- взаємодія з віртуальними об'єктами: після інтерпретації результатів eye-tracking, система може взаємодіяти з віртуальними об'єктами відповідно до дій користувача. Наприклад, система може виконати команду вибору об'єкта, змінити його розташування, змінити розмір, виконати додаткові дії та інші. Це може дозволити користувачу ефективно керувати віртуальними об'єктами без використання додаткових управляючих елементів, таких як миша або клавіатура.

Інформацію про розміщення обличчя можна представити у вигляді 4x4 матриці перетворення. Ця матриця визначає переміщення (перенос), обертання та масштабування обличчя користувача в реальному світі відносно просторових

координат віртуальної реальності [20]. Крім того, інформацію про особливості обличчя, такі як рот, ніс, брови тощо можна представити у вигляді словника з ключами та значеннями. Кожен ключ відповідає певній частині обличчя, а кожне значення в діапазоні від 0 до 1 визначає ступінь зміни цієї частини обличчя. Ця інформація може бути відображена у формі 3D-сітки обличчя, що дозволяє більш детально відобразити його структуру та особливості. Такий підхід до представлення інформації дозволяє більш точно моделювати та відтворювати обличчя в віртуальному середовищі (рис. 3.3).

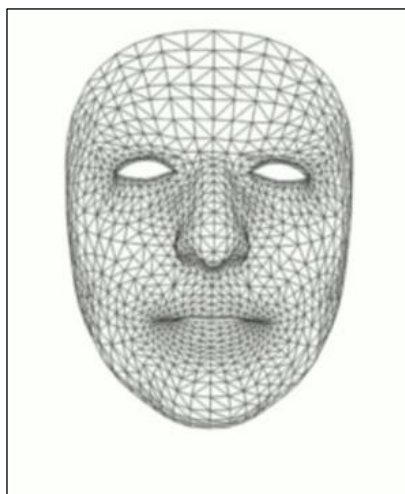


Рисунок 3.3 – Представлення обличчя у вигляді 3D-сітки

Завдяки наявності інформації про положення та трансформацію обличчя користувача, ми можемо наблизитися до визначення розташування очей. Використовуючи інструменти комп'ютерного зору, зокрема фреймворк Vision, який оперує алгоритмами комп'ютерного зору для обробки зображень і відеозаписів, ми можемо шукати зірки (зіниці) в очах. Існує різноманітні методи пошуку зірок, проте для досягнення нашої мети, найефективнішим вважається алгоритм СНТ (Circle Hough Transform). Цей алгоритм використовує перетворення Хафа для виявлення кругових об'єктів на зображенні, а оскільки радіус очей людини становить приблизно 12 мм, він може легко передбачити їх розташування, використовуючи інформацію про сітку обличчя та його положення. Щоб краще проілюструвати цей процес, можна відобразити проекцію сфери очей та зіниць на модель обличчя на рисунку (див.рис.3.4), щоб

візуалізувати ці аспекти більш наглядно. Використання веб-камер і комп'ютерного зору відкриває широкі можливості для розробки різноманітних застосувань, які використовують розпізнавання обличчя та відстеження очей для взаємодії з комп'ютерними системами, доповненою та віртуальною реальністю, безпеки та інших сфер [21].

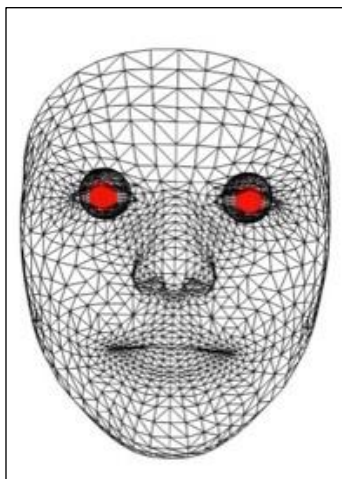


Рисунок 3.4 – Проекція очей та зіниць на 3D-сітці

Знання про положення та орієнтацію очей дозволяє обчислити вектори погляду, які починаються у центрі сфери ока і проходять через центр сфери зіниці. Ці вектори є важливими для визначення напрямку погляду користувача. Використовуючи ці вектори, можна побудувати рівняння, які описують лінії променів погляду в просторі. Ці рівняння можна використовувати для знаходження точок перетину цих променів з площиною або поверхнею пристрою, з яким взаємодіє користувач. (рис. 3.5).

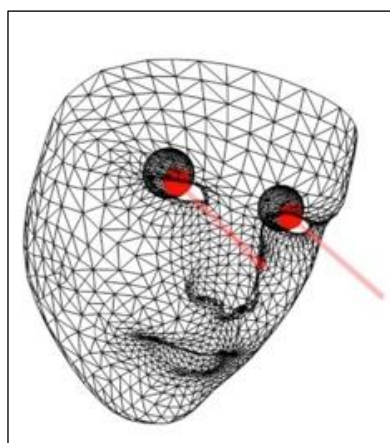


Рисунок 3.5 – Проекція векторів погляду на 3D-сітці

Цей підхід дозволяє використовувати дані про погляд користувача для керування різними системами та додатками. Наприклад, можна використовувати ці дані для інтерактивного керування інтерфейсом, переміщення курсора на екрані, вибір об'єктів або навіть відтворення контенту, що відповідає погляду користувача.

3.3 Геометрична модель

Мета геометричної моделі полягає у визначенні точки, на яку спрямований погляд користувача на екрані комп'ютера за допомогою променевих зліпків. Зліпок променів формується шляхом випромінювання променів з центру очного яблука через зіницю. Точки, де ці промені перетинають площину екрану, вважаються точками погляду. Перетин прямої лінії з площиною можна визначити за допомогою трьох точок на площині екрану, центру очного яблука та зіниці, всі вони обраховуються у тривимірній системі координат. Приймаючи центр очного яблука за c , який належить множині R^3 ($c \in R^3$) та центр зіниці за p ($p \in R^3$), то точки на лінії, що проходить через центр очного яблука, можна описати виразом:

$$c + (p - c)t > 0 \quad (3.1)$$

де $t \in R^3$ та визначає зсув точки на прямій.

Точки на площині екрану можна описати трьома кутами екрану: $tl, tr, br \in R^3$, один з яких використовується в якості опорної точки, а два інших визначають напрямки, в які простирається площина екрану в тривимірному просторі:

$$tl + (tr - tl)u + (br - tl)v, \quad (3.3)$$

де $u, v \in R^3$ – точки зсуву на площині.

Точка перетину рівнянь (3.2) і (3.3) має наступний вигляд (рівняння 3.4 – 3.7):

$$c + (p - c)t = tl + (tr - tl)u + (br - tl)v \quad (3.4)$$

$$c + (p - c)t - tl = (tr - tl)u + (br - tl)v \quad (3.5)$$

$$c - tl = -(p - c)t + (tr - tl)u + (br - tl)v \quad (3.6)$$

$$c - tl = (c - p)t + (tr - tl)u + (br - tl)v \quad (3.7)$$

У більш стислій матричній та векторній формі рівняння (3.1 – 3.6) можна записати наступним чином (формула 3.8):

$$\begin{pmatrix} c_x - tl_x \\ c_y - tl_y \\ c_z - tl_z \end{pmatrix} = \begin{pmatrix} c_x - p_x & tr_x - tl_x & br_x - tl_x \\ c_y - p_y & tr_y - tl_y & br_y - tl_y \\ c_z - p_z & tr_z - tl_z & br_z - tl_z \end{pmatrix} \begin{pmatrix} t \\ u \\ v \end{pmatrix} \quad (3.8)$$

Щоб знайти точку перетину, це матричне рівняння необхідно розв'язати для t , щоб точку перетину можна було обчислити, підставивши t у рівняння (3.1) для кожного з очей незалежно один від одного. Це можна зробити, інвертуючи матрицю 3×3 до тих пір, поки лінія не стане паралельною площині зору, тобто до тих пір, поки вектор лінії погляду залишається лінійним незалежно від векторів напрямку. Складність полягає в тому, щоб знайти всі змінні, щоб рівняння (3.7) можна було розв'язати. Це відбувається на підставі визначення центрів очних яблук за допомогою загальної 3D-моделі голови, про що було сказано раніше. Після того, як виявлено розташування голови відносно камери, оцінюється відстань, щоб розрахувати положення екрану в системі координат моделі. Центри очних яблук, розташування зіниць у координатах моделі та кути екрану в координатах моделі можна підставити в рівняння (3.7), щоб знайти t і обчислити точки погляду. Зрештою, точки погляду перетворюються на цільові координати, наприклад, пікселі екрану.

Перш ніж знайти центри зіниць для проектування їх на 3D-модель, необхідно знайти та виділити обличчя об'єкту. На теперішній час існують понад 15 різних бібліотек комп'ютерного зору, що забезпечують певну функцію

розпізнавання облич. Одним із найбільш поширених методів є використання попередньо навчених класифікаторів OpenCV, які використовують для виявлення облич варіант виявлення ознак Хаара за допомогою AdaBoost [21]. Але цей метод, хоч і популярний, але дозволяє знаходити лише межі обличчя та очей, тоді як Dlib знаходить також ключові точки обличчя (рис.3.6).

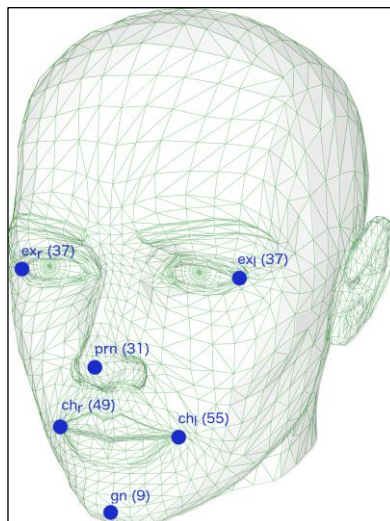


Рисунок 3.6 – Візуалізація ключових точок на 3D-моделі голови

З погляду фізики, зіниця є майже абсолютно чорним тілом, тобто відношення відбитого і поглиненого світлових потоків близько до нуля. Це забезпечує сильний контраст між областями зіниці та райдужної оболонки у кадрі. В залежності від умов освітлення, налаштувань балансу білого у відеокамери та кольору райдужки очного яблука випробуваного контраст є різним на різних колірних каналах.

Щоб відокремити рухи очей від рухів голови, необхідно отримати дві точки відліку. Коли рогівкове відображення джерела світла фіксується в одному положенні, яке не змінюється, обертання ока створює позиційну різницю між цим рогівковим відображенням і центром зіниці. Рухи голови при цьому незначно впливають на різницю позицій.

У реальному світі «образ» описується функцією двох змінних $f(x, y)$, де: пара (x, y) є координатами на зображенні; значення функції $f(x, y)$ є безперервною функцією у часі та просторі та може позначати, наприклад, яскравість, спектр випромінювання, вартість тощо.

При обробці зображень «зображення» описується дискретною функцією двох змінних $f(i, j)$, де: пара (i, j) є дискретними координатами на зображенні; значення функції $f(i, j)$ являє собою, наприклад, значення яскравості, вектор інтенсивності певного спектру випромінювання, а її функціональна вартість має назву апіксель. Ця функція є дискретною у часі, просторі та вартості.

Значення для кожного пікселя зображення можуть бути представлені двійковим значенням, цілим числом або вектором. За типом даних пікселі зображення можна поділити на:

- бінарне зображення, яке також називають дворівневим зображенням;
- зображення за градаціями сірого;
- кольорові зображення;
- спектральні зображення.

Обробку зображень і розпізнавання зображень реального світу можна поділити на декілька основних етапів (цей розподіл не є постійним):

- сканування та оцифрування зображень;
- попередня обробка;
- аналіз і сегментація зображення або коригування зображення;
- опис об'єктів;
- класифікація.

3.4 Методи безконтактного керування з використанням веб-камери

Одним з найбільш поширених методів безконтактного керування об'єктами залишається пересування курсора за допомогою погляду. Цей метод дозволяє користувачам переміщувати курсор на екрані віртуального середовища за допомогою погляду. Веб-камера розпізнає позицію очей користувача і переміщує курсор на відповідну позицію на екрані. Разом з тим існують додаткові методи керування поглядом, що підвищують ефективність їх використання.

Розпізнавання жестів за допомогою погляду. Користувач може виконувати різні жести, наприклад, моргання або моргання визначеним чином, які

розпізнаються програмним забезпеченням веб-камери і використовуються як команди для керування віртуальними об'єктами.

Вибір об'єктів за допомогою погляду. Користувач може вибирати віртуальні об'єкти, дивлячись на них тривалий час. Наприклад, якщо користувач дивиться на віртуальний об'єкт протягом декількох секунд, це може бути сприйнято як вибір цього об'єкта (рис.3.7).

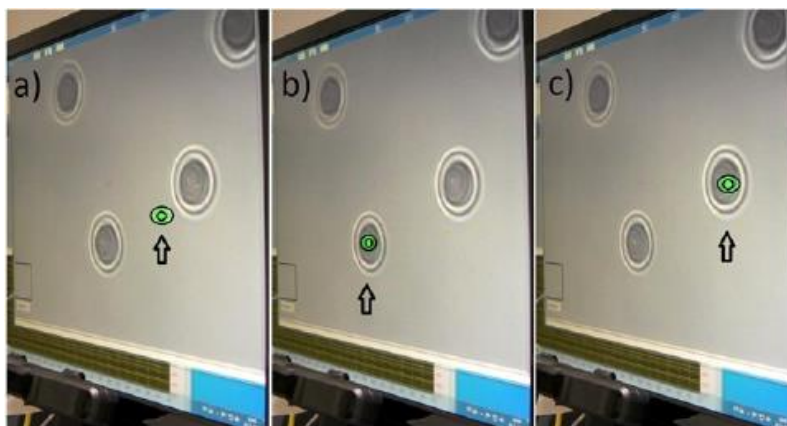


Рисунок 3.7 – Приклад вибору об'єкту поглядом

Взаємодія за допомогою фіксації погляду. Користувач може виконувати різні команди або дії, фіксуючи свій погляд на відповідних областях екрану протягом визначеного проміжку часу. Наприклад, фіксація погляду на кнопці може активувати її дію (рис.3.8).

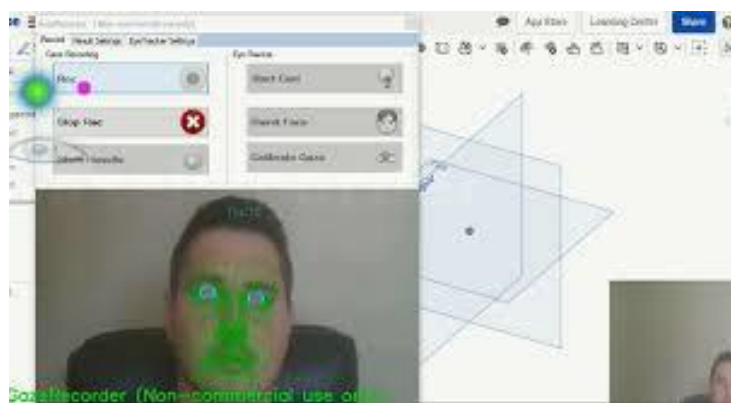


Рисунок 3.8 – Приклад активації дії поглядом

Гестурний контроль. Користувач може виконувати різні жестури або рухи головою, які розпізнаються веб-камерою і використовуються для керування віртуальними об'єктами. Наприклад, нахил голови вліво або вправо може призводити до руху віртуального об'єкта у відповідному напрямку.

Комбінації різних методів. Досить часто використовуються комбінації різних методів eye-tracking технології для більш точного та ефективного керування віртуальними об'єктами. Наприклад, функція пересування курсора за допомогою погляду може бути поєднана з функцією фіксації погляду на елементах інтерфейсу та жестурним контролем, що може забезпечити більш широкий спектр можливих взаємодій користувача з віртуальним середовищем.

3.5 Існуючі програмні рішення

До практичних розробок в галузі безконтактного керування віртуальними об'єктами за допомогою веб-камери та розпізнавання руху очей слід віднести наступні:

GazePlay (<https://gazeplay.github.io/GazePlay/>) – ігрова платформа, яка використовує технологію eye-tracking для керування віртуальними іграми шляхом відстеження взгляду користувача та перетворення його на рухи персонажів гри.

EyeMouse (<https://github.com/Jolton/EyeMouse>) – це програмне забезпечення, яке використовує eye-tracking технологію для керування курсором миші на екрані комп'ютера на основі рухів очей користувача. Воно дозволяє виконувати різноманітні дії, такі як вибір, перетягування, клік та інші лише за допомогою рухів очима.

У дослідженні також було використано бібліотеку OpenCV для розпізнавання напрямку погляду людини. OpenCV (<https://opencv.org/opencv-face-recognition/>) є потужним інструментом комп'ютерного зору, який надає широкий набір функцій для обробки зображень та аналізу даних з веб-камери (див. рис.3.9).

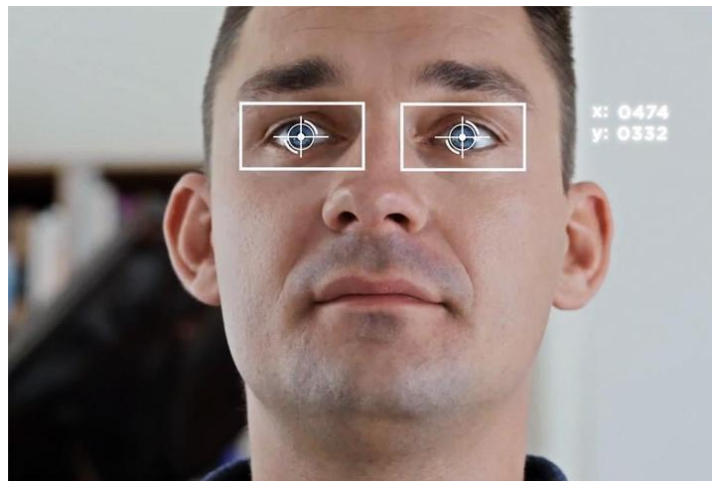


Рисунок 3.9 – Розпізнавання зіниць за допомогою OpenCV

Tobii EyeX (<https://gaming.tobii.com/>) – це ще одна дуже популярна система eye-tracking, яка використовує зовнішню камеру для керування віртуальними об'єктами. Вона має декілька варіантів використання, таких як відстеження погляду, використання рухів очей як інтерфейсних елементів та інші, однак вимагає додаткового обладнання (рис.3.10).

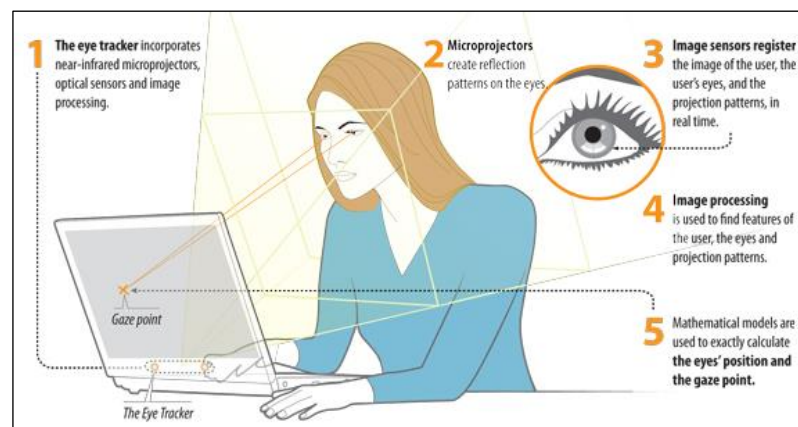


Рисунок 3.10 – Використання Tobii EyeX

GazeRecorder (<https://gazerecorder.com/gazerecorder/>) – це безкоштовна програма, яка використовує веб-камеру для відстеження рухів очей користувача та їх використання для керування віртуальними об'єктами. Вона дозволяє користувачам взаємодіяти з комп'ютером за допомогою рухів очей, таких як вибір, клік, перетягування та інші. Головна проблема полягає у реалізації за

допомогою клієнт-серверній архітектурі – розпізнавання відбувається на хмарному сервісі, що накладає певні обмеження на використання (рис.3.11).

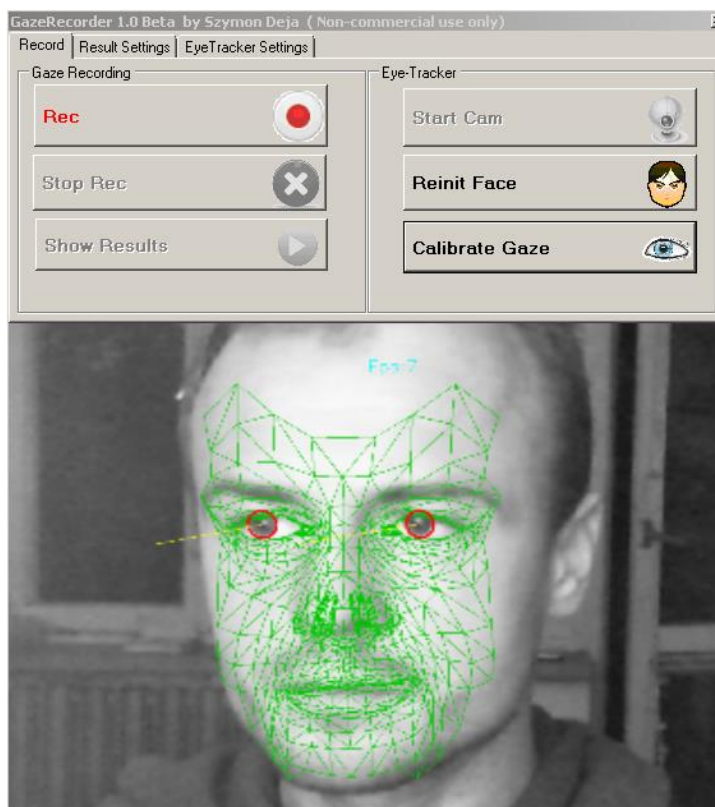


Рисунок 3.11 – Інтерфейс GazeRecorder

Windows Eye Control – це вбудований в операційну систему Windows інструмент для безконтактного керування віртуальними об'єктами за допомогою eye-tracking технології із використанням зовнішніх контролерів. Вона дозволяє користувачам виконувати різні дії за допомогою рухів очима (рис.3.12).



Рисунок 3.11 – Windows Eye Control у середовищі Windows 11

Ці та деякі інші існуючі методи безконтактного керування віртуальними об'єктами з використанням технології eye-tracking мають свої особливості та застосування в різних сферах. Вони можуть бути використані в різних сферах, таких як віртуальна реальність, віртуальна медицина, віртуальна навігація та інші, та дозволяють користувачам взаємодіяти з віртуальними об'єктами за рахунок рухів очей, розпізнавання міміки обличчя та інші.

3.6 Загальні умови проведення досліджень

Враховуючи специфіку запропонованого дослідження, було обрано метод контрольованого експерименту. Базове середовище для проведення виконання має наступний набір характеристик:

- CPU: Intel Core i5-1135G7;
- RAM: 16 Гб;
- VRAM: 4 Гб;
- монітор SyncMaster 225bw (22”), роздільна здатність 1680x1050;
- веб-камера Logitech C615 HD;
- ОС: Windows 10.

Для проведення експериментів було обрано бібліотеки JavaScript GazeRecorder та WebGazer, які надають власний API для використання методів визначення напрямку погляду користувачів, які можуть бути застосовані для керування віртуальними об'єктами у веб-орієнтованому середовищі [12].

GazeRecorder API – це JavaScript-інтерфейс для взаємодії з веб-додатками і відстеження взаємодії користувача з веб-додатками за допомогою відстеження погляду (gaze tracking) і розпізнавання жестів очей (eye gestures). Він надає розширені можливості відстеження погляду, аналізу даних погляду, відслідковування жестів очей та інші функції для реалізації різноманітних сценаріїв взаємодії з користувачем з використанням звичайних веб-камер.

Використання GazeRecorder API базується на сервер-орієнтовній архітектурі: відеоряд з веб-камери доставляється на віддалений сервер для збереження і обробки даних погляду.

GazeRecorder використовує веб-камеру для відстеження та запису того, на яку частину екрану дивиться користувач, а потім створює теплову карту на основі цих даних. GazeRecorder досить точний, але потребує якісного калібрування.

WebGazer.js – це бібліотека JavaScript для використання технології eye-tracking у веб-додатках. Особливістю бібліотеки є той факт, що вона дозволяє виявляти та відстежувати рухи очей користувача за допомогою веб-камери та аналізувати цю інформацію для різних цілей, таких як взаємодія з інтерфейсом додатка або збір статистики про погляд користувача. Вона надає розширені можливості відстеження погляду, аналізу даних погляду, відслідковування жестів очей та інші функції, що можуть бути використані для реалізації різноманітних сценаріїв взаємодії з користувачем.

Загалом можна сказати, що обидві бібліотеки включають:

- відстеження погляду (gaze tracking): вони дозволяють виявляти та відстежувати рухи очей користувача за допомогою веб-камери, яка збирає дані про позицію погляду (координати на екрані), що дозволяє аналізувати та використовувати цю інформацію для взаємодії з додатком;
- розпізнавання жестів очима (eye gestures): API бібліотек надають можливість відстежувати різні жести очей, такі як моргання, відведення погляду, повільне закриття очей та інші, що дозволяє використовувати ці жести для реалізації взаємодії без дотику – для вибору елементів меню, керування функціональністю додатку та реалізації інших дій;
- калібрування погляду (gaze calibration): передбачає можливість виконувати калібрування погляду, тобто налаштування відстежування погляду для конкретного користувача, що використовує програмне забезпечення. Це дозволяє реалізувати більш точне визначення позиції погляду користувача на екрані в залежності від його фізіологічних особливостей, таких як розташування очей, розмір зіниць та інші параметри, що дозволяє досягти більш точного відстеження погляду та покращити якість взаємодії з користувачем;

- реал-тайм взаємодія: обидві бібліотеки дозволяють організувати взаємодію з веб-додатком в реальному часі та ці дані можуть бути використані для миттєвої відповіді на дії користувача, динамічного змінювання веб-інтерфейсу, адаптації функціональності додатку до потреб користувача та інших варіантів взаємодії в реальному часі;
- кросплатформеність: обидві бібліотеки можуть використовуватись на різних платформах, включаючи комп'ютери, планшети, смартфони та інші пристрої, що дозволяє забезпечити єдність досвіду взаємодії незалежно від використаної платформи;
- інтеграція з іншими технологіями: вони можуть бути інтегровані з іншими технологіями та сервісами, такими як машинне навчання, розпізнавання обличчя, розпізнавання мови, розпізнавання жестів та інші, що відкриває широкі можливості для розробки різноманітних застосувань, таких як відеоігри, віртуальна реальність, аналітика поведінки користувачів, системи взаємодії зі смарт-пристроями та інші.

Перейдемо до аналізу використання систем розпізнавання напрямку погляду з метою визначення можливостей їх застосування для управління віртуальними об'єктами за допомогою погляду.

4 АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

4.1 Алгоритм відстеження очей і калібрування

Під час проведення дослідження систем розпізнавання напрямку погляду відеопотік із обличчям особи оброблявся в реальному часі на боці клієнта за допомогою бібліотек JavaScript WebGazer.js та GazeRecorderAPI.js.

Для досягнення мети дослідження обох систем необхідно було передбачити окрему процедуру калібрування [13]. Калібрування – це процедура стеження за поглядом, яка здійснюється за допомогою точок калібрування, що по черзі з'являються у фіксованих положеннях на екрані, по одній точці (рис.4.1).

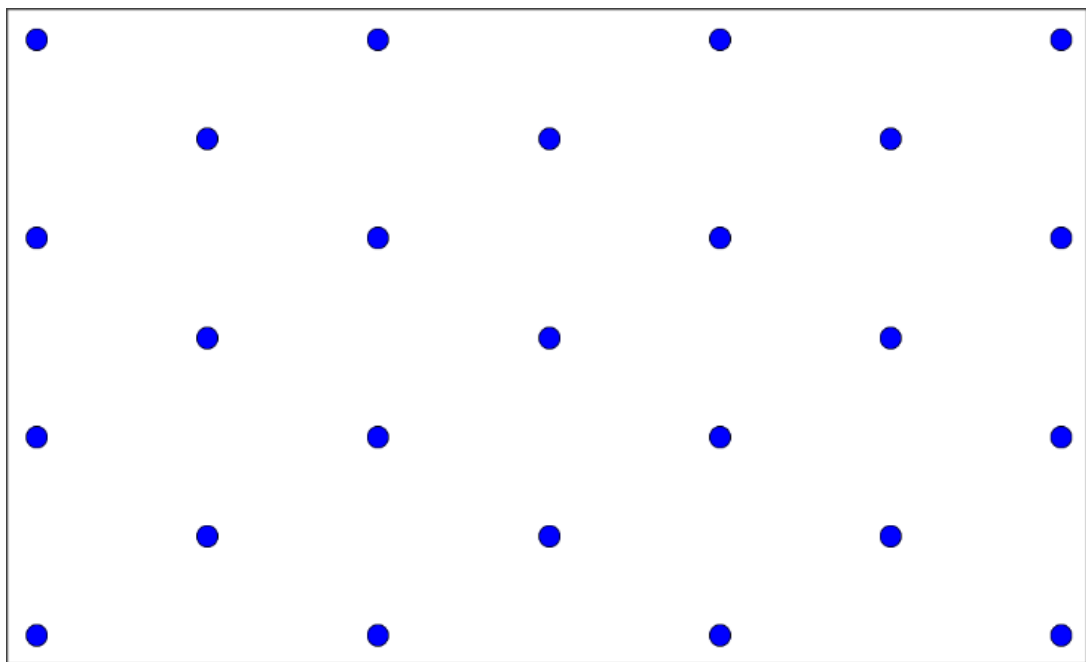


Рисунок 4.1 – Покриття точками калібрування області екрану

На підставі результатів спостереження за напрямком погляду для кожної точки формуються калібрувальні зразки з інтервалом у 100 мс, які завантажуються в алгоритм моделі.

Під час проведення калібрування висувались додаткові вимоги до положення голови користувача системи (див.рис.4.2). Протягом проведення експерименту зміна положення голови (вихід за встановлені межі) призводила до відхилення отриманих результатів [14].



Рисунок 4.2 – Інструкція щодо проведення калібрування

Опишемо процес калібрування для кожної з бібліотек [15].

4.1.1 Калібрування в GazeRecorderAPI

Процес проведення калібрування в GazeRecorder API передбачає наступні кроки:

- встановлення параметрів проведення процесу калібрування: об'єкт повинен знаходитись на відповідній відстані від камери, не мати предметів, що заважають спостереженню за очима та знаходитись у достатньо освітленому приміщенні (рис.4.3);

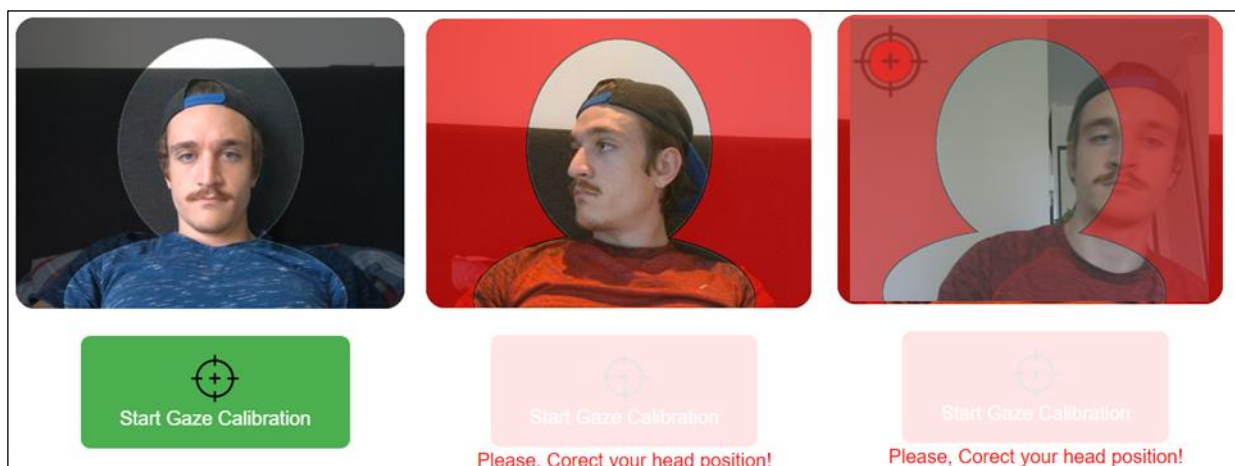


Рисунок4.3 – Інструкції щодо проведення калібрування

- проведення процедури калібрування: користувач послідовно фіксує погляд на об'єктах, що з'являються на екрані, затримуючи погляд на визначений проміжок часу (див.рис.4.4). ;



Рисунок 4.4 – Процедура калібрування GazeRecorder

- по завершенню процедури калібрування результати відразу стають доступними для використання у додатку.

Процес використання GazeRecorderAPI у власному додатку передбачає виконання наступних дій [17]:

- а) до тексту додатку необхідно включити наступний скрипт:

```
<script
src="https://app.gazerecorder.com/GazeRecorderAPI.js"></script>
```

- б) почати процедуру запису відеопослідовності з використанням веб-камери:

- 1) для використання на зовнішньому сайті:

```
GazeRecorderAPI.Rec(url)
```

- 2) для використання на поточній веб-сторінці:

```
GazeRecorderAPI.Rec()
```

- в) для завершення використання необхідно завершити процедуру запису:

```
GazeRecorderAPI.StopRec()
```

- г) зберегти результати сесії:

```
var SesionReplayData = GazeRecorderAPI.GetRecData();
```

Процедура розпізнавання напрямку погляду через хмарні сервіси забезпечується використанням наступного скрипту:

```
<script src="https://api.gazerecorder.com/GazeCloudAPI.js" ></script>
```

Використання розпізнавання відбувається за допомогою функції `GazeCloudAPI.StartEyeTracking()`, виклик якої необхідно включити до коду програми:

```
GazeCloudAPI.OnResult = function (GazeData)
{ GazeData.state // 0 : valid gaze data;
                    -1 : face tracking lost,
                    1 : gaze data uncalibrated
  GazeData.docX // gaze x in document coordinates
  GazeData.docY // gaze y in document coordinates
  GazeData.time // timestamp
}
```

Результат опрацювання за допомогою хмарного сервісу наведено на рис.4.6. Окрім поточних координат напрямку погляду на екрані відображається теплова карта результатів, на якій більш теплим кольором відображаються точки максимальної уваги.

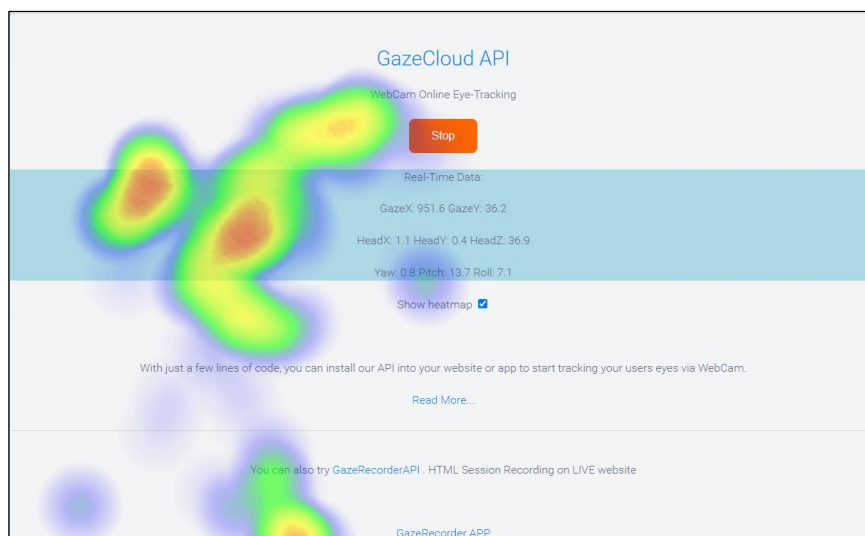


Рисунок 4.5 – Теплова карта GazeRecorder

Перейдемо до аналізу результатів використання бібліотеки WebGazer.

4.1.2 Калібрування в WebGazer

WebGazer [18] розроблений компанією Brown HCI group, ця бібліотека достатньо проста у використанні і може бути налаштована відповідно до обраного сценарію використання. Підключення бібліотеки WebGazer.js до веб-додатку відбувається шляхом додавання наступного тегу скрипту до HTML-сторінки:

```
<script src="webgazer.js"></script>
```

Після підключення бібліотеки її необхідно ініціалізувати за допомогою наступного коду:

```
webgazer.setRegression('ridge') // Вибір алгоритму для регресії
webgazer.setGazeListener(handleGazeData) // Встановлення обробника
даних погляду
webgazer.showPredictionPoints(true) // Відображення позначок
передбачених точок погляду
webgazer.begin() // Запуск відстежування погляду
```

Функція обробки даних погляду handleGazeData повертає дані про погляд, такі як координати (x, y) точки погляду на екрані:

```
function handleGazeData(gazeData) {
  var x = gazeData.x; // Координата x точки погляду
  var y = gazeData.y; // Координата y точки погляду
  // Додаткові дії з даними погляду
}
```

Процес калібрування дозволяє налаштувати систему eye-tracking для конкретного користувача, забезпечуючи більш точне відстеження погляду. З цією метою на екрані монітора відображаються калібрувальні точки, на яких користувач повинен фіксувати свій погляд, коли вона з'являється (рис.4.6).

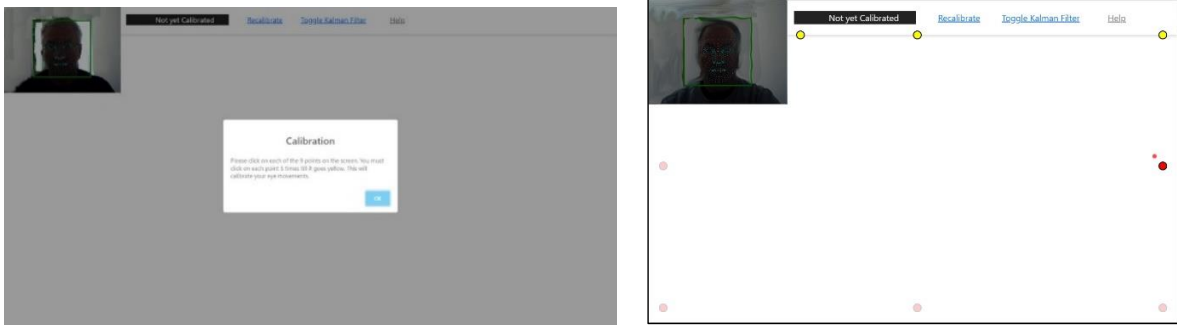


Рисунок 4.6 – Процедура калібрування WebGazer

В додатку виклик процедури калібрування відбувається за допомогою функції `webgazer.begin()`:

```
// виклик функції для початку калібрування
webgazer.begin();

// запуск калібрування при кліку на кнопку
document.getElementById('calibrate-button').addEventListener('click',
function() {
    // зупинити відстеження погляду під час калібрування
    webgazer.pause();
    // викликати функцію калібрування
    webgazer.setGazeListener(calibrationPointCallback).begin();
});

// функція зворотного виклику для отримання координат точки
калібрування
function calibrationPointCallback(x, y) {
    // зберігати координати точки калібрування
    calibrationPoints.push({x: x, y: y});

    // перевірити, чи вже зібрано достатньо точок для калібрування
    if (calibrationPoints.length === 9) {
        // зупинити калібрування
        webgazer.stop();
        // викликати функцію, що обробляє дані калібрування
        analyzeCalibrationData();
    } else {
        // викликати функцію для показу наступної точки калібрування
        showNextCalibrationPoint();
    }
}

// функція для показу наступної точки калібрування
function showNextCalibrationPoint() {
    // отримати наступну точку калібрування з масиву точок калібрування
    var point = calibrationPoints[calibrationPoints.length - 1];

    // встановити положення маркера на точці калібрування
    var marker = document.getElementById('calibration-marker');
    marker.style.left = point.x + 'px';
```

```

marker.style.top = point.y + 'px';

// показати інструкцію користувачеві
var instructions = document.getElementById('calibration-
instructions');
instructions.innerHTML = 'Дивіться на маркер. Натисніть
"Продовжити", коли будете готові.';
}

// функція, яка обробляє дані калібрування
function analyzeCalibrationData() {
// викликати функцію, що обчислює коефіцієнти для корекції позиції
курсора
var coefficients = webgazer.getRegressionCoefficients();

```

Для збереження результатів калібрування використовується функція `saveData()`. Ця функція зберігає дані погляду, отримані під час калібрування, у форматі JSON в локальному сховищі:

```

// Початок процесу калібрування
webgazer.begin();

// Завершення калібрування та збереження результатів
webgazer.end().then(function() {
var data = webgazer.getGazeData(); // отримати дані погляду
var jsonData = JSON.stringify(data); // перетворити в JSON формат
// Зберегти дані у файл
console.log(jsonData);
});

```

Після того, як користувач завершує калібрування, його погляд можна відслідковувати. Функція `getCurrentPrediction()` повертає координати погляду користувача:

```

// Запускаємо відслідковування погляду користувача
webgazer.begin();

// Зупиняємо відслідковування погляду користувача
webgazer.end();

// Отримуємо координати погляду користувача
var prediction = webgazer.getCurrentPrediction();
console.log(prediction);

```

Процес калібрування відбувається без ускладнень – користувачу необхідно декілька разів клацнути на кожне невеличке коло на екрані поки воно не змінить

свій колір з червоного на жовтий, після чого необхідно затримати погляд на цьому колі ще на 5 секунд. Під час калібрування поруч на екрані пересувається червона точка, яка демонструє розпізнаний в поточний момент часу напрям погляду. Користувач повинен постійно перевіряти, що його обличчя знаходиться у необхідному місці та відповідає вимогам роботи бібліотеки (контур зображення зелений). Томі можна зробити висновки, що необхідність постійно контролювати правильність розташування обличчя та наявність червоної позначки погляду під час калібрування значним чином зменшує зручність цієї процедури.

Перейдемо до аналізу отриманих результатів досліджень.

4.2 Порівняння результатів калібрування

Слід зазначити, що обидві бібліотеки не надають високої точності розпізнавання напрямку погляду. На рис.4.7 наведено результати використання систем розпізнавання напрямку погляду. Під час проведення дослідження курсор-хрестик – це місце на екрані монітору, куди насправді був спрямований погляд під час експерименту, а червона крапка – це результат обчислення напрямку погляду бібліотекою WebGazer з урахуванням результатів калібрування.

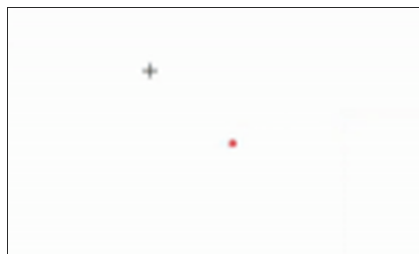


Рисунок 4.7 – Результат розпізнавання напрямку погляду

Результати 5 сесій калібрування та вимірювань відстані від реальної точки, на яку був спрямований погляд, та результатів, які демонструють бібліотеки, наведені у таблицях 4.1–4.2 та 4.3-4.4.

Таблиця 4.1 – Результати обчислення точності GazeRecorder API

Умови проведення досліджень	Точність калібрування (за даними бібліотеки)				Середнє значення
	1	2	3	4	
День, достатнє природнє освітлення	93,608654	76,44667	84,55952	89,37943	86,00
День, достатнє природнє освітлення, окуляри	78,823877	61,46756	50,79794	55,66349	61,69
Вечір, достатнє штучнє освітлення	81,54423	72,68442	85,68805	76,62112	79,13
Вечір, достатнє штучнє освітлення, окуляри	64,355126	72,25504	52,33541	49,36694	59,58
Вечір, недостатнє штучнє освітлення	86,926005	80,76895	76,83057	79,56888	81,02
Вечір, недостатнє штучнє освітлення, окуляри	61,775712	74,84666	75,21043	55,86713	66,92

Таблиця 4.2 – Результати обчислення точності WebGazer

Умови проведення досліджень	Точність калібрування (за даними бібліотеки)				Середнє значення
	1	2	3	4	
День, достатнє природнє освітлення	69	66	80	39	63,50
День, достатнє природнє освітлення, окуляри	72	49	35	30	46,50
Вечір, достатнє штучнє освітлення	81	78	68	69	74,00
Вечір, достатнє штучнє освітлення, окуляри	60	65	73	78	69,00
Вечір, недостатнє штучнє освітлення	45	56	60	56	54,25
Вечір, недостатнє штучнє освітлення, окуляри	86	57	50	41	58,50

Вимірювання відхилення обчисленої точки напрямку погляду від реальної, на яку був спрямований погляд, відбувалось шляхом зняття скріншотів у

випадковий проміжок часу, причому погляд був спрямований на центр екрану, на якому відображалась система координат з кроком в 1 мм (рис.4.8) [16].

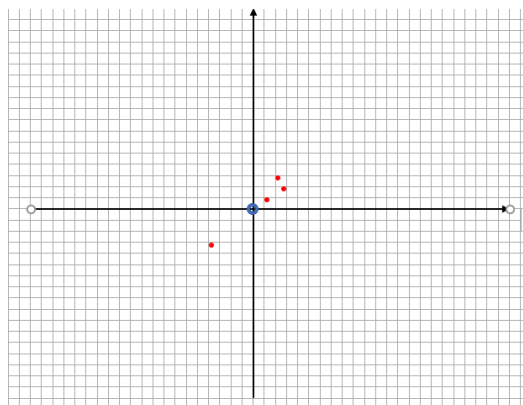


Рисунок 4.8 – Результати захоплення напрямку погляду

Результати вимірювань демонструють трохи кращу продуктивність з боку GazeRecorder API, яка надає менший розкид результатів.

Таблиця 4.3 – Результати вимірювання точності GazeRecorder API

Умови проведення досліджень	Похибка обчислення (за результатами вимірювання координат),%				Середнє значення,%
	1	2	3	4	
День, достатнє природне освітлення	15,0	12,0	18,0	22,0	16,75
День, достатнє природне освітлення, окуляри	18,0	16,0	22,0	27,0	20,75
Вечір, достатнє штучне освітлення	20,0	16	18,0	26,0	20,00
Вечір, достатнє штучне освітлення, окуляри	23,0	30,0	27,0	36,0	29,00
Вечір, недостатнє штучне освітлення	35,0	26,0	25,0	29,0	28,75
Вечір, недостатнє штучне освітлення, окуляри	35,0	38,0	36,0	24,0	33,25

Таблиця 4.4 – Результати вимірювання точності WebGazer

Умови проведення досліджень	Похибка обчислення (за результатами вимірювання координат),%				Середнє значення,%
	1	2	3	4	
День, достатнє природнє освітлення	20,0	22,0	28,0	12,0	20,50
День, достатнє природнє освітлення, окуляри	14,0	26,0	30,0	35,0	26,25
Вечір, достатнє штучнє освітлення	18,0	26,0	28,0	26,0	24,50
Вечір, достатнє штучнє освітлення, окуляри	20,0	28,0	32,0	35,0	28,75
Вечір, недостатнє штучнє освітлення	14,0	20,0	24,0	28,0	21,50
Вечір, недостатнє штучнє освітлення, окуляри	38,0	58,0	25,0	40,0	40,25

Що стосується використання процесора. Під час виконання тестів WebGazer займає до 19,1% ресурсів процесора, тоді як GazeRecorderAPI займає лише 3,1% під час запуску демонстрації в Chrome. Це пов'язано в першу чергу з тим, що WebGazer працює локально в браузері, а GazeRecorderAPI використовує хмарні сервіси.

4.3 Проведення експериментів та порівняння результатів

З метою визначення ефективності використання бібліотек JS для керування віртуальними об'єктами було обрано систему меню, результатом натискання на кнопки якого стає виведення на екран цифри, що відповідає пункту меню (рис.4.9). Дослідження проводились з різними розмірами меню [22]:

- збільшений розмір: меню розміром 1040x1500 пікселів, розмір клавiши 840x200;
- великий розмір: меню розміром 520x750 пікселів, розмір клавiши 420x100;

- нормальний розмір: меню розміром 260x370 пікселів, розмір клавiши 210x50;
- зменшений розмір: меню розміром 130x200 пікселів, розмір клавiши 105x25.



Рисунок 4.9 – Зразок вибору пункту меню поглядом:
а) система меню; б) результат натискання на кнопку «Другий»

Результати вимірювань будуть згруповані у таблицях 4.5–4.12: таблиці 4.5–4.6 містять результати опрацювання збільшеного меню; в таблицях 4.7–4.8 зосереджено результати опрацювання великого меню; таблиці 4.9–4.10 містять результати опрацювання меню нормального розміру; в таблицях 4.11–4.12 наведено результати опрацювання меню зменшеного розміру. Вимірювання відбувалось сетами по 10 спроб, час спрацювання пункту меню було визначено у 3 сек.

Таблиця 4.5 – Результати вимірювання точності GazeRecorder API для збільшеного розміру меню

Умови проведення досліджень	Відсоток спрацювання меню, %		
	Пункт меню співпадає	Пункт меню не співпадає	Не спрацював жодний з пунктів меню
День, достатнє природне освітлення	100,0	0,0	0,0
День, достатнє природне освітлення, окуляри	90,0	10,0	0,0
Вечір, достатнє штучне освітлення	100,0	0,0	0,0
Вечір, достатнє штучне освітлення, окуляри	90,0	10,0	0,0
Вечір, недостатнє штучне освітлення	80,0	20,0	0,0
Вечір, недостатнє штучне освітлення, окуляри	70,0	20,0	10,0

Таблиця 4.6 – Результати вимірювання точності WebGazer для збільшеного розміру меню

Умови проведення досліджень	Відсоток спрацювання меню, %		
	Пункт меню співпадає	Пункт меню не співпадає	Не спрацював жодний з пунктів меню
День, достатнє природне освітлення	100,0	0,0	0,0
День, достатнє природне освітлення, окуляри	90,0	10,0	0,0
Вечір, достатнє штучне освітлення	100,0	0,0	0,0
Вечір, достатнє штучне освітлення, окуляри	90,0	10,0	0,0
Вечір, недостатнє штучне освітлення	90,0	10,0	0,0
Вечір, недостатнє штучне освітлення, окуляри	70,0	30,0	0,0

За результатами проведеного дослідження можна зробити висновки, що обидва фреймворки демонструють приблизно однакові результати.

Таблиця 4.7 – Результати вимірювання точності GazeRecorder API для великого розміру меню

Умови проведення досліджень	Відсоток спрацювання меню, %		
	Пункт меню співпадає	Пункт меню не співпадає	Не спрацював жодний з пунктів меню
День, достатнє природне освітлення	100,0	0,0	0,0
День, достатнє природне освітлення, окуляри	90,0	10,0	0,0
Вечір, достатнє штучне освітлення	100,0	0,0	0,0
Вечір, достатнє штучне освітлення, окуляри	80,0	20,0	0,0
Вечір, недостатнє штучне освітлення	80,0	20,0	0,0
Вечір, недостатнє штучне освітлення, окуляри	70,0	20,0	10,0

Таблиця 4.8 – Результати вимірювання точності WebGazer для великого розміру меню

Умови проведення досліджень	Відсоток спрацювання меню, %		
	Пункт меню співпадає	Пункт меню не співпадає	Не спрацював жодний з пунктів меню
День, достатнє природне освітлення	100,0	0,0	0,0
День, достатнє природне освітлення, окуляри	100,0	0,0	0,0
Вечір, достатнє штучне освітлення	90,0	10,0	0,0
Вечір, достатнє штучне освітлення, окуляри	80,0	10,0	10,0
Вечір, недостатнє штучне освітлення	70,0	30,0	0,0
Вечір, недостатнє штучне освітлення, окуляри	60,0	30,0	10,0

За результатами аналізу вмісту таблицю 4.7–4.8 можна зробити висновки, що при зменшенні розміру WebGazer починає демонструвати трохи гірші результати.

Таблиця 4.9 – Результати вимірювання точності GazeRecorder API для нормального розміру меню

Умови проведення досліджень	Відсоток спрацювання меню, %		
	Пункт меню співпадає	Пункт меню не співпадає	Не спрацював жодний з пунктів меню
День, достатнє природне освітлення	100,0	0,0	0,0
День, достатнє природне освітлення, окуляри	80,0	20,0	0,0
Вечір, достатнє штучне освітлення	90,0	10,0	0,0
Вечір, достатнє штучне освітлення, окуляри	70,0	20,0	10,0
Вечір, недостатнє штучне освітлення	60,0	20,0	20,0
Вечір, недостатнє штучне освітлення, окуляри	50,0	30,0	20,0

Таблиця 4.10 – Результати вимірювання точності WebGazer для нормального розміру меню

Умови проведення досліджень	Відсоток спрацювання меню, %		
	Пункт меню співпадає	Пункт меню не співпадає	Не спрацював жодний з пунктів меню
День, достатнє природне освітлення	90,0	10,0	0,0
День, достатнє природне освітлення, окуляри	90,0	10,0	0,0
Вечір, достатнє штучне освітлення	100,0	0,0	0,0
Вечір, достатнє штучне освітлення, окуляри	80,0	0,0	20,0
Вечір, недостатнє штучне освітлення	60,0	30,0	10,0
Вечір, недостатнє штучне освітлення, окуляри	40,0	30,0	30,0

Аналіз таблиць 4.9–4.10 дозволяє зробити висновки, що при зменшенні розміру обидва фреймворки вже не гарантують 100% правильності результату.

Таблиця 4.11 – Результати вимірювання точності GazeRecorder API для зменшеного розміру меню

Умови проведення досліджень	Відсоток спрацювання меню, %		
	Пункт меню співпадає	Пункт меню не співпадає	Не спрацював жодний з пунктів меню
День, достатнє природне освітлення	80,0	20,0	0,0
День, достатнє природне освітлення, окуляри	70,0	10,0	20,0
Вечір, достатнє штучне освітлення	70,0	20,0	10,0
Вечір, достатнє штучне освітлення, окуляри	60,0	20,0	20,0
Вечір, недостатнє штучне освітлення	40,0	40,0	20,0
Вечір, недостатнє штучне освітлення, окуляри	40,0	30,0	30,0

Таблиця 4.12 – Результати вимірювання точності WebGazer для зменшеного розміру меню

Умови проведення досліджень	Відсоток спрацювання меню, %		
	Пункт меню співпадає	Пункт меню не співпадає	Не спрацював жодний з пунктів меню
День, достатнє природне освітлення	90,0	10,0	0,0
День, достатнє природне освітлення, окуляри	70,0	20,0	1,0
Вечір, достатнє штучне освітлення	80,0	20,0	0,0
Вечір, достатнє штучне освітлення, окуляри	40,0	10,0	30,0
Вечір, недостатнє штучне освітлення	40,0	20,0	20,0
Вечір, недостатнє штучне освітлення, окуляри	30,0	40,0	30,0

На підставі аналізу таблиць 4.11–4.12 можна зробити єдиний висновок – при значному зменшенні розміру пунктів меню жодний з фреймворків вже не може забезпечити правильності результату.

На підсумку проведеного дослідження необхідно зазначити, що застосування відстеження очей та напрямку погляду залежить від того, як вони будуть використовуватися. Якщо намір полягає просто в тому, щоб визначити, чи дивився респондент на певний квадрант екрану, то методи, які засновані на використанні веб-камери, цілком тут спрацьовують. Однак, якщо потрібна більша просторова або часова роздільна здатність, або стоїть питання щодо керування об'єктами, які визначаються певними характеристиками, то необхідні спеціалізовані пристрої стеження за очима. Саме тому на підсумку проведеного дослідження і в осяжному майбутньому можна наполегливо рекомендувати к використанню для керування складними об'єктами спеціальне обладнання для відстеження очей.

ВИСНОВКИ

У ході виконання досліджень, пов'язаних з віддаленим керуванням складними об'єктами за допомогою веб-камери було визначено проблематику, пов'язану з управлінням віртуальними об'єктами, проведено дослідження методів керування віртуальними об'єктами шляхом реалізації контролю за поглядом людини та виконані експериментальні дослідження.

Під час дослідження було проведено аналіз існуючих програмних рішень з використанням бібліотек JavaScript та визначено оптимальні рішення завдання відстеження погляду користувача без використання додаткового обладнання. У порівнянні із першим комерційним досвідом з використанням розширеної реальності у якості розважальних і ігрових продуктів цей метод теоретично може використаний для більш серйозних бізнес-рішень.

Під час виконання кваліфікаційної роботи магістра було запропоновано поєднати графічний інтерфейс користувача із штучною доповненою реальністю. Таке поєднання джерел інформації, на думку автора, надало змогу отримати найкращі результати та мінімізувати похибки.

У досліджуваних методах було визначено ряд важливих моментів, серед яких: швидкість алгоритмів аналізу погляду користувача, застосування технологій доповненої реальності та її зв'язок з віртуальними та реальними об'єктами, проблеми, пов'язані з калібруванням. Це надало можливість дослідити результати визначення точки фокусу погляду користувача з точки зору можливості її застосування для керування об'єктами.

В якості результату дослідження можна вказати на можливість керування віртуальними об'єктами шляхом спрямування погляду без використання додаткового обладнання лише для обмежених типів об'єктів, чий лінійні розміри досягають великих значень.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Puja P. Sorate, Gyankamal J. Chhajed. Computer Vision System for Eye Gaze Tracking. International Journal of Modern Computer Science (IJMCS) ISSN: 2320-7868 (Online) Volume 5, Issue 3, June, 2017. URL: https://www.researchgate.net/publication/353143772_Computer_Vision_System_for_Eye_Gaze_Tracking (дата звернення: 10.03.2023).
2. Orman1 Z., Battal A., Kemer E. A study on face, eye detection and gaze estimation. International Journal of Computer Science & Engineering Survey (IJCSES) Vol.2, No.3, August 2011. URL: <http://dx.doi.org/10.5121/ijcses.2011.2303> (дата звернення: 10.03.2023).
3. Duchowski A. T. Eye Tracking Methodology: Theory and Practice. Springer, 2007, 22. URL: <https://doi.org/10.1007/978-1-84628-609-4> (дата звернення: 10.03.2023).
4. Jones R.E. Fitts, P.M. and J.L Milton. Eye movements of aircraft pilots during instrument landing approaches. Aeronautical Engineering Review, 1950. URL: <https://psycnet.apa.org/record/1950-05519-001> (дата звернення: 10.03.2023).
5. Eye Tracking in Computing Education: DOI: 10.1145/2632320.2632344 URL: https://www.researchgate.net/publication/264574138_Eye_Tracking_in_Computing_Education (дата звернення: 22.03.2023).
6. He, L & Zhu, S. Virtual Reality Technology in Visual Design of Artistic Images: Analysis and Applications. Scientific Programming, 2022, pp.1-6. URL: <https://doi.org/10.1155/2022/2527623> (дата звернення: 02.02.2022).
7. Dionisio, J., Burns III, W.G & Gilbert, R. 3D Virtual worlds and the metaverse. ACM computing surveys, 45(3), pp.1–38. URL: https://digitalcommons.lmu.edu/cgi/viewcontent.cgi?article=1004&context=cs_fac (дата звернення: 07.03.2022).
8. Fernandez, A. The Metaverse: The Future Of Work. Forbes. (Online) 29th March. URL: <https://www.forbes.com/sites/forbesbusinesscouncil/2022/03/18/the-metaverse-the-future-of-work/?sh=2d2d9fe24267> (дата звернення: 17.03.2022).

9. Fortnite. Computer Game. URL: <https://www.fortnite.com>. (дата звернення: 15.03.2022).
10. Richard Szeliski. Computer Vision: Algorithms And Applications. Springer; 2nd ed. 2022.– 947 p.
11. Join us in a full recreation of New York Times Square for NYE in VRChat. VRChat Times Square New Years (назва з екрану) // URL: <https://imgur.com/a/k2oWi> (дата звернення: 15.03.2022).
12. Webgazer eye tracking solution implemented as jsPsych plugin. // [Назва з екрану] URL: https://tsgdoc.socsci.ru.nl/index.php/Webgazer_eye_tracking_solution_implemented_as_jsPsych_plugin (дата звернення: 15.03.2022).
13. A short Eye-Tracking Software Comparison // [Назва з екрану] URL: <https://gazerecorder.com/a-short-eye-tracking-software-comparison/> (дата звернення: 15.03.2022).
14. Webcam-Based Eye Tracking vs. an Eye Tracker [Pros & Cons] // [Назва з екрану] URL: <https://imotions.com/blog/learning/best-practice/webcam-eye-tracking-vs-an-eye-tracker/> (дата звернення: 15.03.2022).
15. Brolly, X.L.C. and Mulligan, J.B. Implicit calibration of a remote gaze tracker, Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop (CVPRW '04), Volume 8, pp. 134. https://www.researchgate.net/publication/264878015_A_Study_On_Face_Eye_Detection_And_Gaze_Estimation
16. Anuradha Kar, Peter Corcoran. A Review and Analysis of Eye-Gaze Estimation Systems, Algorithms and Performance Evaluation Methods in Consumer Platforms. Accepted for publication in IEEE access. DOI 10.1109/ACCESS.2017.2735633.
17. Online webcam-based eye tracking API // [Назва з екрану] URL: <https://gazerecorder.com/gazerecorder-api/> (дата звернення 20.03.2023)
18. WebGazer. Top Level API // [Назва з екрану] URL: <https://github.com/brownhci/WebGazer/wiki/Top-Level-API> (дата звернення 20.03.2023)

19. Bilous N. V., Rakova A. O. Reference points method for human head movements tracking. *Радиоэлектроника, информатика, управление*. - Запоріжжя: ЗНТУ, -2020 №3, pp.121-129 // DOI 10.15588/1607-3274-2020-3-11.

20. К.Л. Хрусталеv, Д.А. Кобеляцкий, А.О. Функендорф. Автоматизация процессов управления автономными робототехническими платформами на основе использования систем технического зрения. // *Технология приборостроения*. 2018, №1.

21. Радченко Г. С. Дослідження методів управління складними об'єктами шляхом реалізації контролю за поглядом людини : атестаційна робота здобувача вищої освіти, пояснювальна записка / Г. С. Радченко ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків, 2020. – 82 с.

22. Pastel, S., Chen, CH., Martin, L. et al. Comparison of gaze accuracy and precision in real-world and virtual reality. *Virtual Reality* 25, 175–189 (2021). <https://doi.org/10.1007/s10055-020-00449-3> (дата перегляду 28.04.2023)

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ЗА НАУКОВИМИ НАПРЯМАМИ
КЕРІВНИКА ТА НАУКОВЦІВ КАФЕДРИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ**

19. Bilous N. V., Rakova A. O. Reference points method for human head movements tracking. Радиоэлектроника, информатика, управление. - Запоріжжя: ЗНТУ,-2020 №3, pp.121-129 // DOI 10.15588/1607-3274-2020-3-11.

20. К.Л. Хрусталеv, Д.А. Кобеляцкий, А.О. Функендорф. Автоматизация процессов управления автономными робототехническими платформами на основе использования систем технического зрения. // Технология приборостроения. 2018, №1.