

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет Комп'ютерних наук
Кафедра Програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

другий (магістерський)

(рівень вищої освіти)

Дослідження моделей та методів комп'ютерного зору
для відстеження рухів людського тіла

Виконав:

Студент 2 курсу групи ІІЗМ-20-2

Кулигін А.С.

(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного
забезпечення

Тип програми Освітньо-наукова

Керівник к.т.н., доц. Мельнікова Р.В.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. Кафедри

_____ проф. Дудар З.В.

2022 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наукКафедра Програмної інженеріїРівень вищої освіти другий (магістерський)Спеціальність 121 – Інженерія програмного забезпеченняТип програми освітньо-наукова програмаОсвітня програма Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«__» _____ 202_ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

Студента Кулигіна Андрія Сергійовича

(прізвище, ім'я, по батькові)

1. Тема роботи: Дослідження моделей та методів комп'ютерного зору для відстеження рухів людського тіла.

затверджена наказом університету від « 24 » березня 2022 р. № 412Ст2. Термін подання студентом роботи до екзаменаційної комісії « 24 » травня 2022 р.

3. Вихідні дані до роботи: Технічне завдання, календарний план, методичні вказівки, моделі та методи відстеження рухів людського тіла

4. Перелік питань, що потрібно опрацювати в роботі: мета роботи, аналіз предметної галузі і постановка задачі, дослідження моделей та методів відстеження рухів людського тіла, розгляд сфери їх використання, програмна реалізація.

КАЛЕНДАРНИЙ ПЛАН-

Номер	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі	17.01.2022- 24.01.2022	<i>Виконано</i>
2	Огляд існуючих методів	25.01.2022- 31.01.2022	<i>Виконано</i>
3	Формування вимог, проектування та розробка ПЗ	01.02-2022- 07.02.2022	<i>Виконано</i>
4	Підготовка пояснювальної записки	08.02.2022- 13.05.2022	<i>Виконано</i>
5	Підготовка доповіді	20.05.2022	<i>Виконано</i>
6	Попередній захист	23.05.2022	<i>Виконано</i>
7	Нормоконтроль та перевірка на плагіат	14.05.2022	<i>Виконано</i>
	Рецензування	19.05.2022	<i>Виконано</i>
	Занесення диплому в електронний архів	19.05.2022	<i>Виконано</i>
	Допуск до захисту у зав. кафедри	23.05.2022	<i>Виконано</i>

Дата видачі завдання 17 січня 2022 р.

Судент _____

(підпис)

Керівник роботи _____

(підпис)

К.Т.Н., доц. Мельнікова Р.В.

(посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Кваліфікаційна робота магістра містить: 68 с., 13 рис., 2 табл., 17 джерел.

ВІДСТЕЖЕННЯ ЛЮДСЬКОГО ТІЛА, МОДЕЛІ, МЕТОДИ, КОМП'ЮТЕРНИЙ ЗІР, НЕЙРОННІ МЕРЕЖІ, ОККЛЮЗІЯ, ЗОБРАЖЕННЯ ГЛИБИНИЮ, UNITY ENGINE

Об'єкт дослідження – відстеження людського тіла на зображенні з пристрою з камерою.

Предмет дослідження – моделі та методи відстеження людського тіла.

Мета роботи – дослідження моделей та методів відстеження людського тіла для їх порівняння та виявлення оптимального використання.

Методи розробки базуються на таких технологіях як .Net, C#, Unity Engine, Barracuda.

У результаті роботи були розглянуті моделі та методи комп'ютерного зору для відстеження рухів людського тіла. Проведені експерименти для порівняння характеристик розглянутих методів. Розроблена програмна система для виконання різноманітних вправ шляхом повторення їх за відео з використанням відстеження рухів тіла для контролю правильності їх виконання.

HUMAN BODY TRACKING, MODELS, METHODS, COMPUTER VISION, NEURAL NETWORKS, OCCLUSION, DEPTH IMAGE, UNITY ENGINE

The research object is human body tracking within the image data received from device's camera.

The research subject is the models and the methods of human body tracking.

Objective of the study is research on models and methods of human body tracking to compare them and determine the optimal way of its use.

Development methods are based on such technologies as .Net, C#, Unity Engine, Barracuda.

As a result of the work, models and methods of computer vision for determination of human body movements were examined. Experiments were carried out to compare

the characteristics of the considered methods. The software system was developed for performing different kinds of exercises by repeating them after a video with the use of a body movement survey to control the correctness of their execution.

Я, Кулигін Андрій Сергійович, студент гр. ПЗм-20-2, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження моделей та методів комп'ютерного зору для відстеження рухів людського тіла», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIAr KhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений 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.....	8
1 Аналітичний огляд	10
1.1 Аналіз предметної галузі	10
1.2 Напрямок дослідження	11
1.3 Аналіз існуючих методів і алгоритмів	12
1.3.1 Відстеження тіла з датчиками.....	13
1.3.2 Метод відстеження положення людського тіла на основі згорткової нейронної мережі.....	15
1.3.3 Метод відстеження людського тіла з врахуванням оклюзії	16
1.3.4 Відстеження положення людського тіла на основі відфільтрованих ключових точок	17
2 Постановка задачі.....	19
3 Математична модель.....	21
3.1 Відстеження тіла VIVE Tracker	21
3.2 Метод згорткової нейронної мережі	23
3.3 Відстеження з врахуванням оклюзії.....	25
3.4 Метод відфільтрованих ключових точок.....	27
4 Проведення експерименту.....	31
4.1 Методологія експерименту	31
4.1 Специфікація програмного забезпечення.....	32
4.1 Проведення експерименту.....	33
4.1 Аналіз результатів експерименту	34
5 Опис розробленої програмної системи.....	38
5.1 Обґрунтування середовища розробки.....	38

5.1 Архітектура програмної системи	40
5.1 Інтерфейс користувача	43
5.1 Майбутні ітерації	46
6 Результати роботи	48
Висновки	51
Перелік джерел посилання	53
Додаток А Перелік джерел посилання на наукові дослідження кафедри	55
Додаток А Тези доповіді	56
Додаток Б Результат перевірки на плагіат	57
Додаток В Слайди презентації	58

ВСТУП

В наш час людям постійно потрібне використання різноманітних розумних технологій, що в свою чергу створює сприятливі умови для швидкого розвитку саме цих технологій та створення нових. Різні алгоритми обробки певної інформації вже набули великої кількості варіацій і кожна з них призначена для вирішення якихось конкретних задач.

Якщо класифікувати різні технології по відношенню до їх використання людьми, то в першу чергу вони поділяються за сферою їх використання. Кожна програмна система розроблена, щоб вирішувати якусь конкретну задачу при роботі з нею. Тобто у людини в сфері її роботи є якась проблема, яку треба вирішувати, і різноманітні технології, додатки дозволяють людині спростити цей процес. На сьогодні дуже велика кількість систем допомагає людям вирішувати питання різноманітної складності у найважливіших сферах життєдіяльності, наприклад, медицини чи сфери навчання.

Все частіше для певних рішень починають використовувати різноманітні алгоритми, методи комп'ютерного зору [1]. Такі методи дозволяються обробляти зображення, отримане з камери. Методи комп'ютерного зору у поєднанні з штучним інтелектом дають можливість аналізувати отримане зображення та робити певні висновки на основі отриманої інформації. Характер цих висновків залежить від характеру задачі, яку потрібно було виконати, та від її мети, яку треба було досягти.

Одною з груп таких методів є методи для відстеження людських тіл, а саме їх руху та змін у просторі. Частіше за все розпізнання рухів людини виконується завдяки використанню різноманітних додаткових датчиків та спеціальних камер, які можуть відстежувати сигнали від цих датчиків у просторі, і таким чином відтворювати рухи людини.

Зазвичай алгоритми для розпізнання рухів працюють з деяким штучним інтелектом, який вже було навчено відстежувати людей. Цей штучний інтелект використовує певні методи та алгоритми для того, щоб виділити людину на

зображенні та знайти позиції ключових точок для відтворення скелету. Результати навчання зберігаються у так звану модель, яка і буде використовуватися потім для розпізнання тіла на зображенні. Тобто алгоритму не потрібно перед використанням кожного разу давати обробляти сотні тренувальних даних, які є фотографіями рухів людей без фонових зображень, а зберегти аналіз обробки цих зображень вперше у вигляді даних, які алгоритм потім сприймає як навчальні і на основі цих даних виконає пошук.

На сьогодні такі алгоритми є достатньо розповсюдженими для використання. Наприклад, шоломи віртуальної та змішаної реальностей використовуються такі алгоритми для відстеження рук, щоб можна було взаємодіяти з віртуальним світом без спеціальних контролерів, а напряму, власноруч, як у реальному світі. Відстеження рухів людини використовується також на підприємствах, наприклад, на великих промислових складах, де роботодавець або власник приміщення зацікавлений у підвищенні виробничої спроможності робітників, тому відстеження рухів використовується для аналізу переміщень людей.

Таких алгоритмів та моделей існує не мало та їх кількість все зростає. Це обумовлено тим, що дана технологія стає все більш доступною та сфер її застосування дуже багато, тому що вона все ж таки ще не є такою масовою, як наприклад звичайна камера у смартфоні кожної людини. Саме тому метою даної роботи є розглянути деякі з них та порівняти їх показники, що дозволить нам виявити найоптимальніші задачі для тих чи інших алгоритмів та сфери їх застосування. Дозволить зробити висновки, щодо можливості застосування таких методів у повсякденному житті та на пристроях, які не мають достатніх потужностей та додаткових технологій для реалізації даної задачі.

1 АНАЛІТИЧНИЙ ОГЛЯД

1.1 Аналіз предметної галузі

Сьогодні людство все більше поринає у вихор комп'ютерних технологій, тому що вони вже майже на кожному кроці навколо нас. Це дозволяє величезній кількості інформації постійно протікати повз нас та потрапляти до наших пристроїв. Певно, майже всі люди тепер мають розумні телефони з постійним доступом до мережі Інтернет. Тому вони постійно знаходяться в полі цієї інформації.

Великі компанії починають всерйоз розглядати ідеї метавсесвітів. Метавсесвіт – віртуальний світ, в якому люди можуть проживати своє або якесь інше життя ніби це насправді, але з функціональними особливостями властивими для саме віртуальних світів.

Такі рішення говорять про те, що людство вже повністю довіряє розумним технологіям, тому зараз, на мій погляд, вони мають найбільші можливості для розвитку.

Саме і в нашому випадку ми маємо стрімкий розвиток технологій комп'ютерного бачення для аналізу зображень [2]. Такі технології дозволяють комп'ютеру розуміти, що відбувається у реальному світі, аналізувати цю інформацію та відтворювати для людини для подальшого аналізу та певною взаємодії. Як вже було сказано, ці алгоритми і використовуються для відстеження людських тіл у просторі з використанням камери користувача для виконання задач, які потрібні в якомусь певному контексті. Звідси витікає певне твердження, що люди таким чином можуть в деякій мірі комунікувати з пристроєм, програмним середовищем на новому рівні, без взаємодії з ним напряду використовуючи інші допоміжні пристосування та вже звичні для нас методи.

Функції відстеження людського тіла вже використовуються у реальному та віртуальному світах для вирішення тих чи інших задач. Ці задачі дуже відрізняються одна від одної саме сферою застосування та шляхом вирішення

проблем. Наприклад, якщо ми говоримо про віртуальні світи, віртуальну реальність, то ми отримуємо задачу у деякому роді обману користувача. Відстеження усього тіла створює додаткове асоціативне відчуття між деяким віртуальним персонажем та користувачем, гравцем. Це робить віртуальний світ більш реальним та змушує наш мозок повірити в реальність віртуального світу, як би це суперечливо не звучало.

З іншого боку, якщо ми розглядаємо, скажімо, застосування відстеження тіла не у віртуальному світі, а у реальному, тобто таке застосування, яке людина використовує прямо зараз у реальному світі, то ми можемо поставити перед собою задачу, наприклад, навчання користувача. Тобто користувач повинен повторювати якісь рухи, а його пристрій в свою чергу повинен відстежувати положення тіла користувача та перевіряти чи правильно рухи виконуються.

У випадку навчальних застосунків ми повинні враховувати те, що для них дуже важливою характеристикою буде компактність рішення, тобто користувач не захоче купляти якісь додаткові датчики, щоб підвищити точність розпізнання, а захоче, щоб все було у нього в кишені, тобто його телефон з камерою. Звідси і впливає потреба у використанні моделей та методів комп'ютерного зору для відстеження людського тіла. Саме ці методи дозволять нам досягти бажаного результату, не звертаючись до використання додаткових датчиків.

1.2 Напрямок дослідження

Тема комп'ютерного зору вже майже у кожного на вустах, а відстеження людських тіл займає почесне місце серед найпопулярніших тем комп'ютерного зору. Про це свідчить те, що починаючи приблизно з 2012-го року регулярно проводяться різноманітні конференції та семінари, стосовно цієї теми [3]. Окрім цих заходів проходять також свого роду змагання з розробки алгоритмів для відстеження людських тіл.

Така зацікавленість темою обумовлена тим, що використання подібних алгоритмів відкриває двері до нових можливостей у розробці програмного забезпечення. Програмні застосунки з використанням трекінгу тіла можуть бути

широко застосовані майже у будь-якій сфері діяльності людей. Ці застосунки можуть бути як навчальними, так і просто розважальними, та також їх можна використовувати у медичній сфері.

Найпоширеніші алгоритми зараз це ті, що використовують різноманітні додаткові пристрої по типу датчиків, чи стаціонарних камер глибини [4]. Ці алгоритми відстежують рухи доволі точно, проте обмежують сферу використання через те, що потрібно використовувати додаткові пристрої.

Метою даного дослідження є розгляд декількох існуючих методів відстеження рухів людського тіла як з використанням додаткових пристроїв чи датчиків, так і без – з використанням лише звичайної RGB камери.

1.3 Аналіз існуючих методів і алгоритмів

Враховуючи вищеописану інформацію ми маємо два загальні види відстеження людського тіла: відстеження з використанням додаткових датчиків та відстеження завдяки алгоритмам комп'ютерного зору для зчитування і аналізу інформації з камери пристрою користувача.

Перший спосіб підходить у більшості для практичного використання лише у технологіях віртуальної реальності, та являє собою достатньо велику конструкцію, проте може надавати більшу точність. Використання подібних алгоритмів виправдовує себе, коли не важлива мобільність застосування методу та мається стаціонарна трекінг зона для роботи.

Алгоритми зчитування рухів з зображення являють собою більш універсальну та компактну версію реалізації функціоналу [5], та більш підходять для практичного повсякденного використання.

У даній роботі розглянуто методи відстеження людських тіл обох видів, але основну увагу виділено саме методам комп'ютерного зору для зчитування рухів з даних камери. Таке рішення було прийняте опираючись на те, що все ж таки відстеження людського тіла з використанням додаткових датчиків доволі обмежене у сферах діяльності, коли відстеження рухів з використанням методів

комп'ютерного зору мають широке охоплення сфер та можуть бути використанні у поточний час у поточному місці.

1.3.1 Відстеження тіла з датчиками

Даний тип відстеження розглянуто на прикладі комплекту системи віртуальної реальності HTC Vive та датчиків для відстеження VIVE Tracker.

У цьому випадку захоплення руху – це деякий процес оцінки абсолютного положення людини та її орієнтації у просторі. Система складається з випромінювачів, які передають у простір синхронізовані світлові коливання, та датчиків відстеження, які називають трекерами, які використовують фотодіоди для отримання влучного світлового променя для замірювання часу світлових імпульсів, що дозволяє розрахувати вертикальні та горизонтальні кути до випромінювачів. Цей кут заміряється технологією, аналогічною технології кута прибуття [6]. Тобто для розрахунку кута сигналу використовуються масиви антен, які розташовуються у робочому просторі користувача.

Світлові дані випромінювача можуть бути двох типів – спалах, який призначений для синхронізації, та інфрачервона розгортка. На рисунку 1.1 представлений схематичний вигляд роботи системи. Кожен новий цикл починається з вже згаданого імпульсу синхронізації, який містить у собі деякі метадані з параметрами калібрування пристроїв [7].

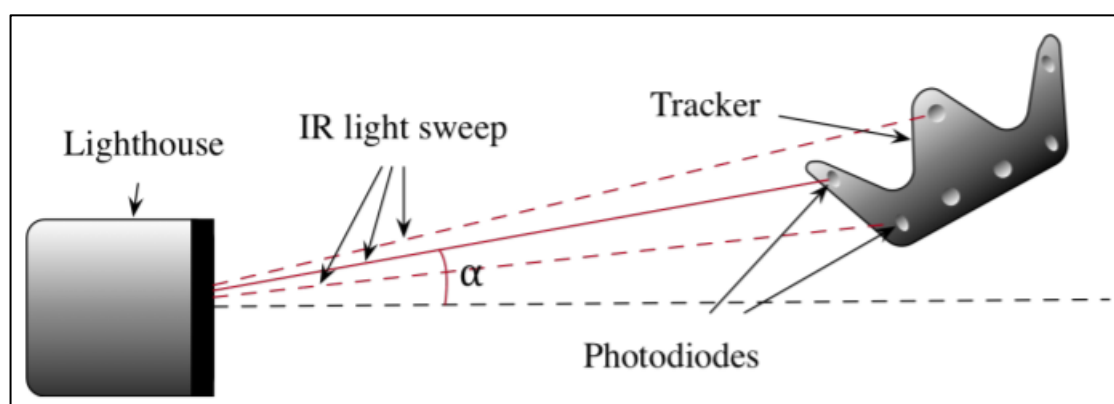


Рисунок 1.1 – Вид збоку на принцип роботи системи HTC Vive з кутом між горизонтальною площиною та вектором нормалі випромінювача

Слідом за цим сигналом йде інфрачервоний плоскісний лазер, що обертається. Датчики фотодіоди трекера зчитують обидва ці сигнали та з отриманої з них інформації можуть розрахувати кут між вектором нормалі випромінювача та фотодіодом, використовуючи різницю у часі, тому що лазер обертається з постійною кутовою швидкістю.

Таким чином дані датчики-трекери розміщуються на тілі людини на спеціальних місцях. Кількість розміщуваних на тілі датчиків залежить від бажаної точності відтворення рухів у віртуальному просторі і вона може складати від одного до семи трекерів (рис. 1.2). У випадку одного додаткового датчику він розміщується в районі стегон людини, а коли датчиків сім – в районі стегон, колін, ліктів та ступнів (рис. 1.2, б). Найоптимальнішим варіантом є комбінація 3 датчиків – датчик на стегнах та два на ступнях (рис. 1.2, а).

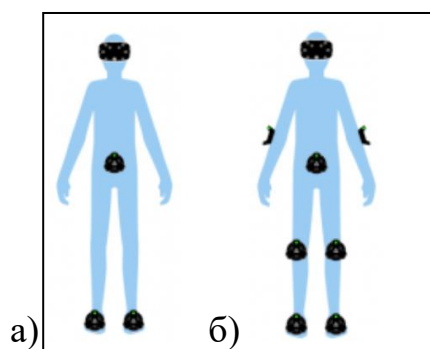


Рисунок 1.2 – Варіанти розміщення датчиків: а) три датчики, б) сім датчиків

Слід враховувати, що ці датчики є додатковими, тому що у користувача завжди є ще три: це безпосередньо шолом віртуальної реальності та два контролери у руках. Якщо відстежувати людину використовуючи лише шолом та два контролери у руках, ми отримаємо лише точні позиції долоней та голови людини, проте багато додатків можуть відтворювати тіло користувача, використовуючи лише ці дані. Ми звичайно враховуємо і ці елементи системи, тому що вони відіграють таку ж саму роль у виявленні користувача у просторі, як і додаткові елементи. Тобто зазначена оптимальна кількість датчиків у

кількості трьох відноситься до кількості додаткових, тому загальна кількість трекерів складатиме шість.

1.3.2 Метод відстеження положення людського тіла на основі згорткової нейронної мережі

Даний метод є монокулярним у своїй реалізації, тобто потребує лише одну камеру, що є великим плюсом. Згорткова нейрона мережа [8] являє ж собою деяку архітектуру штучної нейронної мережі, яка націлена саме на ефективне розпізнання образів. Засновується вона на тому, що є умовно деякі прості клітини в системі, які реагують на деякі прямі лінії під різними кутами, та складні клітини, які пов'язуються з реакцією на одночасну активацію певного набору простих клітин.

Даний метод реалізує новий регресор розрахунку місця знаходження тіла людини у просторі саме на базі згорткової нейронної мережі [9]. Даний метод регресує двовимірні та тривимірні позиції суглобів людини у реальному часі та не потребує занадто великого уточнення поточного кадру.

Метод накладення та налаштування кінематичного скелета в реальному часі також використовує вихідні дані нейронної мережі для отримання стабільних у часі тривимірних глобальних реконструкцій положення людського тіла на базі когерентного кінематичного скелета.

Для досягнення найбільшою точності при використанні цього методу, слід обучати достатньо глибокі мережі, проте це вже становиться недопустимим при використанні методу у реальному часі, тому що глибокі мережі ускладнюють використання методу [10]. Навчання більш глибокої мережі значно збільшить точність, проте даний процес займає також і значно більше часу, а використання методу у реальному часі дещо обмежує нас у часових рамках, тому що на момент навчання моделі додаток не буде працювати. Головна ідея є виявити оптимально навчену модель, яка буде з достатньою точністю відстежувати рухи людини у реальному часі. Тому даний алгоритм має модифікацію, у якій використовується

невелика за глибиною нейрона мережа, але з деякими поліпшеннями для використання у реальному часі.

1.3.3 Метод відстеження людського тіла з врахуванням оклюзії

Даний метод зчитує з монокулярних кольорових зображень двовимірні дані людини та трансформує їх у тривимірні дані з урахуванням оцінки двовимірних суглобів тіла [11].

Метод являє собою класифікатор орієнтації людського тіла та ансамбль нейронних мереж, налаштованих також на орієнтацію. Разом вони регресують тривимірне положення людини. Дані підходи дозволяють розділити отриману модель на верхню та нижню ієрархії, що в свою чергу і надає можливість враховувати оклюзії, що умовно показано на рисунку 1.3 [12].

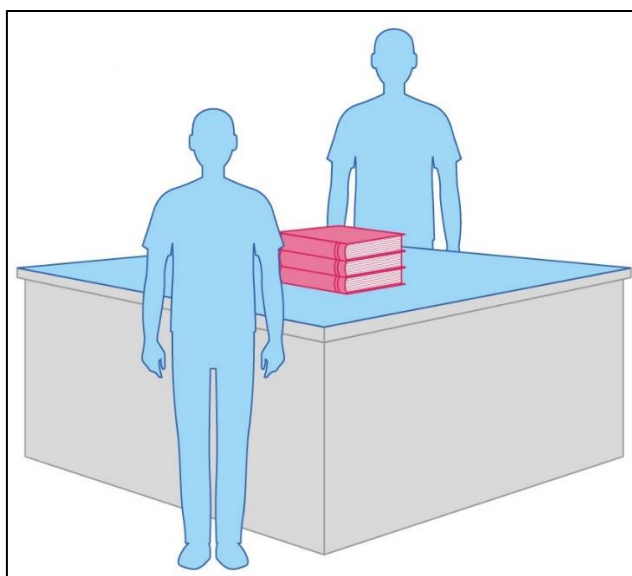


Рисунок 1.3 – Умовне відображення оклюзії

Також метод має деякий уточнюючий механізм, який називається вирішувачем зворотної кінематики. Він призначений для того, щоб уточнити фінальне рішення на основі нейронної мережі, забезпечуючи оцінку позиції людського тіла у відповідність з розмірами цільової функції, якщо та відома. Тобто ми уточнюємо отриманий результат з очікуванням і на основі цього робимо висновок, чи достатньо добре було знайдено людину на зображенні. Це дозволяє

нам позбавитись помилкових результатів, та отримувати стабільно достатньо точно модель.

1.3.4 Відстеження положення людського тіла на основі відфільтрованих ключових точок

Даний метод пропонує метод оцінки тіла людини за допомогою використання RGB-D простору кольорів [13]. Цей колір включає в себе додатковий простір який являє собою зображення глибини. Зображення глибини отримується на пристрої спеціальним датчиком, який заміряє відстані до об'єкта шляхом відправлення йому певних світлових сигналів та очікування поверненого відбитого сигналу. Зображення глибини дозволяє нам з великою точністю виділити людину у кадрі, якщо ніякі більше об'єкти не заважають. Проте звідси і витікає одразу мінус підходу, тому що камера глибини на сьогодні ще не достатньо поширена, і є лише у деяких телефонів, які в свою чергу не є дуже дешевими.

Отже суть методу полягає в тому, що завдяки звичайному зображенню отримуються ключові точки людини у двовимірному просторі та завдяки зображенню глибини воно перетворюється на тривимірне. Наступним кроком нейронна мережа обробляє отримані дані та генерує гіпотези для поточного кадру зображення щодо можливих положень тіла. Для того, щоб виявити кращу гіпотезу, ці можливі положення оцінюються цільовою функцією, яка розраховує наскільки сильно тривимірна модель не відповідає фактичному зображенню глибини, яке дає нам максимально точне положення людини. На рисунку 1.4 зображена робота методу. Тобто ми бачимо ключові точки, які знаходить алгоритм та помічає ними суглоби на тілі людини. Ці точки також називають з'єднаннями, тому що алгоритм знаходить конкретну точку та вказує, що це саме, чи то коліно, чи то лікоть, і потім в залежності від того, що це за точка, малюються з'єднання між сусідніми точками для отримання частин скелету.

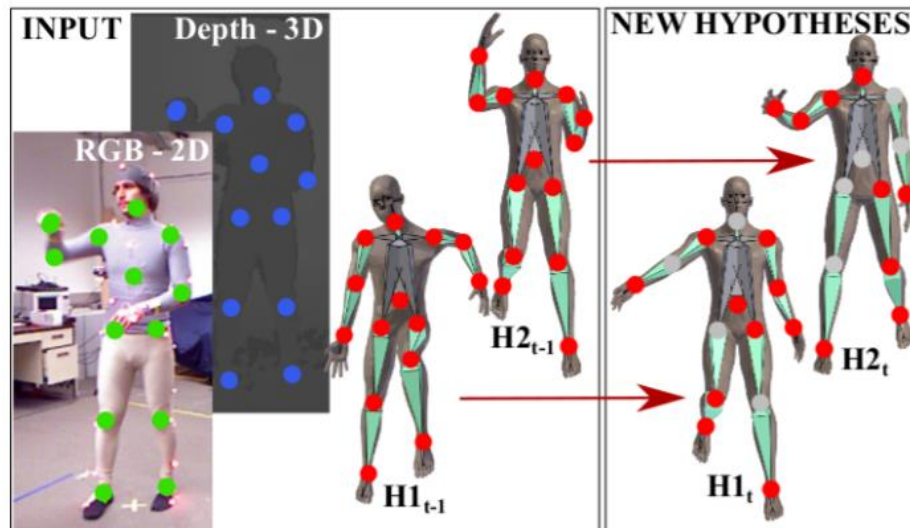


Рисунок 1.4 – Умовне відображення роботи методу

Даний метод певно повинен себе показувати дуже добре на практиці, оскільки, як вже було не раз зазначено, використовує зображення глибини, яке в свою чергу може гарантувати більш точне визначення людського тіла на зображенні. До того ж, зображення глибини у використанні з множинними гіпотезами надає нам можливість отримання зображення людського тіла у найсприятливішому вигляді завдяки фільтрації та згладжуванню отриманих точок. До того ж, проблеми методів, які не підтримують оклюзії, цього методу не стосуються, тому що даний алгоритм вміє їх дуже добре оцінювати та передбачати. Однак, як вже було сказано, даний алгоритм обмежує його дуже поширене використання саме через те, що він опирається на зображення глибини, а RGB-D камери є далеко не у кожному пристрої та не у кожної людини.

2 ПОСТАНОВКА ЗАДАЧІ

Аналіз предметної області надав нам можливість сформулювати конкретні вимоги до програмної системи, що розробляється, та сформулювати саму задачу, яку переслідує дана дослідницька робота.

Основною задачею є аналіз визначених методів. Першим кроком розглядаються методи в своєму формальному вигляді, тобто наводяться їх математичні моделі та їх опис. Розглядаючи методи слід одразу їх поділяти на ті, що використовують якісь додаткові датчики та пристрої, та ті, що використовують лише звичайну камеру та методи комп'ютерного зору.

Після аналізу визначених методів, а саме їх формальної частини, слід провести експеримент-дослідження методів. Експеримент повинен показати, який з методів є найоптимальнішим у використанні та для яких завдань підходить кожний з них. Зауважимо відразу, що у експериментах не буде приймати участь метод відстеження з використанням датчиків, тому що очевидно, що даний метод буде найточнішим, та нас цікавить саме використання методів комп'ютерного зору.

Після проведення експериментів отримані дані аналізуються для того, щоб зробити певні висновки щодо того, який метод влаштовує нас своїми характеристиками та який можна застосувати на практиці при розробці додатку, який продемонструє його роботу у конкретній сфері.

Наступний крок – розробка програмної системи.

Система буде представляти з себе додаток для занять спортом. Він буде мати у собі відеоматеріали з конкретними фізичними вправами. Користувач буде надавати пристрою доступ до власної камери та матиме можливість завдяки відстеженню тіла повторювати вправу на відео та бачити, наскільки правильно він її робить.

Система повинна порівнювати позиції ключових точок на відео з вправою, та на зображенні з камери. Користувач повинен бачити у відсотковому відношенні наскільки правильно він виконує конкретну вправу.

Користувач повинен мати змогу обирати серед наданих йому в прав конкретну та виконувати її. Додаток не повинен спеціалізуватися лише на виконанні, наприклад, присідань.

Зазначена вище вимога призводить до формування вимоги до архітектури програмної системи, а саме вимога масштабованості. Додавання нових вправ до застосунок повинно бути простим та не повинно включати в себе написання нового коду. Компоненти системи не повинні мати жорстких зв'язків, та кожний окремо взятий компонент повинен виконувати свою конкретну задачу та не перетворюватись у суперклас, який контролює увесь застосунок в цілому. Користувацький інтерфейс та код програми також не повинні бути жорстко зв'язаними. Тобто зміни у коді не призведуть до того, що весь інтерфейс доведеться переробляти наново, та навпаки.

Оскільки система, що розроблюється, це демонстрація можливостей методу відстеження рухів у конкретній сфері використання, користувацький інтерфейс повинен бути простим та зрозумілим. На даному етапі немає необхідності витратити час та сили на наведення краси.

Після оформлення результатів експериментів та розробки додатка настає етап збору та аналізу результатів роботи, які мають за мету пересвідчитись у доцільності використання того чи іншого методу відстеження рухів людського тіла у конкретних сферах діяльності людини, та переконатись, що обрані методи можуть бути використані у запропонованій реалізації.

3 МАТЕМАТИЧНА МОДЕЛЬ

Формалізація зазначених методів є дуже важливим етапом, оскільки це дозволяє нам побачити повну картину того, що буде робити той чи інший метод. Розписавши математичні моделі методів, ми будемо впевнені в тому, які кроки виконувати при реалізації алгоритмів. Це в свою чергу дасть змогу написати максимально ефективний алгоритм, який буде виконувати тільки те, що від нього потребується та не буде витрачати ресурси системи та пристрою на обробку якихось зайвих даних.

3.1 Відстеження тіла VIVE Tracker

Окрім отриманих даних завдяки аналізу лазерних променів датчик також має інформацію щодо даних руху у просторі через інерційний блок, вбудований в нього. Інерційні дані складаються з лінійного прискорення та кутових швидкостей.

Що випромінювач, що датчик мають свої кадри l і t відповідно (рис. 3.1). Позначимо випромінювач як l_i для наочності, де i – його індекс. З рисунку ми бачимо, що VIVE Tracker оцінює кут випромінювача з різницею часу отримання сигналу від нього. Допоміжний кадр v обирається так, щоб він завжди співпадав з одним з кадрів випромінювача, обраного під час процедури калібрування. Ця процедура дозволяє системі зіставляти позиції випромінювачів, якщо їх використовується декілька. Кінцевий результат показаний червоною лінією на рисунку.

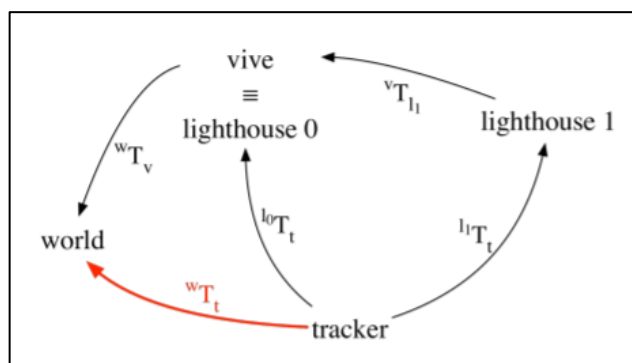


Рисунок 3.1 – Кадри оцінки місцезнаходження

Для перетворень використовуються певні умовні позначення, де aT_b – перетворення тіла з кадру b в кадр a . Ми також будемо використовувати aP_b та ${}^a\tilde{P}_b$, які являють собою положення кадру b відносно a у Декартових та однорідних координатах відповідно.

Алгоритм оцінює повні позиції трекара в режимі реального часу, використовуючи лише світлові дані. Він знаходить найкращу відповідність між позицією та доступними даними за допомогою нелінійного методу найменших квадратів, що включає модель системи. Також записуються інерційні дані, однак ми не використовували їх, тому що спочатку шукаємо задовільний результат.

Модель цієї системи є функцією, яка повертає горизонтальний і вертикальний кути залежно від відносного положення фотодіода трекара і випромінювача. Це положення виходить за допомогою серії перетворень положення тіла, які перетворюють координати фотодіода з кадру трекара в координати випромінювача:

$${}^l\tilde{P}_p = {}^lT_v {}^vT_t {}^t\tilde{P}_p, \quad (3.1)$$

де p – фотодіод, l – випромінювач, v – кадр віде, lT_v – різниця в часі між кадром віде та кадром випромінювача, t – кадр датчика, vT_t – різниця в часі між кадром віде та кадром датчика. Позиція фотодіода ${}^t\tilde{P}_p$ вже відома, тому що у нас є можливість її дізнатися під час калібрування пристрою, яке описувалось вище.

Для того, щоб отримати відносні горизонтальний та вертикальний кути між фотодіодом та вектором нормалі випромінювача, ми повинні використовувати координати трьох фотодіодів окремо:

$$h({}^lP_p) = \begin{cases} \arctan\left(\frac{{}^lP_y}{\frac{{}^lP_z}{p}}\right) \\ \arctan\left(\frac{{}^lP_x}{\frac{{}^lP_z}{p}}\right) \end{cases}. \quad (3.2)$$

де ${}^lP_p^x, {}^lP_p^y, {}^lP_p^z$ – є координатами x, y, z lP_p та верхній вираз саме для горизонтальної осі, а нижній для вертикальної.

Щоб розрахувати позицію датчика, ми використовуємо суму квадратів різниць між кутами фотодіода, які були визначені (α_p) і можливими кутами. Це задача оптимізації, та якщо ми будемо використовувати оптимізатор, то зможемо отримати дані швидко. Формалізована модель функції:

$$f = \sum_{l=1}^M \sum_{p=1}^N [h_{p,l}({}^vT_t) - \alpha_p]^2, \quad (3.3)$$

де $h_{p,l}$ – функція (3.2) з (3.1) в якості вхідного аргументу після перетворення у Декартові координати.

Щоб а ні старі дані, а ні викиди ніяк не впливали на оцінку, ми включили ще такі обмеження, які говорять про те, що: всі вимірні кути з величиною більше 60 градусів відхиляються та має бути не менше 4 вимірних кутів від останнього виявленого випромінювача і всіх зразків старше 50 мс не використовуються, якщо вони отримані не від останнього виявленого випромінювача, інакше пропускаємо оцінку цієї позиції.

Функція (3.3) є досить складною, тому оптимізатор може розходитися або сходиться до локального мінімуму. Щоб запобігти неправильним оцінкам, ми включаємо ще одну перевірку перед наданням рішення користувачеві: він перевіряє значення функції, і якщо воно перевищує порогове значення, лінійно пов'язане з кількістю кутів, що спостерігаються, алгоритм відхиляє позицію і чекає на нові дані.

3.2 Метод згорткової нейронної мережі

На рисунку 3.2 наведено огляд даного методу вирішення задачі. Він складається з двох основних етапів. Перший – згорткова нейронна мережа для регресії двох- та трьохвимірних положень суглобів в умовах монокулярного

аналізу. Враховується повнорозмірне зображення у конкретному кадрі. Згорткова нейронна мережа генерує теплові мапи для визначення людини. Ключові точки отримуються шляхом фільтрації отриманих теплових мап.

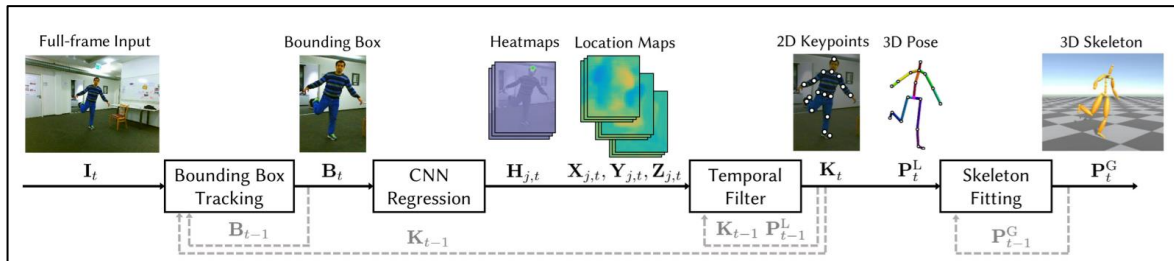


Рисунок 3.2 – Огляд методу використання згорткової нейронної мережі

Зазначена мережа повинна бути навчена на моделях даних позицій людини в трьохвимірному просторі та додатково на двовимірних даних положень тіла людини для більш кращих результатів у природніх умовах. Другий етап поєднує в собі відновлені положення суглобів з методом накладення кінематичного скелета для створення стабільної в часі тривимірної позиції скелета.

У якості вхідних даних методу ми отримуємо потік зображень RGB $\{\dots, I_{t-1}, I_t\}$. Для конкретного кадру t з вхідного потоку кінцевий результат методу – P_t^G , який є повною тривимірною позицією скелета. Регресор положення нейронної мережі оцінює двовимірне K_t та тривимірне P_t^L позиціонування суглобів. Після цього аналізу обох вимірів положень зіставляються для згладжування потрібної позиції $P_t^G(\theta, d)$, де θ – кути суглобів кінематичного скелета S , d – глобальне положення у просторі камери. J вказує на кількість суглобів.

Отримання суглобів у двовимірному просторі зображення зводиться до задачі передбачення достовірної теплової мапи $H_{j,t}$ на площині зображення для кожного суглобу $j \in \{1 \dots J\}$.

Для побудови трьохвимірного простору розширяється метод теплової мапи двовимірному простору шляхом використання трьох додаткових мап положення X_j, Y_j, Z_j для кожного суглоба j , захоплюючи також кореневі положення x_j, y_j, z_j

відповідно. Для того, щоб прогнозування тривимірної позиції більш суворо було пов'язане з двовимірним видом, значення x_j, y_j, z_j зчитуються з їх відповідних мап положення у позиції максимуму двовимірної теплової мапи H_j та зберігається в $P^L = \{x, y, z\}$, де $x \in R^{1 \times J}$ – вектор, у якому зберігається координата x позиції кожного максимуму суглобів.

Даний метод описує по кадрову оцінку положення людського тіла у просторі, проте такі методи не використовуються для аналізу відео контенту, оскільки сканування кожен кадр через неточності може викликати небажані ефекти у вигляді посмикування моделі, що не є достатньо необхідним для нашої реалізації. Проте тому даний алгоритм і об'єднує двох- та трьохвимірні положення суглобів людини в деяку загальну структуру оптимізації, наряду з фільтрацією у часі та згладжуванням, для отримання максимально точної та стабільної моделі.

Перш за все спочатку над передбаченнями K_t проводиться фільтрація за часом. Для того, щоб забезпечити стабільність моделі, довжини кожних кінцівок P_T^L замінюються на іншу довжину, яка зберігає напрямлення кістки P_T^L . Отримані передбачення об'єднуються за рахунок мінімізації цільової енергії

$$E_{\text{загальна}}(\theta, d) = E_{IK}(\theta, d) + E_{\text{проекту}}(\theta, d) + E_{\text{згладжування}}(\theta, d) + E_{\text{глибини}}(\theta, d)$$

Як уже було зазначено, даний метод розрахований на використання лише з одною камерою без яких би то не було додаткових сенсорів. Він обіцяє показувати гарні результати навіть на телефонах з дуже поганими камерами.

3.3 Відстеження з врахуванням оклюзії

Даний метод дозволяє розрахувати положення тіла в просторі за три кроки: знаходження двовимірних суглобів та їх кодування з використанням матриці нормалізованого обернення знаків, класифікація двовимірної людської моделі та використання відповідних нейронних мереж для регресії у тривимірну модель, та

обернену кінематику для більш точного налаштування отриманих результатів. Алгоритм базується на моделі Body25, яка має 25 ключових точок.

Двовимірна модель суглобів, що попередньо була оцінена, кодується у дві матриці нормалізованого обертання зі знаком (NSRM): одну для точок верхньої частини тіла та одну для точок нижньої частини тіла. NSRM - це нова схема кодування двовимірних положень тіла. Вона є інваріантною до трансляції положення та його масштабування та є ефективною з точки зору кількості елементів, необхідних для кодування двовимірної моделі. Проте є і недолік, який полягає в тому, що для них потрібні два канали кодування: один для X, а інший для спільних координат Y.

На рисунку 3.3 можна побачити схематичний огляд алгоритму, який був вже описаний вище. Даний рисунок ілюструє сказану раніше інформацію.

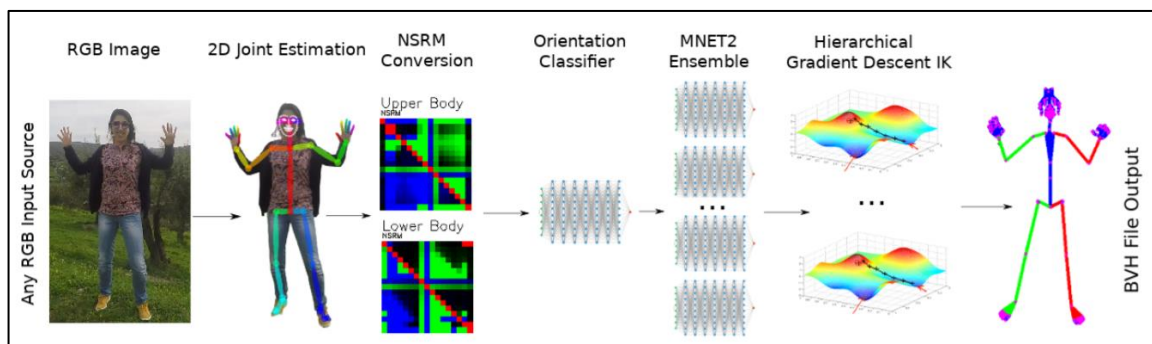


Рисунок 3.3 – Огляд методу використання матриці нормалізованого обертання зі знаком

Для кожної пари двовимірних точок a та b можемо назвати нову точку $c = (b_x, b_y - |b - a|)$ що є точкою b , яку було здвинуто по вертикалі. Ми можемо вивести формулу, яка буде обумовлювати взаємовідносини між точками a та b та їх відносний поворот щодо вертикальної осі.

$$NSRM^h(a, b, c) = \begin{cases} atan2(A_x B_y - A_y B_x, A_x B_x + A_y B_y) \\ a \neq b \\ 0 \end{cases},$$

де $A_x = b_x - a_x$, $A_y = b_y - a_y$, $B_x = c_x - b_x$, $B_y = c_y - b_y$.

Надалі використовується нейрона мережа для відтворення трьохвимірного відображення тіла. У попередніх методах це вже розглядалося, тому повторювати це немає сенсу.

Даний метод є дуже гарним прикладом методу типу «розділяй та володарюй». Метод дуже гарний тим, що має дуже гарне співвідношення між точністю та продуктивністю. Також безпосереднім плюсом є те, що метод поділяє людське тіло на умовні дві частини, що в свою чергу дозволяє нам використовувати оклюзію, тобто коли користувач може ховатися за перешкодами, а алгоритм буде продовжувати відстежувати його, хоч і лише половину. У випадках, коли частина тіла схована іншим об'єктом, попередня фільтрація точок [14] може згладити результат, та не звжди. Майбутнє даного алгоритму може заключатися у вивченні відстеження рук та обличчя.

3.4 Метод відфільтрованих ключових точок

Метод, як і багато інших схожих методів, використовує глибокі мережі OpenPose для отримання вихідного двомірного положення тіла людини. Проте як вже було сказано, даний метод використовує датчик глибини, який дозволяє таким чином перетворити двомірне представлення на трьохвимірне. Однак ми не можемо виключати помилки як самого датчика, там і алгоритму, який генерує двомірну модель.

На рисунку 3.4 можна побачити приклад таких помилок. Камера у точці С дивиться з боку на плече людини і виходить, що друге плече перекривається тілом, тому його відстеження може бути помилковим. Такі помилки є невід'ємною частиною методів відстеження тіла, тому що не часто буває, що людина дивиться прямо у камеру. Такі помилки не є суттєвими у випадку, коли відстеження людини потрібно як факт її знаходження на зображенні, а не коли треба відстежувати усі кінцівки.

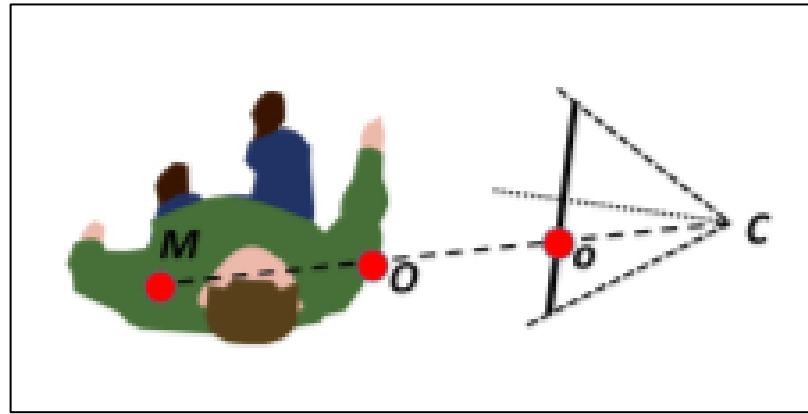


Рисунок 3.4 – Приклад помилки

Саме тому для оцінки позиції людини в цьому методі використовується трекер множинних гіпотез. Кожна гіпотеза, що генерується, визначається як підмножина доступних орієнтирів на тілі. Таким чином ми можемо навіть позбавитись від деяких шумів. Підмножини отримуються шляхом випадкової зваженої вибірки. Для того, щоб розрахувати ваги кожного з орієнтирів, використовуються набори геометричних обмежень та критеріїв неперервності плину часу.

Як вже було сказано, тривимірні позиції орієнтирів отримуються шляхом накладання двовимірної моделі, зчитаної людини, на її зображення глибини. У рамках цієї структура виконуються такі два підходи:

- LVM – базовий метод, у якому для створення трьохвимірної моделі використовуються усі ключові точки двовимірної моделі;
- МНТ – підхід, який пропонує даний метод – використання множинних гіпотез.

Враховуючи x позицію, яка визначається параметрами моделі K тіла людини та її функцією прямої кінематики, ми розраховуємо ключові точки суглоби у тривимірному просторі $M_i = (X_i, Y_i, Z_i), i \in [1, I]$ у деякій системі координат камери, а їх проекції $m_i = (u_i, v_i), i \in [1, I], I = 18$, на площинні зображенн.

Припустимо, що ми маємо знайдені двовимірні стики, перетини o_i та деякий двійковий прапорець, який приймає вигляд одиниці, якщо ключова точка

таки була знайдена, та 0, якщо цього не відбулось. Хай $O_i = (X_i, Y_i, Z_i), i \in [1, I]$ є тривимірними точками пов'язаними з o_i . Точки O_i отримані лише для заходження тих ключових точок, які мають певну конкретну та дійсну інформацію глибини.

Для даного методу відстеження людського тіла повне розходження $S(x, O, o)$ розраховується як:

$$S(x, O, o) = \sum_{i=1}^I F_i \|M_i - O_i\| + (f_i - F_i) \|m_i - o_i\|.$$

Та для того, щоб отримати тривимірну модель x , яка найбільш підходить, ми можемо мінімізувати цільову функцію, що і дасть нам значення:

$$x^* = \arg \min_x \{S(x, O, o)\}.$$

Далі настає шлях оцінки гіпотез. Оцінка гіпотези виконується з використанням ймовірності визначення глибини зображення. Ми говоримо про ступінь відповідності між візуалізованою моделлю та глибиною.

Крок попередньої обробки – використання оціночного положення моделі з попереднього кадру у якості еталону та збереження тільки того спостереження, яке знаходиться в межах заздалегідь визначеного діапазону. Щоб визначити вірогідність гіпотези ми виконуємо рендеринг з обліком того, що треба буде відкалібрувати камеру.

Приведемо також функцію, яка використовується для оцінки невідповідності між гіпотезою x та спостереженням z .

$$\Delta(z, x) = \lambda \frac{\sum_{p \in P} \min(|z_{d,p} - r_{d,p}|, d_M)}{d_M |P_i|} + (1 - \lambda)$$

Діапазон значень від 0 до 1 – від ідеального збігу до повної збіжності відповідно. На кожному кадрі у якості рішення обирається гіпотеза з мінімумом.

Якщо порівнювати два підходи, які були описані вище, то можна зробити все ж такі деякі висновки щодо даного методу. Виявлення базовим методом не будуть точними у більшості своїй через специфіку методу. Основною помилкою буде помилка в передбачуваних трьохвимірних спільних точках. У методі гіпотез немає такої помилки як раз все ж таки через специфіку його реалізації, а саме завдяки зображенню глибини та самих гіпотез.

Серед усіх розглянутих методів даний метод має найбільший потенціал у виробничій спроможності. Це обумовлено тим, що для розрахунків окклюзії потрібно проводити менше саме підрахунків, які базуються на передбаченнях та прогнозуваннях саме через використання даного методу у комбінації з камерою глибини, яка надає так зображення, де можна легше виділити людське тіло.

Проте все одно на сьогодні не усі телефони мають вбудовані датчики глибини. Можна використовувати звичайно камери з датчиками типу Microsoft Kinect, та вони самі не дуже поширені і їх використання створює обмеження мобільності розробленої системи. Тому ми можемо казати про те, що не кожна людина зможе використовувати додаток, у якому використовується відстеження тіла з використанням даного методу.

4 ПРОВЕДЕННЯ ЕКСПЕРИМЕНТУ

Найважливішим етапом роботи після дослідження є безпосереднє проведення експерименту. Даний етап допомагає зрозуміти у правильності проведення дослідження та обраного шляху його проведення. Сам експеримент дозволить провести аналіз отриманих результатів та зробити конкретні висновки щодо виконаної роботи.

4.1 Методологія експерименту

Для порівняння вищезазначених методів комп'ютерного зору для відстеження рухів людського тіла треба провести експерименти, у яких будуть визначені певні характеристики цих методів, які покажуть їх окремі сильні та слабкі сторони. Проведення експериментів – необхідна складова аналізу запропонованих методів, оскільки кожен окремий метод досягає виконання поставленої задачі різними шляхами, тому нам необхідно визначитись з методологією експерименту.

Розглядаючи методику експерименту, можна визначити його структуру, формулювання та послідовність виконання дій, кожна з яких ґрунтується на обсязі. Суть експерименту в цій роботі полягає в тому, щоб порівняти наявні методи, пояснити їх основні принципи та порівняти їх загальну ефективність у виконанні завдання. Мета експерименту – знайти найбільш ефективний з зазначених метод розпізнання рухів людського тіла.

У проведенні експерименту прийматимуть участь три методи з наведених, а саме метод згорткової нейронної мережі, метод відстеження з врахуванням оклюзії та метод відфільтрованих ключових точок. Вибір саме цих методів обумовлений тим, що їх реалізацію можливо запустити майже на будь-якому сучасному пристрої, який має процесор для обробки даних та камеру для їх зчитування.

Експеримент буде проводитись з використанням одного і того ж набору даних для кожного методу. Цей набір даних використовувався на змаганні з розробки методів відстеження «ImageNet Large Scale Visual Recognition Challenge

2012», який після проведення був публічно опублікований та поширився серед людей, які займаються комп'ютерним зором.

Після проведення експериментів і отримання значень певних характеристик, вони будуть занесені до порівняльних таблиць. Це надасть змогу одразу візуально побачити різницю між тим чи іншим алгоритмом.

Далі отримані дані будуть порівняні у кількісних значеннях та буде виявлено найоптимальніший з методів.

4.2 Специфікація програмного забезпечення

За для проведення експерименту так чи інакше треба реалізувати отримані методи програмно та надати користувачу можливість їх тестування. Слід зазначити, що для проведення експерименту нам не важлива візуальна складова додатку, а нам потрібен максимально чистий додаток, який буде виконувати лише одну конкретну задачу, і нічого не буде витрачати ресурси пристрою. Тому у цьому розділі розроблений додаток для тестів буде лише відображати зображення з камери, помічати ключові точки, які будуть знайдені алгоритмом, та записувати конкретні показники. Звідси, вимоги від користувача:

- користувач повинен бачити зображення з своєї камери;
- користувач повинен бачити відмітки ключових точок на зображенні з камери;
- користувач повинен отримувати числові значення результатів роботи алгоритму;
- користувач повинен мати можливість переключитись на виконання іншого алгоритму.

Дана версія додатку буде запускатися на комп'ютері з наступною конфігурацією:

- процесор – Intel Core i5-7200U з вбудованим графічним чипом HD Graphics 620;
- ОЗУ – 8 Гб, де до 2 Гб може резервувати графічний чип;
- Windows 11 версії 21H2 збірки 10.0.22000.

Тобто даний комп'ютер можна назвати комп'ютером середнього класу. Це говорить про те, що ми очікуємо на задовільну працю додатку на не самих сильних пристроях.

4.3 Проведення експерименту

Оцінка роботи методів виконується як якісно, так і кількісно. Для оцінки точності відстеження позицій тіла використовуються дві метрики. Перша метрика – середня відстань у міліметрах з послідовності прогнозованих точок від їх вихідних положень. Друга метрика представляє собою показник середньої кількості кадрів у секунду при роботі програмної реалізації методу.

У таблиці 4.1 наведено результати вимірювання для першого показника. Для експерименту було виділено три види рухів: повільні, середньої інтенсивності та швидкі. Повільні рухи – це майже статичне положення тіла, середні – звичайна поведінка людини, а швидкі рухи – це виконання тих чи інших вправ, наприклад, стрибків.

Таблиця 4.1 – Результати вимірювання першого показника

Інтенсивність рухів	Метод згорткової нейронної мережі	Метод відстеження з врахуванням окклюзії	Метод відфільтрованих ключових точок
Повільні	48	46	45
Середні	79	78	66
Швидкі	98	95	85

Оскільки нас цікавить виробнича спроможність кожного з алгоритмів при використанні на не самих потужних комп'ютерах, то також була заміряна середня кількість кадрів у секунду (FPS) з якими працює кожний з алгоритмів. Ці дані наведено у таблиці 4.2. Слід зауважити, що для вимірювання показників операційна система максимально звільнялась від фонових процесів, щоб отримані дані були незалежними від інших процесів.

Таблиця 4.2 – Показники кадрів в секунду для кожного методу

Метод	Кадри в секунду
Метод згорткової нейронної мережі	25
Метод відстеження з врахуванням окклюдії	15
Метод відфільтрованих ключових точок	30

З наведених даних випливає висновок, що розрахунок окклюдії – затратна задача, тому алгоритм починає використовувати підвищені ресурси пристрою. З наведених алгоритмів лише алгоритм згорткової мережі не підтримує можливості відтворення окклюдії.

4.4 Аналіз результатів експерименту

Розглянемо отримані результати та порівняємо алгоритми за наведеними показниками.

Перш за все з таблиці 4.1 ми бачимо, що метод відфільтрованих точок має найменшу відстань між ключовими точками. Цей показник і обумовлює точність алгоритму. В цьому випадку середня помилка відстані, тобто відстань від знайденої ключової точки до її очікуваного місця знаходження, складає 65 мм, коли середня відстань інших двох алгоритмів знаходиться у значенні близько 80 мм, тобто приблизно на 18 % менше. Метод відфільтрованих точок має вищий коефіцієнт успіху для всіх порогових значень проти інших варіантів.

Залежно від інтенсивності руху також змінюється точність методів: чим рухи швидші та чим їх характер є більш різким, тим точність знаходження ключових точок нижче. При повільних рухах точність алгоритмів є майже однаковою. Проте з підвищенням інтенсивності рухів точність методів згорткової мережі та методу з розрахунком окклюдії зменшується більше ніж у методу відфільтрованих ключових точок. Це обумовлюється тим, що даний метод окрім саме фільтрації точок використовує також зображення з камери глибини, що дає

можливість більш швидше отримувати зображення людини, тому що пошук людського контуру на зображенні саме з камери займає менше часу.

Якщо ж ми подивимось на результати замірів при інтенсивних рухах, то ми побачимо, що усі методи суттєво втрачають точність, оскільки алгоритми не встигають обробляти кадр за кадром, тому попередній кадр значно відрізняється від іншого, тому конкретні рухи відстежити стає важко.

Показник кадрів у секунду для нас є дуже важливим, оскільки ми розглядаємо дані методи для практичного їх використання, до того для використання на пристроях, які не мають великих потужностей. З таблиці 4.2 ми бачимо, що найменше значення кадрів за секунду показує метод, який програмно розраховує оклюзію. Саме з її розрахунком і пов'язана така затримка у відтворенні кадрів. Слід зазначити, що звичайно, якщо стояти, майже не рухатись, та не закривати частини тіла, то показник кадрів буде на довільному рівні. Проте у протилежному випадку алгоритм починає прораховувати приблизне значення кінцівок людини та намагатись визначити, чи закриває їх щось.

У випадку методу відфільтрованих точок ситуація позитивна навіть якщо закривати кінцівки. Це знову частково обумовлено камерою глибини, але більший вплив має як раз саме фільтрація точок. Метод найкраще працює, коли не відбувається випадкового виключення ключових точок, оскільки критерії геометричної та тимчасової безперервності гарантують гарний вибір підмножини орієнтирів. А, як вже було зазначено, використання зображення глибини у комбінації з монокулярним зображенням звичайної камери полегшує безпосередній аналіз. Ми представили гібридний метод оцінки положення тіла з використанням введення зображення глибини. Щоб впоратися з шумом, використовується надійний трекер множинних гіпотез для оцінки та поширення гіпотез про позу, згенерованих підмножинами орієнтирів. Підмножини отримують шляхом випадкової виваженої вибірки, при цьому вага кожного орієнтиру розраховується з використанням набору критеріїв геометричної та часової безперервності. Експерименти показують, що цей метод значно перевершує (з погляду точності) базовий підхід, коли за оптимізації

використовуються всі ключові точки. Різниця ще більша, коли послідовність містить значні оклюзії.

Метод нейронної згорткової мережі показує також доволі непогані результати кадрів за секунда (мається на увазі, що використання застосунку з такою кількістю кадрів є прийнятним). Проте даний алгоритм взагалі не враховує оклюзії, тому у випадку, коли з тієї чи іншої причини немає можливості відстежити якусь з ключових точок з'єднання, відстеження просто припиняється до моменту відновлення можливості відстеження усіх необхідних точок. Тому даний алгоритм не підійде для виконання задач, де між камерою та людиною можуть виникати якісь об'єкти, які будуть заважати відстеженню тіла.

З отриманих результатів аналізу проведених експериментів можна зробити висновок, що найоптимальнішим з наведених методів є метод відфільтрованих ключових точок [15]. Певна частина переваги досягається завдяки використанню камери глибини, що є цілком логічним, проте у комбінації з методом фільтрації точок досягається найкращий результат. Даний метод доцільно буде використовувати у випадках, коли ми не можемо бути впевненні, що жодна частина тіла не буде перекрита якимось об'єктом. Однак обмеження у вигляді камери глибини є суттєвим, оскільки не так багато пристроїв мають дану технологію. Найкраще ця технологія у мобільних пристроях на сьогодні розвинена у компанії Apple у їх останній лінійці мобільних телефонів iPhone.

Не погані характеристики показує метод згорткової нейронної мережі. Хоч даний метод і не має підтримки оклюзії, проте ми можемо спостерігати стабільну поведінку кадрів за секунду та достатню їх кількість. Ми спостерігаємо підвищення точності, особливо у разі значної оклюзії суглобів. Випадки збою виникають при введенні помилкових фокусних відстаней в модуль інверсної кінематики, який потім змушує людей нахилитися вперед, щоб задовольнити обмеження 2D-зчленування, дотримуючись при цьому наданих фокусних відстаней.

Тому для більшості застосунків, де, наприклад, відстеження рухів потрібно для контролю повторень якихось вправ та ми впевнені, що шанс того, що щось

або хтось закриє людину на зображенні, мінімальний, метод згорткової нейронної мережі підійде і виконає поставлену перед ним задачу (рис. 4.1).

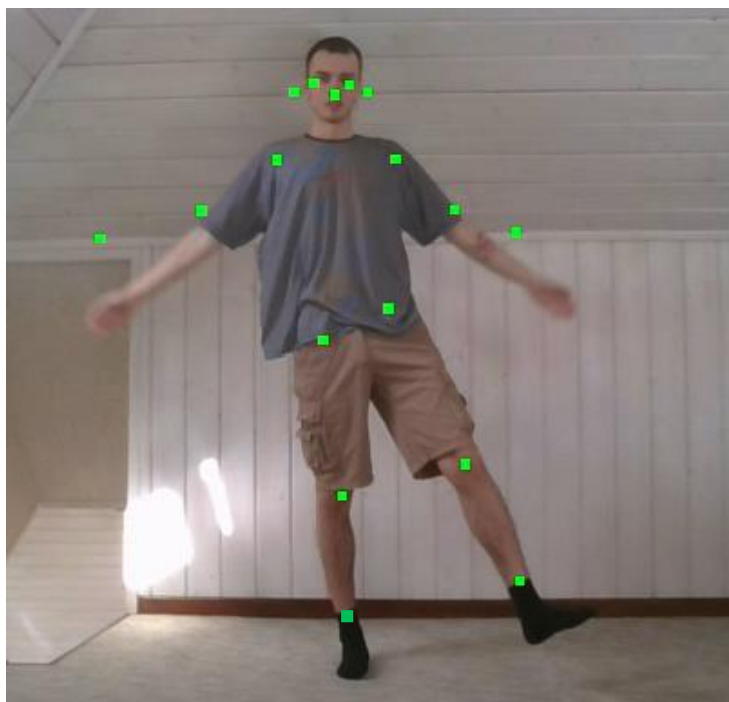


Рисунок 4.1 – Приклад роботи методів

Ми представили метод, який оцінює тривимірну кінематичну позу людини, включаючи глобальне положення, стабільним узгодженим у часі способом з одного відеопотоку RGB. Цей підхід поєднує в собі повністю згорткову нейронну мережу, яка регресує положення суглобів у 2D та 3D, та метод підбору кінематичного скелета, виробляючи стабільну у часі 3D-реконструкцію руху в реальному часі. На відміну від більшості існуючих підходів, ця мережа може працювати без строгих рамок, що обмежують, і забезпечує не дуже затратне відстеження обмежуючих рамок.

Якісні та кількісні оцінки показують, що цей підхід можна порівняти з автономними сучасними монокулярними методами RGB та він наближається за якістю до методів які використовують камери глибини.

5 ОПИС РОЗРОБЛЕНОЇ ПРОГРАМНОЇ СИСТЕМИ

Для демонстрації результатів дослідження та практичного застосування отриманої інформації було вирішено розробити додаток з використанням методу відстеження рухів людського тіла, у якому людина зможе виконувати різноманітні вправи, повторюючи їх за відео. Суть програми є у порівнянні отриманих ключових точок відстеження рухів на відео та на зображенні з камери користувача. Користувачу буде надаватись список із доступних вправ, які можна виконати. Обрана вправа буде відображатись на відео яке буде також відстежуватись для того, щоб користувач повторював вправу та бачив, наскільки якісно він це робить.

У даному розділі будуть описані етапи проектування та побудування програмної системи та отриманий результат.

5.1 Обґрунтування середовища розробки

Для додатку, який було розроблено, стояли дві найголовніші вимоги: можливість побудувати підсумкову версію додатку для різних систем, включаючи веб та мобільну системи, та можливість використання навченої моделі комп'ютерного зору.

У якості середовища для розробки кросплатформного застосунку був обраний Unity Engine. Unity – гнучкий інструмент, який дозволяє розробляти універсальні застосунки для різноманітних платформ. Але чому саме цей інструмент, якщо існують інші?

Відстеження людського тіла зараз дуже поширене у технологіях віртуальної, доповненої та змішаної реальностей. Unity в свою чергу має гнучку та дуже глибоку підтримку даних технологій, тому їх розробка з використанням цього інструмента є максимально зручною. До того ж, повертаючись до кросплатформності, Unity залежно від обраної платформи, на яку буде збиратись додаток, сам використовує потрібні варіації тих чи інших бібліотек, тому розробник переймається лише написанням безпосередньо коду та не повинен тримати у голові мільйони залежностей чи купу інших проектів, кожен під окрему

платформу. В свою чергу, якщо розробник хоче виконати більш тонке налаштування проекту, або конкретної платформ, Unity надає можливість також і особистого втручання у процес збірки додатку. Тобто при великому бажанні, розробник може працювати і з нативними класами та методами конкретної платформи.

Кросплатформність Unity досягається завдяки тому, що дане середовище працює на платформі .Net. Використовуючи цю платформу ми отримуємо не тільки кросплатформність, а й високу виробничу спроможність. З кожною новою версією .Net розробники приносять в неї багато нових функцій та покращують роботу старих. Застосунки створенні з використанням цієї платформи можуть переносити великі навантаження на систему та спокійно їх підтримувати.

В свою ж чергу мова програмування .Net платформи – C#. Оскільки C# – покращена версія мови C++, то дуже зручний, функціональний та розширюваний інструмент для створення програмного забезпечення. На мою думку, одна з найсильніших сторін C# – його синтаксис. Програми, написані на цій мові програмування, легко читаються зверху вниз, що дозволяє краще та швидше розуміти чужий код, а для досвідчених програмістів дана мова має багато конструкцій для пришвидшення написання типових речей.

У якості моделі комп'ютерного зору використовувалась технологія від Google під назвою Tensorflow. А саме його розширення написане спеціально для платформи .Net – TensorFlow.NET. Оскільки первісно TensorFlow – Python бібліотека, то і версія для платформи .Net використовує правила іменування Python, що насправді дозволяє не плутатись у документації та отримувати широку підтримку операцій більш високого рівня.

Для перевірки застосунок буде розроблено для WebGL платформи – JavaScript API яка дозволяє відображати на веб-сторінці інтерактивну графіку. Частіше за все це використовується для створення якихось простеньких онлайн ігор, що зручно для користувача, тому що йому не потрібно нічого завантажувати. Чому простеньких? Тому що WebGL має обмежені ресурси, що впливає на

виробничу спроможність технології, і дуже складну гру там буде важко запустити, тому це є найкращий варіантом тесту застосунку.

5.2 Архітектура програмної системи

Певно майже кожний застосунок розроблений з допомогою Unity [16] побудований з використанням компонентної архітектури. Саме середовище налаштовано саме так, що тобі буде зручно використовувати дану архітектуру.

Під компонентною архітектурою мається на увазі, що дизайн розкладається на окремі функціональні або логічні елементи. Так і в нашому випадку: ми створюємо компонент, описуємо його функціональну логіку, створюємо для нього об'єкт у робочому просторі і він виконує якісь конкретні задачі. На рисунку 5.1 ми бачимо що у нас є об'єкт MatchDetector з однойменним компонентом закріпленим на ньому. Даний компонент займається розрахунком збігу повторення вправи з відео користувачем і більше нічим, тобто його сфера діяльності обмежена у рамках однієї функціональної діяльності.

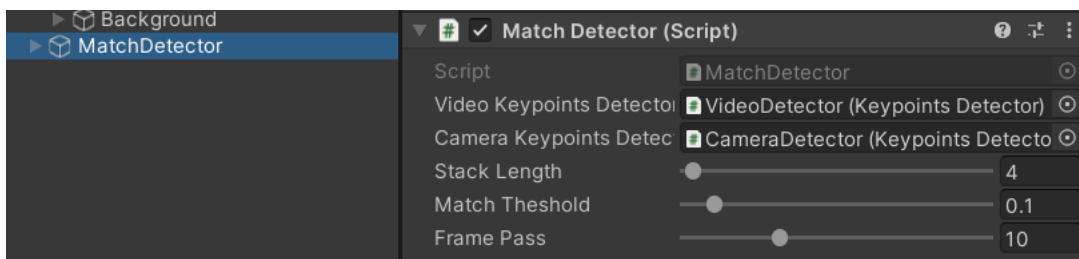


Рисунок 5.1 – Приклад компонентної архітектури проекту

Таким чином і будуються застосунки з використанням середовища Unity – кожний окремий компонент має свою конкретну задачу, і це правильний підхід, бо коли один компонент починає виконувати усі можливі операції проекту він перетворюється у суперклас, тобто отримує повний доступ до всього проекту, а це не повинно допускатись, тому що це порушує основні принципи ООП.

Якщо компонентна архітектура – майже невід’ємна частина кожного Unity проекту, то є деякі особливі рекомендації, щодо створення таких проектів, і яких я дотримуюсь.

Для підтримки компонентної архітектури і збереження принципу єдиної відповідальності використовується фреймворк Zenject – розширення для використання принципу впровадження залежностей у вашому проекті. Багато розробників Unity використовують такий шаблон проектування як Singleton, тобто конкретний компонент повинен існувати у проекті лише в одному екземплярі та до нього повинні мати доступ усі інші компоненти з будь-якого місця програми (рис. 5.2).

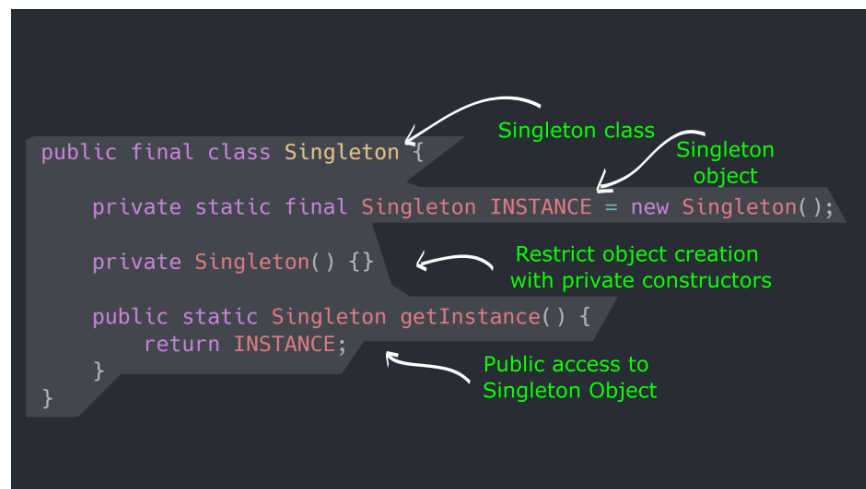


Рисунок 5.2 – Шаблон проектування Singleton

І здається, це вирішує проблему, проте це погана практика. Singleton тримається на статичних полях та значеннях, а це означає, що вони не вивантажуються з пам'яті і завантажують її поки існує об'єкт. Завжди можна захопитись та наробити армію Singleton-ів і потім не розуміти, чому основний потік завантажений та застосунок тормозе. Саме тут допоможе принцип впровадження залежностей, тобто у проекті буде існувати умовний контейнер залежностей, який і буде мати посилання на той чи інший компонент, який може комусь знадобитись, і цей контейнер буде надавати компонентам, які того просять, посилання на потрібний компонент (рис. 5.3). Таким чином ми лише зберігаємо у пам'яті посилання, а не сам об'єкт в цілому. Цей принцип дозволяє нам писати менше коду, а менше коду означає менше шансів, що щось піде не так.

```

private ICameraService cameraService;

@Inject
new *
public void Construct(ICameraService cameraService)
{
    this.cameraService = cameraService;
}

```

Рисунок 5.3 – Приклад використання принципу впровадження залежностей

Ще непоганим є використання такого поведінкового шаблону проектування як команда. Команда – це клас, що створюється під час виконання програми, який використовується для обгортання виклику методу та знищується після завершення. Перевага його полягає в тому, що ми можемо зберігати тимчасову інформацію під час асинхронного виклику як об'єктних змінних. Команди починаються і закінчуються чистим станом і повертаються лише тоді, коли вони мають остаточний результат (дані, успіх/невдача).

Сигнали або події замість безкінечних постійних перевірок. Що мається на увазі. Ну наприклад, ми маємо якусь змінну, значення якої для нас важливо, тому що коли воно більше якогось значення, ми повинні завершити програму. Звичайна практика – кожного кадру перевіряти значення цієї змінної. Проте це дуже сповільнює виконання програми, оскільки уявіть собі коли тридцять разів за секунду ви порівнюєте одне значення з іншим. Тому краще в таких випадках використовувати події. Тобто компонент буде підписуватись на подію, яка відбувається під час зміни значення, або навіть на подію, яка викликається у конкретному стані того самого значення, якщо таких станів потрібно декілька.

Та останнє, проте не менш важливе, чого треба дотримуватись при розробці додатків, це чистота коду. По-перше, у кожній мові програмування є свої правила написання коду та іменування полів, класів та ін. Цих правил треба дотримуватись обов'язково, тому що зазвичай нам доводиться працювати у команді з іншими розробниками, або потім наш код потрапляє у руки інших розробників. Дотримання правил іменування надає змогу швидкого розуміння

коду і допомагає повертатись до дуже давно написаних систем без особливих складнощів. По-друге, треба постійно проводити рефакторинг. Коли здається, що задача виконана, та все начебто працює, слід братися за перегляд усього написаного коду для усунення якихось помилок, спрощення даремно ускладненого коду, та ін.

Така архітектура проекту і дотримання зазначених рекомендацій дозволяє постійно розуміти, що відбувається у проекті, не губитись в ньому, та підвищувати його виробничу спроможність.

5.3 Інтерфейс користувача

Для кожного розробника, звичайно, є дуже важливою архітектура системи, тому що розробник більшу частину часу проводить як раз саме у кодї свого проекту. Однак з боку звичайного користувача, йому абсолютно все одно, що там в середині та як воно працює. Його цікавить те, що він бачить, а саме користувацький інтерфейс.

Користувацький інтерфейс для будь-якої системи не менш важливий ніж її архітектура. Головне для інтерфейсу – зрозумілість та зручність. Користувач повинен тільки побачивши якусь кнопку одразу зрозуміти, яку дію вона виконає та який результат він отримає, та ця кнопка не повинна бути а ні дуже малою, а ні дуже великою. Враження користувача від інтерфейсу дасть нам зрозуміти, чи буде він і далі використовувати додаток чи ні. Також невдало спроектований інтерфейс може призвести до того, що користувач взагалі не знайде якихось важливих функцій, передбачених системою.

У даному додатку, в якості демонстраційної версії, особливого користувацького інтерфейсу немає. Він розроблений мінімалістичним, щоб було максимально зрозуміло. Але все одно розглянемо його.

На рисунку 5.4 зображено початкову сторінку додатку. Дана сторінка є доволі примітивною та в той же час достатньо інформативною. Вона не містить зайвих елементів та зайвої інформації і надає користувачу лише те, що йому потрібно.

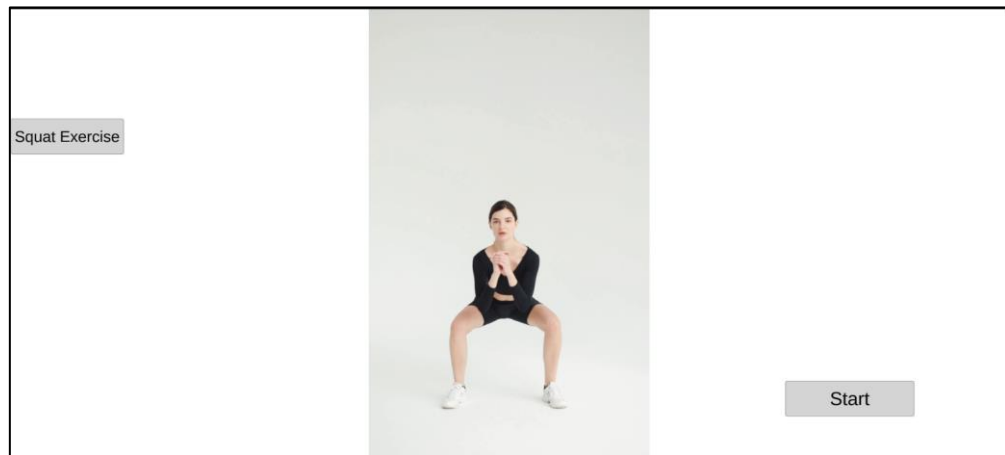


Рисунок 5.4 – Початкова сторінка додатку

Ліворуч ми маємо список кнопок, які відповідають за вибір конкретної вправи. На рисунку ми бачимо, що поки до застосунку додано лише одну вправу, проте система написана так, що додавання нових вправ не матиме ніяких складних дій. Уся інформація про вправу зберігається у об'єктах `ScriptableObject` – це компонент Unity, який є контейнером, що дозволяє нам зберігати достатньо великі розміри даних не маючи прив'язок до конкретного екземпляру класу. Тому додавання нової вправи це лише створення нового `ScriptableObject` та додавання його до списку вправ. Потім програма ініціює створення кнопки, яка матиме зв'язок з конкретною вправою.

По-середині сторінки відображається попередній показ відео з вправою, щоб користувач розумів, що він буде робити. Та праворуч знаходиться одна кнопка “Start” яка відповідає за розпочаток виконання вправи.

На рисунку 5.5 зображена основна сторінка додатку. На цій сторінці ми також маємо показ відео з вправою, але ще до нього додається відстеження рухів: на зображення виконуваної вправи додано відстеження ключових точок, які відображаються у вигляді квадратів зеленого кольору. Праворуч від відео розташовується показ зображення з камери користувача, на якій також застосовується відстеження рухів тіла і відображається таким же чином у вигляді квадратів. Як ми можемо бачити з рисунку шкала виконання знаходиться у відмітці 69%. Таке значення досягається через не зовсім правильне виконання

вправи, тому відношення відстаней точок на відео та зображенні з камери наближається один до одного, просто все ж таки відрізняється.

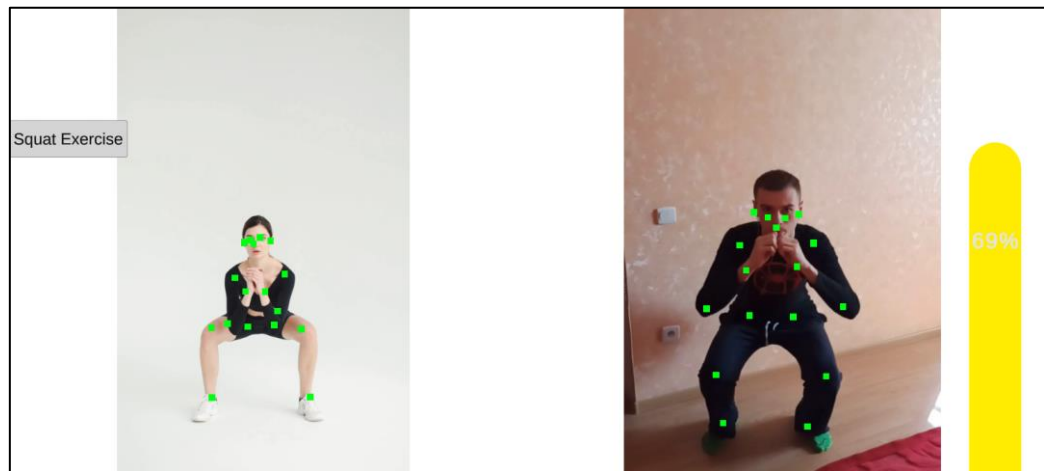


Рисунок 5.5 – Основна сторінка додатку

Перед початком виконання вправи, йде зворотній відлік у три секунду, щоб користувач встиг зайняти стартове положення перед виконанням вправи. Біля правого краю екрану розташований індикатор, який відповідає за правильність виконання вправи, тобто він розраховує позиції ключових точок на відео відносно один одного та на зображенні з камери користувача. Якщо коефіцієнт влучення складає 80% або більше, вправа зараховується. Індикатор також змінює колір від червоного до зеленого залежно від коефіцієнту влучень. Кожного разу, як зараховується вправа, користувач бачить зарахування у вигляді кульки з текстом «+1». Тобто програма розрахована також на підрахунок кількості повторень вправи, тому можна вказувати мінімальну кількість, скільки користувач повинен зробити.

Система написана таким шляхом, що додавання нових елементів як до інтерфейсу так і до архітектури додатку не матиме жодних труднощів, тому доповнення та розширення функціоналу, який вже є, не тільки не буде складним, а і буде підтримуватись. Тобто показник масштабованості системи є доволі великим. Це дозволить підтримувати та розширювати програмний застосунок у майбутньому без якихось труднощів. У випадку, коли додаток знайде свою

аудиторію та буде регулярно використовуватись, це дасть змогу цю аудиторію не втратити.

5.4 Майбутні ітерації

Оскільки програмна система написана таким чином, щоб мати можливість її постійно доповнювати, можна припустити, що він буде покращуватись, шляхом зміни вже існуючих компонентів та додавання нових.

Планується перш за все розширити список вправ додатку, тому що це є основною функціональністю додатку. Також розглядається можливість розділення вправ на класи за групами м'язів, або іншими характеристиками.

Оскільки зараз це просто відображення вправи і повторення за відео, слід перед вибором вправи додати також відображення опису вправи: яка саме правильна техніка виконання у теорії, які групи м'язів працюють, як людина повинна себе почувати під час виконання та ін.

Також оскільки зараз усі вправи зберігаються безпосередньо у додатку, то з ростом їх кількості буде також рости розмір самого додатку, що не є гарною ідеєю, бо користувачу важливий простір, який займає додаток. Тому розглядається можливість переносу усієї можливої інформації на хмарні сервіси, які дозволять завантажувати лише потрібну інформацію безпосередньо під час роботи. Також потім планується надати можливість користувачу зберігати обрані вправи, щоб наступного разу їх не потрібно було завантажувати знов.

Щоб алгоритм краще себе поведив на мобільних пристроях та з швидкою інтенсивністю рухів, можливо у майбутньому розглянути можливість заміни поточного алгоритму на інший. Наприклад, розглянути вже готові методи відстеження зображень які пропонуються у Tensorflow MobileNet, що розроблені спеціально для мобільних пристроїв. Особливістю даної архітектури є відсутність max pooling-шарів. Замість них зниження просторової розмірності використовується згортка з параметром stride рівним 2. Цей алгоритм налаштовується параметрами ширини сітки та її роздільної спроможності. Їх налаштування дозволяє коригувати точність та швидкість алгоритму: чим вони

менше, тим менше точність, проте більше швидкість. Це дозволить обирати оптимальні варіанти залежно від задач, для яких використовуються ці алгоритми. З MobileNetV2 мобільні розробники отримали майже необмежений інструментарій в області комп'ютерного зору - крім простих моделей для класифікації зображень, тепер ми можемо використовувати прямо на мобільному пристрої алгоритми детектування об'єктів і семантичної сегментації.

На даний час інтерфейс додатку є максимально примітивним з використанням стандартних вбудованих Unity UI елементів, які добре працюють, та не дуже виглядають. Тому у майбутньому планується повна переробка інтерфейсу, щоб зробити його не тільки зрозумілим та функціональним, а ще й гарним. Для цього Unity має новий інструмент, який називається UI Toolkit (UI Builder). Даний інструмент дозволяє гучко налаштовувати елементи користувацького інтерфейсу по типу розробки інтерфейсу веб-застосунку, використовуючи аналог мови розмітки HTML. До того ж цей інструмент має глибоку підтримку таблиць стилів, що дозволить створити майже будь-який елемент інтерфейсу, майже будь якого вигляду. Це дозволить створити максимально сучасний інтерфейс, який буде залучати нових користувачів. До того ж використання UI Toolkit дозволить використовувати ще і нову систему анімацій для користувацького інтерфейсу, яка зробить поведінку окремих елементів більш привабливою та приємною для зору користувача, тому він на підсвідомому рівні буде хотіти використовувати цей додаток для отримання певного естетичного задоволення.

На мій погляд додаток має потенціал до розширення, тому і запропоноване саме таке використання розглянутих алгоритмів відстеження рухів людського тіла.

6 РЕЗУЛЬТАТИ РОБОТИ

У результаті виконання роботи були розглянуті чотири методи відстеження рухів людського тіла, один з яких використовується лише для технологій віртуальної реальності та потребують додаткових датчиків для відстеження позиції. З останніми трьома методами були проведені експерименти їх працездатності та якості їх роботи.

Було отримано, що найточнішим є метод відфільтрованих ключових точок, тому що у комбінації з камерою глибини цей алгоритм надає найточніше визначення точок, та найбільш плавний результат роботи, тобто найбільшу кількість кадрів за секунду. Однак також було визначено, що точність методів сильно залежить від інтенсивності рухів людини. Пониження точності відстеження рухів з ростом інтенсивності пов'язане з неможливістю алгоритмів так швидко обробляти кожен кадр, тому зображення на кожному сусідньому кадрі, що аналізуються, дуже відрізняються, та немає можливості точно передбачити місцезнаходження ключової точки.

Представлений підхід використовує моделі руху, що дозволяє сформулювати завдання відстеження як задачу мінімізації диференціальної цільової функції щодо невеликої кількості параметрів. Ми приймаємо їх за перші кілька коефіцієнтів основних компонентів простору суглобових кутів для рухів, захоплених за допомогою оптичного пристрою захоплення руху.

Використовуючи різну інтенсивність руху, було показано, що це уявлення, хоча має досить низьку розмірність, проте має досить багату диференціальну структуру, щоб забезпечити гарну продуктивність при низьких обчислювальних витратах. Він також може захоплювати перехід від одного руху до іншого. Цей метод ідеально підходить для аналізу спортивних дій, таких як замахи у гольфі чи подача у тенісі [17]. Те саме можна сказати і про реєстрацію руху ортопедичних пацієнтів, коли їх просять виконати певну процедуру, призначену для оцінки їхнього стану.

Також було виявлено, що найменше виробнича спроможність у методи відстеження з урахуванням окклюзії. Даний метод має поганий показник середньої кількості кадрів за секунду, оскільки алгоритм визначення окклюзії займає багато ресурсів, тому метод не підходить для використання у застосунках, де потрібне постійне відстеження тіла.

Ми можемо з повною впевненістю казати, що якби ми проводили експерименти також з методом відстеження з датчиками Vive Tracker, то саме цей би метод показав найкращі результати. Та в цьому немає нічого дивного, оскільки технологія отримання позиції людини зовсім інша. Висока точність досягається тим, що по периметру тестової кімнати розставляються випромінювачі, їх ще називають маяками, та датчики, які людина розміщує на собі, отримує сигнали випромінювача. Але такі технології не мають дуже великого практичного застосування, тому що не є достатньо мобільними. Дана перевага у точності може бути значної, якщо розглядати стаціонарну середу для відстеження.

Метод згорткової мережі використовувався у розробці тестового додатку.

Вибір пав саме на цей метод з запропонованих, тому що все одно метод відфільтрованих точок хоч і точніший і має кращі показники роботи, та він потребує даних з камери глибини, а такі пристрої є не у кожного.

Хоча метод згорткової мережі не має можливостей для розрахунку окклюзії, нам цього і не потрібно, оскільки відсутність подібних розрахунків прискорює роботу методу та як результат дає більш стабільну кількість кадрів за секунду.

Загалом, результати роботи дають нам зрозуміти, що для кожної окремої задачі можуть підходити різні методи та алгоритми відстеження тіла. Все залежить від конкретних цілей, які система повинна досягати та у яких умовах її будуть використовувати.

Розглянуті методи є не найточнішими серед методів відстеження рухів людського тіла, та їх виконання не потребує великих ресурсів від пристрою, на якому вони запускаються. Тому дані алгоритми непогано підходять для непотужних комп'ютерів та різноманітних мобільних пристроїв.

Такий опір на методи відстеження рухів людського тіла які дуже не навантажують систему дозволив нам створити застосунок, який може бути використаний як на ноутбучі так і на мобільному пристрої.

Розроблений додаток був зібраний для платформи WebGL, яка у свою чергу має певні обмеження у ресурсах, та програма показали добру працездатність. Що свідчить про те, що програма не має жорстких обмежень з потужності. Проте слід зауважити, що програма має налаштування роздільної здатності вихідного формату зображення, і якщо значно збільшити це значення, то і кадри за секунду де що втраяться.

Обрана архітектура програмної системи дозволила без зайвих труднощів вносити зміни налаштувань компонентів та користувацького інтерфейсу.

Створений програмний застосунок – результат аналіз обраних методів та проведення над ними експериментів. Додатки подібного формату, як вже і було зазначено, можуть бути використані у найрізноманітніших сферах діяльності. Якщо навіть у розробленому додатку замінити відео з вправою на трансляцію з камери з іншого пристрою, вже вийде інший додаток, який спеціалізувався на заняттях з репетитором.

Тому отримані результати є доволі важливими у розумінні того, як може бути застосований той чи інший метод відстеження рухів людського тіла.

ВИСНОВКИ

У даній роботі розглянуто методи, що використовуються для відстеження тіла людини у просторі. Кожен метод був детально описаний з наведенням всіх формальних моделей та зв'язків. Також кожен метод несе за собою окреме призначення та буде використовуватися у різних задачах.

Було показано, що у відповідності до поставленого завдання для вирішення, необхідно обирати конкретні методи, які поділяються на ті, що використовують різноманітні додаткові датчики та пристрої, та ті, яким достатньо лише однієї камери.

Отримано дані, що методи відстеження тіла за допомогою додаткових датчиків, надають найбільш точні результати. Проте також було виявлено під час виконання роботи, що і з використанням методів зчитування тіла з зображення шляхом використання методів комп'ютерного зору можна досягти достатньо великої точності.

Були проведені експерименти з трьома методами комп'ютерного зору для відстеження рухів людського тіла: метод згорткової нейронної мережі, метод відстеження з врахуванням оклюзії та метод відфільтрованих ключових точок. У результаті проведення експериментів були отримані певні характеристики, якими порівнюються зазначені методи.

З результатів було зроблено висновок, що точність кожного з алгоритмів напряму залежить від інтенсивності рухів людини: чим швидше рухи, тим нижче є точність.

Найбільш точним алгоритмом комп'ютерного зору з наведених для відстеження людського тіла виявився метод, який використовував зображення з камери глибини у комбінації з використанням множинних гіпотез. Це обумовлено тим, що зображення глибини дозволяє нам значно скорегувати двовимірне відображення тіла, та накласти його на наше зображення, перетворивши у тривимірне, а в свою чергу множинні гіпотези використовуються для згладжування даних та підгонки їх до більш точних та сприятливих.

Метод відстеження рухів з врахуванням оклюзії виявився самим несприятливим до використання через його малу виробничу спроможність, тобто низьку кількість кадрів за секунду під час використання. Найкращий результат показав також метод відфільтрованих ключових точок, та його використання обмежується потребою використовувати його у комбінації з камерою глибини.

Метод згорткової нейронної мережі показав не самі погані результати точності, та непогані результати середньої кількості кадрів за секунду, що сприяло його використанню під час розробки тестового додатку.

Був розроблений додаток для виконання різноманітних фізичних вправ шляхом повторення їх за відео. Було отримано, що дійсно метод згорткової нейронної мережі є достатнім для реалізації додатку подібного функціоналу. Метод добре виконує поставлену задачу та надає можливість точно повторювати елементи вправ з відео.

Були визначені майбутні ітерації проекту, які обумовлені його перспективністю та гарно спроектованою архітектурою, яка робить додаток добре масштабованим та дозволяє легко додавати нову функціональність.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Путятін Є. П. Методи та алгоритми комп'ютерного зору : навч. посіб. / Є. П. Путятін, В. О. Гороховатський, О. О. Матат. – Харків : СМІТ, 2006. – 236 с. : іл. – ISBN 966-8530-74-8. – 28,00
2. Szeliski, R. Computer Vision: Algorithms and Applications [Текст] / R. Szeliski. – Springer, 2010. – 979 с.
3. Klette R. Concise Computer Vision: An Introduction into Theory and Algorithms [Текст] / Reinhard Klette. – Лондон.: Springer, 2014. – 429 с.
4. Шапиро Л., Стокман Д. Компьютерное зрение / Пер. с англ.-М.:БИНОМ. Лаборатория знаний, 2006. – 752 с
5. Aggarwal, J. K., & Cai, Q. (1997, June). Human motion analysis: A review. Proceedings of the Nonrigid and Articulated Motion Workshop '97 (pp. 90-102).
6. Rong P., Sichitiu M.L., Angle of Arrival Localization for Wireless Sensor Networks // 3rd Annual IEEE SECON, vol. 1, 2006, pp. 374–382.
7. Miguel B., Symington A., Coltin B. HTC Vive: Analysis and Accuracy Improvement // International Conference on Intelligent Robots and Systems, October 1-5, 2018. – Madrid, Spain, 2018.
8. Fausett, L. V. Fundamental of Neural Networks: Architectures, Algorithms, and Applications – Paris: Prentice-Hall, 1993. – 463 с
9. Agarwal A., Triggs B. Recovering 3D human pose from monocular images // IEEE Transactions on Pattern Analysis and Machine Intelligence, 28. 2006 – pp. 44-58.
10. Ruban I., Smelyakov K., Martovytskyi V., Pribyl'nov D., Bolohova N. Method of neural network recognition of ground-based air objects // IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT), 24-27 May. – 2018. – P. 589-592.
11. Qammaz A., Argyros A.A. Occlusion-tolerant and personalized 3D human pose estimation in RGB images. International Conference on Pattern Recognition (ICPR 2020), January, 2021

12. Moravec H. Rover visual obstacle avoidance // In International Joint Conference on Artificial Intelligence, Vancouver, Canada, 1981, pp. 785-790.

13. Makris A, Argyros A.A. Robust 3D Human Pose Estimation Guided by Filtered Subsets of Body Keypoints // Machine Vision Applications. – Tokyo, Japan, 2019.

14. Білоус Н. В., Красов А. И., Власенко В. П. Видалення низькочастотної складової зображення з використанням медіанного фільтру // Журнал інженерних наук. 2016. Том 3 №2.

15. Мельнікова Р.В., Кулигін А.С. Дослідження моделей та методів комп'ютерного зору для відстеження рухів людського тіла. Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління. Тези доповідей дванадцятої міжнародної науково-технічної конференції. Том 2: секція 5. 27-28 квітня 2022, м. Харків, Україна, с. 142.

16. Unity – Manual: Unity User Manual 2020.3 (LTS). URL: <https://docs.unity3d.com/2020.3/Documentation/Manual/index.html> (дата звернення 07.02.2022)

17. Bertini, M., Del Bimbo, A., & Nunziati, W. (2003, November). Model checking for detection of sport highlights. Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval (pp. 215-222)