

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
Завідувач кафедри
_____ Скарга-Бандурова І.С.
« ____ » _____ 20__ р.

МАГІСТЕРСЬКА РОБОТА

НА ТЕМУ:

Методи та модель розпізнавання жестів рук для керування комп'ютером

Освітній ступінь “Магістр”
Спеціальність 123 “Комп’ютерна інженерія”

Науковий керівник роботи:

(підпис)

В.М.Барбарук

(ініціали, прізвище)

Консультант з охорони праці:

(підпис)

Я.О.Критська

(ініціали, прізвище)

Студент:

(підпис)

О.О.Швачка

(ініціали, прізвище)

Група:

КІ-17ДМ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки

Кафедра Комп'ютерних наук та інженерії

Освітній ступінь магістр

Напрямок підготовки _____

(шифр і назва)

Спеціальність 123 "Комп'ютерна інженерія"

(шифр і назва)

ЗАТВЕРДЖУЮ:

Завідувач кафедри _____

І.С. Скарга-Бандурова

« _____ » _____ 20 ____ р.

**З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ**

Швачці Олександру Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Методи та модель розпізнавання жестів рук для керування комп'ютером

керівник проекту (роботи) Барабрук Віктор Миколайович, к.т.н., доц.

(прізвище, м. 'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «18» 10 2018 р. № 221/48

2. Строк подання студентом роботи 10.01.2018

3. Вихідні дані до роботи Матеріали науково-дослідної практики, алгоритми по розпізнаванню жестів, сенсор Kinect, середа розробки Microsoft Visual Studio 2013, бібліотека OpenCV

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Огляд предметної області та постановка завдання, огляд існуючих методів та розроблення нового методу, машинне навчання для побудови класифікатора, програмна реалізація, охорона праці та безпека в надзвичайних ситуаціях, висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Електронні плакати

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці та безпека в надзвичайних ситуаціях	Критська Я.О. ст. викл. кафедри КНІ		

7. Дата видачі завдання 18.10.2018

Керівник

_____ (підпис)

Завдання прийняв до виконання

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Отримання завдання на дипломне проектування	10.09.2018-15.09.2018	
2	Аналіз завдання, робота з літературою	16.09.2018-22.09.2018	
3	Розробка власного методу	26.09.2018-06.10.2018	
4	Розробка програмної системи	07.10.2018-25.10.2018	
5	Тестування програмної системи	25.10.2018-25.11.2018	
6	Розробка частини проекту "Охорона праці та безпеки в надзвичайних ситуаціях"	26.11.2018-1.12.2018	
7	Оформлення пояснювальної записки, автореферату та презентації	2.12.2018-09.01.2019	

Студент

_____ (підпис)

Швачка О.О.

_____ (прізвище та ініціали)

Науковий керівник

_____ (підпис)

Барбарук В.М.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Швачка О.О. Методи та модель розпізнавання жестів рук для керування комп'ютером.

Робота присвячена розробці та аналізу існуючих методів розпізнавання жестів. Були досліджені комплексні системи для взаємодії людини з комп'ютером. Проведено аналіз по дослідженим системам. Розроблено метод розпізнавання жестів. В основі методу лежить пошук дескрипторів на карті глибини, які однозначно описують жест та використання машинного навчання для класифікації жестів. Розроблений метод використано для безконтактної взаємодії з комп'ютером.

Ключові слова: розпізнавання жестів, сегментація, карта глибини, дескриптори, випукла множина, контур, центр мас, людина – комп'ютерний інтерфейс.

ABSTRACT

Shvachka O.O. Methods and model of recognition of hand gestures for computer control.

The work is devoted to the development and analysis of existing methods for recognizing gestures. Integrated systems for human interaction with the computer were explored. The analysis on the investigated systems is carried out. A method for recognizing gestures has been developed. At the heart of the method is the search for descriptors on the depth map that uniquely describe the gesture and the use of machine learning for the classification of gestures. The developed method is used for contactless interaction with the computer.

Keywords: gesture recognition, segmentation, depth map, descriptors, convex set, contour, center of mass, human - computer interface.

ЗМІСТ

ВСТУП.....	4
1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ	5
1.1 Аналіз існуючих рішень.....	5
1.1.1 Microsoft Kinect.....	5
1.1.2 Leap Motion	7
1.1.3 Intel Perceptual Computing SDK.....	8
1.1.4 Nintendo Wii Remote	9
1.1.5 Розпізнавання жестів по WiFi.....	9
1.1.6 Програмні засоби.....	10
1.2 Аналіз і висновки по існуючих рішеннях	11
1.3 Постановка задачі дослідження.....	12
2 ОГЛЯД ІСНУЮЧИХ МЕТОДІВ ТА РОЗРОБЛЕННЯ НОВОГО МЕТОДУ	14
2.1 Розпізнавання позиції та орієнтації за допомогою моментів зображення.....	14
2.2 Розпізнавання рухів.....	15
2.21 Розпізнавання рухів на основі аналізу різниць зображень.....	15
2.22 Розпізнавання рухів на основі метода Віюли - Джонса	15
2.23 Розпізнавання рухів на основі кольорових рукавичок	18
2.24 Розпізнавання рухів на основі аналізу контуру зображення руки 19	19
2.25 Розпізнавання жестів руки із застосуванням штучних ШНМ... 19	19
2.26 Розпізнавання жестів руки із застосуванням прихованих моделей Маркова.....	21

2.2.7 Розпізнавання позиції і конфігурації руки методом випадкових лісів	21
2.2.7 Розпізнавання жестів як задача оптимізації	22
2.3 Аналіз досліджених методів	24
2.4 Запропонований підхід	25
2.4.1 Сегментація зображення	25
2.4.2 Дескриптори жесту	27
3 МАШИННЕ НАВЧАННЯ ДЛЯ ПОБУДОВИ КЛАСИФІКАТОРА	33
3.1 Основні підходи	33
3.2 Дерево рішень	35
4 ПРОГРАМНА РЕАЛІЗАЦІЯ	38
4.1 Реалізація основних частин	38
4.2 Робота з Kinect	40
4.3 Аналіз результатів	43
4.4 Навчання класифікатора	54
5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ. ЕКОЛОГІЯ	58
5.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проектowanego об'єкту, що мають вплив на персонал	58
5.2 Заходи щодо техніки безпеки	60
5.3 Заходи, що забезпечують виробничу санітарію і гігієну праці	62
5.4 Рекомендації по пожежній безпеці	65
ВИСНОВКИ	70
ПЕРЕЛІК ПОСИЛАННЯ	71
ДОДАТОК А. Електронні плакати	73

ВСТУП

З часів появи персональних комп'ютерів, була необхідність розробити можливість простого взаємозв'язку з ним, як наслідок розробили комп'ютерну мишу (механічний маніпулятор, що перетворює рух у керуючий імпульс), з тих пір пройшло досить багато часу, але світ не стоїть на місці і тому розробляються нові методи інтерактивного зв'язку користувача з комп'ютером. Так проста кулькова однокнопочна миша переросла в бездротову лазерну з безліччю кнопок, що дозволяють добитися простоти управління комп'ютером. Але у миші, є ряд недоліків, наприклад для роботи потрібна рівна гладка поверхня, небезпека виникнення синдрому зап'ястного каналу, нестійка до вібрацій, не зручно носити з собою, останню проблему вирішили за допомогою тачпада. У процесі еволюції взаємодії людини з комп'ютером з'явилася нова технологія - Touchscreen (пристрій введення, що представляє собою екран, який реагує на дотики до нього), яку пізніше вдосконалили до Multitouch (дозволяє розпізнавання відразу декількох дотиків). Але й у цих технологій є ряд недоліків: на горизонтальному екрані руки перекривають огляд, використання затруднюється з забрудненими руками і т.д. Сьогодні дуже прогресивно розвиваються такі технології як віртуальна реальність, голосове управління і управління за допомогою жестів рук.

Жести руки - основа людино-машинної взаємодії, управління за допомогою жестів може доповнити стандартні пристрої введення або взагалі їх замінити, тому ця тема дуже актуальна.

Актуальність теми зумовлена тим, що світ постійно розвивається і людство намагається вирішити проблеми які трапляються перед ним, тому людство намагається знайти кращий спосіб взаємодіяти з комп'ютером. При цьому розвиваються ще й математичний апарат.

1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Аналіз існуючих рішень

1.1.1 Microsoft Kinect

Kinect[1]— безконтактний сенсорний ігровий контролер, спочатку представлений для консолі Xbox 360, і значно пізніше для персональних комп'ютерів під керуванням ОС Windows. Розроблений фірмою Microsoft. Заснований на додаванні периферійного пристрою до гральної консолі Xbox 360, Kinect дозволяє користувачеві взаємодіяти з нею без допомоги контактного ігрового контролера через усні команди, пози тіла, об'єкти або малюнки. Мета проекту — збільшити число користувачів Xbox 360. Kinect для консолі Xbox був вперше представлений 1 червня 2009 року на виставці E3. У той же день фірма Microsoft розіслала розробникам програм більше 1000 комплектів засобів розробки Kinect. Kinect вийшов для консолі 4 листопада 2010 року в США і 10 листопада в Європі. Поставки версії для Windows були розпочаті 1 лютого 2012 року.



Рисунок 1.1 - Kinect сенсор

Kinect — це горизонтально розташована коробка на невеликій круглій підставці, яку розташовують вище або нижче екрана. Розміри — приблизно 23 см в довжину і 4 см у висоту. Складається з двох сенсорів глибини, кольорової відеокамери та мікрофонної решітки. Програмне забезпечення здійснює повне 3-х вимірне розпізнавання рухів тіла, міміки особи та голосу. Датчик глибини складається з інфрачервоного проєктора, об'єднаного з

монохромною КМОП-матрицею, що дозволяє датчику Kinect отримувати тривимірне зображення при будь-якому природному освітленні.

Діапазон глибини та програмне забезпечення дозволяють автоматично калібрувати датчик з урахуванням умов гри і навколишніх умов, наприклад меблів, що знаходяться в кімнаті.



Рисунок 1.2 - Це ІЧ зображення, показане лазерною сіткою Kinect, яка використовується для розрахунку глибини

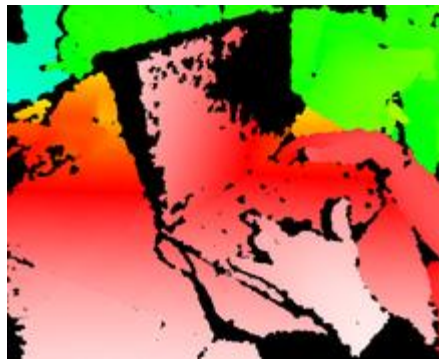


Рисунок 1.3 - Карта глибини; використовується для відображення: колірний градієнт від білого (ближче) до синього (далі)

Судячи з патентом Microsoft, Kinect буде здатний розпізнавати мову жестів. Поки патент стосується лише ASL, але, можливо, інші мови будуть додані пізніше. Очікується, що це розширить аудиторію користувачів та допоможе навчати німих мові жестів. Однак, згідно з офіційним коментарем, ця особливість не буде включена в першу версію Kinect за зниженою ціною. З іншого боку, Microsoft не відмовляється від використання патенту — але чи буде це поліпшена версія Kinect, або окремий продукт, поки невідомо.

1.1.2 Leap Motion

Датчики руху зробили великий стрибок вперед завдяки Leap Motion[2]. Leap Motion створив новий пристрій, що використовує датчик руху, який зводить нанівець використання комп'ютерної миші. Все, що необхідно для управління комп'ютером є те, чим ви користуєтеся щодня - ваші руки.

Контролер Leap Motion являє собою невеликий пристрій, який підключено до комп'ютера за допомогою кабелю USB. Контролер становить близько 3 дюймів у довжину, 1-дюймовий широкий, і 1/2-дюймовий товщиною і має nonskid гумове дно. Вона складається з трьох камер і інфрачервоних датчиків, які виявляють рух десяти пальців до 1/100-го міліметра і має 150° діапазон зору. З виявлення, що точна, вона здатна створювати точні картини і малюнки на екрані комп'ютера. 150° діапазон зору дозволяє користувачам переміщати свої руки в 3D. 3D - це означає, що вони в змозі використовувати додатки Leap Motion, щоб захопити і переміщати об'єкти, як якщо б у реальному житті. Існує вісім кубічних футів простору над контролером, який використовується для управління комп'ютером. Контролер також переміщається при переміщенні тому він відслідковує рух зі швидкістю більше 200 кадрів в секунду (Leap Motion, 2013).

Процес установки простий і зручний. Підключіть контролер Leap Motion в порт USB, а потім помістити контролер або в передній частині клавіатури для ноутбука або між екраном і клавіатурою для настільного комп'ютера (Stein, 2013). Після цього, завантажити програмне забезпечення з сайту Leap Motion. Потім програмне забезпечення показує швидкий вступ про те, як контролер відстежує всі десять пальців і віртуальний простір над пристроєм. Після вступу його потім переходить до використання повітряного простору будинку, який є воротами в повітряному просторі магазину, який є магазин додатків спеціально для контролера Leap Motion. Повітряний простір має близько 75 додатків, які варіюються від ігор, музичних інструментів, і продуктивності. Після завантаження програми на комп'ютер він здатний використовувати контролер Leap Motion.

1.1.3 Intel Perceptual Computing SDK

Intel perceptual computing sdk - це нова технологія, яка дозволить користувачеві взаємодіяти зі своїм мобільним пристроєм за допомогою голосу, рук, обличчя, тобто більш природних для людини способів взаємодії. В даний час компанія intel веде роботу в цьому напрямку, на сайті компанії з'явилася перша версія perc sdk, яка підтримувала такі можливості:

- розпізнавання жестів, положення рук, пальців, рівень відкритості долоні;
- відстеження положення обличчя, визначення контрольних точок (губи, ніс, очі), ідентифікація («впізнавання») особи;
- відстеження положення 2-х і 3-х мірних об'єктів.

У 2013 році Perc випустила свій перший продукт, Senz3d, камера глибини у партнерстві з Creative Technology. Камера схожа на Microsoft kinect, але вона працює на невеликих відстанях. До складу камери входять два мікрофони, що дозволяють значно поліпшити якість розпізнавання мови, вбудовану RGB і інфрачервону камери.



Рисунок 1.4 – Камера Creative Senz3d

1.1.4 Nintendo Wii Remote

Wii Remote (іноді назив. “Wiimote”) - основний контролер для гальної консолі Wii компанії Nintendo. Головною особливістю Wii Remote в порівнянні з конкурентами є детектор руху, що дозволяє гравцям управляти персонажами гри і предметами на екрані рухом руки, або «вказуючи» на об'єкти, що відображаються на екрані. Це забезпечується роботою акселерометра і світлочутливої матриці. Крім цього, до контролера можна підключати різні пристрої, розширюючи таким чином його можливості.

Wii Remote був вперше продемонстрований 16 вересня 2005 року на виставці Tokyo Game Show. З тих пір до контролера було залучено багато уваги через його унікальні особливості і значні відмінності від звичайних ігрових контролерів. Він також отримав значну увагу від хакерів, які використовують його для управління різноманітними пристроями, не пов'язаними з Wii, за допомогою Wii homebrew.

1.1.5 Розпізнавання жестів по WiFi

Не секрет, що випромінювання WiFi дозволяє використовувати високоточні радари для вимірювання руху людей: на цьому принципі був заснований перший пасивний WiFi-радар, створений у минулому році фахівцями з Лондонського університетського коледжу. Радар сканує приміщення крізь стіни і визначають точне місце розташування людей. Передбачалося, що пристрій стане в нагоді правоохоронним органам і армії.

Таким чином, достатньо всього лише точно реєструвати частоту відбитого сигналу і час його повернення, щоб знати швидкість і відстань до об'єкта, що рухається. Якщо навчитися робити це досить точно, то можна навіть реєструвати пульс і частоту дихання людини!

В Вашингтонському університеті продовжили роботу англійських фахівців - і зробили більш просунутий варіант пасивного WiFi-радара, який здатний розпізнавати навіть окремі жести людини в зоні дії хотспот.

Іншими словами, технологія може знайти мирне застосування в інтерфейсах управління комп'ютерною технікою і побутовими електронними приладами, освітленням та іншими системами електронного будинку.

В даний момент система WiSee[3] розпізнає дев'ять жестів з точністю 94%.

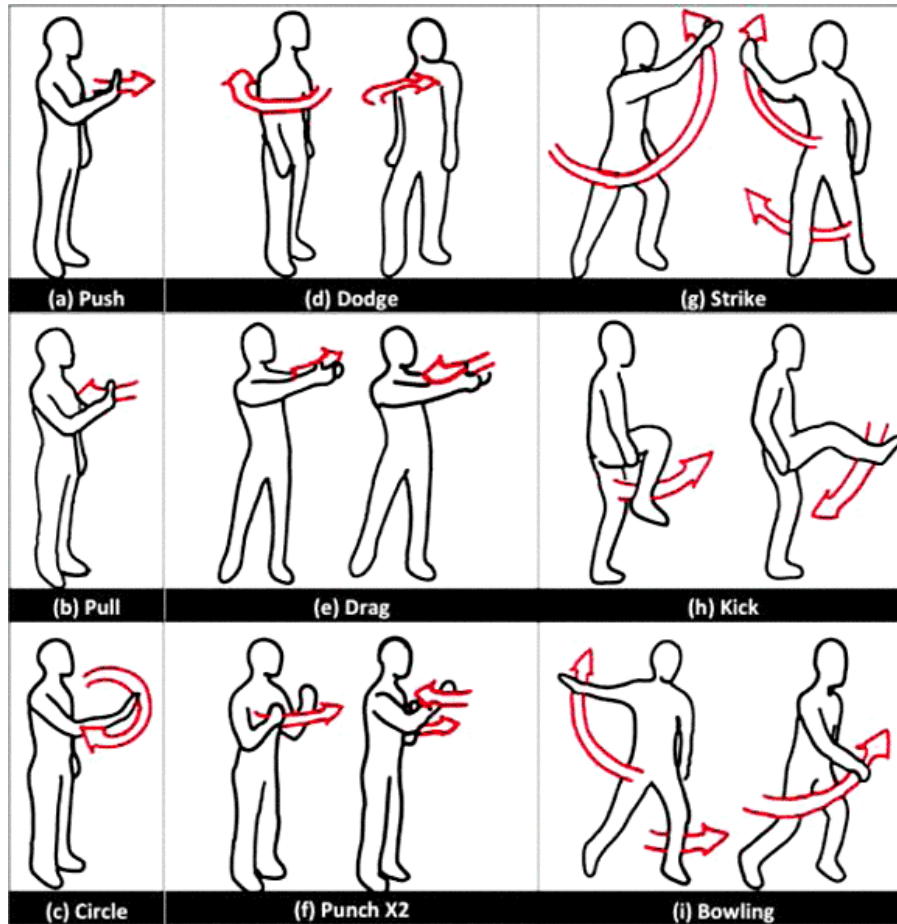


Рисунок 1.5 – Жести які розпізнає WiSee система

В експериментальній установці приймач зібраний з декількох універсальних приймачів USRP-N210, кожен працює на своїй частоті, сигнал обробляється в програмі GNURadio.

1.1.6 Програмні засоби

Flutter - програма яка дозволить вам контролювати iTunes або Windows Media (і інші сумісні медіаплеєри, такі як Winamp, QuickTime, VLC, Spotify і т. д.) за допомогою веб-камери. У жовтні 2013 стартап Flutter був куплений компанією Google, так що, ймовірно, він отримає бурхливий розвиток в найближчому майбутньому. Існують і альтернативи програмі Flutter: PointGrab HGRS, XTR3D, EyeSight PC і NPointer .

За допомогою NPointer ви зможете управляти не тільки музикою і відео але і використовувати її замість миші або тачпада, виконуючи жести руками, головою і іншими частинами тіла. NPointer здатний виявити будь-який рухомий об'єкт і синхронізувати його координати з покажчиком миші. Програма пропонує масу налаштувань, за допомогою яких ви зможете змінити швидкість руху, прискорення і інші параметри. Однак суттєвий її мінус в тому, що вона працює тільки під Windows і не підтримується Mac OSX. .

1.2 Аналіз і висновки по існуючих рішеннях

Після ознайомлення з системами які є на ринку, можна помітити те, що, взагалі не так багато рішень які вирішують проблеми взаємодії людини з комп'ютером. Навіть ті що є, важко порівняти, тому що, кожна з цих систем вирішує якусь одну проблему, точніше сказати вони рішають трохи різні задачі.

Наприклад Kinect дуже якісна система, але на даному етапі свого розвитку вона має не так багато функцій. Основні це skeletal tracking, seated skeletal tracking и face tracking. Насправді вона може ще розпізнавати голос, але розпізнавання голосу виходить за рамки даної роботи, тому я не буду на цьому зупинятися. Найбільше всього цікавить функція трекінгу людини та трекінгу обличчя.

Трекінг людини - це не просто розпізнавання людини на кадрі, а так-же спостереження за рухом. Kinect може відстежувати до шести осіб, але тільки для двох будувати скелет. Ми отримуємо ключові точки людини, тобто скелет.

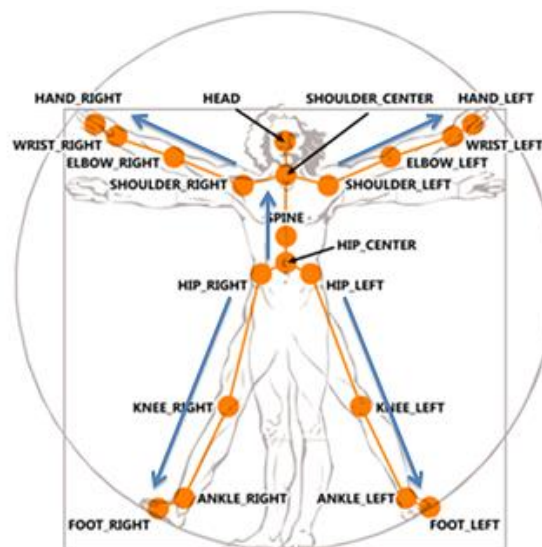


Рисунок 1.6 – 20 основних точок

Хочу звернути увагу на те, що для того щоб Kinect зміг відстежувати людину, людині потрібно стояти на певній відстані від Kinect, що іноді може бути не зручно. Побудувати всі 20 точок іноді не представляється можливим. Так з'явився режим, званий трекінг сидячої фігури (seated skeletal tracking). У цьому режимі сенсор будує не повний 10-ти вузловий скелет. Крім відстеження всього тіла, можна відстежувати обличчя. Kinect будує 89 точок для обличчя. До речі, кажучи точніше, відстеженням займається не сама приставка Kinect, а SDK яке входить до неї і яке можна скачати на сайті виробника.

Якщо вам потрібно специфічні функції їх можна реалізувати на базі системи Kinect, використовуючи відкриту бібліотеку комп'ютерного зору – OpenCV. Використовуючи офіційне SDK не можна буде розпізнати жести руки, можливо тільки отримати її координати. Алгоритми розпізнавання окремих жестів потрібно реалізовувати самому. У цілому система хороша тільки для відстеження фігури людини та її руху.

Intel Perceptual Computing SDK розробка дуже схожа на розробку Microsoft Kinect, але чомусь мало відома, можливо це пов'язано з маркетингом.

Leap Motion спрямована на відстеження мікро-жестів руки. Як на мене це дуже прогресуюча розробка для відстеження руки, але вона не може працювати на великих відстанях, руки потрібно тримати над контролером.

Програмні засоби такі як – Flutter, PointGrab HGRS, XTR3D, EyeSight PC і NPointer взагалі вирішують дуже вузькі проблеми, їх навіть не порівняти з системами такими як Kinect та Leap Motion.

Отже як бачимо проблема взаємодії людини з комп'ютером досить таки не тривіальна, різні розробники по різному підійшли до цієї проблеми, але комплексного рішення ще нема, це також зумовлено тим, що обробка зображень дуже складна область та її потрібно продовжувати досліджувати.

1.3 Постановка задачі дослідження

Людино-комп'ютерна взаємодія – це міждисциплінарний науковий напрям, що вивчає і вдосконалює методи розробки інтерактивних комп'ютерних систем, призначених для зручного використання людиною.

HCI (англ. human-computer interaction) отримало розвиток в контексті різних дисциплін: комп'ютерної графіки, когнітивної науки, інформатики і багато ін.

У зв'язку з тим що, HCI вивчається як з людського боку, так і з боку комп'ютера, знання які отримані в дослідженні, опираються як на людський фактор, так і на комп'ютерний. З комп'ютерного боку важливі технології: комп'ютерна графіка, мови

програмування, операційні системи . З людського боку: теорія комунікації, когнітивна психологія, соціологія. Так само має значення інженерія та проектування.

Основне завдання НСІ поліпшити взаємодію між людиною і комп'ютером. Кінцева мета НСІ, створити якісний інтерфейс, який можна назвати точкою зв'язку між людиною і комп'ютером.

В останні роки широке поширення отримав один з методів НСІ, заснований на розпізнаванні жестів людини. Жест - рух (дія) для вираження емоції чи інформації замість розмови, або під час розмови. Дія або рух людського тіла, або його частини, що має певне значення, або сенс, тобто є знаком або символом.

Рух пальців, рук, голови, плечей, міміка обличчя, все це - жести. У НСІ, жестами передають інформацію в комп'ютер. Розпізнавання жестів можна застосовувати в управлінні комп'ютером і побутовими приладами, створення природних людино-машинних інтерфейсів для глухонімих, маніпуляція тривимірними об'єктами, додатки віртуальної реальності. Існуючі методи розпізнавання жестів руки можна розбити на дві великі категорії:

- методи, засновані на аналізі зовнішніх ознак жесту;
- методи, засновані на аналізі тривимірної моделі руки.

Ціль роботи - це дослідити існуючі методи та розробити новий метод розпізнавання жестів руки.

Використовуючи карту глибини (depth map) потрібно розпізнати 11 жестів (Рис. 1.7).

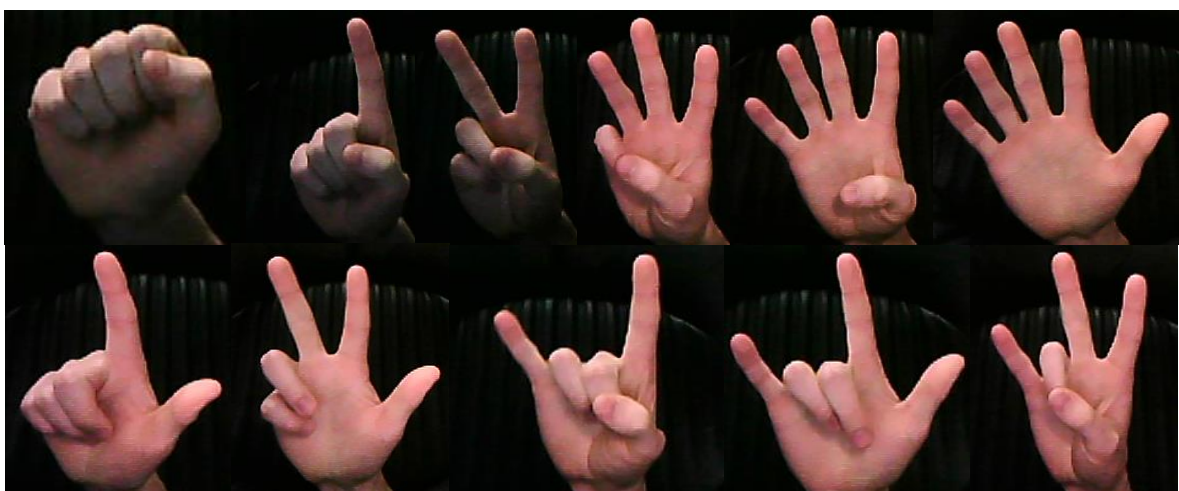


Рисунок 1.7 – Жести які потрібно розпізнати

2 ОГЛЯД ІСНУЮЧИХ МЕТОДІВ ТА РОЗРОБЛЕННЯ НОВОГО МЕТОДУ

2.1 Розпізнавання позиції та орієнтації за допомогою моментів зображення

Позначимо через $M_{0,0}$, $M_{1,0}$, $M_{0,1}$ та $M_{2,0}$, $M_{1,1}$, $M_{0,2}$ моменти нульового, першого та другого порядку зображення с функцією інтенсивності $f(x, y)$:

$$M_{0,0} = \sum_x \sum_y f(x, y),$$

$$M_{0,1} = \sum_x \sum_y y \times f(x, y), \quad M_{1,0} = \sum_x \sum_y x \times f(x, y),$$

$$M_{1,1} = \sum_x \sum_y x \times y \times f(x, y), \quad M_{0,2} = \sum_x \sum_y y^2 \times f(x, y),$$

$$M_{2,0} = \sum_x \sum_y x^2 \times f(x, y).$$

При деяких обмеженнях моменти зображення можуть бути використані для розпізнавання простих жестів рук і створення на їх основі додатків НСІ.

Розпізнавання позиції $f(x_c, y_c)$ та орієнтації руки в бінарному зображенні здійснюється за допомогою обчислення моментів зображення, за умови, що фон зображення однорідний і рука є домінуючим об'єктом у зображенні:

$$x_c = \frac{M_{1,0}}{M_{0,0}}, \quad y_c = \frac{M_{0,1}}{M_{0,0}}, \quad \theta = \frac{\arctan(b, (a-c))}{2}, \quad a = \frac{M_{2,0}}{M_{0,0}} - x_c^2, \quad c = \frac{M_{0,2}}{M_{0,0}} - y_c^2,$$

$$b = 2 \times \left(\frac{M_{1,1}}{M_{0,0}} - x_c \times y_c \right).$$

$$\arctan(i, j) = \begin{cases} \arctan\left(\frac{i}{j}\right) & j > 0 \\ \arctan\left(\frac{i}{j}\right) + \pi & i \geq 0, j < 0 \\ \arctan\left(\frac{i}{j}\right) - \pi & i < 0, j < 0 \\ \quad + \frac{\pi}{2} & i > 0, j = 0 \\ \quad - \frac{\pi}{2} & i < 0, j = 0 \\ \text{невизначеність} & i = 0, j = 0 \end{cases}.$$

Розмір зображення руки оцінюється значеннями l_1, l_2 де,

$$l_1 = \sqrt{\frac{(a+c) + \sqrt{(b^2 + (a-c)^2)}}{2}},$$

$$l_2 = \sqrt{\frac{(a+c) - \sqrt{(b^2 + (a-c)^2)}}{2}}.$$

2.2 Розпізнавання рухів

2.2.1 Розпізнавання рухів на основі аналізу різниць зображень

На практиці задовольняти умовам однорідного фону вдається не завжди. Для таких випадків, зазвичай використовується метод, заснований на аналізі центру маси різниць зображень (motion energy image - MEI).

Використання різниць кадрів відеоряду (MEI) дозволяє в реальному часі аналізувати рух об'єкта у відеоряді при стабільному, але необов'язково однорідному фоні зображення. На практиці дана технологія та її вдосконалені види (наприклад, motion history image - MHI) знайшли застосування в таких додатках як інтерактивний віртуальний тренер з аеробіки та інтерактивна кімната для розповіді історій.

2.2.2 Розпізнавання рухів на основі метода Віюли - Джонса

Метод був розроблений і представлений в 2001 р. Полом Віолою і Майклом Джонсом[4] і досі ефективний для пошуку об'єктів на цифрових зображеннях і відео-поточці в режимі реального часу. Основною його ідеєю є використання каскаду простих класифікаторів - детекторів характеристик, замість одного складного класифікатора. На базі цієї ідеї можлива побудова детектора, здатного працювати в режимі реального часу.

Характеристики використовуються замість безпосередніх значень пікселів з багатьох причин. Основною причиною є те, що характеристики можуть описувати ті знання про клас об'єктів, які важко виявити на кінцевому числі навчальних даних. Друга важлива причина використання характеристик: системи, побудовані на їх основі, працюють набагато швидше, ніж системи, що працюють безпосередньо з пікселями.

Для того щоб розрахувати яскравість прямокутної ділянки зображення, використовується інтегральне представлення. Воно часто використовується і в інших методах, наприклад у вейвлет - перетвореннях, SURF), фільтрах Хаара та багатьох розроблених алгоритмах. Інтегральне подання дозволяє швидко розрахувати сумарну яскравість довільного прямокутника на даному зображенні, причому час розрахунку не залежить від площі прямокутника.

Інтегральне представлення зображення являє собою матрицю, збігається за розмірами з вихідним зображенням. У кожному її елементі зберігається сума інтенсивностей всіх пікселів, що знаходяться лівіше і вище даного елемента.

Елементи матриці розраховуються за такою формулою:

$$I(x,y) = \sum_{x' \leq x, y' \leq y} i(x',y'),$$

де, $I(x,y)$ - значення точки (x,y) інтегрального зображення; $i(x,y)$ - значення інтенсивності вихідного зображення.

На основі застосування інтегрального представлення зображення обчислення ознак однакового виду, але з різними геометричними параметрами, відбувається за однаковий час. Кожен елемент матриці $I(x,y)$ являє собою суму пікселів в прямокутнику від $i(0,0)$ до $i(x,y)$, тобто значення кожного елемента $I(x,y)$ дорівнює сумі значень всіх пікселів лівіше і вище даного пікселя $i(x,y)$. Розрахунок матриці займає лінійний час, пропорційне числу пікселів у зображенні, і його можна обчислювати за такою формулою:

$$I(x,y) = i(x,y) - I(x-1,y-1) + I(x,y-1) + I(x-1,y).$$

Інтегральне уявлення має цікаву особливість. За інтегральною матрицею можна дуже швидко (за константний час) обчислити суму пікселів довільного прямокутника.

З точки зору необхідності використання досить простих алгоритмів отримання ознак, перспективним є застосування хаар подібних характеристик (Haar wavelet - like features), що представляють собою результат порівняння яркостей у двох прямокутних областях зображення. Зокрема, як вже зазначалося вище, Віола і Джонс запропонували використовувати три види характеристик. Значенням характеристики з двох прямокутників є різниця між сумою пікселів в цих прямокутних областях. Області мають однаковий розмір і форму по горизонталі та по вертикалі.

Припустимо, що задано безліч об'єктів A і безліч допустимих відповідей B . Нехай $g:A \rightarrow B$ називається вирішальною функцією. Вирішальна функція g повинна допускати ефективну комп'ютерну реалізацію, з цієї причини її також називають алгоритмом. Ознака (feature) f об'єкта A - відображення $f:A \rightarrow D_f$, де D_f - безліч допустимих значень ознаки. Зокрема, будь-який алгоритм $g:A \rightarrow B$ також можна розглядати як ознаку. Якщо заданий набір ознак f_1, \dots, f_n , то вектор $x = (f_1(a), \dots, f_n(a))$ називається ознаковим описом об'єкта $a \in A$. Ознакові описи допустимо ототожнювати з самими об'єктами. При цьому множину $A = D_{f_1} \times \dots \times D_{f_n}$ називають просторі ознак. Обчислюваним значенням такої ознаки буде

$$F = U - V,$$

де U - сума значень яскравостей точок, що закриваються світлою частиною ознаки; V - сума значень яскравостей точок, що закриваються темною частиною ознаки.

Для їх обчислення використовується поняття інтегрального зображення. Хаар[5] - подібні ознаки описують значення перепаду яскравості по осі X і Y зображення відповідно.

Бустінг - комплекс методів, що сприяють підвищенню точності аналітичних моделей. Бустінг (boosting) - це процедура послідовної побудови композиції алгоритмів машинного навчання, коли кожен наступний алгоритм прагне компенсувати недоліки композиції всіх попередніх алгоритмів. Ідея бустінга була запропонована Робертом Шапіро (Scharire) наприкінці 90 -х рр. минулого століття, коли треба було знайти рішення питання про те, яким чином, маючи безліч поганих (незначно відрізняються від випадкових) алгоритмів навчання, отримати один хороший.

В результаті роботи алгоритму бустінга на кожній ітерації формується простий класифікатор виду:

$$h_j(z) = \begin{cases} 1, & \text{если } p_j f_j(z) < p_j \theta_j, \\ 0, & \text{иначе.} \end{cases}$$

де p_j - показує напрямок знака нерівності; θ_j - значення порога; $f_j(z)$ - обчислене значення ознаки; z - вікно зображення розміром 20×20 .

Отриманий класифікатор має мінімальну помилку по відношенню до поточних значень ваг, використаним у процедурі навчання для визначення помилки.

Розвитком даного підходу є розробка більш досконалого сімейства алгоритмів бустінга AdaBoost (адаптивне поліпшення), здійснена Йоав Фройнд і Робертом Шапіро в 1999 р. У AdaBoost можна використовувати довільне число класифікаторів і виробляти

навчання на одному наборі прикладів, по черзі застосовуючи їх на різних кроках. У методі Віюли - Джонса варіант AdaBoost використовується як для вибору особливостей, так і для навчання класифікатора. У його оригінальній формі навчальний алгоритм AdaBoost використовується для підвищення ефективності класифікації простого (іноді званого слабким) повчального алгоритму. Для підвищення швидкості виявлення використовується каскадна структура, що фокусує свою роботу на найбільш інформативних областях зображення. Каскад складається з шарів, які представляють собою класифікатори, навчені за допомогою процедури бустінга.

2.2.3 Розпізнавання рухів на основі кольорових рукавичок

Для розпізнавання жестів руки часто застосовують кольорові рукавички. За допомогою однієї відеокамери в реальному часі розпізнають конфігурацію руки і відстежують рухи долоні в тривимірному просторі. Рукавичка сконструйована з різних кольорів. Використання невеликої кількості кольорів дозволяє розпізнати колір обраної точки зображення рукавички при різному освітленні, а спеціальне розміщення колірних сегментів не дозволяє отримати ідентичні зображення при різних конфігураціях руки. Розпізнавання конфігурації руки здійснюється згладжуванням початкового зображення алгоритмом двостороннього розмиття (англ. Bilateral filter), видаленням фонових точок, де фонові точки класифікуються на основі моделі Гауссових сумішей (англ. Gaussian mixture models) і перетворенням отриманого зображення в растрове зображення меншого розміру. Отримане растрове зображення порівнюється з зберігаються в базі даних еталонними зображеннями, відповідними різним конфігураціям руки, де відстань між растровими зображеннями r^i та

r^j оцінюється за допомогою наступної формули:

$$d(r^i, r^j) = \tilde{d}(r^i, r^j) + \tilde{d}(r^j, r^i),$$

$$\tilde{d}(r^i, r^j) = \sqrt{\frac{1}{|C_i|} \sum_{(x,y) \in C_i} \min_{(u,v) \in S_{xy}} (u-x)^2 + (v-y)^2},$$

$$S_{xy} = \{(u, v) | r^i(x, y) = r^j(u, v)\},$$

$$C_i = \{(x, y) | r^i(x, y) \neq \text{фон}\}.$$

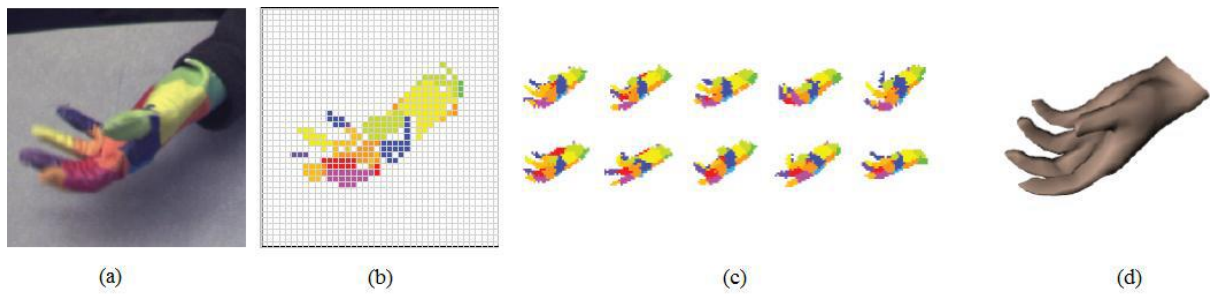


Рисунок 2.1 - Застосування кольорових рукавичок в задачі розпізнавання жестів руки : (a) Вхідний зображення руки; (b) Растрове зображення руки розміром 40 x 40; (c) Подібні зображення руки в базі даних; (d) Знайдена конфігурація

Дана система знайшла застосування в задачах управління анімаційним персонажем і розпізнавання жестів азбуки глухонімих.

2.2.4 Розпізнавання рухів на основі аналізу контуру зображення руки

В якості дескриптора конфігурації руки часто обирають контур зображення руки. У кольоровому зображенні руки, видаляються всі точки, що не відповідають кольору шкіри людини. Отримане зображення перетворюється в бінарне зображення, після чого згладжується фільтром Гаусса. Рука виділяється як зв'язаний компонент. Контури руки виділяються за допомогою оператора Собеля. Обчислюється описаний прямокутник контуру руки, центр якого відповідає початку координат. Дескриптором жесту є вектор (x_c, y_c) та площа контуру руки, де $(x_c, y_c) = \frac{1}{n} (\sum_{i=1}^n x_i, \sum_{i=1}^n y_i)$, а

(x_c, y_c) являє собою координати точок контуру, $i = 1, 2, \dots, n$. Схожість жестів оцінюється відстанню Евкліда між векторами дескрипторів жестів.

2.2.5 Розпізнавання жестів руки із застосуванням штучних ШНМ

Штучні нейронні мережі (ШНМ) представляють собою математичну модель, побудовану на принципах організації та функціонування біологічних нейронних мереж. У задачах розпізнавання жестів, ШНМ зазвичай використовують як інструмент машинного навчання, де на вхід ШНМ передаються характеристики жесту і на виході виходить розпізнаний жест. ШНМ відрізняються між собою структурою і методом навчання. Найбільш поширеними видами ШНМ, використовуваними в задачах розпізнавання, є мережі прямого поширення (англ. feed forward neural networks), рекурентні нейронні

мережі, самоорганізуючі карти Кохонена і т.д. На рис. 2.2 зображена рекуррентна нейронна мережа, використовувана для розпізнавання статичних жестів руки азбуки глухонімих Японії. В якості вхідного пристрою використовується рукавичка з вбудованими сенсорами, яка повертає характеристики показаного жесту (10 точок і 3 кута). Характеристики жесту нормалізуються і передаються на вхід нейронної мережі. На виході мережа видає результат у вигляді масиву з 42 елементів, які відповідають 42 жестам руки. Результати розпізнавання - 71.4 % для зареєстрованих користувачів і 47.8 % для незареєстрованих.

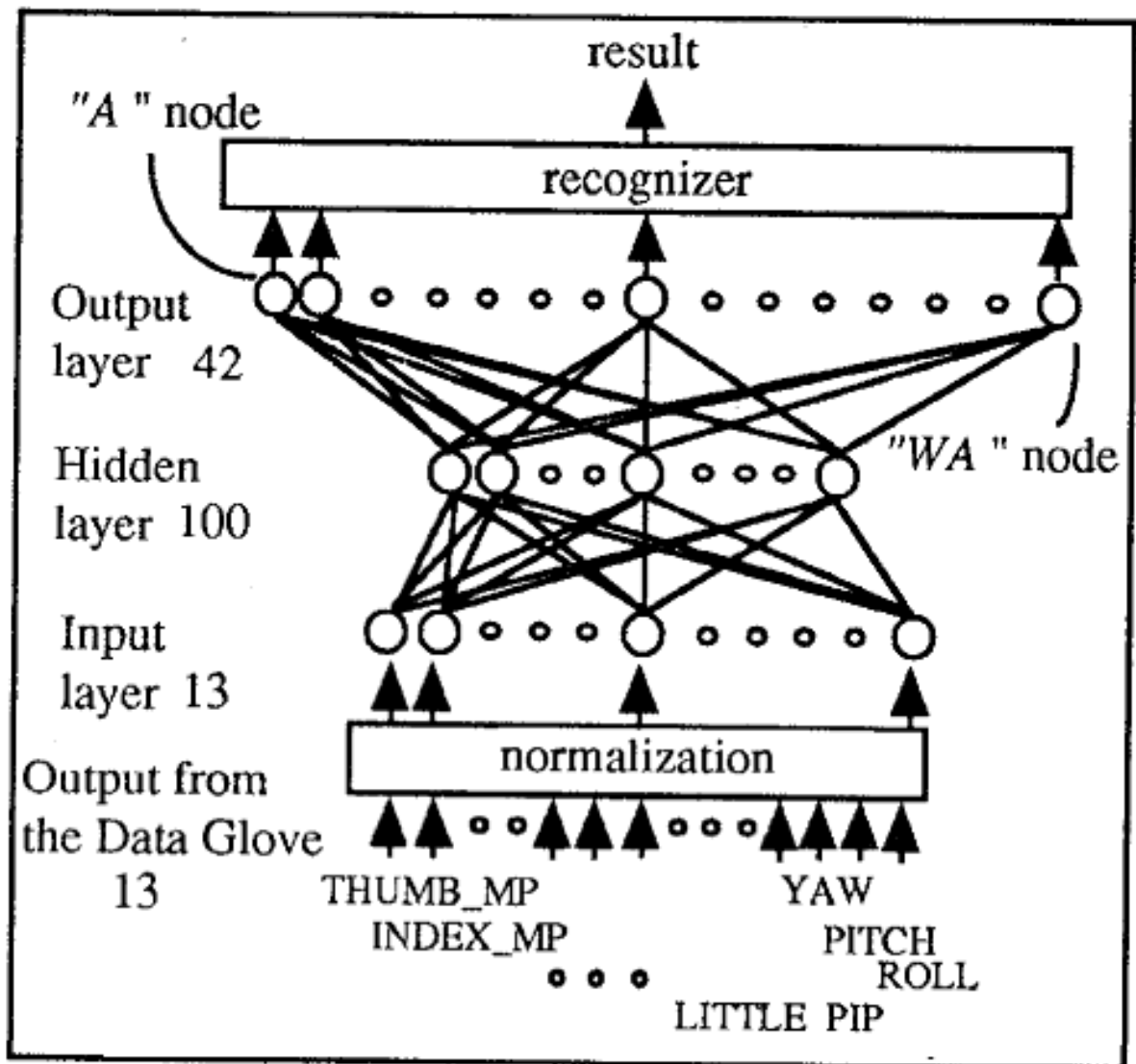


Рисунок 2.2 - Рекуррентна нейронна мережа, використовувана для розпізнавання статичних жестів руки азбуки глухонімих Японії

2.2.6 Розпізнавання жестів руки із застосуванням прихованих моделей Маркова

Прихована Марковська модель (ПММ) - це статистична модель, яка знайшла широке застосування як у задачах розпізнавання мовлення та письма, так і в задачах розпізнавання жестів. ПММ - це безліч станів, змінних і залежностей, де кожне стан має імовірнісний розподіл, серед усіх можливих вихідних значень.

Позначимо через q_t состояние ПММ в момент часу t , O_t - спостережувану змінну в момент часу t , P - ймовірність. Формально ПММ можна представити як:

$$\lambda = (A, B, \pi).$$

Де, $A = \{a_{ij}\}$ є матрицею розміром $N \times N$, де N дорівнює кількості станів $\{S_1, \dots, S_N\}$, $a_{ij} = P(q_{t+1} = s_j \mid q_t = s_i)$, $B = \{b_i(k)\}$ є матрицею розміром $N \times M$ де M число спостережуваних змінних $\{V_1, \dots, V_N\}$,

$b_j(k) = P(O_t = v_k \mid q_t = s_j)$, $\pi = \{\pi_i\}$ - є масивом з N елементів, де $\pi_i = P(q_1 = s_i)$.

У задачах розпізнавання жестів, зазвичай для кожного окремого жесту руки створюється СММ, яка навчається розпізнаванню саме цього жесту. На етапі розпізнавання кожної моделі на вхід передаються характеристики показаного жесту. Обчислюється функція відповідності кожної моделі та переданих характеристик. Розпізнається жест, для моделі якого ця функція має найбільше значення. ПММ використовується для розпізнавання, як ізольованих жестів, так і безперервних.

2.2.7 Розпізнавання позиції і конфігурації руки методом випадкових лісів

Алгоритм випадкових лісів (англ. randomized decision forests) є алгоритмом машинного навчання, використовуваним в задачах класифікації, регресії і кластеризації. У задачах розпізнавання жестів даний алгоритм був використаний для розпізнавання

позицій ключових характеристик тіла людини та класифікації конфігурації руки. Саме на цій технології заснована система розпізнавання жестів людини, випущена компанією Microsoft в 2010 році. Запропонований метод дозволяє розпізнавати позиції 20 частин тіла в тому числі позиції долонь обох рук отримуючи на вході одне дальностное зображення людини. Позиціонування цікавлять точок тіла здійснюється за допомогою випадкових лісів рішень, які навчаються на вибірці зображень відповідних 100000 різних поз тіла людини. Алгоритм розпізнавання складається з наступних кроків:

- відділення зображення людини в дальностном зображенні;
- поділ зображення людини на 31 кластер допомогою застосування алгоритму випадкових лісів;

знаходження позицій частин тіла на основі отриманих кластерів.

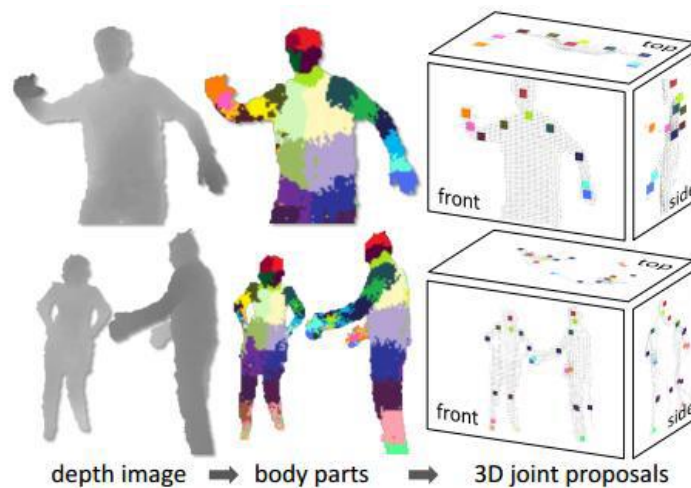


Рисунок 2.3 – Кроки розпізнавання

2.2.7 Розпізнавання жестів як задача оптимізації

В роботі задача розпізнавання жестів руки зводиться до задачі оптимізації пошуку параметрів, які задають конфігурацію моделі руки і мінімізують відстань між вхідним зображенням жесту і зображенням проекції моделі руки. Модель руки будується із сукупності еліптичного циліндра, сфер і конусів, які мають 26 ступенів свободи руху та 27 параметрів, які задають конфігурацію руки.

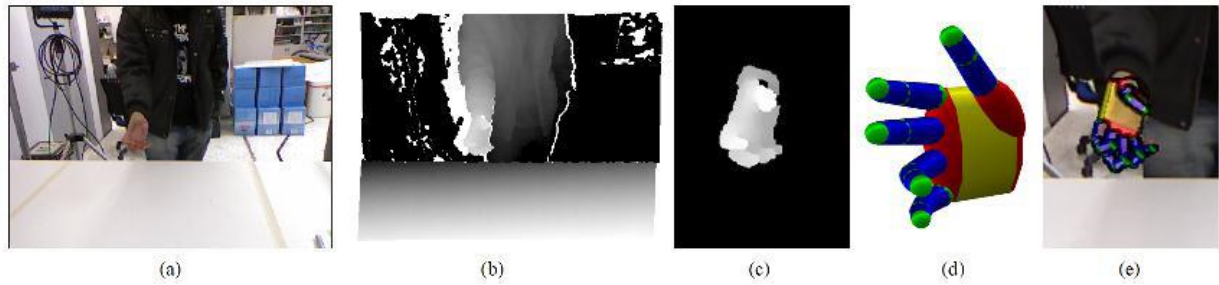


Рисунок 2.4 - (a) Вхідне кольорове зображення; (b) Вхідне дальностне зображення; (c) Відокремлене зображення руки; (d) тривимірна модель руки; (e) Розпізнана конфігурація, прикладена до вхідного зображенню

Як пристрій введення інформації про жести руки використовується сенсор Kinect [1], що дозволяє отримувати кольорове і дальностне зображення

людини зі швидкістю 30 кадрів в секунду. В отриманому кольоровому зображенні видаляються всі фонові точки. Решта точки групуються, створюючи пов'язані компоненти, з яких в якості зображення руки вибирається компонента, що має максимальні розміри. використовуючи позицію руки в кольоровому зображенні, отримують пару дескрипторів вхідного жесту $O = (o_s, o_d)$, де o_s – кольорове зображення руки, а o_d – карта глибини.

Передбачається, що в кожному кадрі відеоряду конфігурація руки збігається або незначно відрізняється від конфігурації руки, розпізнаний в попередньому кадрі. Для кожного кадру відеоряду за попередньою конфігурації руки h і калібрування камери C , створюється проекція моделі руки у вигляді дальностного зображення $r_d(h, C)$. Відстань між вхідним зображенням $O = (o_s, o_d)$ і конфігурацією руки h оцінюється функцією $E = (h, O)$, де

$$E(h, O) = D(O, h, C) + \lambda_k kc(h),$$

$$D(O, h, C) = \frac{\sum \min(|o_d - r_d|, d_m)}{\sum (o_s \vee r_m) + \varepsilon} + \lambda \left(1 - \frac{2 \sum (o_s \wedge r_m)}{\sum (o_s \wedge r_m) + \sum (o_s \vee r_m)} \right).$$

В наведених формулах λ_k є фактором нормалізації, $r_m(h, c)$ - бінарне зображення, піксель якого приймає значення 1, якщо значення відповідної точки в зображеннях o_d та

r_d відрізняється не більше, ніж заданий поріг d_m . λ, d_m, d_M – є константами, а функція $kc(h)$ використовується для збільшення відстані, якщо конфігурація руки не є кінематично правдоподібною.

Оптимальна конфігурація руки h для мінімізації функції $E = (h, O)$

визначається за допомогою методу рою частинок. У роботі наводиться швидкість розпізнавання 15 кадрів в секунду, при використанні ресурсів графічного процесора.

Перевагою даного методу є розпізнавання детальної конфігурації руки без початкового навчання, але в той же час, щоб використовувати даний підхід для розпізнавання жестів ручної азбуки глухонімих, потрібно початкове навчання.

2.3 Аналіз досліджених методів

В попередніх пунктах були описані самі основні методи розпізнавання жестів на основі зовнішніх ознак. Звичайно використовують й інші, наприклад дескриптори Surf або «Скелетизація». Так як, розпізнавати потрібно по звичайній RGB камері, не були розглянуті методи які будують руку в 3D, для цього потрібні інші системи захвату кадру, основані на карті глибини, наприклад Kinect. Також за допомогою лише однієї камери не можливо розпізнати деякі жести, наприклад коли пальці перекривають самі себе чи долоню, через ці ж обмеження не можливо розпізнати коли руки перекривають одну-одну.

Деякі методи, які були досліджено можна проаналізувати. На приклад метод Віола-Джонса спочатку використовувався для розпізнавання обличь, але його можна використовувати і для розпізнавання різних об'єктів в реальному часі, отже можна розпізнавати і жести, але через свою особливість його важко використовувати для цієї задачі, тому що, на кожний окремий жест потрібно обучати окремо. Тобто скільки жестів – стільки і потрібно обучати, а так як обучати потрібно на великій вибірці, чим більша вибірка тим точніше він працює, то навчання витрачає багато часу, якщо звичайно у вас не має обчислювального кластеру. Чим більше жестів які схожі між тим більше він буде помилятися. Отже метод доволі швидкий але його краще використовувати там, де потрібно просто знайти руку на зображенні.

За допомогою метода який оснований на кольоровій рукавичці можна детально розпізнати жест, але в нього є головний недолік – потрібно використовувати рукавичку,

що не дуже зручно для звичайної взаємодії. Цей метод гарний в комп'ютерній графіці чи кінематографі коли потрібно управляти персонажем або іншим предметом.

Метод оснований на моментах дуже примітивний, за допомогою нього можна доповнити якийсь інший метод.

Використовуючи метод оснований на різниці зображень важко відстежити деталі жесту, а також не можливо розпізнати статичний жест. Метод можна використовувати як доповнення до якогось іншого методу.

2.4 Запропонований підхід

На основі методів які були досліджені можна зробити деякі висновки, а саме, що серед них не має універсального метода, кожен із них має плюси та мінуси, точніше сказати, що вибрати їх потрібно виходячи із задачі. Тому метод який буде застосовано в нашій задачі - комбінування деяких методів які оглянуто раніше.

Розіб'ємо метод на осн овні складові:

- сегментація зображення та виділення руки на карті глибини;
- отримання дескрипторів які описують жест;
- збирання статистики дескрипторів по кожному жесту;
- навчання системи та отримання моделі розпізнавання;
- - використовуючи модель розпізнавати жести.

2.4.1 Сегментація зображення

З комп'ютерної точки зору, сегментація - це процес розділення цифрового зображення на декілька сегментів. Сегментація зображень зазвичай використовується для того, щоб виділити об'єкти і межі (лінії, криві, і т. д.) на зображеннях. Більш точно, сегментація зображень - це процес присвоєння таких міток кожному пікселю зображення. Пікселі з однаковими мітками мають спільні візуальні характеристики.

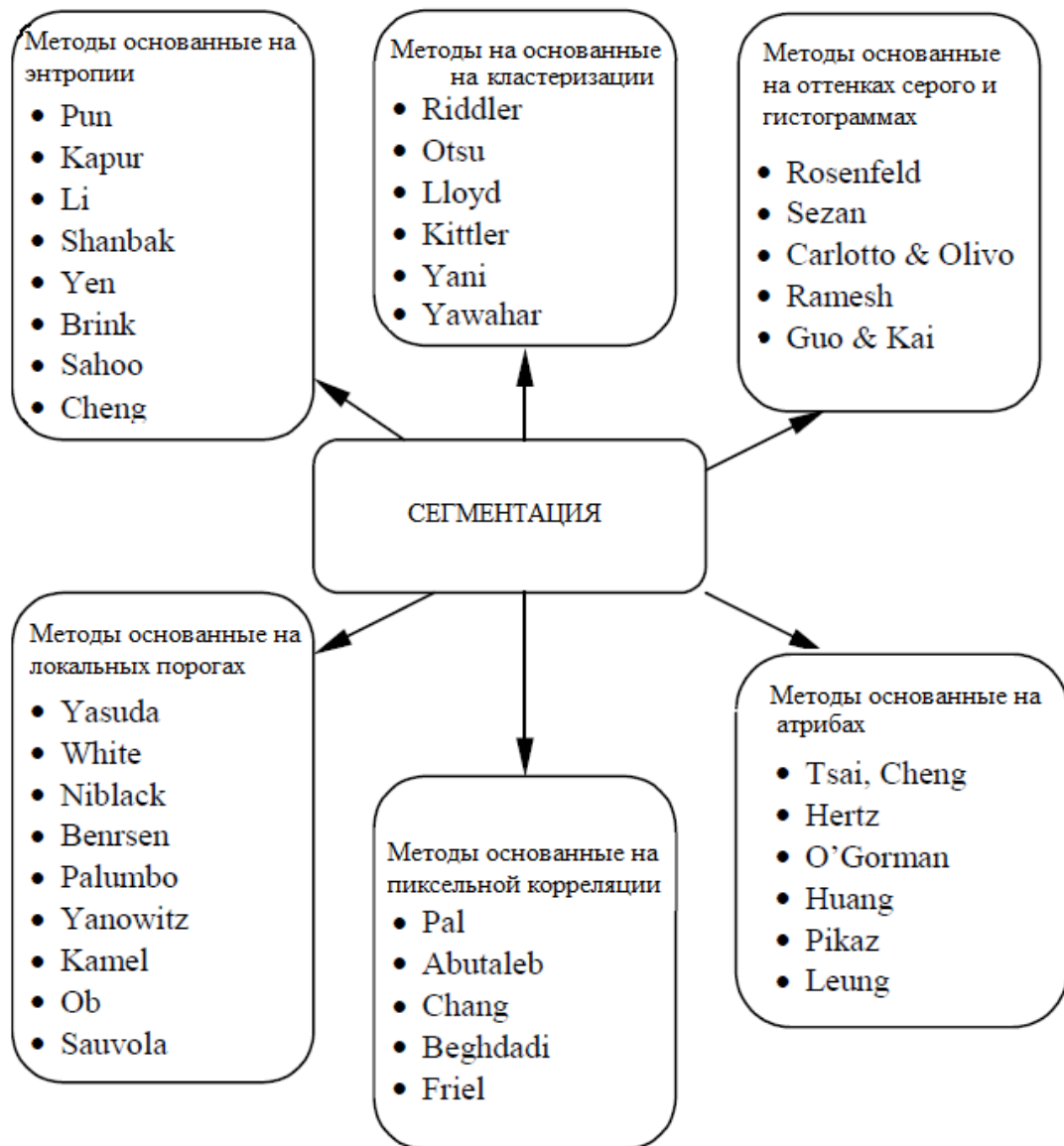


Рисунок 2.5 – Методи сегментації

Один з основних і простих способів - це побудова сегментації за допомогою порога. Поріг - це ознака (властивість), яка допомагає розділити шуканий сигнал на класи. Операція порогового поділу полягає в зіставленні значення яскравості кожного пікселя зображення із заданим значенням порога. Операція порогового поділу, яка в результаті дає бінарне зображення, називається бінаризацією.

Карта глибини - це монохромне зображення в якому вихідне діапазонне зображення, містить в якості значень пікселів відстань до конкретних точок зображення. Якщо датчик, який використовується для отримання зображення був правильно відкалібрований, значення пікселів можуть вказуватись безпосередньо в фізичних одиницях вимірювання, таких як метри.

Робота з картою глибини потребує другого підходу ніж робота з звичайним зображенням. Будемо використовувати дві стратегії для отримання руки на карті глибини. Перша стратегія – це використовувати діапазони, тобто розпізнавати жест будемо в заданому діапазоні l_1, l_2 . Де l_1 – ближня відстань до камери, а l_2 – дальня відстань. Все що буде в цьому діапазоні будемо вважати рукою.

В другому підході будемо шукати перший об'єкт до камери, площа якого буде не менше деякої заданої S .

В результаті отримаємо карту глибини на якому лише одна рука

2.4.2 Дескриптори жесту

Після етапу сегментації ми отримали карту глибини на якій лише рука, тепер потрібно отримати дескриптор. Дескриптор – це вектор $X \in R^n$ в просторі, який однозначно описує жест, координати якого є ознаками (атрибутами) жесту. Тобто дескриптор це вектор - елементи якого є ознаками для одного жесту. Різні жести можуть мати різну довжину вектора.

Щоб отримати дескриптор для одного жесту будемо використовувати таку послідовність:

1) Виділення контуру руки [5]. Деякі задачі обробки та аналізу зображень передбачають виділення контуру зображення, будь-то бінарного, кольорового чи карти глибини, для його подальшого використання в задачах розпізнавання та класифікації.

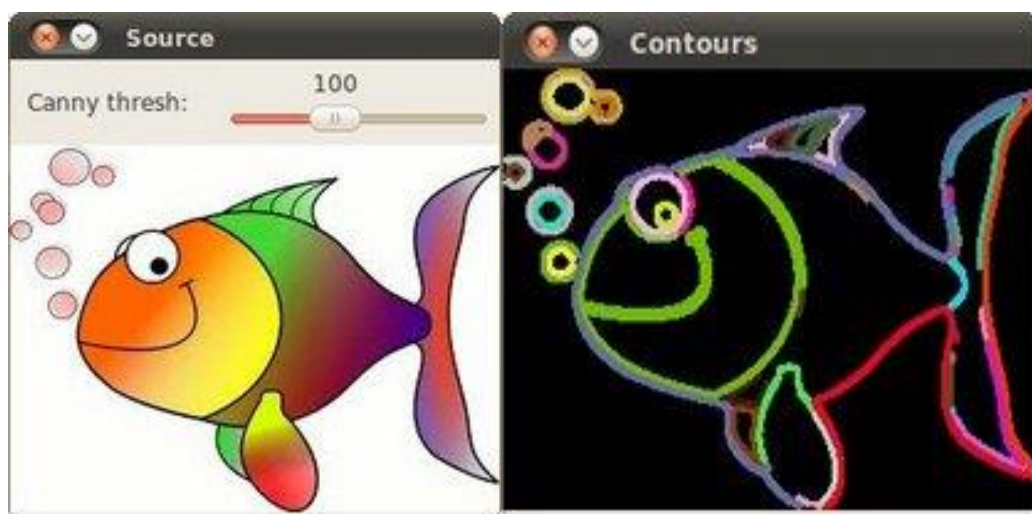


Рисунок 2.6 – Виділення контуру по кольоровому зображенні

Для знаходження контурів використовують ознаки зв'язності. Для побудови зв'язнісної моделі можна скористатися поняттями "сусідства". Відомо, що при квадратному растрі можлива чотиризв'язність (тобто сусідніми вважаються елементи, що прилягають до сторін); восьмизв'язність (тобто елементи, що дотикаються у кутах, також вважаються "сусідами").

Одиничний елемент $b(m, n)$ бінарної матриці $B(M, N)$ є зв'язаним, як що хоча б один з сусідніх з ним елементів $b(m, n + 1)$, $b(m + 1, n)$, $b(m + 1, n + 1)$,

$b(m - 1, n)$, $b(m, n - 1)$, $b(m - 1, n - 1)$, $b(m + 1, n - 1)$, $b(m - 1, n + 1)$ є також одиничним, де m, n — координати елементу b . Потрібно знайти всі зв'язні точки в яких хоча б один сусід не належав області.

Для цієї задачі використовуємо алгоритм відстеження. Алгоритм який відстежує ґрунтуються на тому, що на зображенні відшукується об'єкт (перша точка об'єкта, яка зустрілась), контур об'єкта відстежується і векторизується. Перевагою даних алгоритмів є їх простота, до недоліків можна віднести їх послідовну реалізацію і деяку складність при пошуку й обробці внутрішніх контурів. Приклад відслідковуємого алгоритму - "алгоритм обходу контуру", або "алгоритм жука", - наведеного на рисунку 2.7.

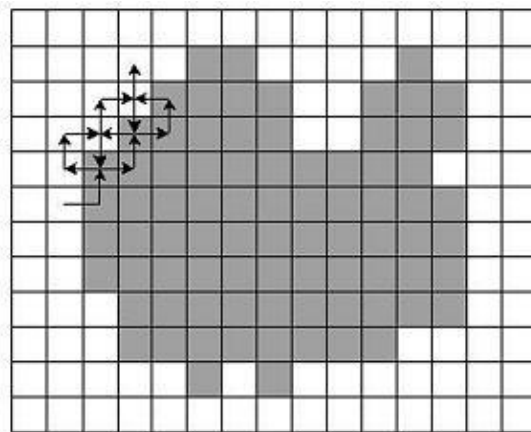


Рисунок 2.7 - Схема роботи алгоритму "жука"

"Жук" починає рух з білою області у напрямку до чорної. Як тільки він потрапляє на чорний елемент, він повертає ліворуч і переходить до наступного елементу. Якщо цей елемент білий, то жук повертається направо, інакше - наліво. Процедура повторюється до тих пір, поки жук не повернеться у вихідну точку. Координати точок переходу з чорного на біле і з білого на чорне і описують кордон об'єкта.

2) Знаходження опуклої оболонки. Опукла оболонка [6] множини точок X на евклідовій площині або у просторі — це мінімальна опукла множина, що містить X . В обчислювальній геометрії, прийнято використовувати термін “опукла оболонка” для границі мінімальної опуклої множини, що містить дану не порожню скінченну множину точок на площині. Для скінченної множини точок, опукла оболонка являє собою ламану лінію.

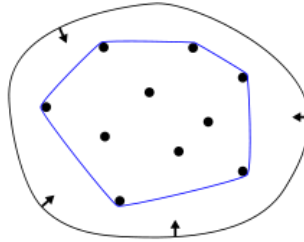


Рисунок 2.8 – Опукла оболонка: аналог еластичної пов’язки

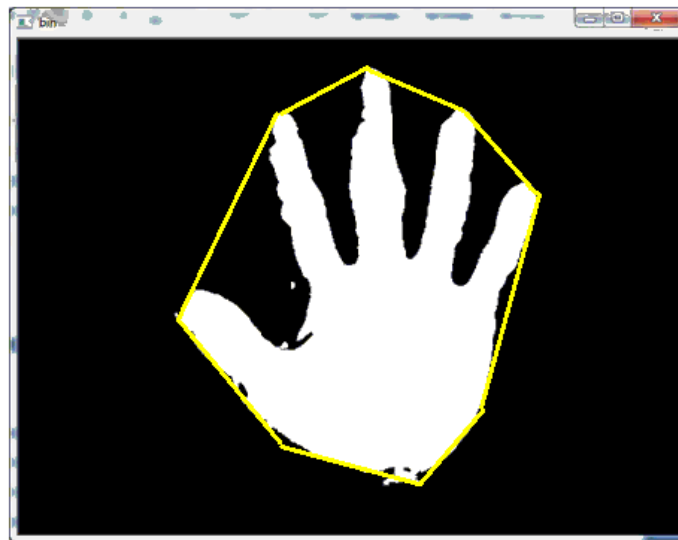


Рисунок 2.9 – Жовтою лінією показано опуклу оболонку

Знаходити опуклу оболонку можна різними алгоритмами, наприклад:

- алгоритм Грехема[7];
- алгоритм Ендрю[8];
- алгоритм Чана[9];
- алгоритм швидкої оболонки[10].

Використаємо алгоритм швидкої оболонки, тому що його легко розпаралелити та узагальнити на довільну розмірність. В середньому складність алгоритму буде $O(n \cdot \log(n))$, якщо кількість елементів підзадач буде лінійно зменшуватись зі сталим

коефіцієнтом $k < 1$. В найгіршому випадку швидкість буде $O(n^2)$ (квадратична). Сам алгоритм простий. Потрібно знайти точки з мінімальною і максимальною x координатою, вони зобов'язані бути частиною опуклої оболонки. Використовуючи лінію, утворену двома точками розділити всю множину точок на дві підмножини, які будуть оброблятися рекурсивно. Визначити точку, на одній стороні лінії, з максимальною відстанню від лінії. Знайдені до цього дві точки утворюють з цією точкою трикутник з найбільшою площею. Точки, що лежать всередині цього трикутника не може бути частиною опуклої оболонки і, отже, можуть бути проігноровані в наступних кроках. Повторимо попередні два кроки для двох ліній, утвореного трикутника (окрім початкової лінії). Продовжуємо робити так доти, поки більше точок не залишиться. В кінці рекурсії, вибрані точки, складуть опуклу оболонку.

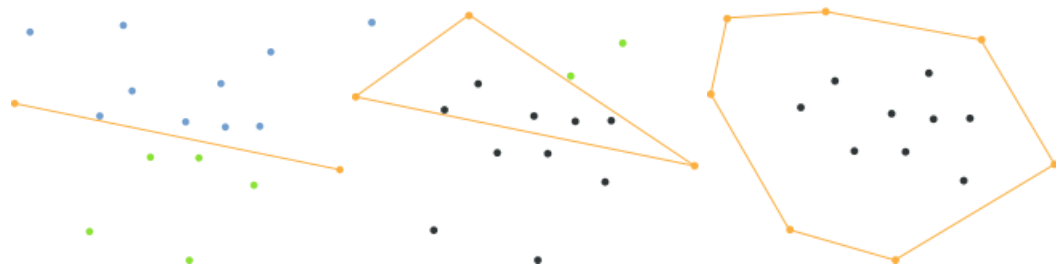


Рисунок 2.10 – Послідовність алгоритму швидкої оболонки

3) Знаходження дефектів опуклості. Рисунок 3.8 ілюструє концепцію дефекту опуклості на прикладі зображення руки.

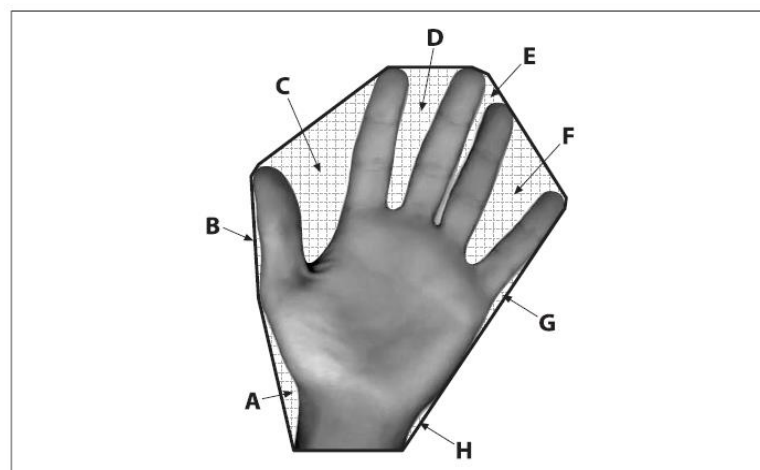


Рисунок 2.11 - Дефекти опуклості

Каркас опуклості зображений темної лінії навколо руки, а дефекти, пов'язані з цим каркасом, помічені буквами від А до Н. Ці дефекти опуклості дозволяють характеризувати не тільки саму руку, але також і її стан.

Будемо знаходити ті дефекти опуклості, які відповідають певним умовам а саме:

- - кут між двома векторами які утворені, повинен буди не більше 90 градусів;
- - довжина цих векторів повинна не перевищувати відношення рамки в якій знаходиться рука.

В результаті повинно буди від 0 до 4 таких дефектів, в залежності від жесту.

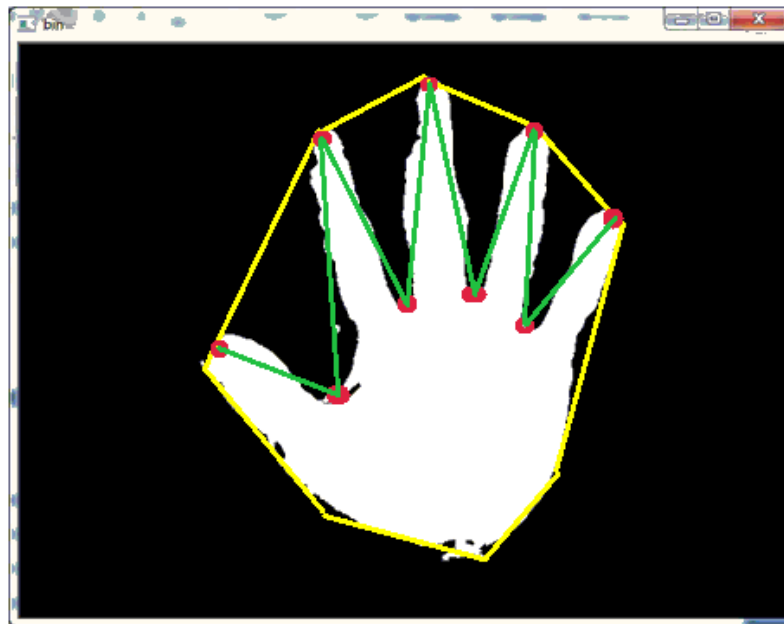


Рисунок 2.12 – Дефекти опуклості

4) Знаходження центру мас, та радіусу і центру описаного кола.

Моменти зображення корисні для опису об'єктів після сегментації. Прості властивості зображення, які можна знайти за допомогою моментів, включають площу (або сумарну інтенсивність), її геометричний центр та інформацію про орієнтацію.

Момент - це характеристика контуру, об'єднана з усіма пікселями контуру. Момент (p, q) визначається як:

$$m_{p,q} = \sum_{i=1}^n I(x,y) x^p y^q.$$

Отже маючи ознаки які були перераховані вище можемо сформувати вектор(дескриптор).

Обхід всіх кутів відбувається зліва на право.

n – кількість дефектів;

x_1, \dots, x_n – це кути між двома сусідніми прямими – пряма, координати якої починаються з першої кінцівки та закінчується координатою першого дефекту, а друга пряма, початок якої в координаті першого дефекту, та кінець в координаті наступної кінцівки;

x_{n+1}, \dots, x_{2n} - це кути між двома сусідніми прямими – пряма, координати якої починаються з першої кінцівки та закінчується координатою центра описаного кола, а друга пряма, початок якої в координаті центра описаного кола, та кінець в координаті наступної кінцівки;

$x_{2n+1}, \dots, x_{2n+n}$ - це нормована довжина прямих від координати пальця до координати дефекту. Нормуємо так – знаходимо максимальну довжину, а потім ділимо максимальну довжину на поточну;

x_{2n+n+1} - це відстань між центром мас та описаним колом.

Потрібно також зробити зауваження на рахунок того, що не для кожного жесту будуть дефекти.

3 МАШИННЕ НАВЧАННЯ ДЛЯ ПОБУДОВИ КЛАСИФІКАТОРА

3.1 Основні підходи

Машинне навчання [11] - узагальнена назва штучної генерації знань з досвіду. Штучна система навчається на прикладах і після закінчення фази навчання може узагальнювати. Тобто система не просто вивчає наведені приклади, а розпізнає певні закономірності в даних для навчання. Серед багатьох програмних продуктів варто згадати системи автоматичного діагностування, розпізнавання шахрайства з кредитними картками, аналіз ринку цінних паперів, класифікація ланцюжків ДНК, розпізнавання мовлення тексту, автономні системи.

Практичне використання відбувається, переважно, за допомогою алгоритмів. Різноманітні алгоритми машинного навчання можна грубо поділити за такою схемою:

- навчання з вчителем: алгоритм вивчає функцію на основі наданих пар вхідних та вихідних даних. При цьому, в процесі навчання, «вчитель» вказує вірні вихідні дані для кожного значення вхідних даних. Одним з розділів навчання з вчителем є машинна класифікація. Такі алгоритми застосовуються для розпізнавання текстів;
- навчання без вчителя;
- навчання з закріпленням: алгоритм навчається за допомогою тактики нагороди та покарання для максимізації вигоди для агентів (систем до яких належить компонента, що навчається).

Машинне навчання, розділ штучного інтелекту, має за основу побудову та дослідження систем, які можуть самостійно навчатись з даних. Наприклад, система машинного навчання може бути натренована на електронних повідомленнях для розрізнення спам і не спам-повідомлення. Після навчання вона може бути використана для класифікації нових повідомлень електронної пошти на спам та не-спам папки.

В основі машинного навчання розглядаються уявлення та узагальнення. Представлення даних і функцій оцінки цих даних є частиною всіх систем машинного навчання, наприклад, у наведеному вище прикладі повідомлення по електронній пошті, ми можемо уявити лист як набір англійських слів, просто відмовившись від порядку слів. Узагальнення є властивістю, яку система буде застосовувати добре на невидимих екземплярах даних; умови, за яких це може бути гарантовано є ключовим об'єктом вивчення в полі обчислювальної теорії навчання. Існує широкий спектр завдань машинного навчання та успішних застосувань.

Розпізнавання образів — це розділ теорії штучного інтелекту що вивчає методи класифікації об'єктів. За традицією об'єкт, що піддається класифікації, називається образом. Образом може бути цифрова фотографія (розпізнавання зображень), буква або цифра (розпізнавання символів), запис мови (розпізнавання мови) тощо. В межах теорії штучного інтелекту розпізнавання образів включається в більш широку наукову дисципліну — теорію машинного навчання, метою якої є розробка методів побудови алгоритмів, що здатні навчатися. Існує два підходи до навчання: індуктивне і дедуктивне. Індуктивне навчання, або навчання за прецедентами, засноване на виявленні загальних властивостей об'єктів на підставі неповної інформації, отриманих емпіричним шляхом. Дедуктивне навчання передбачає формалізацію знань експертів у вигляді баз знань (експертних систем тощо). Слід зауважити, що, як кожна математична дисципліна, розпізнавання образів має власний математичний апарат, який включає математичну статистику, методи оптимізації, дискретну математику, алгебру і геометрію. Розпізнавання образів має широке застосування і використовується при створенні усіх комп'ютерних систем, на які покладаються інтелектуальні функції, тобто функції, пов'язані із прийняттям рішень замість людини: медична діагностика, криміналістична експертиза, пошук інформації та інтелектуальний аналіз даних тощо. Прецедент — це об'єкт, приналежність якого до заданого класу визначена заздалегідь. Прецедентом може бути, наприклад, набір ознак пацієнта із відомим діагнозом, з яким слід порівнювати набір ознак людини, діагноз якої ще невідомий. Кожний образ являє собою набір чисел, що описують його властивості і називаються ознаками. Упорядкований набір ознак об'єкта називається вектором ознак. Вектор ознак — це точка в просторі ознак. Класифікатор, або вирішальне правило — це функція, яка ставить у відповідність вектору ознак образу клас, до якого він належить. Задачу розпізнавання образів можна розділити на ряд під задач:

- генерування ознак — вимірювання або обчислення числових ознак, що характеризують об'єкт;
- вибір ознак — визначення найбільш інформативних ознак для класифікації (в цей набір можуть входити не лише первинні ознаки, але й функції від них);
- побудова класифікатора — конструювання вирішального правила, на підставі якого здійснюється класифікація;
- оцінка якості класифікації — обчислення показників правильності класифікації (точність, чутливість, специфічність, помилки першого та другого роду).

3.2 Дерево рішень

В даній роботі використаємо апарат машинного навчання, а саме дерева рішень. Існує безліч алгоритмів, вони мають свої недоліки та переваги. Поставлену задачу інтерпретуємо як задачу класифікації. Маємо низку жестів які необхідно розпізнати, кожний жест є одним класом зі своїми характеристиками. Сформулюю математичну постановку задачі.

Нехай Y — простір жестів; y – жест, $y \in Y$; $m = \{1, 2, \dots, \}$ — номери жестів; X - простір ознак, тобто векторний простір, точками якого є вектори ознак образів; Існує невизначена залежність – відображення $y^*: X \rightarrow Y$, значення якої відомі тільки на об'єктах кінцевої навчальної вибірки. Потрібно побудувати алгоритм $a: X \rightarrow Y$, здатний класифікувати довільний жест $x \in X$. В якості данного алгоритма будемо використовувати дерева рішень.

Під терміном "дерева рішень" [11] мається на увазі сімейство алгоритмів, заснованих на створенні ієрархічної структури, яка базується на відповіді "Так" або "Ні" до набору питань. Такі алгоритми досить популярні: в даний час вони реалізовані практично у всіх комерційних засобах Data Mining.

Сімейство алгоритмів побудови дерев рішень дозволяє передбачити значення будь-якого параметра для заданого випадку. Наприклад: чи поверне людина вчасно виданий йому кредит. На основі великої кількості даних про інші подібні випадки (зокрема, на основі відомостей про інших осіб, яким видавалися кредити). Зазвичай алгоритми цього сімейства застосовуються для вирішення завдань, що дозволяють розділити всі вихідні дані на кілька дискретних груп.

Коли один з алгоритмів побудови дерев рішень застосовується до набору вихідних даних, результат відображається у вигляді дерева. Подібні алгоритми дозволяють здійснити кілька рівнів такого поділу, розбиваючи отримані групи (гілки дерева) на більш дрібні на підставі інших ознак до тих пір, поки значення, які передбачається пророкувати, не стануть однаковими (або, в разі безперервного значення пророкує параметра, близькими) для всіх отриманих груп (листя дерева). Саме ці значення і застосовуються для здійснення прогнозів на основі даної моделі.

Дія алгоритмів побудови дерев рішень базується на застосуванні методів регресійного і кореляційного аналізу. Один з найпопулярніших алгоритмів цього сімейства – CART (Classification and Regression Trees), заснований на поділі даних у гілки

дерева на дві дочірні гілки, при цьому подальший поділ тієї чи іншої гілки залежить від того, чи багато вихідних даних описує дана гілка. Деякі інші подібні алгоритми дозволяють розділити гілку на більшу кількість дочірніх гілок. У даному випадку поділ проводиться на основі найбільш високого для описуваних гілкою даних коефіцієнта кореляції між параметром, згідно з яким відбувається поділ, і параметром, який надалі повинен бути передбачений.

Деякі жести мають різну кількість характеристик, тому було прийнято розділити жести на групи, простір груп – G , де в кожній групі лише ті жести, для яких кількість характеристик однакова. Потрібно задати функцію $g:Y \rightarrow G$ яка буде відображати простір жестів на простір груп. Для кожної групи будуюмо окремо дерево, тобто тепер добавимо простір який складається з окремих дерев.

Як же відбувається сама побудова дерева, існує безліч алгоритмів побудова дерев рішень: CART, C4.5, NewId, ITrule, CHAID, CN2 і т.д. Але найбільш розповсюджені наступні два:

- CART - це алгоритм побудови бінарного дерева рішень дихотомічної класифікаційної моделі. Кожен вузол дерева при розбитті має тільки двох нащадків. Як видно з назви алгоритму, вирішує завдання класифікації і регресії.

- C4.5 - алгоритм побудови дерева рішень, кількість нащадків у вузла не обмежена. Не вміє працювати з безперервним цільовим полем, тому вирішує тільки завдання класифікації.

Більшість з відомих алгоритмів є "жадібними алгоритмами". Якщо один раз був обраний атрибут, і по ньому було вироблено розбиття на підмножини, то алгоритм не може повернутися назад і вибрати інший атрибут, який дав би краще розбиття. Тому на етапі побудови не можна сказати чи дасть обраний атрибут, зрештою, оптимальне розбиття. Атрибут – це одна з ознак жесту.

Для побудови дерева на кожному внутрішньому вузлі необхідно знайти таку умову (перевірку), яка б розбивала множину, асоційовану з цим вузлом на підмножини. В якості такої перевірки повинен бути вибраний один з атрибутів. Загальне правило для вибору атрибута можна сформулювати наступним чином: обраний атрибут повинен розбити множину так, щоб одержувані внаслідок підмножини склалися з об'єктів, що належать до одного класу, або були максимально наближені до цього, тобто кількість об'єктів з інших класів ("домішок") у кожному з цих множин була як можна менше.

Розглянемо критерії розбиття лише двох алгоритмів :

- Статистичний критерій, використовує індекс Gini, оцінює "відстань" між розподілами класів.

$$Gini(c) = 1 - \sum_j p_j^2,$$

де, c – поточний вузол, а p_j – ймовірність класу j в вузлі c .

– теоретико-інформаційний критерій, алгоритм C4.5, вдосконалена версія алгоритму ID3 (Iterative Dichotomizer), використовує теоретико-інформаційний підхід. Для вибору найбільш відповідного атрибуту, пропонується наступний критерій:

$$Gain(X) = Info(T) - Info_x(T),$$

де, $Info(T)$ – ентропія множини T , а

$$Info_x(T) = \sum_{i=1}^n \frac{|T_i|}{|T|} * Info(T_i).$$

Множини T_1, T_2, \dots, T_N отримані при розбитті початкової множини T з перевірки X . Вибирається атрибут, який дає максимальне значення за критерієм $Gain(X)$.

Дерева рішень мають ряд переваг від інших алгоритмів:

– алгоритм конструювання дерева рішень не вимагає від користувача вибору вхідних атрибутів (незалежних змінних). На вхід алгоритму можна подавати всі існуючі атрибути, алгоритм сам вибере найбільш значущі серед них, і тільки вони будуть використані для побудови дерева. У порівнянні, наприклад, з нейронними мережами, це значно полегшує користувачеві роботу, оскільки в нейронних мережах вибір кількості вхідних атрибутів істотно впливає на час навчання;

– точність моделей, створених за допомогою дерев рішень, порівнянна з іншими методами побудови класифікаційних моделей (статистичні методи, нейронні мережі);

– швидкий процес навчання. На побудову класифікаційних моделей за допомогою алгоритмів конструювання дерев рішень потрібно значно менше часу, ніж, наприклад, на навчання нейронних мереж чи svm;

– більшість алгоритмів конструювання дерев рішень мають можливість спеціальної обробки пропущених значень;

– багато класичних статистичних методів, за допомогою яких вирішуються завдання класифікації, можуть працювати тільки з числовими даними, в той час як дерева рішень працюють і з числовими, і з категоріальними типами даних.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ

4.1 Реалізація основних частин

Для реалізації задачі будемо використовувати мову програмування C++ та бібліотеку OpenCV[12].

OpenCV - Open Source Computer Vision Library, бібліотека комп'ютерного зору з відкритим вихідним кодом. Бібліотека функцій та алгоритмів комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення. OpenCV застосовують у таких сферах:

- аналіз та обробка зображень;
- системи з розпізнавання обличчя;
- розпізнавання жестів;
- система взаємодії людини з комп'ютером;
- аналіз руху;
- ідентифікація об'єктів;
- сегментація зображень;
- трекінг відео;
- машинному навчанні.

Реалізація складається з двох модулів:

- за допомогою першого модуля ми можемо отримати дані для кожного жесту, тобто закономірностями;
- за допомогою другого модуля можна використовувати дані с закономірностями та побудувати модель закономірностей для використання її при класифікації жестів.

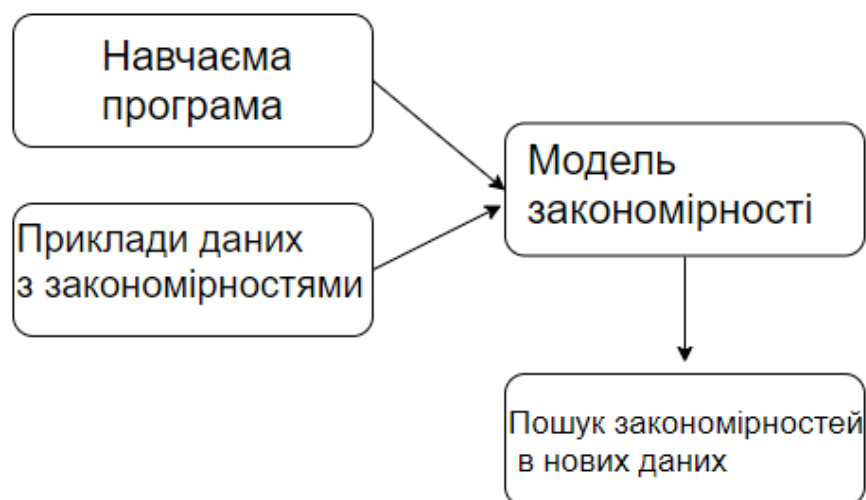


Рисунок 4.1 – Основні частини

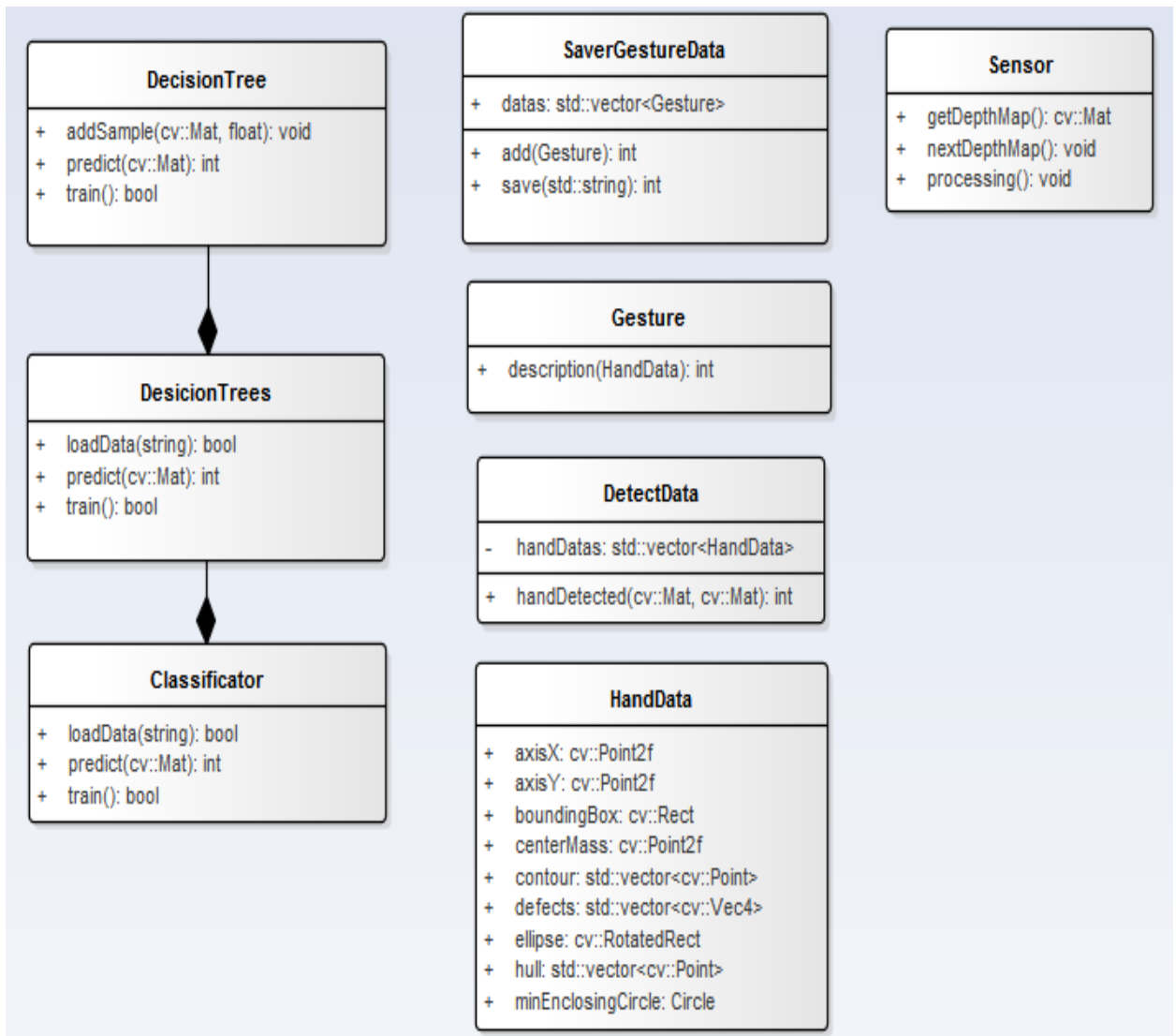


Рисунок 4.2 – Діаграма основних класів

Так, як багато класів використовується в обох модулях, необхідно зробити один виконавчий файл. За допомогою флагу(-train) який він приймає, файл зможе працювати в двох режимах – отримання даних по жесту та навчання с розпізнавання.

Опис декількох основних класів:

Клас Classifier відповідає за класифікацію жестів, спочатку необхідно загрузити дані с закономірностями, а потім передаючи йому дані які не ввійшли до закономірностей, дізнатися до якого класу належать нові дані. Під собою він скриває масив дерев рішень, а сам алгоритм побудови дерева використовується з бібліотеки, де кожне дерево для своєї групи жестів, довжина векторів яких однакова.

Клас Sensor інкапсулює роботу з сенсором глибини, у проекті використовується Kinect. За допомогою цього класу можна отримувати послідовність карт глибини та робити попередню обробку яка може бути потрібна для конкретного сенсору.

Клас DetectData обробляє карту глибини та на виході дає сирі дані для знайденого жесту.

Клас Gesture перетворює дані отримані за допомогою класу DetectData, та на виході дає дескриптор, який описує жест.

Після того як отримали об'єкт класу Gesture, можемо використовувати ці дані для створення закономірностей чи використовувати класифікатор для розпізнавання жесту який описує цей дескриптор.

Так-як ми використовуємо навчання з «вчителем», тому в режимі отримання інформації про жест потрібно спочатку вказати номер(клас) жесту, а потім показувати сам жест перед сенсором.

4.2 Робота з Kinect

В якості камери глибини використовуємо камеру Kinect, хоча для нашої задачі можна використовувати і інші. Головне щоб робота з камерою була швидка.

Щоб працювати з камерою, потрібно завантажити SDK[13] для неї з офіційного сайту Microsoft.

Після завантаження та установки SDK можна працювати з камерою. В SDK входить драйвер та приклади по роботі з камерою.

Щоб підключити камеру до свого проекту, потрібно прописати шлях до заголовочних файлів с розширенням «*.h», які знаходяться в папці «inc». Та прописати шлях до бібліотек «.dll», які знаходяться в папці «lib».

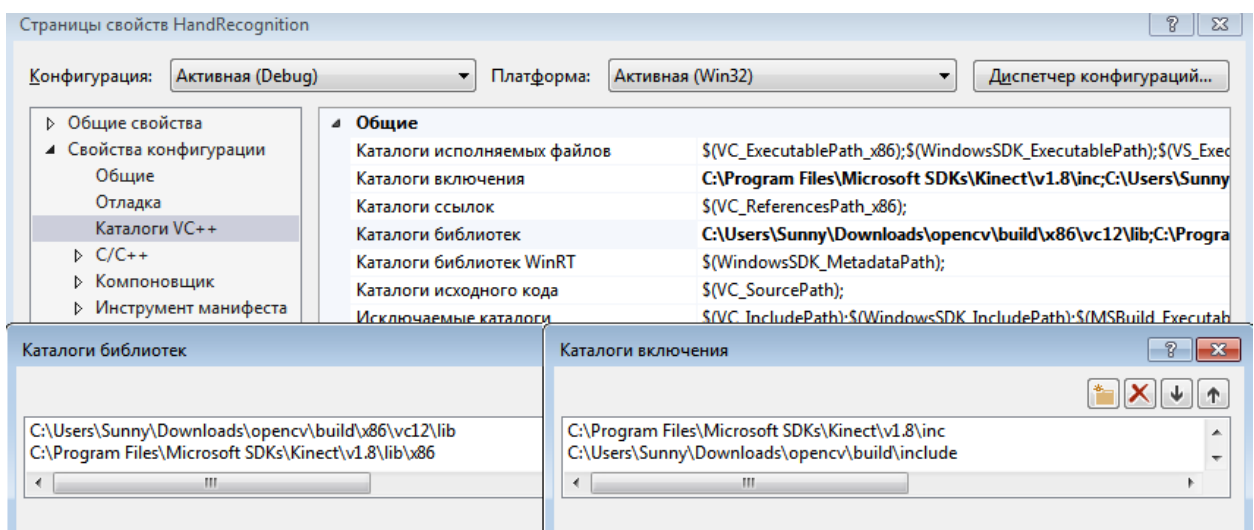


Рисунок 4.3 - Шлях до заголовків та динамічної бібліотеки

Після того як налаштована робота з камерою, приступимо до захоплення зображення з пристрою, для цього його необхідно ініціалізувати. Робиться це за допомогою методу `NuiInitialize`, який у якості параметра приймає бітову маску, що описує список підсистем пристрої, які ми плануємо використовувати (сенсор глибини, камеру або пошук гравців на відео).

Для того, щоб отримувати зображення з Kinect необхідно ініціалізувати потік захоплення зображення. Для цих цілей використовується метод `NuiImageStreamOpen`, який у якості параметрів приймає:

- тип зображення (кольорове зображення, буфер глибини і т.д.);
- здатність (від 80x60 до 1280x1024);
- прапори обробки зображення (в beta-версії SDK не використовується);
- кількість кешованих кадрів (максимальне значення `NUI_IMAGE_STREAM_FRAME_LIMIT_MAXIMUM` на даний момент дорівнює 4);
- дескриптор події, яка буде виникати при отриманні нового кадру (необов'язковий параметр, але на ділі виявилось, що якщо передати `NULL`, то потік захоплення може і не запуститися);
- покажчик на змінну, в яку буде записаний дескриптор потоку захоплення зображення при успішному завершенні роботи функції.

Для того щоб зупинити захоплення зображень з пристрою, необхідно викликати метод `NuiShutdown`, а після завершення роботи з екземпляром `INuiInstance` потрібно звільнити пам'ять за допомогою функції `MSR_NuiDestroyInstance`, в параметр якої передати покажчик на об'єкт `INuiInstance`.

Для того, щоб почати отримання буфера глибини, необхідно викликати метод `NuiImageStreamOpen` в якості першого параметра передати значення, що містить прапор `NUI_IMAGE_TYPE_DEPTH_AND_PLAYER_INDEX` або `NUI_IMAGE_TYPE_DEPTH`. Найбільш відповідний для подальшої обробки буфер виходив з використанням прапора `NUI_IMAGE_TYPE_DEPTH_AND_PLAYER_INDEX`. У вихідному коді подібний виклик буде виглядати таким чином:

```
hr = mNuiSensor->NuiImageStreamOpen(NUI_IMAGE_TYPE_DEPTH,
                                     NUI_IMAGE_RESOLUTION_640x480,
                                     0,
                                     2, mNextDepthFrameEvent,
                                     &mDepthStreamHandle);
```

Рисунок 4.4 – Виклик метода `NuiImageStreamOpen`

В змінній `mDepthStreamHandle` буде дескриптор потоку захвата карти глибини. Коли нове зображення буде готове буде викликано подію `mNextDepthFrameEvent`. Очікування цієї події можна реалізувати за допомогою функції `WaitForMultipleObjects` або `WaitForSingleObject`.

Отримати ж сам буфер з пристрою можна за допомогою методу `NuiImageStreamGetNextFrame`, в якому в якості параметрів потрібно передати:

- дескриптор потоку захоплення;
- період очікування буфера в мілісекундах;
- покажчик на структуру `NUI_IMAGE_FRAME`, в яку буде записана інформація про отриманий буфер.

```

HRESULT hr;
NUI_IMAGE_FRAME imageFrame;

hr = mNuiSensor->NuiImageStreamGetNextFrame(mDepthStreamHandle,
                                             0,
                                             &imageFrame);

if (FAILED(hr))
{
    return false;
}

```

Рисунок 4.4 – Виклик методу `NuiImageStreamGetNextFrame`

В отриманому екземплярі `NUI_IMAGE_FRAME` нас, на даний момент, найбільше цікавить поле `NuiImageBuffer * pFrameTexture`.

Для роботи безпосередньо з даними буфера, необхідно викликати метод `LockRect` в екземплярі `NUI_IMAGE_FRAME`. У методі `LockRect` чотири параметри.

В якості першого параметра потрібно передати `0`, в якості другого - покажчик на структуру `KINECT_LOCKED_RECT`, в яку після успішного завершення роботи функції, будуть записані дані для роботи з буфером. В якості третього параметра передаємо `NULL`, як четвертого - `0`.

Далі в структурі `KINECT_LOCKED_RECT` нас цікавить поле `pBits`, яке містить безпосередньо дані глибини. Для кожного пікселя зображення в буфері відводиться 2 байти.

Для того, щоб завершити роботу з отриманим зображенням, необхідно звільнити пам'ять, виділену під буфер. Зробити це можна за допомогою методу `NuiImageStreamReleaseFrame`, який у якості параметрів приймає дескриптор потоку і покажчик на екземпляр `NUI_IMAGE_FRAME`.

Підсумок:

- для того, щоб почати захоплення карти, потрібно ініціалізувати пристрій методом `NuiInitialize`;
- потім потрібно стартувати потік захоплення за допомогою методу `NuiImageStreamOpen`;
- при отриманні нового зображення викликається подія, дескриптор якої ми передали в `NuiImageStreamOpen`;
- після виклику події можна отримати фрейм за допомогою методу `NuiImageStreamGetNextFrame`;
- потім виконати захоплення буфера за допомогою методу `LockRect`;
- після цього пройтися по буферу і отримати колір кожного пікселя, нормалізував значення глибини.
- звільнити буфер за допомогою методу `NuiImageStreamReleaseFrame`;
- для того, щоб зупинити захоплення зображень з пристрою, треба звільнити його методом `NuiShutdown`.

4.3 Аналіз результатів

Після того, як ми використали програму з першим модулем для отримання інформації по кожному жесту ми зберегли цю вибірку в файл. В кожній строчці якого була інформація (дескриптор) про жест та номер жесту. Для кожного жесту було більше ніж 40 таких строчок. В результаті вибірка була з 500 строк.

Потім використали другий модуль програми, в який передали цей файл, обучили множину дерев на цій вибірці, та перевірили точність розпізнавання.

Перевірка відбувалась таким чином: запустили програму та перед камерою показували жест, програма виводила той жест, який вона розпізнала, звірялась правильність розпізнаного жесту. Потім показувався довільний жест та знову показували перед камерою жест який перевіряли. Таких повторень зроблено від 20 до 30, з різним нахилом жесту. Так як, це камера глибини то немає ніяких обмежень на умови

використання, її можна використовувати навіть в темряві. Одинадцять жестів які піддавалися розпізнаванню зображені на рисунку 6.1

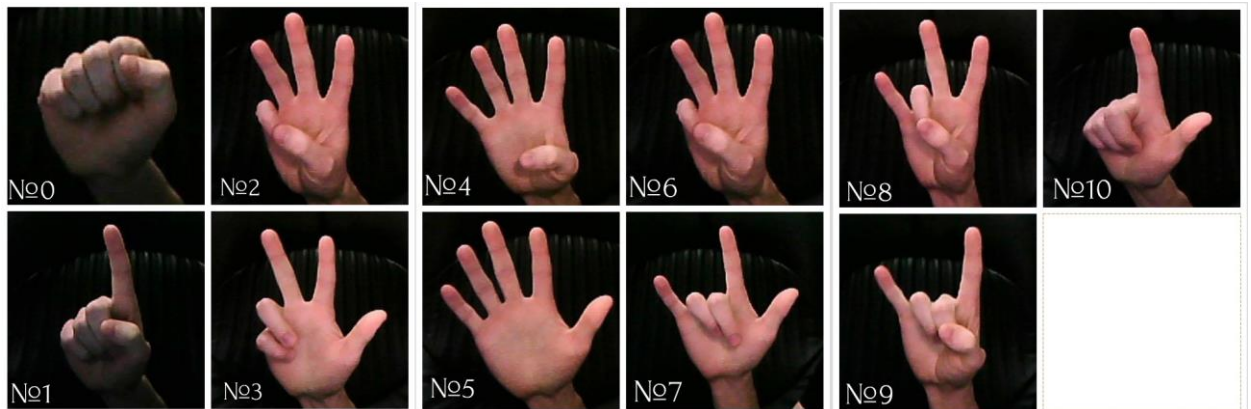


Рисунок 4.5 – Жести які піддавалися розпізнаванню

Результати розпізнавання наведені у таблиці 6.1.

Таблиця 4.1 – Таблиця результатів

Номер жесту	Точність розпізнавання
0	100%
1	100%
2	100%
3	100%
4	100%
5	100%
6	100%
7	100%
8	100%
9	100%
10	100%

В таблиці 4.1 зведений результат по перевірці точності програми. Показано номер жесту та точність розпізнавання. Така ймовірність вийшла тому, що значення дескрипторів для різних жестів не пересікаються між собою. .

Можна зробити одне зауваження, жести були тільки для правої руки і в програмі на даний час немає можливості розпізнати – права чи ліва це рука.

Наведемо далі опис дескрипторів, які було використано при класифікації для кожного жесту.

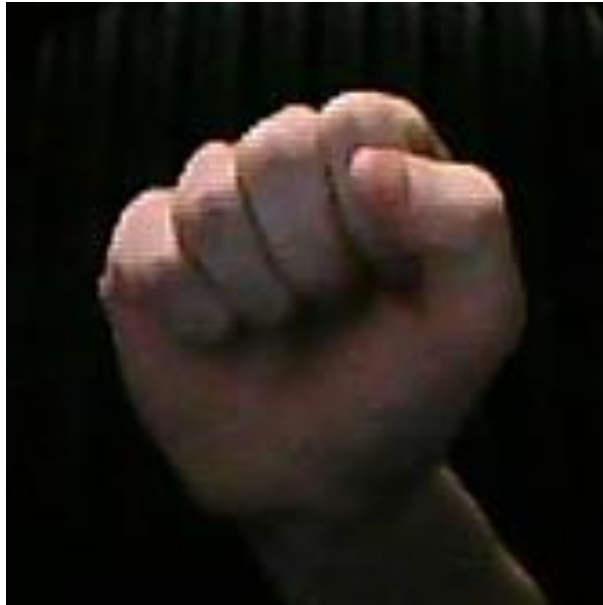


Рисунок 4.6 – Жест номер 0

Таблиця 4.2 – Дескриптор для 0 жесту

Змінна дескриптора	Значення дескриптора
x_1	0.665

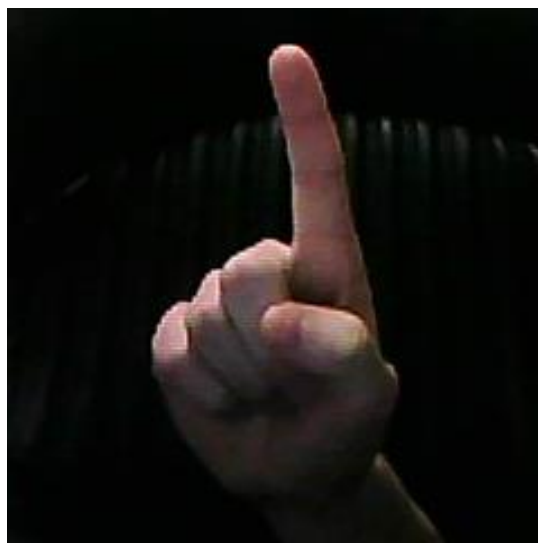


Рисунок 4.7 – Жест номер 1

Таблиця 6.3 – Дескриптор для жесту номер 1

Змінна дескриптора	Значення дескриптора
x_1	17.8015



Рисунок 4.8 – Жест номер 2

Таблиця 4.4 – Дескриптор для 2 жесту

Змінна дескриптора	Значення дескриптора
x_1	40.2854
x_2	27.4925
x_3	1
x_4	0.896548
x_5	12.8901



Рисунок 4.9 – Жест номер 3

Таблиця 4.5 – Дескриптор для 3 жесту

Змінна дескриптора	Значення дескриптора
x_1	41.5283
x_2	60.5095
x_3	27.5385
x_4	40.3338
x_5	0.974993
x_7	0.885856
x_8	1
x_9	0.484831
x_{10}	15.804



Рисунок 4.10 – Жест номер 4

Таблиця 4.6 – Детриптор для 4 жесту

Змінна дескриптора	Значення дескриптора
x_1	46.9552
x_2	29.9728
x_3	39.5238
x_4	31.3875
x_5	21.1899
x_6	28.132
x_7	0.628723
x_8	0.873725
x_9	12.9689
x_{10}	0.84609
x_{11}	0.985442
x_{12}	1
x_{13}	0.880368
x_{14}	17.3107

Таблиця 4.7 – Дескриптор для 5 жесту

Змінна дескриптора	Значення дескриптора
x_1	30.6498
x_2	34.7558
x_3	66.1713
x_4	46.4591
x_5	20.0895
x_6	23.6904
x_7	51.0512
x_8	26.7903
x_9	0.759495
x_{10}	0.84609
x_{11}	0.865839
x_{12}	0.903179
x_{13}	0.880368
x_{14}	0.784618
x_{15}	1
x_{16}	0.524918
x_{17}	0.483009
x_{18}	0.762044
x_{19}	13.1829

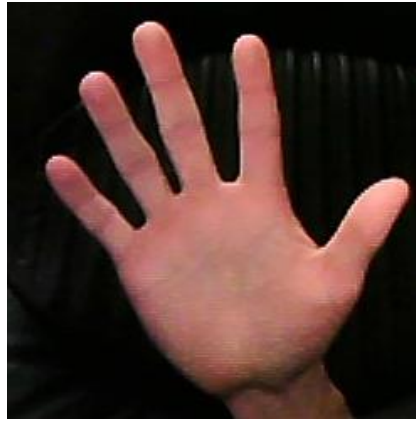


Рисунок 4.12 – Жест номер 5

Таблиця 4.8 – Детриптор для 6 жесту

Змінна дескриптора	Значення дескриптора
x_1	35.2049
x_2	40.5965
x_3	24.7694
x_4	31.5406
x_5	0.730743
x_7	0.909744
x_8	1
x_9	0.908746
x_{10}	12.9689



Рисунок 4.13 – Жест номер 6

Таблиця 4.9 – Дескриптор для 7 жесту

Змінна дескриптора	Значення дескриптора
x_1	75.0923
x_2	77.7245
x_3	45.3545
x_4	54.6468
x_5	0.79436
x_7	0.623295
x_8	1
x_9	0.451664
x_{10}	8.55199



Рисунок 4.14 – Жест номер 7

Таблиця 4.10 – Дескриптор для 8 жесту

Змінна дескриптора	Значення дескриптора
x_1	52.9922
x_2	48.603
x_3	38.574
x_4	33.129
x_5	0.563803
x_7	1
x_8	0.93026
x_9	0.807285
x_{10}	10.6917



Рисунок 4.15 – Жест номер 8

Таблиця 4.11 – Дескриптор для 9 жесту

Змінна дескриптора	Значення дескриптора
x_1	71.7708
x_2	42.2682
x_3	1
x_4	0.872412
x_5	15.7692



Рисунок 4.16 – Жест номер 9



Рисунок 4.17 – Жест номер 10

Таблиця 6.12 – Дескриптор для 10 жесту

Змінна дескриптора	Значення дескриптора
x_1	70.6067
x_2	51.569
x_3	1
x_4	0.543624
x_5	19.4015

4.4 Навчання класифікатора

Перед використанням програми ми змінили нашу вибірку так, щоб кожний дескриптор мав однакову довжину, ми записали нулі в пропущених місцях. У вибірці 322 строки. Використовуючи програму Deductor [14] побудували дерево рішень для нашої вибірки.

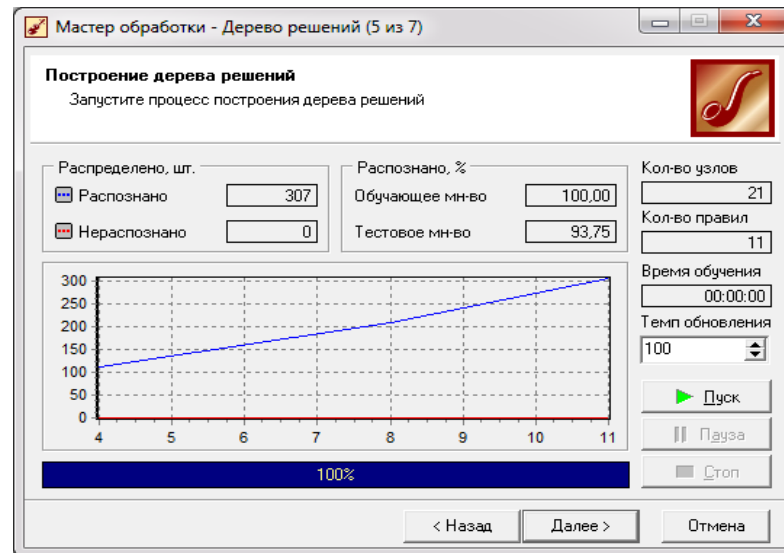


Рисунок 4.18 – Побудова дерева рішень

Отримали 21 вузол та 11 правил. Після того як побудували дерево рішень, ми можемо проаналізувати наскільки добре можна використовувати дерево для класифікації, та наскільки добре підходить наша вибірка.

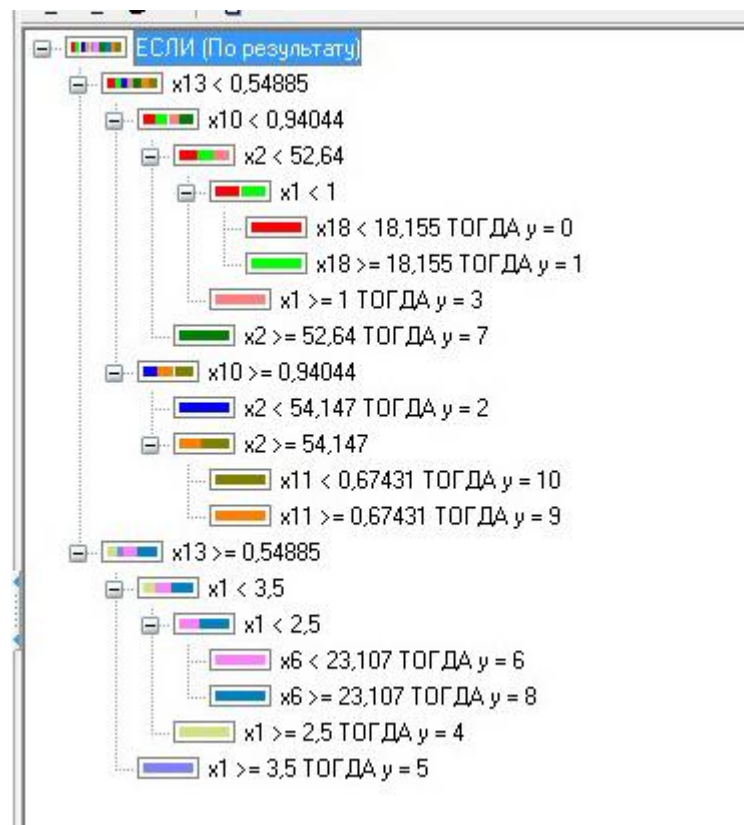


Рисунок 4.19 – Вузли дерева

Таблиця 6.13- Правила

	Умови	Слідство(y)	Підтримка		Достовірність	
			%	Кількість	%	Кількість
1	$x_{13} < 0,54885$ И $x_{10} < 0,94044$ И $x_2 < 52,64$ И $x_1 < 1$ И $x_{18} < 18,155$	0	9,12	28	100,00	28
2	$x_{13} < 0,54885$ И $x_{10} < 0,94044$ И $x_2 < 52,64$ И $x_1 < 1$ И $x_{18} \geq 18,155$	1	9,12	28	100,00	28
3	$x_{13} < 0,54885$ И $x_{10} < 0,94044$ И $x_2 < 52,64$ И $x_1 \geq 1$	3	7,82	24	100,00	24
4	$x_{13} < 0,54885$ И $x_{10} < 0,94044$ И $x_2 \geq 52,64$	7	10,10	31	100,00	31
5	$x_{13} < 0,54885$ И $x_{10} \geq 0,94044$ И $x_2 < 54,147$	2	8,47	26	100,00	26
6	$x_{13} < 0,54885$ И $x_{10} \geq 0,94044$ И $x_2 \geq 54,147$ И $x_{11} < 0,67431$	10	10,47	33	100,00	33
7	$x_{13} < 0,54885$ И $x_{10} \geq 0,94044$ И $x_2 \geq 54,147$ И $x_{11} \geq 0,67431$	9	9,12	28	100,00	28
8	$x_{13} \geq 0,54885$ И $x_1 < 3,5$ И $x_1 < 2,5$ И $x_6 < 23,107$	6	10,10	31	100,00	31
9	$x_{13} \geq 0,54885$ И $x_1 < 3,5$ И $x_1 < 2,5$ И $x_6 \geq 23,107$	8	14,01	43	100,00	43
10	$x_{13} \geq 0,54885$ И $x_1 < 3,5$ И $x_1 \geq 2,5$	4	7,17	22	100,00	22

В результаті ми отримали 11 правил за допомогою яких можемо однозначно класифікувати жести. Ймовірність збігається з той ймовірністю, яку ми одержали в своїй програмі. Це підтверджує коректність нашої роботи.



















Целевой атрибут: y			
№	Атрибут	Значимость, %	/
13	x13		27,513
1	x1		19,339
10	x10		18,706
2	x2		16,367
6	x6		6,932
11	x11		5,797
18	x18		5,347
17	x17		0,000
16	x16		0,000
15	x15		0,000
14	x14		0,000
12	x12		0,000
5	x5		0,000
4	x4		0,000
3	x3		0,000
9	x9		0,000
8	x8		0,000
7	x7		0,000

Рисунок 4.20 – Значимість атрибутів (елементів вектора)

На рисунку 4.20 видно які атрибути більш значимі а які менш. Отже самі значимі X_{13} та X_1 – це кількість дефектів. Отже можна викинути ті атрибути які не впливають на результат.

5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ. ЕКОЛОГІЯ

В даному розділі проведено аналіз потенційних небезпечних та шкідливих виробничих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

Завданням даної магістерської роботи було проаналізувати існуючі методи керування комп'ютером за допомогою жестів та розробка власного універсального методу на основі переваг з декількох інших, і як результат було створено універсальний метод керування комп'ютером за допомогою жестів. За цим методом в подальшому розроблятиметься реальна система, яка значно полегшить процес керування комп'ютером за допомогою жестів. Так як в процесі проектування використовувалися комп'ютерні технології, то аналіз потенційно небезпечних і шкідливих виробничих чинників виконується для персонального комп'ютера на якому буде розроблятися/використовуватися розроблена програма.

5.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проектного об'єкту, що мають вплив на персонал

У даному дипломному проекті розробляється програмне забезпечення. Розроблене програмне забезпечення орієнтоване на роботу з персональним комп'ютером. Експлуатовані для вирішення внутрішньовиробничих завдань ПЕОМ типу IBM PC мають наступні характеристики:

споживана потужність	220 Вт;
робоча напруга	220 В;
напруга джерел живлення	+12 В; - 12 В; +5 В;
робоча частота	50 Гц.

Виходячи з приведених характеристик, вочевидь, що для людини існує небезпека поразки електричним струмом, унаслідок недбалого поводження з комп'ютером і порушення правил експлуатації, залишення частин ПЕОМ, що знаходяться під напругою, відкритими або знятих для ремонту вузлів.

Відповідно до ГОСТ 12.1.005-88 [15], до легкої фізичної роботи відносяться всі види діяльності, виконувані сидячи і ті, що не потребують фізичної напруги. Робота користувача ПК відноситься до категорії 1а.

При роботі на ПЕОМ користувач піддається ряду потенційних небезпек. Унаслідок недотримання правил техніки безпеки при роботі з машиною (невиконання огляду відкритих частин ПЕОМ, що знаходяться під напругою або знятих для ремонту вузлів) для користувача існує небезпека поразки електричним струмом.

Джерелами підвищеної небезпеки можуть служити наступні елементи:

- розподільний щит;
- джерела живлення;
- блоки ПЕОМ і друку, що знаходяться в ремонті.

Ще одна проблема полягає у тому, що спектр випромінювання комп'ютерного монітора включає рентгенівську, ультрафіолетову і інфрачервону області, а також широкий діапазон хвиль інших частот. Небезпека рентгенівського проміння низька, оскільки цей вид випромінювання поглинається речовиною екрану. Проте велику увагу слід приділяти біологічним ефектам низькочастотних електромагнітних полів (аж до порушення ДНК).

Відповідно до ГОСТ 12.0.003-74 [16], при обслуговуванні ПЕОМ мають місце фізичні і психофізичні небезпечні, а також шкідливі виробничі чинники:

- підвищене значення напруги в електричному ланцюзі, замикання якої може відбутися через тіло людини;
- підвищений рівень статичної електрики;
- підвищений рівень електромагнітних випромінювань;
- підвищена або знижена температура повітря робочої зони;
- підвищений або знижений рух повітря;
- підвищена або знижена вологість повітря;
- відсутність або недостатність природного світла;
- підвищена пульсація світлового потоку;
- недостатня освітленість робочого місця;
- підвищений рівень шуму на робочому місці;
- розумове перенапруження;
- емоційні навантаження;
- монотонність праці.

5.2 Заходи щодо техніки безпеки

Основним небезпечним чинником при роботі з ЕОМ є небезпека ураження людини електричним струмом, яка посилюється тим, що органи чуття людини не можуть на відстані знайти наявності електричної напруги на устаткуванні.

Проходячи через тіло людини, електричний струм чинить на нього складну дію, що є сукупністю термічної (нагрів тканин і біологічних середовищ), електролітичної (розкладання крові і плазми) і біологічної (роздратування і збудження нервових волокон і інших органів тканин організму) дій.

Тяжкість ураження людини електричним струмом залежить від цілого ряду чинників:

- значення сили струму;
- електричного опору тіла людини і тривалості протікання через нього струму;
- роду і частоти струму;
- індивідуальних властивостей людини і навколишнього середовища.

Розроблений дипломний проект передбачає наступні технічні способи і засоби, що застерігають людину від ураження електричним струмом:

- заземлення електроустановок;
- занулення;
- захисне відключення;
- електричне розділення мережі;
- використання малої напруги;
- ізоляція частин, що проводять струм;
- огорожа електроустановок.

Занулення зменшує напругу дотику і обмежує години, протягом яких людина, торкнувшись до корпусу, може потрапити під дію напруги.

Струм однофазного короткого замикання визначається по наближеній формулі:

$$I_K = \frac{U_\phi}{Z_\Pi + \frac{Z_\Gamma}{3}}, \quad (5.1)$$

де U_ϕ - номінальна фазна напруга мережі, В;

Z_Π - повний опір петлі, створене фазними і нульовими дротами, Ом;

Z_Γ - повний опір струму короткого замикання на корпус, Ом.

Згідно таблиці 4, ДСТУ 7237:2011 [17]: $Z_T / 3 = 0,1 \text{ Ом}$.

Для провідників і жил кабелю для розрахунку повного опору петлі використовуємо формулу(5.2.) :

$$Z_{\Pi} = \sqrt{R_{\Pi}^2 + X_{\Pi}^2} , \quad (5.2)$$

де $R_{\Pi} = R_{\phi} + R_0$ - сумарний активний опір фазного R_{ϕ} і нульового R_0 дротів, Ом;
 X_{Π} - індуктивний опір паяння дротів, Ом.

Перетин 1 км мідного дроту $S = 2.5 \text{ мм}$, тоді згідно таблицям 5 і 6, ДСТУ 7237:2011 [17], має такий опір:

$$X_{\Pi} = 0,11 \text{ Ом};$$

$$R_{\phi} = 7,55 \text{ Ом};$$

$$R_0 = 7,55 \text{ Ом}.$$

$$\text{Отже, } R_{\Pi} = 7,55 + 7,55 = 15,1 \text{ Ом}.$$

Тоді по формулі (5.2) знаходимо повний опір петлі :

$$Z_{\Pi} = \sqrt{15,1^2 + 0,11^2} \approx 15,1 \text{ (Ом)}.$$

Струм однофазного короткого замикання рівний:

$$I_k = \frac{220}{15,1 + 0,1} = 14,47 \text{ (А)}.$$

Дія плавкої вставки на ПЕОМ забезпечується, якщо виконується співвідношення:

$$I_k \geq k * I_n , \quad (5.3)$$

де I_n - номінальний струм спрацьовування плавкої вставки, А;

k - коефіцієнт кратності нелінійного струму I_n , А.

Коефіцієнт кратності нелінійного струму I_n розраховується по формулі (5.4.) :

$$I_n = P / U , \quad (5.4)$$

де $P = 220 \text{ Вт}$ - споживана потужність;

$U = 220 \text{ В}$ - робоча напруга;

$k = 3 \text{ A}$ - для плавких вставок.

Отже, $I_n = 220 / 220 = 1 \text{ A}$.

Підставивши значення у вираз (5.3), одержимо:

$$14,47 > 3 * 1.$$

Таким чином, доведено, що апарат забезпечить спрацьовування(і захист) при підвищенні номінального струму.

5.3 Заходи, що забезпечують виробничу санітарію і гігієну праці

Вимоги до виробничих приміщень встановлюються ДСП 173-96 [18], ДБН, відповідними ГОСТами і ОСТами з урахуванням небезпечних і шкідливих чинників, що утворюються в процесі експлуатації електроустаткування.

Підвищення працездатності людини і збереження її здоров'я забезпечується стабільними метеорологічними умовами.

Мікроклімат виробничих приміщень визначається діючими на організм людини поєднаннями температури, вологості і швидкості руху повітря, а також температури навколишніх поверхонь. Значне коливання параметрів мікроклімату приводить до порушення систем кровообігу, нервової і потовидільної, що може викликати підвищення або пониження температури тіла, слабкість, запаморочення і навіть непритомність.

Відповідно до ГОСТ 12.1.005-88 [15] встановлюють оптимальну і допустиму температуру, відносну вологість і швидкість руху повітря в робочій зоні. За відсутності надмірного тепла, вологи, шкідливих речовин в приміщенні досить природної вентиляції.

У приміщенні для виконання робіт операторського типу(категорія 1а), пов'язаних з нервово-емоційною напругою, проектом передбачається дотримання наступних нормованих величин параметрів мікроклімату (табл.5.1).

Таблиця 5.1 - Санітарні норми мікроклімату робочої зони приміщень для робіт категорії 1а.

Пора року	Температура, С	Відносна вологість, %	Швидкість руху повітря, м/с
Холодна	22...24	40...60	0,1
Тепло	23...25	40...60	0,1

У приміщенні, де знаходиться ПЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (з пристроєм вентиляційних каналів в перекриттях будівлі і вертикальних шахт) й установленого промислового кондиціонера фірми Mitsubishi, який дозволяє вирішити переважну більшість завдань по створінню та підтримці необхідних параметрів повітряного середовища. Цей метод забезпечує приток потрібної кількості свіжого повітря, визначеного в ДБН (30 м³ в годину на одного працівника).

Шум на виробництві має шкідливу дію на організм людини. Стомлення операторів через шум збільшує число помилок при роботі, призводить до виникнення травм. Для оператора ПЕОМ джерелом шуму є робота принтера. Щоб усунути це джерело шуму, використовують наступні методи. При покупці принтера слід вибирати найбільш шумозахисні матричні принтери або з великою швидкістю роботи(струменеві, лазерні). Рекомендується принтер поміщати в найбільш віддалене місце від персоналу, або застосувати звукоізоляцію та звукопоглинання(під принтер підкладають демпфуючі підкладки з пористих звукопоглинальних матеріалів з листів тонкої повсті, поролону, пеноплєну).

При роботі на ПЕОМ, проектом передбачені наступні методи захисту від електромагнітного випромінювання : обмеження часом, відстанню, властивостями екрану.

Обмеження годині роботи на ПЕОМ складає 3,5-4,5 години. Захист відстанню передбачає розміщення монітора на відстані 0,4-0,5 м від оператора. Передбачений монітор 20" TFT, Samsung 2043BW відповідає вимогам стандарту ТСО'03.

ТСО'03 пред'являє жорсткі вимоги в таких областях: ергономіка (фізична, візуальна і зручність користування), енергія, випромінювання (електричних і магнітних полів), навколишнє середовище і екологія, а також пожежна та електрична безпека, які відповідають всім вимогам ДСанПіН 3.3.2.007-98 [21].

Для зниження стомлюваності та підвищення продуктивності праці обслуговуючого персоналу в колірній композиції інтер'єру приміщень для ПЕОМ дипломним проектом пропонується використовувати спокійні колірні поєднання і покриття, що не дають відблисків.

У проекті передбачається використання сумісного освітлення. У світлий час доби приміщення освітлюватиметься через віконні отвори, в решту часу використовуватиметься штучне освітлення.

Як штучне освітлення необхідно використовувати штучне робоче загальне освітлення. Для загального освітлення необхідно використовувати люмінесцентні лампи.

Вони володіють наступними перевагами: високою світловою віддачею, тривалим терміном служби, хоча мають і недоліки: високу пульсацію світлового потоку.

При експлуатації ПЕОМ виробляється зорова робота. Відповідно до ДБН В.2.5-28:2015 [22] ця робота відноситься до розряду 5а. При цьому нормоване освітлення на робочому місці(E_n) при загальному освітленні рівна 200 лк.

Приміщення завдовжки 12 м, шириною 10 м, заввишки 4 м обладнується світильниками типу ЛПО2П, оснащеними лампами типу ЛБ зі світловим потоком 3120 лм кожна.

Виконаємо розрахунок кількості світильників в робочому приміщенні завдовжки $a=12$ м, шириною $b=10$ м, заввишки $z=4$ м, використовуючи формулу (5.5) розрахунку штучного освітлення при горизонтальній робочій поверхні методом світлового потоку:

$$n = (E \cdot S \cdot Z \cdot k) / (F \cdot U \cdot M), \quad (5.5)$$

де F - світловий потік = 3120 лм;

E - максимально допустима освітленість робочих поверхонь = 200 лк;

S - площа підлоги = 120 м²;

Z - поправочний коефіцієнт світильника = 1,2;

k - коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації світильників = 1,5;

n - кількість світильників;

U - коефіцієнт використання освітлювальної установки = 0,6;

M - кількість ламп у світильнику = 2.

З формули (5.5) виразимо n (5.6) і визначимо кількість світильників для даного приміщення:

$$n = (E \cdot S \cdot Z \cdot k) / (F \cdot U \cdot M), \quad (5.6)$$

Отже, $n = (200 \cdot 120 \cdot 1,2 \cdot 1,5) / (3120 \cdot 0,6 \cdot 2) = 12$.

Виходячи з цього, рекомендується використовувати 12 світильників. Світильники слід розміщувати рядами, бажано паралельно стіні з вікнами. Схема розташування світильників зображена на рис. 5.1.

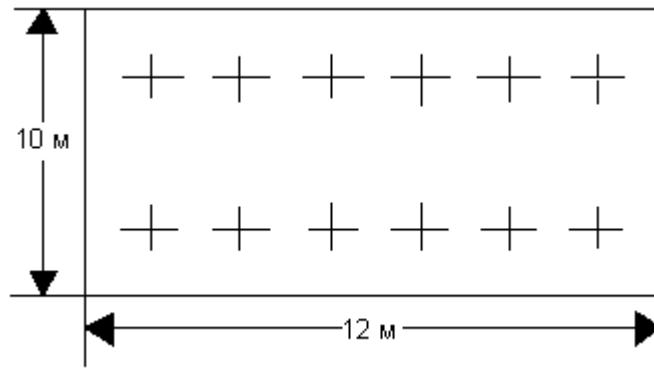


Рисунок 5.1 - Схема розташування світильників

5.4 Рекомендації по пожежній безпеці

Пожежі в приміщеннях, де встановлена обчислювальна техніка, представляють небезпеку для життя людини. Пожежі також пов'язані як з матеріальними втратами, так і з відмовою засобів обчислювальної техніки, що у свою чергу спричиняє за собою порушення ходу технологічного процесу.

Пожежа може виникнути при наявності горючої речовини та внесення джерела запалювання в горюче середовище. Пальними матеріалами в приміщеннях, де розташовані ПЕОМ, є:

- поліамід - матеріал корпусу мікросхеми, горюча речовина, температура самозаймання аерогелю 420 °С ;
- полівінілхлорид - ізоляційний матеріал, горюча речовина, температура запалювання 335 °С, температура самозаймання 530 °С, кількість енергії, що виділяється при згоранні - 18000 - 20700 кДж/кг;
- стеклотекстоліт ДЦ - матеріал друкарських плат, важкозаймистий матеріал, показник горючості 1.74, не схильний до температурного самозаймання;
- пластика кабельний №489 - матеріал ізоляції кабелю, горючий матеріал, показник горючості більш 2.1;
- деревина - будівельний і обробний матеріал, матеріал з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, теплота згорання 18731 - 20853 кДж/кг, температура запалювання 399 °С, схильна до самозаймання.

Згідно ДСТУ Б В.1.1-36:2016 [24] приміщення відносяться до категорії В (пожежовибухонебезпечним) і згідно правилам побудови електроустановок простір усередині приміщення відноситься до вогнебезпечної зони класу П - Па (зони, розташовані в приміщеннях, в яких зберігаються тверді горючі речовини).

Потенційними джерелами запалення при роботі ПЕОМ є:

- іскри при замиканні і розмиканні ланцюгів;
- іскри і дуги коротких замикань;
- перегриви від тривалого перевантаження і наявності перехідного опору.

Продуктами згорання, що виділяються при пожежі, є : оксид вуглецю, сірчистий газ, оксид азоту, синильна кислота, акролеїн, фосген, хлор та ін. При горінні пластмас, окрім звичайних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол та ін., що шкідливо впливають на організм людини.

Для захисту персоналу від дії небезпечних і шкідливих чинників пожежі проектом передбачається застосування промислового протигазу з коробкою марки В(жовта).

Пожежна безпека об'єктів народного господарства регламентується ГОСТ 12.1.004-91 [19] і забезпечується системами запобігання пожежам і протипожежному захисту. Для успішного гасіння пожеж вирішальне значення має швидке виявлення пожежі і своєчасний виклик пожежних підрозділів до місця пожежі.

Зменшити горюче навантаження не представляється можливим, тому проектом передбачається застосувати наступні способи і їх комбінації для запобігання утворенню(внесення) джерел запалення :

- застосування устаткування, що задовольняє вимогам електростатичної безпеки;
- застосування в конструкції швидкодіючих засобів захисного відключення можливих джерел запалення;
- виключення можливості появи іскрового заряду статичної електрики в горючому середовищі з енергією, рівної і вище мінімальної енергії запалення;
- підтримка температури нагріву поверхні машин, механізмів, устаткування, пристроїв, речовин і матеріалів, які можуть увійти до контакту з палим середовищем, нижче гранично допустимої, становить 80% якнайменшої температури самозаймання пального.
- заміна небезпечних технологічних операцій більш безпечними;
- ізольоване розташування небезпечних технологічних установок і устаткування;
- зменшення кількості палих і вибухонебезпечних речовин, що знаходяться у виробничих приміщеннях;
- запобігання можливості утворення палих сумішей на лінії, вентиляційних системах і ін.;
- механізація, автоматизація та справність(потокова) виробництва;

- суворе дотримання стандартів і точне виконання встановленого технологічного режиму;
- запобігання можливості появи в небезпечних місцях джерел запалення;
- запобігання розповсюдженню пожеж і вибухів;
- використання устаткування і пристроїв, при роботі яких не виникає джерел запалення;
- виконання вимог сумісного зберігання речовин і матеріалів;
- наявність громовідводу;
- ліквідація можливості самозаймання речовин і матеріалів .

Для запобігання пожежі в обчислювальних центрах проектом пропонується виконання наступних вимог :

- електроживлення ЕОМ повинно мати автоматичне блокування відключення електроенергії на випадок зупинки системи охолодження і кондиціонування;
- система вентиляції обчислювальних центрів повинна бути обладнана блокуючими пристроями, що забезпечують її відключення на випадок пожежі;
- робочі місця повинні бути оснащені пожежними щитами, сигналізацією, засобами для сповіщення про пожежну небезпеку (телефонами), медичними аптечками для надання першої медичної допомоги, розробленим планом евакуації.

Для зниження пожежної небезпеки в приміщеннях використовуються первинні засоби гасіння пожеж, а також система автоматичної пожежної сигналізації, яка дозволяє знайти початкову стадію загоряння, швидко і точно оповістити службу пожежної охорони про час і місце виникнення пожежі.

Відповідно до правил пожежної безпеки для промислових підприємств приміщення категорії В підлягають устаткуванню системами автоматичної пожежної сигналізації. Проектом передбачається застосування датчика типу ІДФ - 1(димовий фотоелектричний датчик), оскільки специфікою пожеж обчислювальної техніки і радіоапаратури є, в першу чергу, виділення диму, а потім - підвищення температури.

При виникненні пожежі в робочому приміщенні обслуговуючий персонал зобов'язаний негайно вжити заходи по ліквідації пожежі. Для ліквідації пожежі використовують вогнегасники (хімічно-пінні, пінні для повітря ОП-5, ОП-6, ОП-9, вуглекислотні ОУ-5), пісок, пожежний інвентар (сокири, ломы, багри, шерстяну або азбестову ковдри). Як засіб індивідуального захисту проектом передбачається використання промислового протигаза з маскою, фільтруючої коробки В.

В якості організаційно-технічних заходів рекомендується проводити навчання робочого персоналу правилам пожежної безпеки.

5.4 Вплив на навколишнє середовище

В даний час зростає кількість комп'ютерної техніки в усіх галузях діяльності людини. Багато користувачів і виробників помиляються, вважаючи, що зі зменшенням і удосконаленням комп'ютерів, зменшиться їх негативний вплив на навколишнє середовище.

На даний момент найбільш суворим з існуючих світових стандартів екологічності для комп'ютерної техніки є стандарт ТСО-99. У порівнянні з попередніми він містить додаткові обмеження по частині екології, ергономіки, енергоспоживання і емісії пристроїв.

Організація по захисту навколишнього середовища Greenpeace з 2006 року оцінює виробників електроніки за кількістю важких металів і отруйних речовин, наприклад інгібіторів горіння, використовуваних ними при виробництві (інгібітор - речовина, присутність якого в невеликих кількостях призводить до запобігання або уповільнення процесів горіння або корозії; інгібітори знижують швидкість хімічних реакцій або пригнічують їх). Однак навіть оцінки такої організації, як Greenpeace, не можуть претендувати на об'єктивність. Адже в одних випадках вона використовує перевірену інформацію, що стосується, наприклад, заходів щодо утилізації відходів, а в інших спирається тільки на дані виробника. А якщо компанія не повідомляє ніяких відомостей, то автоматично опиняється на нижніх рядках рейтингу. Крім того, енергетичні витрати на виробництво і перевезення продукції також необхідно враховувати при оцінці екологічної ефективності. Адже часи, коли техніка виготовлялася тільки на одному заводі, давно пройшли. Сьогодні окремі комплектуючі закупаються на різних підприємствах по всьому світу, після чого здійснюється складання пристроїв. Тому найчастіше навіть самі компанії не можуть знати, які шкідливі речовини потрапляють в атмосферу при виготовленні їх продукції і які саме метали або токсини в ній містяться.

ЖК-екрани - один з джерел парникових газів, які набагато шкідливіше діоксиду вуглецю. Рідкокристалічні монітори швидко знайшли популярність, прийшовши на зміну громіздким ЕПТ-моделям. І це не дивно, адже вони мають тонкі корпуса і споживають значно менше електроенергії. За іншим аспектам екологічної безпеки дисплеї на основі рідких кристалів також вважалися проривом, тому що в них не використовувався газ, що містить свинець. Досить довго ніхто не звертав уваги на застосовуваний для чищення РК-панелей тріфтористий азот (NF₃), і тільки в середині 2008 року вченими було доведено наявність даної хімічної речовини в атмосфері. Відкриття було вражаючим: порівняно з діоксидом вуглецю (CO₂) NF₃ має в 17 000 разів більше активного парникового газу, а

його атмосферний час напіврозпаду може складати від 550 до 740 світлових років (у CO₂ - від 30 до 40 років). Закону, який обмежував би рівень викиду NF₃, поки не існує.

Виявлення енерговитрат є таким же проблематичним процесом, як і визначення кількості матеріалів, придатних для вторинної переробки, і важких металів, що містяться в пристроях. Таким чином, надійним показником екологічності залишається тільки рівень енергоспоживання.

Полівінілхлорид, що позначається зазвичай аббревіатурою ПВХ, - це різновид пластику, що застосовується в самих різних цілях. З нього зроблена зовнішня оболонка кабелів, якими з'єднуються пристрої, він оточує електричний провід портативного комп'ютера. Це дешевий, міцний і вельми поширений матеріал. Разом з тим, за словами IT-аналітика «Грінпіс» Кейсі Харрелл, «ПВХ - найгірший з пластиків». Він є причиною виникнення гормонального дисбалансу, проблем в репродуктивній сфері та різних форм раку. Полівінілхлорид практично неможливо правильно утилізувати. Внаслідок старий матеріал виявляється зазвичай на звалищі з відходами або, того гірше, спалюється з метою вилучення мідних жил і інших цінних компонентів. При його згорянні утворюється вкрай шкідливий канцерогенний діоксин. Звалища і хімічні поховання забруднюють джерела води. Єдиний спосіб правильно утилізувати ПВХ полягає в тому, щоб відправити його в центр небезпечних відходів.

Залишається лише сподіватися, що настане час, коли технології будуть допомагати людині, не завдаючи незворотної шкоди здоров'ю навколишнього середовища.

ВИСНОВКИ

У результаті виконання магістерської атестаційної роботи було досліджено тему людино – комп'ютерної взаємодії. Аналіз показав, що це бурно розвиваюча область. Можливо в недалекому майбутньому, це новий крок в еволюції інтерфейсу для взаємодії з комп'ютером. Було проведено аналіз існуючих рішень в цій області. Kinect та Leap Motion одні з таких систем, які показали, що не тільки теоретично можливо створити такий інтерфейс, а що можливо і на практиці. Також досліджено низку алгоритмів розпізнавання жестів руки.

Було проаналізовано сенсори для отримання карти глибини. Використовуючи карту глибини було розроблено та реалізовано метод розпізнавання жестів руки оснований на дескрипторах та з використанням машинного навчання. Не зважаючи на простоту метода, він дає гарний результат для заданих жестів. Зрозуміло що такий підхід не буде працювати на деяких жестах, хоча все залежить від того як ми будемо отримувати дескриптори.

За допомогою цих жестів можна управляти комп'ютером, але таке управління буде менш комфортне ніж за допомогою динамічних жестів. Точніше, за допомогою динамічних жестів можливо керувати 3D інтерфейсом. Тому наступним кроком є розпізнавання динамічних жестів котрі в сумі дали б більшу можливість взаємодіяти з комп'ютером, та розробка 3D інтерфейсу.

У розділі «Охорона праці» виконано аналіз потенційних небезпек при роботі із засобами обчислювальної техніки і механізмами, розроблені заходи щодо техніки безпеки, заходи, які забезпечують виробничу санітарію і гігієну праці, розраховане штучне освітлення, виконані рекомендації по пожежній безпеці, розглянутий можливий вплив на навколишнє середовище.

ПЕРЕЛІК ПОСИЛАННЯ

- 1) Опис сенсора Kinect[Електронний ресурс] – Режим доступу: [www/ URL : http://uk.wikipedia.org/wiki/Kinect/](http://uk.wikipedia.org/wiki/Kinect/) - 15.10.2018
- 2) Офіційний сайт Leap Motion[Електронний ресурс] – Режим доступу: [www/ URL : https://www.leapmotion.com/](https://www.leapmotion.com/) - 15.10.2018
- 3) Опис технології WiSee[Електронний ресурс] – Режим доступу:[www/ URL : http://wisee.cs.washington.edu/wisee_paper.pdf](http://wisee.cs.washington.edu/wisee_paper.pdf) - 15.10.2018
- 4) Oikonomidis I. Efficient Model-based 3D Tracking of Hand Articulations using Kinect. [Text] / I.Oikonomidis, N.Kyriazis, A.A.Argyros In BMVC 2011. BMVA, 2011.
- 5) Виділення границь[Електронний ресурс] – Режим доступу: [www/ URL https://ru.wikipedia.org/wiki/Выделение_границ/](https://ru.wikipedia.org/wiki/Выделение_границ/) - 15.10.2018
- 6) Алгоритм отримання опуклої оболонки[Електронний ресурс] – Режим доступу [www/ URL : http://e-maxx.ru/algo/convex_hull_graham/](http://e-maxx.ru/algo/convex_hull_graham/) - 15.10.2018
- 7) Опис алгоритму Грехема[Електронний ресурс] – Режим доступу: [www/ URL : uk.wikipedia.org/wiki/Алгоритм_Грехема/](http://uk.wikipedia.org/wiki/Алгоритм_Грехема/) - 15.10.2018
- 8) Опис алгоритму Ендрю[Електронний ресурс] – Режим доступу: [www/ URL : uk.wikipedia.org/wiki/Алгоритм_Ендрю/](http://uk.wikipedia.org/wiki/Алгоритм_Ендрю/) - 15.10.2018
- 9) Опис алгоритму Чена[Електронний ресурс] – Режим доступу: [www/ URL : uk.wikipedia.org/wiki/Алгоритм_Чена/](http://uk.wikipedia.org/wiki/Алгоритм_Чена/) - 15.10.2018
- 10) Опис алгоритму швидкої оболонки[Електронний ресурс] – Режим доступу: [www/ URL : uk wikipedia.org/wiki/Алгоритм_швидкої_оболонки/](http://uk.wikipedia.org/wiki/Алгоритм_швидкої_оболонки/) - 15.10.2018
- 11) Шапиро, Л. Компьютерное зрение [Текст] / Л. Шапиро, Дж. Стокман; Пер. с англ. – М.: Бинум. Лаборатория знаний, 2006. – 752 с.
- 12) Офіційний сайт документації бібліотеки OpenCV[Електронний ресурс] – Режим доступу: [www/ URL : http://docs.opencv.org/](http://docs.opencv.org/) - 15.10.2018
- 13) Сайт загрузки SDK Kinect[Електронний ресурс] – Режим доступу: [www/ URL : https://www.microsoft.com/en-us/kinectforwindows/](https://www.microsoft.com/en-us/kinectforwindows/) - 15.10.2018
- 14) Програма Deductor[Електронний ресур] – Режим доступу: [www/ URL : http://www.basegroup.ru/download/deductor/](http://www.basegroup.ru/download/deductor/) - 15.10.2018
- 15) ГОСТ 12.1.005-88. Міждержавний стандарт. Система стандартів безпеки праці. Загальні санітарно-гігієнічні вимоги до повітря робочої зони
- 16) ГОСТ 12.0.003-74 Небезпечні і шкідливі виробничі фактори. Класифікація
- 17) ДСТУ 7237:2011 Національний стандарт України. Система стандартів безпеки праці. Електробезпека. Загальні вимоги та номенклатура видів захисту

18) ДСанПіН 3.3.2.007-98. Державні санітарні правила і норми. Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин.

19) ГОСТ 12.1.004-91. Пожежна безпека. Загальні вимоги .

20) ДБН В.2.5-67:2013. Опалення вентиляція та кондиціонування.

21) ГОСТ 12.1.006-84. Електромагнітні поля радіочастот. Допустимі рівні на робочих місцях і вимоги до проведення контролю

22) ДБН В.2.5-28:2015. Природне і штучне освітлення.

23) ГОСТ 12.4.009-83. Пожежна техніка для захисту об'єктів. Основні види. Розміщення і обслуговування.

24) ДСТУ Б В.1.1-36:2016. Визначення категорії приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною безпекою.

25) ДСП 173-96. Державні санітарні правила планування та забудови населених пунктів

26) Симметрон. Электронные компоненты. Каталог 2002, 2002г. – 192с.

27) Симметрон. Электронные компоненты. Каталог 2004, 2004г. – 192с.

ДОДАТОК А.
Електронні плакати