Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Програмної інженерії
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти – другий (магістерський)

Дослідження методів виявлення і розпізнавання жестів.
(тема)

Виконав: студент 2 курсу, групи ПЗСм-19-1

Черепухін Г.О.
(прізвище, ініціали)

Спеціальності 121 – Інженерія програмного забезпечення

Освітньо-професійної програми
(тип програми)

Програмне забезпечення систем

Керівник проф. Смеляков К.С.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри , проф _____ З.В. Дудар
(підпис) (прізвище, ініціали)

2020 р.

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет Комп'ютерних наук

Кафедра Програмної Інженерії

Рівень вищої освіти другий (магістерський)

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

Тип програми Освітньо-професійна програма

Освітня програма Програмне забезпечення систем

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 20 ___ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові Черепухіну Глібу Олексійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів виявлення і розпізнавання жестів
затверджена наказом по університету від 30 жовтня 2020 р. № 1490 Ст
2. Термін подання студентом роботи до екзаменаційної комісії 16 листопада 2020 р.
3. Вихідні дані до роботи метод розпізнавання жестів, програмна реалізація методу розпізнавання жестів
4. Перелік питань, що потрібно опрацювати в роботі актуальність роботи та постановка задачі, методи розпізнавання жестів руки в системах людино-машинної взаємодії, дослідження і розробка методів розпізнавання жестів руки в дальностних зображеннях людини, розробка та адаптація методів людинно-машинної взаємодії за допомогою жестів
5. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Спецчастина	проф. Смеляков К.С.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на дипломну роботу		виконано
2	Актуальність роботи та постановка задачі		виконано
3	Методи розпізнавання жестів руки в системах		виконано
4	Дослідження і розробка методів розпізнавання		виконано
	жестів руки в дальностних зображеннях		
5	Розробка та адаптація методів		виконано
	людинно-машинної взаємодії за допомогою жестів		
6	Оформлення пояснювальної записки		виконано
7	Спецчастина		виконано
8	Підготовка презентації та доповіді		виконано
9	Нормоконтроль, рецензування		виконано
10	Попередній захист		виконано
11	Занесення диплома в електронний архів		виконано
12	Допуск до захисту у зав. кафедри		виконано

Дата видачі завдання _30_____жовтня____2020 р.

Студент _____
(підпис)

Керівник роботи _____ _____проф. Смеляков К.С
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка до атестаційної роботи: 106 с., 42 рис., 3 додатки, 52 джерел.

КОМП'ЮТЕРНИЙ ЗІР, РОЗПІЗНАВАННЯ ЖЕСТІВ, АЛГОРИТМИ ВИДІЛЕННЯ ДІСКРЕТНИХ ФІГУР, СТАТИЧНІ ЖЕСТИ, ДАЛЬНОСТНІ ЗОБРАЖЕННЯ, НЕЙРОННА МЕРЕЖА, PYTHON, OPENCV, KERAS.

Метою роботи є дослідження методів розпізнання жестів та створення інтелектуального додатку, що буде розпізнавати жести та мати можливість навчання новим еталонним жестам за прикладом використовуючи методи машинного навчання.

Об'єктом дослідження є методи комп'ютерного розпізнавання жестів та образів. Основним завданням є впровадження методів машинного навчання у додаток для розпізнавання статичних жестів руки.

COMPUTER VISION, GESTURE RECOGNITION, DISCRETE FIGURES ALGORITHMS, STATIC GESTURES, DISTANCE IMAGES, NEURAL NETWORK, PYTHON, OPENCV, KERAS.

The aim of the work is to study the methods of gesture recognition and create an intelligent application that will recognize gestures and be able to learn new reference gestures by example using machine learning methods.

The object of research is the methods of computer recognition of gestures and images. The main task is to introduce machine learning methods in the application for recognizing static hand gestures.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, ... 7
скорочень і термінів 7
Вступ 9
1 Актуальність роботи та постановка задачі 13
2 Методи розпізнавання жестів руки в системах 16
людино-машинної взаємодії 16
2.1 Завдання розпізнавання жестів руки 16
2.2. Аналіз існуючих методів розпізнавання жестів руки на 24
основі аналізу зовнішніх ознак жесту 19
2.2.1 Розпізнавання позиції і орієнтації за допомогою моментів.. 24
зображення 19
2.2.2 Розпізнавання рухів на основі аналізу різниць зображень ... 20
2.2.3 Розпізнавання конфігурації на основі аналізу гістограм 25
напрямків 21
2.2.4 Розпізнавання конфігурації і позиції із застосуванням 28
кольорових рукавичок 23
2.2.5 Розпізнавання конфігурації і позиції на основі аналізу 29
контуру зображення руки 24
2.2.6 Розпізнавання позиції і конфігурації руки методом 30
випадкових лісів 25
2.2.7 Розпізнавання жестів руки із застосуванням штучних 32
нейронних мереж 27
2.2.8 Розпізнавання жестів руки із застосуванням прихованих 34
моделей Маркова 29
2.3 Метод розпізнавання жестів руки на основі аналізу тривимірної 34
моделі руки 29
2.4 Підсумок розглянутих методів 31

3 Дослідження і розробка методів розпізнавання жестів руки в……34
дальностних зображеннях людини……34
3.1 Розгляд характеристик дальностних зображень……34
3.2 Основні визначення……37
3.3 Витяг зображення руки……38
3.3.1 Обчислення позиції довільної точки долоні руки в……43
дальностном зображенні……38
3.3.2 Витяг зображення руки по позиції долоні……41
3.4 Розпізнавання позицій кінчиків пальців і точок з'єднання……47
пальців з долонею руки……42
3.5 Розпізнавання конфігурації руки……47
3.5.1 Скелетизації зображення руки за допомогою……52
планарного уявлення дальностного зображення……47
4.1 Обрані технології та бібліотеки програмної мови……56
4.1.2 OpenCV……58
4.1.3 Keras……60
4.1.3 Numpy……61
4.2 Згорткова нейронна мережа (Convolutional Neural Network)……63
4.3 Програмна реалізація додатку по розпізнаванню……70
статичних жестів……65
Висновки……68
Перелік посилань……70
Додаток А……73
Додаток Б……74

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AR — Augmented Reality (англ.) — розширена реальність. Набір технологій, вид віртуальної реальності, при якому віртуальні об'єкти відображаються в реальному оточенні, видимому користувачами

VR — Virtual Reality (англ) — віртуальна реальність. Набір технологій і апаратних засобів для відображення, перегляду та взаємодії з віртуальними об'єктами.

ПЗ — Програмне Забезпечення

NUI — Natural User Interfaces (англ.) — природний користувальницький інтерфейс. Різновид людино-комп'ютерних інтерфейсів, побудованих на фіксації і розпізнаванні будь-якої комбінації рухів людини або активності його органів.

GUI — Graphical User Interface (англ.) — графічний користувальницький інтерфейс. Різновид чоловік-компьютерного інтерфейсу, в якому елементи інтерфейсу, представлені користувачеві, виконані в вигляді графічних зображень.

CPU — Central Processing Unit (англ.) — центральне обробляє пристрій. Електронний блок або інтегральна схема (мікропроцесор), виконуюча машинні інструкції (код програм), головна частина апаратного забезпечення комп'ютера або програмованого логічного контролера. Іноді називають мікропроцесором або просто процесором. Так само використовується в якості позначення головного логічного юніта.

API — Application Programming Interface (англ.) – інтерфейс програмування додатків. Іноді також інтерфейс прикладного програмування. Набір готових класів, процедур, функцій, структур і констант, що надаються додатком (бібліотекою, сервісом, обладнанням) для використання в зовнішніх програмних продуктах.

ID — Identifier (англ.) — ідентифікатор. Унікальний номер, присвоюється об'єкту для його ідентифікації. Зазвичай вдає із себе число, але може бути різних типів. У тому числі і складені, коли ідентифікатор складається з декількох частин.

SDK — Software Development Kit (англ.) — комплект засобів розробки. Дозволяє фахівцям з програмного забезпечення створювати додатки для певного пакета програм, програмного забезпечення базових засобів розробки, апаратної платформи, комп'ютерної системи, ігрових консолей, операційних систем і інших платформ.

FPS — Frame Per Second (англ.) — Кадри в секунду

CNN — Convolutional Neural Network (англ.) — Згорткова Нейронна Мережа

ВСТУП

У сучасній науці і техніці для вирішення багатьох проблем необхідно використовувати природний метод взаємодії людина-комп'ютер, так званий природний інтерфейс. Однак досі немає єдиного визначення такого інтерфейсу. Однак існують різні методи класифікації та ранжування природних інтерфейсів. У цьому розділі наведено більш детальний опис таких методів.

Тепер ми вказуємо на позицію проблеми, яку потрібно вирішити в загальній класифікації природних інтерфейсів. Для цього скористайтеся схемою, зображеною на рисунку 1.1.

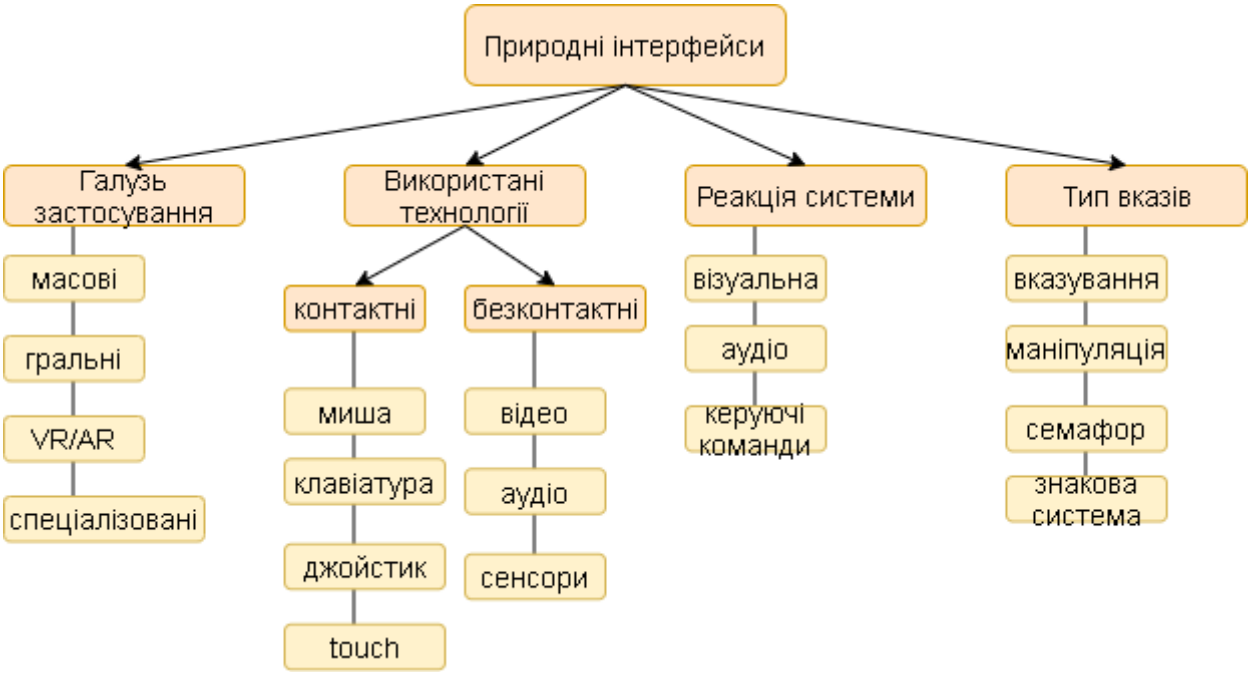


Рисунок 1.1 — Систематизація природних інтерфейсів за галузями застосування, використовуваних технологій, виду реакції системи і типу вказівок.

У цьому випадку інтерфейс ділиться на:

- Області застосування. Тут поділ йде, в основному, за вимогами, що накладається на інтерфейс. Очевидно, що ці вимоги будуть різними для, наприклад, масових та індивідуальних інтерфейсів;

- Використовуваних технологій. Тут поділ йде «фізичної частини» інтерфейсів. Чи використовуються, наприклад, «класичні» пристрої введення або передбачаються спеціальні датчики;

- Реакції системи. Тут поділ йде за видами зворотного зв'язку;

- Системи. Наприклад, генерується чи у відповідь на дії користувача візуальний образ і аудіо сигнал або ж генерується керуючий сигнал CPU.

- Типу вказівок. Тут поділ йде за типом вказівок системі. Чи використовується, наприклад, маніпулятивна система взаємодії, вказівний або знакова, і так далі.

Ключовими напрямами досліджень та розробок є автоматизація операційних та розробка інтерфейсів управління медичним обладнанням. Другою за важливістю сферою є взаємодія з роботою платформи.

Тому головною сферою застосування, що розробляється є спеціалізовані вмонтовані системи.

Так само, з огляду на цінність природних інтерфейсів для задач неавтоматизованої навігації, взаємодії з віртуальними об'єктами і психологічних завдань дослідження станів присутності, як області застосування системи слід відзначити використання в середах AR і VR.

Однією з вимог до розробки систем для медичних виробів є те, що оператори повинні дотримуватися стерильних умов. Найпростіший спосіб - використовувати безконтактну технологію.

Оскільки взаємодія користувача з комп’ютером - це трудомістка операція, слід враховувати історію та контекст взаємодії. Тому методи та методи аналізу відеопотоків кращі, ніж аналіз окремих зображень. Порівняно з прямим аналізом відеопотоку для безпосередньої взаємодії користувача, аналіз глибини сцени в будь-який момент часу забезпечує вищу надійність, оскільки він не покладається на можливі зміни умов освітлення, особливо під час тривалої роботи.

Тому використовується датчик глибини, а інформація від датчика глибини обробляється безперервним потоком, подібним до відеопотоку. Слід зазначити, що датчиком глибини сцени можуть бути різні пристрої, такі як рейтинг камер RGBD, звичайні камери, об'єднані в (стерео) масиви, далекоміри або спеціальні пристрої..

Реакція системи. Головним завданням взаємодії користувача з системою в даному випадку є управління обладнанням і роботизованими системами, що має на увазі наявність кінетичного відгуку на успішну дію користувача.

Тому основна реакція системи полягає у формуванні команд управління для остаточного апаратного чи програмного забезпечення. Жодних інших стимулів у вигляді аудіо чи відеозв’язку не потрібно. Більше того, у разі застосування в середовищах AR та VR цей додатковий стимул може бути небажаним.

Система призначена для обробки жестів користувача. Виберіть тип руху та тип інструкції відповідно до завдання, яке вирішить система. Такі жести, як "операція" та "команда", найкраще підходять для завдань управління пристроєм. Для вирішення проблеми неавтоматичної навігації потрібні такі жести, як «команда». Так само в обох завданнях повинні використовуватися тривимірні жести.

Апаратне забезпечення сучасної взаємодії людина-комп’ютер з’явилося 50 років тому і вперше з’явилося на дисплеї. До середини 1960-х років існував набір певних пристроїв введення - буквено-цифрові та функціональні дисплеї клавіатури, друкарські машинки, світлове перо, яке здійснювало введення в різних режимах, миша (первинна назва — bug (англ. жук). Трохи пізніше з'явилися джойстики (joystick), трекболи (trackball), тач-скріни(Touch-screen — сенсорні екрани). Такі інтерфейси, реалізовані за рахунок операцій з будь-якими пристроями, можна назвати девайсними інтерфейсами (Device Interfaces) на відміну від природних призначених для користувача інтерфейсів (Natural User Interfaces — NUI).

Використовується кілька визначень природних інтерфейсів. Деякі з них зосереджуються на тому, що в рамках природного інтерфейсу розроблені

користувачем операції інтуїтивно зрозумілі і засновані на природній щоденній поведінці. Інші інтерфейси, засновані на природних елементах, майже непомітні (або стають такими після того, як користувач освоїв їх). Коли ми говоримо про природні інтерфейси, ми маємо на увазі інтерфейси, які побудовані на фіксації та розпізнаванні будь-якої комбінації людського руху чи його діяльності органів (рисунок 1.2).

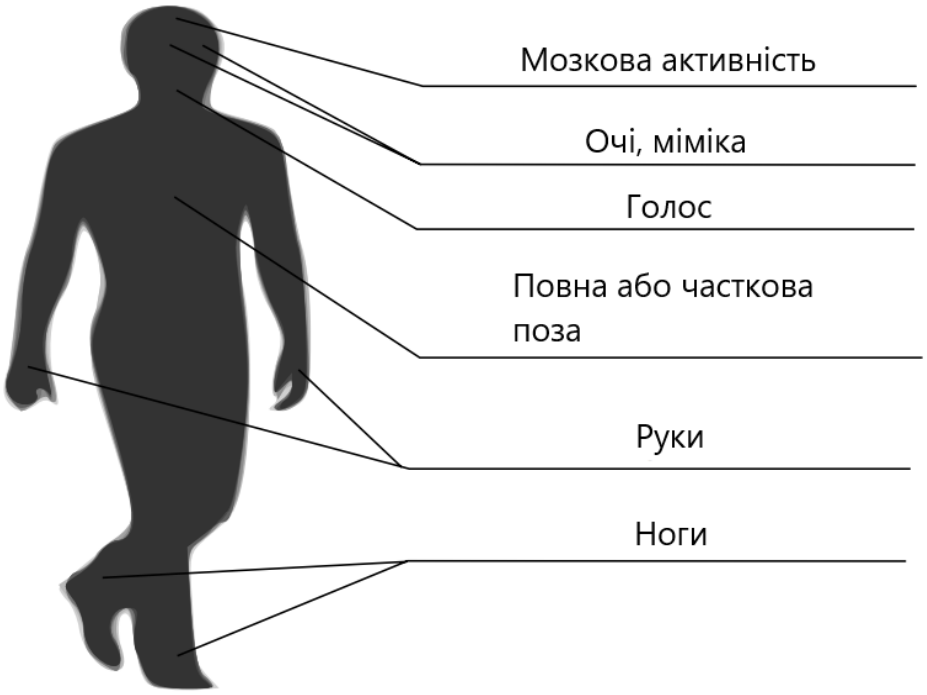


Рисунок 1.2 — Природні інтерфейси побудовані на фіксації і розпізнаванні комбінації рухів людини або активності його органів.

1 АКТУАЛЬНІСТЬ РОБОТИ ТА ПОСТАНОВКА ЗАДАЧІ

Актуальність дослідження. Створення природних, легких в управлінні людино-машинних інтерфейсів для різних додатків є актуальною науковою задачею. В даний час проводиться досить багато досліджень по створенню методів розпізнавання образів, що дозволяють безконтактно взаємодіяти з комп'ютером за допомогою жестів рук.

На основі розроблених методів створені нечисленні практичні використання, найбільш значимі з яких:

- Програмні бібліотеки Kinect SDK (Microsoft), OpenNI / NITE (PrimeSense), PMD SDK і SoftKinetic IISU для розпізнавання ключових точок тіла людини за допомогою застосування тривимірних сенсорів;

- Сенсори компаній LeapMotion, SoftKinetic для розпізнавання ключових точок руки людини;

- Комп'ютерні програми Flutter, PointGrab HGRS, XTR3D, EyeSight PC для управління комп'ютером за допомогою простих жестів рук.

Незважаючи на окремі успіхи, якість розроблених алгоритмів розпізнавання жестів рук і пальців, як динамічних, так і статичних, з використанням кольорових відеокамер і тривимірних сенсорів все ще залишається недостатнім для побудови практичних систем людино-машинного взаємодії. Головними недоліками існуючих методів є чутливість до змін освітлення, потреба навчання системи для кожного оператора, невисока якість розпізнавання жестів і невелика швидкість розпізнавання.

Таким чином, актуальною є задача створення нових моделей, методів і алгоритмів розпізнавання жестів руки, які можуть бути використані для створення систем безконтактного людино-машинного взаємодії.

Мета магістерської роботи. Метою роботи є дослідження і розробка методів розпізнавання статичних і динамічних жестів руки, які можуть бути використані

для безконтактного взаємодії людини з комп'ютером. Засобом досягнення мети служить вирішення таких основних завдань:

- Аналіз існуючих методів відстеження та розпізнавання жестів руки, а також безконтактного людино-машинного взаємодії;

- Дослідження і розробка методів опису, вилучення і розпізнавання конфігурацій руки;

На захист виносяться такі нові наукові результати:

- Метод вилучення зображення руки з дальностного зображення людини на основі аналізу пов'язаності точок зображення в тривимірному просторі, зі складністю O(n+wm), де n — кількість точок вихідного зображення. m – максимальна кількість вершин багатокутника що апроксимує контур фігури руки;

- Метод розпізнавання позиції кінчиків пальців і точок з'єднання пальців з долонею руки на основі аналізу контуру зображення руки, зі складністю O(n), де n — кількість точок зображення руки;

- Метод скелетизації дальностного зображення, заснований на безперервному скелетному поданні бінарного зображення, пошуку граничних точок фігури і створенні діаграми Вороного для цих точок;

- Демонстрація комп’ютерної програми розпізнавання статичних жестів руки.

Наукова новизна полягає в побудові нових методів і алгоритмів, які забезпечують вирішення завдань розпізнавання статичних жестів руки по дальностним зображенням людини, створенні на їх основі природних людино-машинних інтерфейсів.

- Запропоновано новий метод скелетизації дальностного зображення, забезпечує, у порівнянні з алгоритмами обчислення двовимірного скелета, побудова більш інформативного графа і меншу обчислювальну складність щодо алгоритмів обчислення тривимірного криволінійного скелета, оцінювану як (nlogn), де n — число вершин багатокутника що апроксимує контур фігури;

- Запропоновано нову міру оцінки відстані між двома тривимірними скелетами, для якої виконуються всі аксіоми метрики, за винятком нерівності трикутника;

- Розроблено та досліджено метод розпізнавання конфігурацій руки на основі запропонованої заходи з навчанням по одному прецеденту, який відрізняється від відомих методів високою надійністю і низькою обчислювальною складністю — O(n+wm), де n — кількість точок у кадру відеоряду, w — число еталонних жестів, m – максимальне кількість вершин багатокутника що апроксимує контур фігури руки;

- Досліджено та розроблено метод пошуку ключових точок руки по дальностним зображень на основі аналізу контуру фігури руки, дозволяє в реальному часі розпізнавати позиції кінчиків пальців і точок з'єднання пальців з долонею руки, незалежно від кількості і напрямки пальців;

- Розроблено експериментальні програми для демонстрації розробленого методу.

Методи дослідження. В роботі використовуються методи теорії графів, обчислювальної геометрії, математичного аналізу, теорії розпізнавання образів. Робота несе теоретико-експериментальний характер. Метод розпізнавання статичних жестів реалізований у вигляді комп'ютерної програми для декількох платформ.

Обґрунтованість і достовірність результатів. Отримані в роботі результати обґрунтовані використанням строгих і апробованих математичних методів алгебри матриць, комп'ютерної графіки. Достовірність підтверджується обчислювальним експериментом і імітаційним моделюванням, проведеним з використанням реальних вихідних даних, а також шляхом зіставлення результатів, отриманих в процесі роботи, з доступними у відкритій пресі результатами.

Практична значимість. На основі розробленого методу створено програмний додаток, що дозволяє взаємодіяти з комп'ютером за допомогою декількох пальців руки. Розроблені алгоритми порівняння об'єктів можуть бути використані для розпізнавання різних цільових об'єктів по дальностним зображень у реальному часі.

2 МЕТОДИ РОЗПІЗНАВАННЯ ЖЕСТІВ РУКИ В СИСТЕМАХ ЛЮДИНО-МАШИННОЇ ВЗАЄМОДІЇ

2.1 Завдання розпізнавання жестів руки

Людино-машинна взаємодія (Human-computer interaction — HCI) — це міждисциплінарний науковий напрямок, що вивчає взаємодію між людьми і машинами. Предметом HCI є вивчення, планування і розробка методів взаємодії людини з машиною, де в ролі машини може виступати персональний комп'ютер, комп'ютерна система великих масштабів, система управління процесами і т.д. [6]. під взаємодією розуміється будь-яка комунікація між людиною і машиною. Одним із методів HCI, який отримав широке поширення в останні роки, є взаємодія, засноване на жестах людини [7, 8].

Жести — це різні рухи тіла, які є невербальною способом передачі інформації. Рух пальців, рук, голови, плечей, міміка особи: всі перераховані є жестами. За допомогою жестів людина може передати самостійну інформаційну одиницю, доповнити словесний ряд, передати почуття і пр. Звичайно жести поділяються на статичні (сприймаються одномоментно) і динамічні (що сприймаються в деякий період часу) жести, які мають певну інтерпретацію в ручних абетках і безконтактних людино-машинних інтерфейсів. У HCI жести використовуються для передачі інформації в комп'ютер, який в подальшому може використовуватися для ідентифікації людини, управління комп'ютером, літальним апаратом, ігровим аватаром і т.д. Розглянемо основні поняття, які будуть зустрічатися в цьому розділі.

В задачах розпізнавання образів і обробки цифрових зображень

часто застосовується функція інтенсивності виду f(x, y), Що залежить від цілочисельних координат x і y належать так званої растрової решітці W^2[11]. Пара (x, y) задає позицію точки зображення в системі координат OXY, Початком якої є лівий верхній кут решітки. Функція f(x, y) зазвичай задає яскравість, освітленість, насиченість, ступінь поглинання і інші спектральні характеристики в точці (x, y).

Нехай задані просторова точка M з координатами (x, y, z) і її ортогональна проекція m з координатами (x, y) на площину спостереження Z. Під дальностним зображенням будемо розуміти цифрове зображення d(x, y), В кожній точці (x, y) якого d(x, y) приймає невід'ємні цілі значення, рівні величині відстані від точки M доточки m.

Дальностне зображення завжди можна перетворити в безліч (Хмара) точок {(x, y, d(x, y))} в тривимірному просторі, що є моделлю об'єкта, що спостерігається.

Під конфігурацією руки будемо розуміти встановлене положення пальців і кисті руки.

У літературі завдання розпізнавання жестів руки трактують по-різному,

як, наприклад, обчислення позиції долоні, плечей і кінчиків пальців руки, ідентифікацію конфігурації і траєкторії руху руки і т.д. У цій роботі під завданням розпізнавання жестів руки будемо розуміти такі три підзадачі:

- Розпізнавання позицій долоні і кінчиків пальців руки в тривимірному просторі;

- Розпізнавання статичного жесту руки по еталонним конфігурацій;

- Розпізнавання динамічних жестів по траєкторії руху долоні руки.

Система розпізнавання жестів руки — це сукупність комп'ютерних технологій і математичних алгоритмів, яка дозволяє вирішувати задачу розпізнавання певної групи жестів руки.

Розпізнавання жестів можна застосовувати в таких областях діяльності людини, як наприклад:

- Управління комп'ютером і побутовими приладами де у комп'ютерному додатку кожної конфігурації руки зіставляється певна команда. Людина показує жест, система розпізнає конфігурацію руки і відправляє відповідну команду в комп'ютер. Позиція долоні зіставляється з позицією курсору миші на екрані

Рухи руки призводять до рухів курсору. команди натискання кнопок мишки зіставляються з різними конфігураціями руки. Розпізнані позиції кінчиків пальців

руки можуть бути використані для управління комп'ютером або побутових приладів одночасно кількома пальцями руки;

- Створення природних людино-машинних інтерфейсів для глухонімих;

- Систему розпізнавання жестів можна застосувати для введення тексту в комп'ютер за допомогою жестів руки, що для глухонімих людей простіше і природніше, ніж введення тексту за допомогою клавіатури комп'ютера;

- Систему розпізнавання жестів можна застосувати для створення додатків онлайн зв'язку, коли пропускна здатність мережі мала і відеозв'язок неможлива. Маючи розпізнані жести руки на одному кінці мережі, можна показати анімацію цих же жестів, на іншому кінці передаючи по мережі тільки характеристики показаних жестів;

- Маніпуляція тривимірними моделями об'єктів. На сьогоднішній день, для роботи з тривимірними моделями зазвичай використовують комп'ютерну мишу, яка не дуже зручна для цієї завдання. Маючи тривимірні координати руки і кінчиків пальців руки, можна створити систему HCI, яка дозволить керувати моделями по усіх напрямках тривимірного простору;

- Додатки віртуальної реальності. Доповнивши систему розпізнавання жестів пристроями, наприклад стереоскопічними окулярами, можна створити додатки віртуальної реальності, де користувач зможе «доторкатися» до віртуальних об'єктів.

Наведений список можна доповнити системами, супутніми реабілітації пацієнтів [12], ігровими додатками [13] і т.д. частина наведених додатків вже використовується в різних пристроях і програмах [14], а частина з них ще знаходиться на етапі досліджень.

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.

Далі розглянемо дослідження, що проводяться в області розпізнавання жестів руки людини.

2.2. Аналіз існуючих методів розпізнавання жестів руки на основі аналізу зовнішніх ознак жесту

Особливістю методів, заснованих на аналізі зовнішніх ознак жесту, є аналіз тільки зовнішнього вигляду (форми, позиції і т.д.) цільового об'єкта. Для розпізнавання не зберігається ніякої інформації про фізичних властивості даного об'єкту.

Розглянемо відомі роботи і методи, присвячені розпізнаванню жестів руки людини на основі аналізу зовнішніх ознак жесту.

2.2.1 Розпізнавання позиції і орієнтації за допомогою моментів зображення

Позначимо через, $M_{0,0}$ $M_{0,1}$, $M_{1,0}$ і $M_{2,0}$ $M_{1,1}$ $M_{0,2}$ моменти нульового та першого порядку зображення з функцією інтенсивності f(x, y) – формула 2.1:

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

$$2.\ M_{0,1} = \sum_x \sum_y y \times f(x,y), M_{1,0} = \sum_x \sum_y x \times f(x,y) \qquad (2.1)$$

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

Наприклад, в роботі [15] розглядається додаток, що дозволяє управляти іграшковим роботом за допомогою рухів руки, де орієнтація руки визначає напрямки руху робота (рисунок. 2.1).

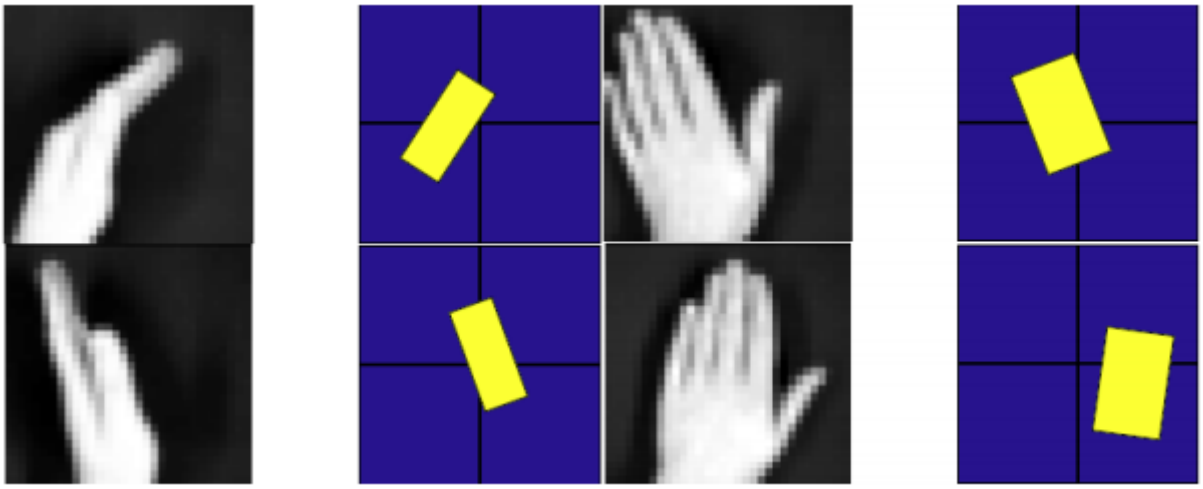


Рисунок 2.1 — Управління простими жестами руки Прямокутник показує розпізнане напрямок, позицію і розмір зображення руки;

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

Функція arctan задовольняє умові arctan(i, j)$\in(-\pi,\pi]$. Розмір зображення руки оцінюється значеннями l_1 та l_2

Де $l_1 = \sqrt{\frac{(a+c)+\sqrt{b^2+(a-c)^2}}{2}}, l_2 = \sqrt{\frac{(a+c)-\sqrt{b^2+(a-c)^2}}{2}}$. На рисунку 2.1а для кожного зображення приведені прямокутники з центром (x_c y_c)та розміром l_1 x l_2.

2.2.2 Розпізнавання рухів на основі аналізу різниць зображень

На практиці відповідати умовам однорідного фону вдається не завжди. Для таких випадків, зазвичай використовується метод, заснований на аналізі центру маси різниць зображень (motion energy image — MEI) руки в кадрах відеоряду (рисунок. 2.2) [14].

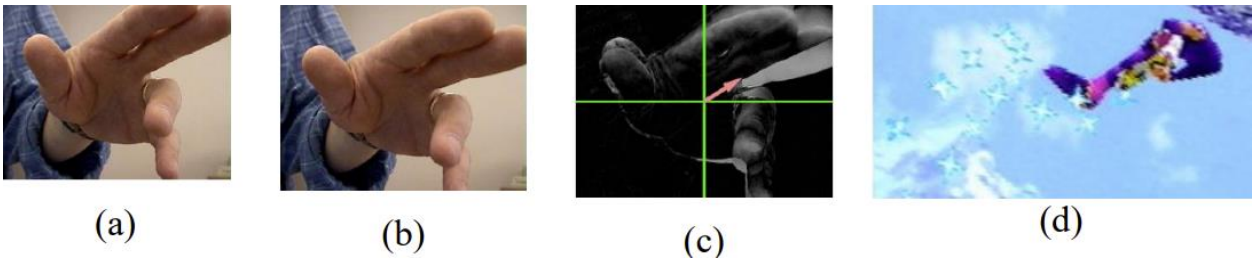


Рисунок 2.2 — Управління аватаром в комп'ютерній грі [15]: (a) і (b) Два кадри з відеоряду; (c) Різниця зображень (a-b); (d) Центр маси (стріла в (c)) управляє напрямком польоту аватара в грі

Використання різниць кадрів відеоряду (MEI) дозволяє в реальному часу аналізувати руху об'єкта в відеоряді при стабільному, але необов'язково однорідному тлі зображення. На практиці дана технологія і її вдосконалені види (наприклад, motion history image — MHI) знайшли застосування в таких додатках як інтерактивний віртуальний тренер з аеробіки [16] і інтерактивна кімната для розповіді історій [17].

2.2.3 Розпізнавання конфігурації на основі аналізу гістограм напрямків

У більшості додатків комп'ютерного зору, крім позиції і орієнтації руки людини, потрібна додаткова інформація про її конфігурації. Дане завдання в роботах [18,19] вирішується аналізом, так званих гістограм напрямків (orientation histograms) і карт напрямків (orientation maps) зображення, які менш чутливі до змін освітлення спостерігається середовища.

Нехай f(x, y) є інтенсивність бінарного або напівтонового зображення руки. локальне напрямок (x, y) в точці (x, y) визначається наступним чином (2.2):

$$\theta(x,y) = \arctan\big(f(x,y)\big) - f(x-1,y), f(x,y) - f(x,y-1) \qquad (2.2)$$

Відстань між двома зображеннями руки визначається як євклидова відстань між векторами цих зображень. На рисунку 2.3 наведені зображення жестів руки і відповідних гістограм напрямків при N = 36.

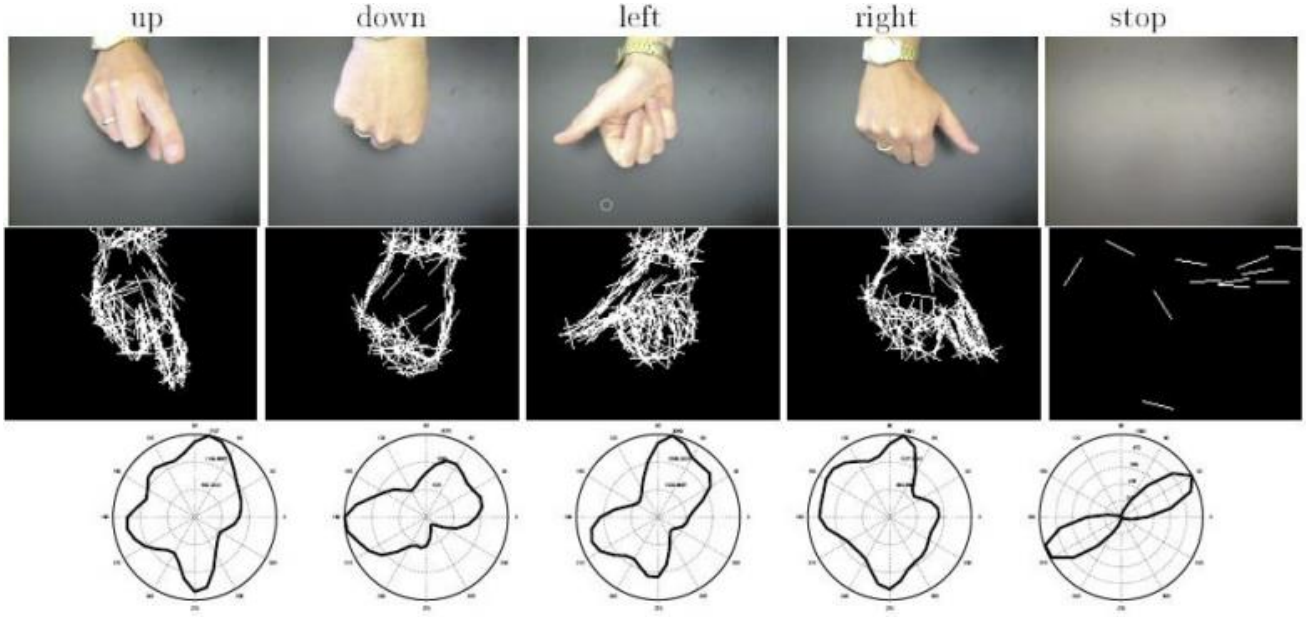


Рисунок 2.3 — Жести рук, карти напрямків і гістограми напрямків [18].

Наведені жести і технологія розпізнавання конфігурацій руки на основі гістограм напрямків в роботі [18] були використані в додатку управління анімаційним краном. У роботі відзначається, що додаток працює в реальному часі, невідчутно до незначних змін розміру руки, але відчутно до змін напрямку руки. Додаток вимагає навчання для кожного оператора.

Дана технологія дозволяє в режимі реального часу розпізнавати конфігурацію руки, якщо виконуються наступні умови:

- Фон зображення однорідний;

- Рука є домінуючим об'єктом в зображенні;

- Жести обрані таким чином, що їх гістограми напрямків мають значущі відмінності один від одного.

2.2.4 Розпізнавання конфігурації і позиції із застосуванням кольорових рукавичок

Для розпізнавання жестів руки часто застосовують кольорові рукавички [20-22]. Запропонований в роботі [20] метод дозволяє за допомогою однієї відеокамери в реальному часі розпізнавати конфігурацію руки і відстежувати рухи долоні в тривимірному просторі (рисунок 2.4).

Використовувана рукавичка сконструйована з двадцяти сегментів десяти 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.

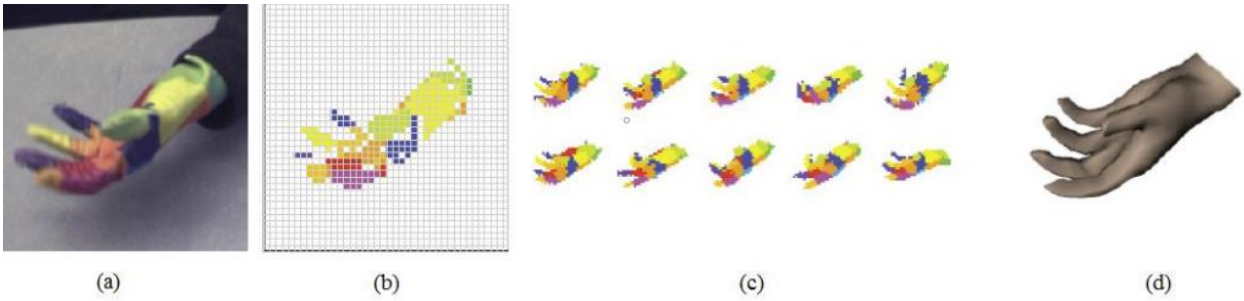


Рисунок 2.4 — Застосування кольорових рукавичок в задачі розпізнавання жестів руки [20]: (a) Вхідний зображення руки; (b) Растрове зображення руки з обхватом 40x40; (c) Схожі зображення руки в базі даних; (d) Знайдена конфігурація.

Розпізнавання конфігурації руки здійснюється згладжуванням початкового зображення алгоритмом двостороннього розмиття (англ. Bilateral filter), видаленням фонових точок, де фонові точки класифікуються на основі моделі Гаусових сумішей (англ. Gaussian mixture models) і перетворенням отриманого зображення в растрове зображення меншого розміру (рисунок 2.4.b). Отримане растрове зображення порівнюється з зберігаються в базі даних 18000 еталонними

зображеннями, відповідними різним конфігурацій руки де відстань $d(r^i, r^j)$ між растровими зображеннями r^i та r^j оцінюється за допомогою формули 2.3

$$\tilde{d}(r^j, r^i), \quad \tilde{d}(r^i, r^j)\sqrt{\frac{1}{|C_i|}\sum \min(u-x)^2 + (v-y)^2} \qquad (2.3)$$

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

2.2.5 Розпізнавання конфігурації і позиції на основі аналізу контуру зображення руки

Як дескриптора конфігурації руки часто вибирають контур зображення руки. Наприклад, в роботі [23] пропонується система розпізнавання жестів ручної азбуки глухонімих ASL на основі аналізу контуру зображення руки. Для полегшення завдання два динамічних жесту розглядаються, і частина жестів замінюється новими. У кольоровому зображенні руки, видаляються всі точки, які не відповідають кольору шкіри людини.

Отримане зображення перетворюється в бінарне зображення, після чого згладжується фільтром Гауса. Рука виділяється як пов'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 площина контуру руки де $(x_c, y_c) = \frac{1}{n}(\sum_{i=1}^{n} x_i, \sum_{i=1}^{n} y_i)$ а (x_i, y_i) являють собою координати точок контуру. Схожість жестів оцінюється відстанню Евкліда між векторами дескрипторів жестів. В роботі наводиться оцінка якості розпізнавання — 90%. Якість розпізнавання, при показі жестів різними людьми, не розглядається.

В роботі [24] розглядається задача розпізнавання позиції кінчиків пальців руки на основі аналізу контуру руки в кольоровому зображенні. Першими

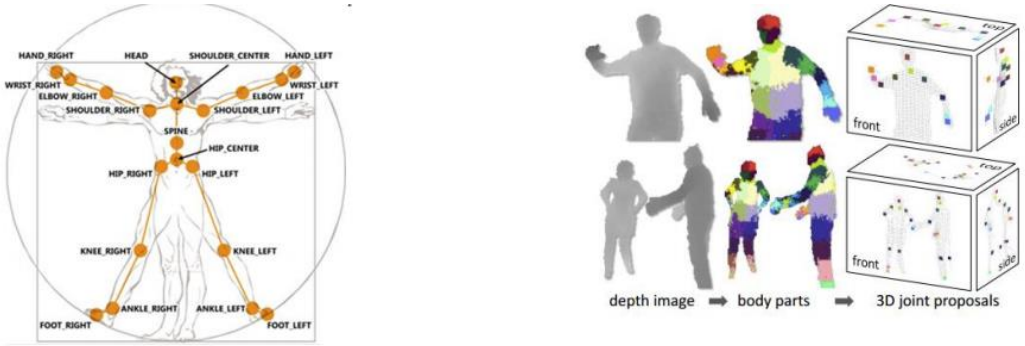
кроками алгоритму, як і в випадку [23], є видалення фонових точок, згладжування і виділення зображення руки. Маючи контур руки, пальці руки виділяються за допомогою аналізу локальних вигинів контуру. Даний алгоритм використовувався для управління веб-камерою персонального комп'ютера.

2.2.6 Розпізнавання позиції і конфігурації руки методом випадкових лісів

Алгоритм випадкових лісів (англ. Randomized decision forests) є алгоритмом машинного навчання, що використовуються в задачах класифікації,регресії і кластеризації. У завданнях розпізнавання жестів даний алгоритм був використаний для розпізнавання позицій ключових характеристик тіла людини і класифікації конфігурації руки [26]. Саме на цій технології заснована система розпізнавання жестів людини, випущена компанією Microsoft в 2010 році. Запропонований метод [25] дозволяє розпізнавати позиції 20 частин тіла (рис. 1.5.а) в тому числі позиції долонь обох рук отримуючи на вході одне дальностне зображення людини. Позиціонування цікавлять точок тіла здійснюється з 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 за допомогою застосування алгоритму випадкових лісів;
- Знаходження позицій цікавлять частин тіла на основі отриманих кластерів.

Візуальну репрезентацію наведено на рисунку 2.5.



(a) (b)

Рисунок 2.5 — (a) Розпізнавані точки тіла людини; (b) Кроки розпізнавання.

Згідно з наведеними даними в [25] даний алгоритм дозволяє розпізнавати жести людини зі швидкістю 200 кадрів в секунду при виконанні на платформі Xbox 360. Для навчання трьох дерев з глибиною рівній двадцяти на безлічі з мільйона зображень і на суперкомп'ютері з процесором з 1000 ядер потрібно один день.

В роботі [26] випадкові лісу використовуються для розпізнавання конфігурації руки для безлічі статичних жестів з ручної азбуки ASL. На рисунку 2.6 показана архітектура розробленої системи.

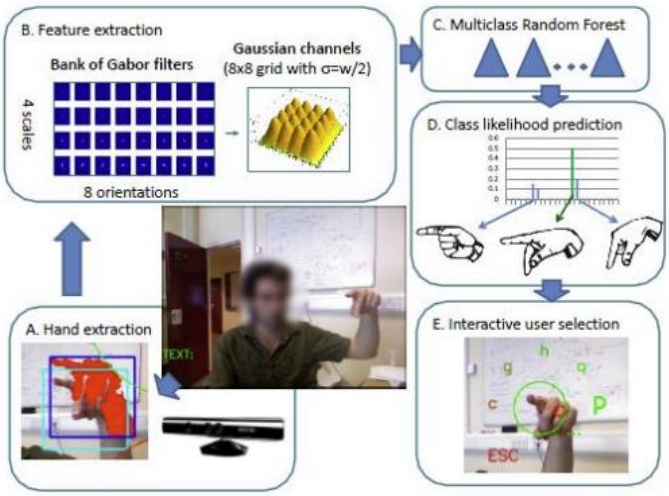


Рисунок 2.6 — Архітектура системи розпізнавання конфігурації руки для множини статичних жестів

Використовуючи в якості пристрою введення сенсор Kinect [27] система на вході отримує дальностне і кольорове зображення людини. За допомогою програмних платформ OpenNI і Nite [12] шукається рука людини, після чого рука сегментується як зв'язаний об'єкт в дальностном зображенні (рис. 2.6.А).

Після нормалізації зображень руки до них застосовується фільтр Габора. Отримані зображення перетворюються в матриці з розмірами 8x8, які виступають як дескриптор показаного жесту. Система навчається із застосуванням алгоритму випадкових лісів. розроблена система дозволяє в реальному часі розпізнавати 24 жесту ручної азбуки ASL. Середня якість розпізнавання запропонованим алгоритмом при тестуванні системи на жестах п'яти різних людей становить 75%. При використанні тільки кольорового зображення точність розпізнавання становить 73% і при використанні тільки дальностного зображення — 69%.

2.2.7 Розпізнавання жестів руки із застосуванням штучних нейронних мереж

Штучні нейронні мережі (ШНМ) представляють собою математичну модель, побудовану на принципах організації і функціонування біологічних нейронних мереж. У завданнях розпізнавання жестів, ІНС зазвичай використовують як інструмент машинного навчання, де на вхід ІНС передаються характеристики жесту і на виході виходить розпізнаний жест. ІНС відрізняються між собою структурою і методом навчання.

Найбільш поширеними видами ІНС, використовуваними в задачах розпізнавання, є мережі прямого поширення (англ. Feed forward neural networks), рекурентні нейронні мережі, самоорганізовані карти Кохонена та згорткові нейронні мережі т.д. На рисунку 2.7 зображена рекурентна нейронна мережа, використовувана для розпізнавання статичних жестів руки азбуки глухонімих Японії. В якості вхідного пристрою в роботі 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 мережу 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.

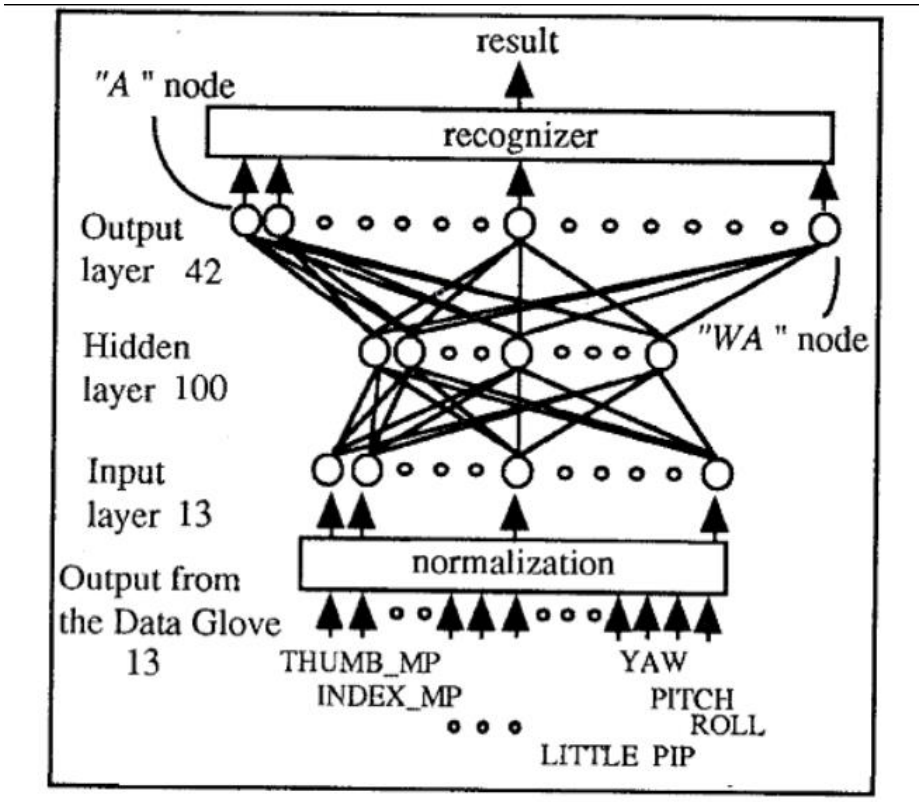


Рисунок 2.7 — Рекурентна нейронна мережа, яка використовується для розпізнавання статичних жестів

Розпізнавання конфігурацій руки здійснюється за допомогою алгоритму «нейронний газ». Для навчання використовується конкурентну навчання Хебба. Стверджується, що якість розпізнавання — 90.45% і швидкість — 1.5 секунди. В якості вхідного пристрою використовується відео камера. Для полегшення завдання розпізнавання, в запропонованій системі застосовується маркована різними кольорами рукавичка. Стверджується, що якість розпізнавання — 95.11% при використанні повністю рекурентною нейронної мережі, і 89.67%, при використанні ІНС Елмана.

2.2.8 Розпізнавання жестів руки із застосуванням прихованих моделей Маркова

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

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

ПММ використовується для розпізнавання десяти динамічних жестів руки. Якість розпізнавання складає 98.94% для ізольованих жестів і 95.7% — для безперервних.

2.3 Метод розпізнавання жестів руки на основі аналізу тривимірної моделі руки

Дана технологія використовується в комп'ютерному зорі для розпізнавання детальної тривимірної конфігурації руки при наявності на вході одного або декількох зображень жесту. Під детальної конфігурацією розуміється позиція і орієнтація долоні і ключових точок пальців руки в тривимірному просторі. На рисунку 2.8.а зображена анатомія руки людини, де DOF означає мірою важливо

свободи руху (англ. degrees of freedom). При побудові тривимірної моделі руки, зазвичай створюються точки артикуляції, які моделюють суглоби руки людини. Залежно від конкретного завдання точки артикуляції і їх ступеня свободи руху можуть збігатися або бути підмножиною суглобів руки і їх ступенями свободи руху.

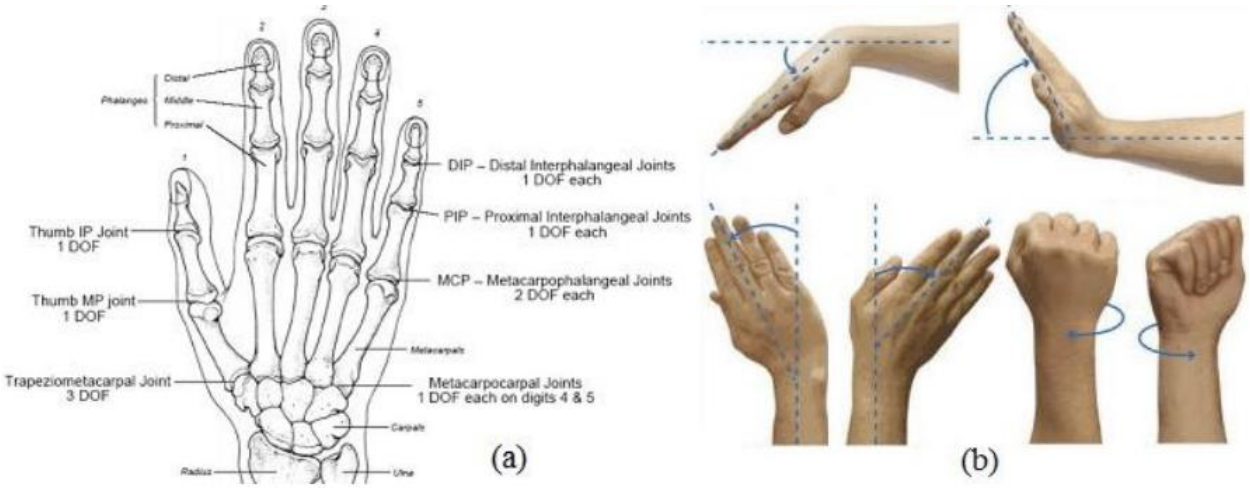


Рисунок 2.8 — Структура руки людини: (a) 17 точок артикуляції руки з 23 ступенями свободи руху; (b) 6 ступенів свободи руху кисті

Особливістю методів розпізнавання жестів на основі моделі є порівняння проекцій тривимірної моделі руки з вхідними зображеннями. За вхідного двовимірного зображення руки, на основі різних гіпотез модель руки наводиться в певну конфігурацію. Тривимірна модель руки проектується на площину, тим самим, виходить двовимірне зображення, яке порівнюється з вхідним зображенням. Якщо відстань між зображеннями менше заданого порогу, то жест розпізнано, в іншому випадку конфігурація моделі руки змінюється і заново порівнюється з вхідним зображенням.

2.4 Підсумок розглянутих методів

У попередніх розділах були описані найбільш значущі роботи і поширені методи, використовувані в задачах розпізнавання жестів руки людини. Звичайно, застосовуються і інші методи, наприклад методи, засновані на використанні кінцевих автоматів, фільтра Кальмара, класифікації Хаара, Байєсової мережі і т.д. В роботі [15] розпізнавання динамічних жестів здійснюється за допомогою використання нечітких кінцевих автоматів. Розроблена система дозволяє розпізнавати 15 динамічних жестів руки зі швидкістю 15 кадрів в секунду. В роботі [14] розпізнавання динамічних жестів здійснюється в реальному часі з допомогою метричного класифікатора із застосуванням RGB камер і тривимірних сенсорів. Запропонована система заснована на застосуванні динамічної Байєсової мережі. Розроблена система дозволяє розпізнати 10 ізольованих жестів з якістю розпізнавання 99.59%. при розпізнаванні безперервних жестів якість розпізнавання становить 80.77%. На рисунку 2.9 наведені характеристики деяких методів вирішення завдання розпізнавання жестів руки.

Метод	Пристрій вводу	Жести що розпізнаються	Якість	Швидкість	Обмеження	Работа
Моменти зображення	Відеокамера	Позиція і орієнтація руки, розмір долоні	-	Реальний час	однорідний фон	[15]
Разності зображення	Відеокамера	Напрямок руху руки	-	Реальний час	стабільний фон	[15]
Гістограми напрямків	Відеокамера	конфігурація руки	-	Реальний час	навчання для кожного оператора, однорідний фон	[18] [19]
Відстань зображень	Відеокамера	Позиція і розмір руки, конфігурація руки	-	Реальний час	-	[20]
Відстань контурів руки	Відеокамера	конфігурація руки	90%	0.5 секунд	навчання для кожного оператора	[23]
Випадкові ліси	3D сенсор	20 частин тіла людини	-	Реальний час	-	[25]
Нейронний газ	3D сенсор + Відеокамера	конфігурація руки	90.45%	1.5 секунд	чутливість до змін освітлення	[30]
Згорткова нейронна мережа	Відеокамера	конфігурація руки	95%	30 кадрів/секунда	-	[30]
тривимірна модель руки	Відеокамера	конфігурація руки (3D обертання кожного жесту)	-	Реальний час	навчання для кожного оператора	[35]
Байєсова мережа	Відеокамера	динамічні жести	99.59%	Реальний час	чутливість до змін освітлення	[39]

Рисунок 2.9 — Порівняння різноманітних методів розпізнання жестів

Таким чином, по використовуваних пристроїв для введення інформації про жестах руки, існуючі методи можна розділити на 3 основні групи:

- Методи, засновані на використанні відеокамер;

- Методи, засновані на використанні спеціальних рукавичок, оснащеними сенсорами;

- Методи, засновані на використанні тривимірних сенсорів.

Перша група методів дозволяє розпізнавати статичні і динамічні жести руки, і в деяких випадках з точністю розпізнавання більше 90%. Недоліком цих методів є в основному чутливість до змін освітлення, яку в різних роботах намагаються усунути за допомогою використання кольорових рукавичок або створенням однорідного фону, що робить людино-машинне взаємодію незручною і неприродною.

Методи другої групи дозволяють з високою якістю розпізнавати позиції ключових точок руки людини, але натомість вимагають одягання спеціальних дорогих рукавичок. Є мало поширеними.. Проведені дослідження по розпізнаванню конфігурацій руки дозволили отримати якість розпізнавання не більше 75%. Огляд літератури дозволяє зробити наступні основні висновки:

- Завдання розпізнавання жестів руки з використанням широко поширеною апаратури залишається на сьогоднішній день невирішеною;

- Існуючі пристрої і методи дозволяють вирішувати лише невелику частина актуальних завдань, наприклад розпізнавання позицій долоні і кінчиків пальців руки;

- Майже всі методи, які використовують відео камеру, чутливі до освітленості, вимагають надягання рукавичок, або ж ставлять обмеження на фон зображень;

- Тривимірні сенсори дозволяють отримати порівняно стабільні зображення, але на сьогоднішній день немає методів, здатних в реальному часі розпізнавати позиції кінчиків пальців, конфігурації і динамічні жести руки на основі дальностних зображень людини.

3 ДОСЛІДЖЕННЯ І РОЗРОБКА МЕТОДІВ РОЗПІЗНАВАННЯ ЖЕСТІВ РУКИ В ДАЛЬНОСТНИХ ЗОБРАЖЕННЯХ ЛЮДИНИ

3.1 Розгляд характеристик дальностних зображень

На рисунку 3.1 представлений приклад дальностного зображення людини в вигляді бінарних зображень, отриманих за допомогою спроектованого на площину хмари точок, під різними кутами спостереження. Зображення отримані за допомогою сенсора Kinect[27] і програмної бібліотеки OpenTK. Можна помітити, що відбулося відділення точок людини від заднього фону (рис. 3.1.a). На рисунку 3.1. (d-e) чітко видно точки, що створюють поверхню руки. Помітні нерівності поверхні і контурів руки, які властиві всім об'єктам в дальностном зображенні.

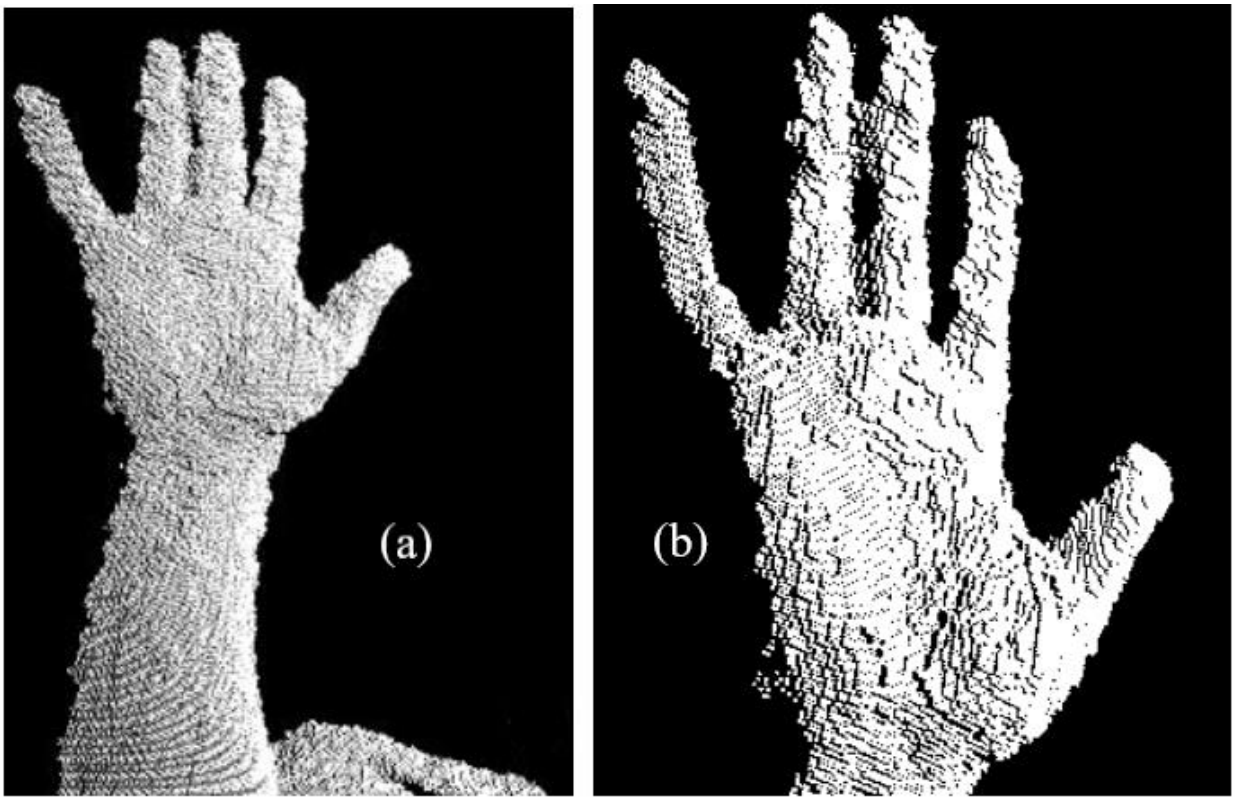


Рисунок 3.1 — (a, b) Укрупнені зображення руки у дальностному зображенні.

Для тривимірних сенсорів (далі сенсор) визначено максимальна целочисельна відстань M від площини спостереження до сцени. У точках, відстань яких більше M, визначаємо d (x, y) = 0, таким чином d (x, y) {0,1, ..., M}. У разі сенсорів Kinect і Asus Xtion Pro Live M =10000, що відповідає 10000 мм.

Нехай задано деякий дальностне зображення d(x, y), яке в одній точці приймає значення відмінне від нуля. тоді йому можна поставити у відповідність півтонування з функцією інтенсивності p(x, y), де

$$p(x,y) = \left[(B-1) * \frac{d(x,y)}{M}\right] \qquad (3.1)$$

На рисунку 3.2.a приведено півтонове зображення, отримане на основі дальностного зображення людини 3.1.a за допомогою перетворення (3.1).

Рисунок 3.2 — (a) Півтонове зображення людини, отримане за допомогою перетворення (3.1); (b) Півтонове зображення людини, отримане за допомогою перетворення (3.1)

Як видно на наведеному малюнку, в напівтоновому зображенні відсутня інформація про форму особи, одягу і т.д., яка була спочатку в дальностном зображенні (рис. 2.1 b, c). Для отримання більш інформативного півтонування, пропонується метод перетворення з використанням гістограм зображень.

У цьому випадку p(x,y) = [$h''_{d(x,y)}$] приймає цілі значення від 0 до B =1 і показує інтенсивність півтонування в кожній точці (x, y). На рисунку 3.2.b представлено півтонування людини, отримане на основі дальностного зображення 3.1.a за допомогою перетворення (3.2). В напівтоновому зображенні чітко видно обличчя і вигини одягу людини, які розмиті в зображенні 3.2.a. У зображеннях 3.1.a і 3.1.b можна помітити нерівні контури об'єктів і «тінь» на тлі зображення. При наявності перед тривимірним сенсором об'єктів, які заступають один одного, виникає смуга, яка відбиває світло проектора сенсора. Причиною є схема, яка містить один проектор, який випромінює інфрачервоне світло, і сенсор, який отримує відбитий від сцени світло (рис. 3.3).

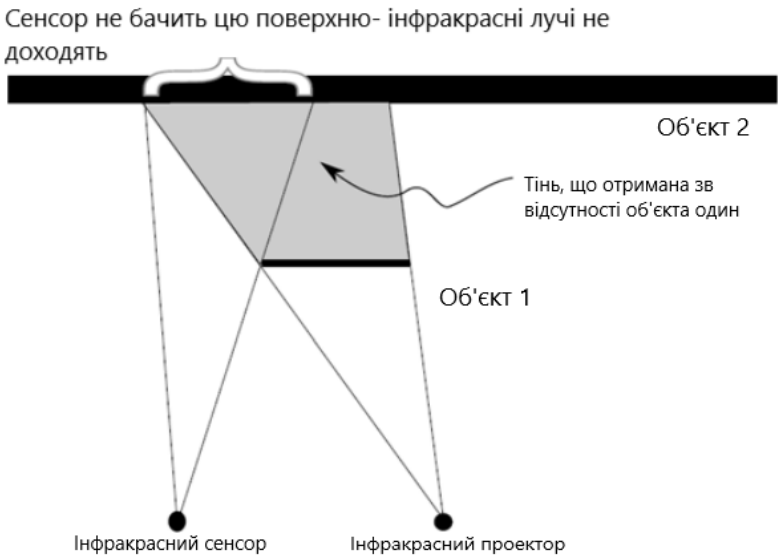


Рисунок 3.3 – тінь що отримується у дальностному зображенні

У точках дальностного зображення, відповідних даній смузі визначаємо d(x, y) = 0.

Таким чином, розглядаються тривимірним сенсором притаманні такі особливості:

- Наявність відстані M від площини спостереження до сцени таке, що в точках, для яких відстань більше M, d(x, y) = 0;

- Наявність, в разі закритої об'єктів, смуги, в точках якої d(x, y) = 0 ;

- Наявність у об'єктів дальностного зображення нерівних контурів і поверхонь;

- Можливість перетворення дальностного зображення в півтонування.

3.2 Основні визначення

Введемо основні визначення, які будуть використані в запропонованих методах вилучення зображення и розпізнавання жестів руки. Нехай є бінарне зображення з функцією інтенсивності g(x, y), де g(x, y) приймає значення 0 (для чорних крапок) и 1 (для білих точок), та дальностне зображення d(x, y).

Назвемо точки 4-Сусідами (далі просто «Сусідами») зображення будь-якого типу, якщо у них відрізняється тільки одна з координат и притому тільки на 1. Загально прийнято, що точка (X, y)в бінарному зображенні називається граничної, якщо у неї є сусід Іншого кольору.

Крапку (X, y) в бінарному зображенні назвемо ізольованою, якщо існують більше двох сусідів іншого кольору. У бінарному зображенні точка (x′, y′) називається пов'язаним сусідом точки (x, y), якщо вони сусідні і одного кольору

Назвемо точки (x_0,y_0) і(x_n,y_n) пов'язаними в множині точок A бінарного зображення, якщо існує послідовність точок (x_0, y_0), (x_1, y_1),..., (x_n, y_n) що належать A, таких, що (x_{i-1}, y_{i-1}) та (x_i, y_i) є пов'язаними сусідами для всіх значень i = {1,2, ..., n}.

Множина точок називається зв'язковим безліччю, якщо будь-які пари його точок((x, y), (x′, y′)) є пов'язаними в цій множині.

Дискретної фігурою в бінарному зображенні назвемо чітка множина, що не є підмножиною іншої зв'язкової множини.

Контуром дискретної фігури бінарного зображення назвемо множину граничних точок даної фігури.

3.3 Витяг зображення руки

Пропонований в даній роботі метод вилучення зображення руки з дальностного зображення розглядається як сукупність двох підзадач:

- Обчислення позиції довільної точки долоні руки в дальностном зображенні;
- Витяг зображення руки по позиції точки долоні.

3.3.1 Обчислення позиції довільної точки долоні руки в дальностном зображенні

Можливі три різні способи обчислення позиції довільної точки долоні руки:

- Обчислення позиції руки як ключовий точки тіла людини [26];
- Обчислення позиції за допомогою показу початкового жесту
- Обчислення позиції за допомогою порогової обробки.

Як зазначалося в розділі 1, на сьогоднішній день існують програмні бібліотеки, які дозволяють в реальному часі розпізнавати 20 ключових точок тіла людини в дальностном зображенні. Зазначені бібліотеки можуть бути використані для обчислення позиції долоні руки в випадках, коли велика частина тіла людини знаходиться в області видимості сенсора.

Обчислити позицію долоні можна за допомогою показу початкового жесту — руху рукою вперед, струшування рукою і т.д. За інформацією про початковому жесті, у відеоряді шукається об'єкт, чия траєкторія руху збігається з певним жестом. Знайдений об'єкт розпізнається як долоня руки і відстежується в наступних кадрах відеоряду. Дана технологія реалізована в деяких програмних платформах, і може бути використана для розпізнавання позиції руки.

При деяких обмеженнях, порогова обробка може бути використана для розпізнавання позиції долоні. Ставиться обмеження — жести руки розглядаються тільки в тому випадку, коли відстань долоні руки до сенсора знаходиться в фіксованому діапазоні $[r_1, r_2]$. Для кожного кадру відеоряду виконуються наступні дії:

- Видалення всіх точок, які не входять у розглянуту зону;
- Пошук дискретних фігур в дальностном зображенні;
- Обчислення центрів знайдених дискретних фігур;
- Фільтрація дискретних фігур.

Видалення точок (x, y) здійснюється шляхом порівняння значень d(x, y) всіх точок зі значеннями r_1, r_2. Якщо значення d(x, y) точки (x, y) не входить в діапазон $[r_1, r_2]$, то в даній точці d(x, y) обнуляється. Пошук дискретних фігур здійснюється в два проходи, за допомогою алгоритму пошуку пов'язаних компонент в графі [11].

Під час першого проходу всі крапки зображення позначаються тимчасовими мітками, де мітки являють собою цифрові значення. Паралельно створюється безліч еквівалентних міток. Наприклад, на рисунку 3.4 мітки 2 і 3 є еквівалентними. У другому проході все тимчасові мітки змінюються на мітку з мінімальним значенням з числа еквівалентних міток. Наприклад, на рисунку 3.4 мітка 3 зміниться на мітку 2. Після другого проходу, безліч точок, позначених еквівалентними мітками, буде являти собою дискретну фігуру в заданому дальностном зображенні.

Наприклад, на рисунку 2.4 безліч точок, помічених міткою 1 і міткою 2 є дискретними фігурами.

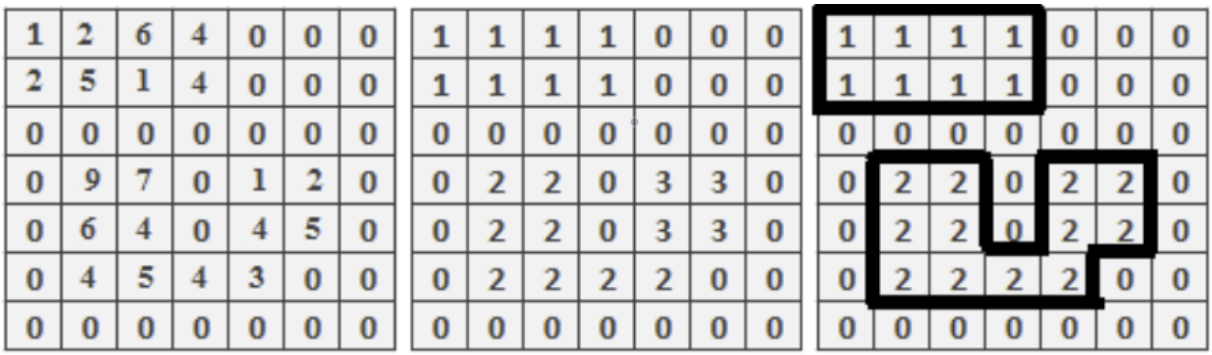


Рисунок 3.4 — Приклад маркування точок дискретних фігур в дальностном зображенні при виборі r_0=5. Зліва направо показані відповідно дальностне зображення; маркування після першого проходу і маркування після другого проходу

Позначимо через K(x, y) мітку точки (x, y) в зображенні d(x, y). Наступним кроком алгоритму є обчислення центрів знайдених дискретних фігур. Центри обчислюються окремо для кожної дискретної фігури за допомогою моментів. Нехай C — дискретна фігура в зображенні d(x, y),

Наступним кроком алгоритму є фільтрація знайдених дискретних фігур. З усіх фігур видаляються:

- фігури, розмір яких занадто малий, щоб бути зображенням долоні людини;
- фігури, центри яких не змінюють розташування протягом часу.

Для відстежування місцеположення центрів розглядається фіксоване число попередніх кадрів відеоряду. Розпізнані центри дискретних фігур, вибираються як позиції долоні. Залежно від конкретного завдання, для обчислення позиції долоні руки застосовується один з наведених методів. Далі будемо припускати, що позиція долоні відома.

3.3.2 Витяг зображення руки по позиції долоні

Позначимо через dist(a, b) відстань між точками $a, b \in R^n$. Нехай є розпізнана точка долоні (x_c, y_c) в дальностном зображенні d(x, y). Розглянемо модель спостережуваного об'єкта у вигляді. Множини просторових точок

{(x, y, d(x, y))}.

- Створюється сфера навколо точки (x_c, y_c, d(x_c, y_c)) з радіусом $\lambda = \left\lceil \frac{C}{d(x_c, y_c)} \right\rceil$, де C — деяка константа, що залежить від характеристик конкретного сенсора (в експериментах C = 80000);

- Позначаються всі крапки (x, y) дальностного зображення, які не є r_0-пов'язаними з розпізнаної точкою долоні (x_c, y_c) множини непомічених точок;

- Позначаються всі крапки (x, y) дальностного зображення, для яких (x, y, d(x, y)) не входять до побудованої сфери;

- На базі вихідного зображення створюється нове дальностне зображення з розміром (2λ+1) x (2λ +1), Причому, в помічених точках визначається d(x, y) = 0

Витяг зображення руки здійснюється за час, що оцінюється O(n), де n — число точок вихідного зображення. На рисунку 3.5.a показані дальностне зображення людини у вигляді півтонування, Незрозумілий точка долоні і сфера навколо цієї точки. На малюнку 3.5.b показано виділене зображення руки. Зауважимо, що в наведеному зображенні рука знаходиться досить близько до обличчя людини і закриває частину обличчя, що не є перешкодою для отримання зображення руки.

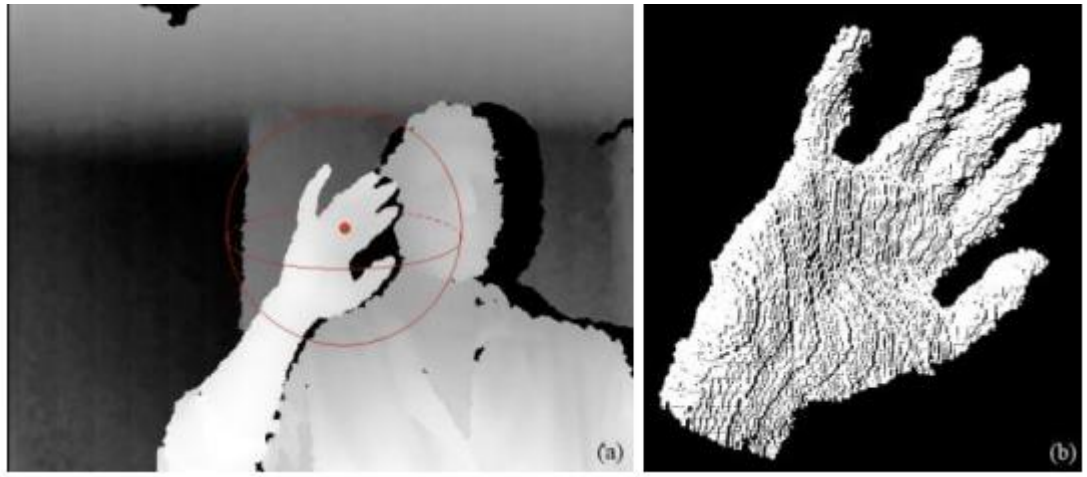


Рисунок 3.5 — (a) Дальностне зображення людини, розпізнана точка долоні і сфера навколо знайденої точки; (b) Витяг зображення руки.

3.4 Розпізнавання позицій кінчиків пальців і точок з'єднання пальців з долонею руки

Представимо дальностне зображення руки у вигляді бінарного зображення з функцією інтенсивності g(x, y), де

$$g(x,y) = 1 - sg\big(d(x,y)\big) \qquad (3.2)$$

На рисунку 3.6.a наведено бінарне зображення, отримане з дальностного зображення (рис. 3.5.b) за допомогою перетворення (3.2).

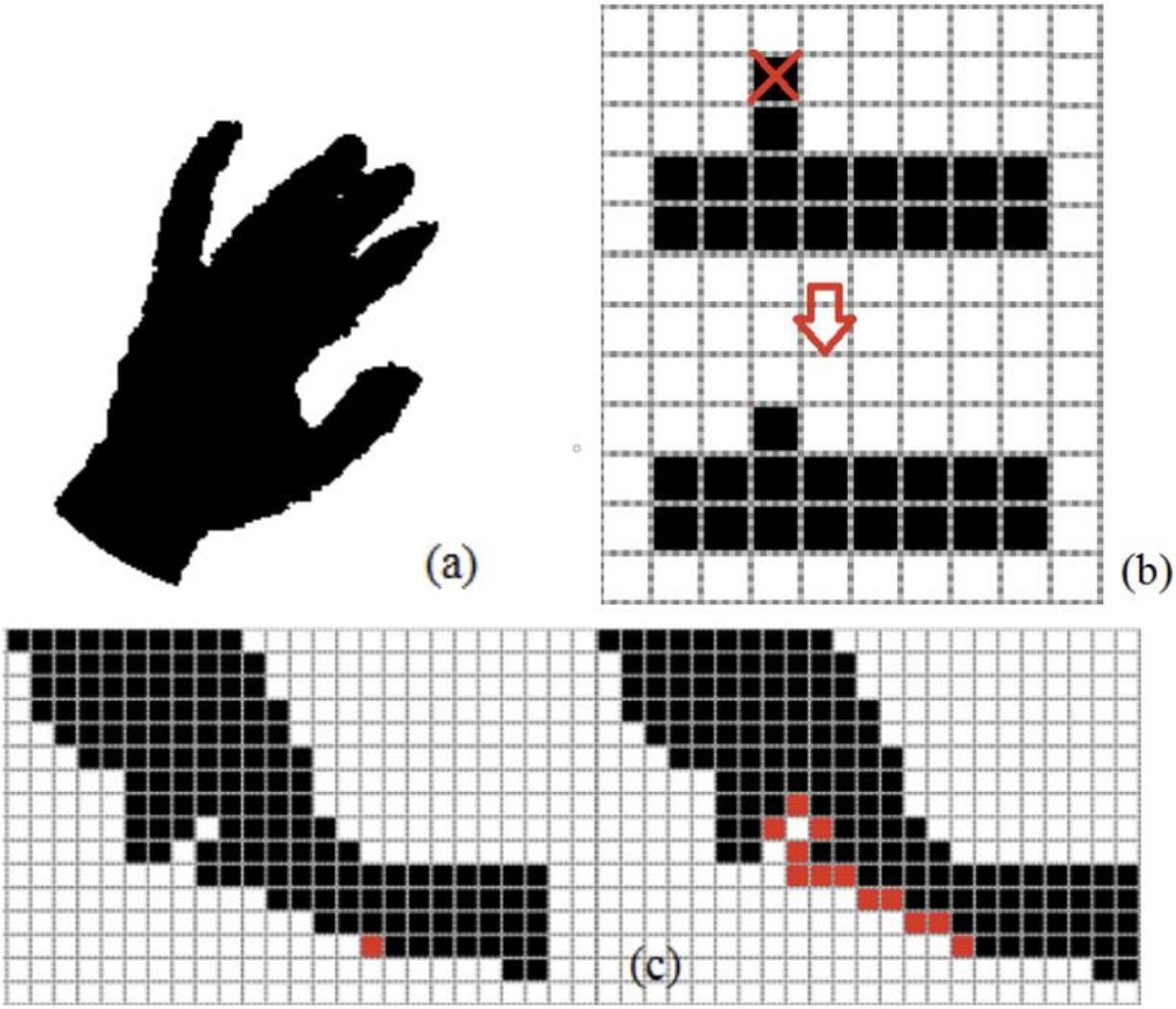


Рисунок 3.6 — (a) Бінарне зображення руки; (b) Видалення ізольованою точки призводить до появи нової ізольованою точки; (c) Необхідність попереднього згладжування. У лівому кутку показана початкова точка, а в правом — результат обчислення точок контуру руки.

Розпізнавання позицій кінчиків пальців руки здійснюється за допомогою наступних кроків:

- Згладжування бінарного зображення g(x, y);
- Пошук довільної граничної точки кисті руки;
- Обчислення позицій точок контуру руки;
- Аналіз зміни відстаней точок контуру щодо точки кисті.

Згладжування зображення руки здійснюється шляхом видалення всіх ізольованих точок. Зауважимо, що видалення ізольованою точки, може привести

до появи нової ізольованою точки, яка до цього була граничної (рис. 3.6.b). процес видалення триває доти, доки всі всіх ізольованих точок.

Гранична точка кисті руки $s_0=(x_0,y_0)$ вибирається на кроці алгоритму вилучення зображення руки як гранична точка фігури руки, яка має r_0- пов'язаного сусіда, який не входить в побудовану сферу.

Позначимо через $S=\{s_0, s_1, ..., s_k\}$, $si=(x_i,y_i)$, $i=0,1,...,k$ отримана впорядкована множина точок контуру фігури руки (див. приклад на рис. 3.7.a).

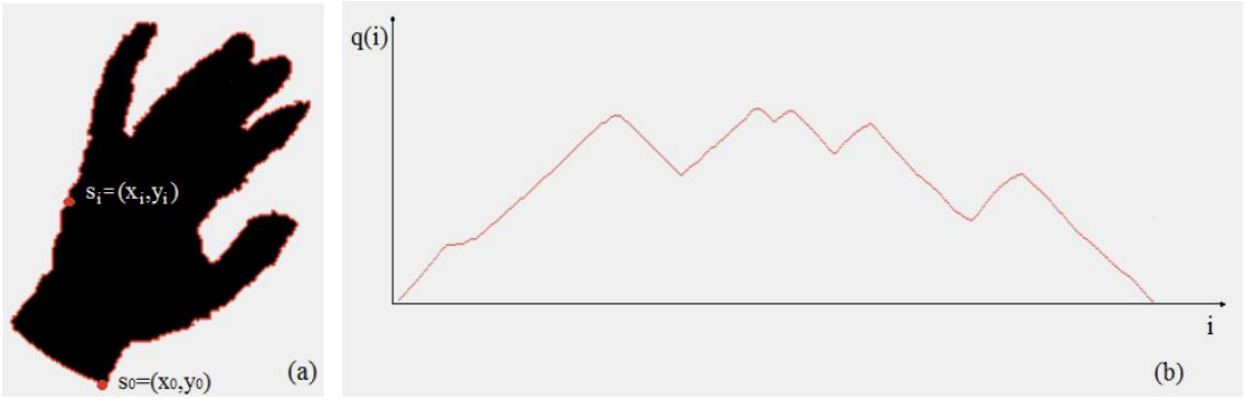


Рисунок 3.7 — (a) Контур фігури руки; (b) Поведінка функції q(i) в точках контуру фігури руки.

Зауважимо, що без попереднього згладжування зображення руки, можна отримати неточні результати. Такий приклад показаний на рисунку 3.6.c.

Розглянемо дискретну функцію відстаней(3.4) на рисунку 3.7.b.

$$q(i) = dist\big((x_i, y_i), (x_0, y_0)\big), i = 0,1, \dots, k \qquad (3.3)$$

На рисунку 3.7.b. можна помітити, що точки контуру руки, що знаходяться на кінчиках пальців руки відповідають локальним максимумів функції q(i), але не всі локальні максимуми є точками кінчиків пальців руки (рис. 2.8.a)

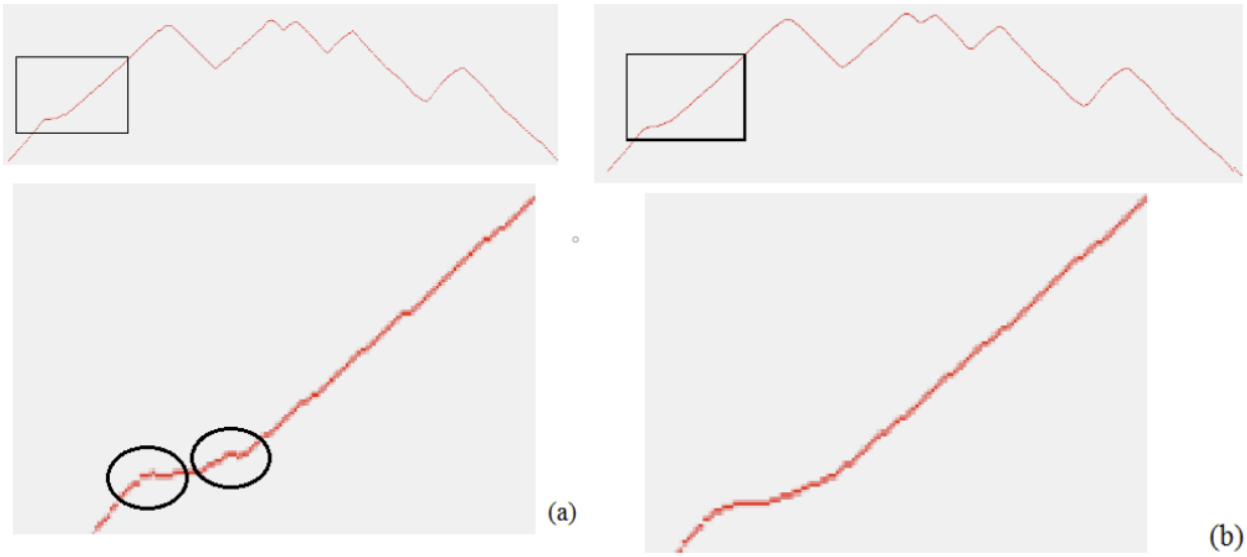


Рисунок 3.8 — (a) Локальні максимуми функції q(i) отримані за причини нерівностей контуру руки; (b) Функція Q(j) після згладжена.

Виконаємо згладжування функції виду (3.4) методом ковзного вікна за допомогою такої формули:

$$Q(j) = \frac{1}{m} \sum_{i=j}^{j+m-1} q(i), j = 0,1, \dots, k - m + 1, k > m \qquad (3.4)$$

Де m — кількість точок ковзного вікна (в програмній реалізації алгоритму m=10). Зауважимо, що функції не згладжується для останніх m точок, що не впливає на процес розпізнавання, так як для всіх жестів руки кінчики пальців знаходяться досить далеко від кисті руки і останні m точок не можуть відповідати кінчиків пальців руки. З цієї причини в подальшому ці точки просто не розглядаються.

На рисунку 3.9 показана функція b(i) відповідна згладженої функції Q з рисунка 3.8.b

Рисунок 3.9 — Поведінка функції b(i).

Всі точки з номерами i, для яких b(i)=1, b(i +1)=0, є точками локальних максимумів, а для яких b(i)=0, b(i+1)=1 — точками локальних мінімумів. У деяких випадках, в функції b(i) з'являються поодинокі точки екстремумів. Для запобігання цьому, функція b(i) згладжується аналогічно (3.5).

Позначимо через $M_1=\{m_1,m_2,...,m_l\}$ множину значень номерів точок локальних максимумів і через $N_1=\{n_1,n_2,...,n_p\}$ — множину значень номерів точок локальних мінімумів функції Q(j), причому $i\leq5$, $p\leq4$, що визначається природними обмеженнями жесту руки людини.

Аналогічно точки з'єднання пальців з долонею визначимо як $(x_j, y_j, d(x_j, y_j))$, де $(x_j, y_j)=s_{nj}m$ j=1,2,...,p. Розпізнані точки, які знаходяться близько до точки кисті руки, не враховуються. Розроблений метод дозволяє за час, що оцінюється як O(n) обчислювати позиції кінчиків пальців і точок з'єднання пальців з долонею руки в кожному кадрі відеоряду, де n — загальна кількість точок витягнутого зображення руки. Результати запропонованого алгоритму для деяких жестів руки приведені на рисунку 3.10.

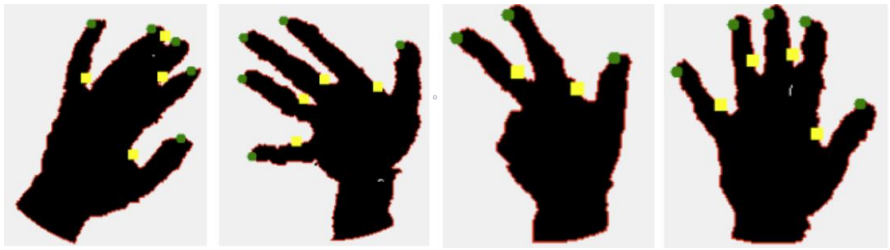


Рисунок 3.10 — Розпізнані кінчики пальців руки і точки з'єднання пальців з долонею.

3.5 Розпізнавання конфігурації руки

Завданням розпізнавання конфігурації руки є ідентифікація показаного статичного жесту при наявності безлічі еталонних конфігурацій руки. Перерахуємо труднощі, з якими стикаються системи розпізнавання конфігурацій руки:

- Необхідність розпізнавання жестів в реальному часі;
- Відмінності в розмірі долоні / руки у різних людей;
- Особливості жестів, властиві різним людям;
- Шуми, присутні в зображеннях руки.

Порівняння дальностних зображень руки пропонується здійснити за допомогою скелетного подання дальностного зображення. Скелетне уявлення дозволяє зберігати фігуру руки у вигляді плоского графа, що значно зменшує вимоги до ресурсів, необхідних для зберігання і порівняння жестів руки. Іншою перевагою скелетного подання є значуща инваріантність до розмірів долоні руки. Перетворити дальностне зображення руки в скелет можна двома різними способами:

- За допомогою планарного уявлення дальностного зображення;
- За допомогою просторового уявлення дальностного зображення;

3.5.1 Скелетизації зображення руки за допомогою планарного уявлення дальностного зображення

Розглянемо бінарне зображення руки з функцією інтенсивності g(x, y), Отримане на основі дальностного зображення руки d(x, y) за допомогою перетворення (3.3).

Скелетом дискретної фігури в бінарному зображенні називається пов'язане безліч точок, шириною в один піксель, які рівновіддалені від граничних пікселів фігури. Для коректного математичного визначення скелета фігури використовують безперервну модель фігури, яка є апроксимацією дискретної моделі, де фігура визначається як пов'язана замкнута область на площині, обмежена кінцевим числом неперетинних Жорданових кривих. Жорданової кривої називають образ кола при безперервному ін'єкційних її відображенні в євклидову площину, де відображення вважається ін'єкційних, якщо дві різні точки прообразу відображаються в дві різні точки образу. Нехай F є безперервна апроксимація дискретної фігури, тоді можна дати наступні визначення.

Порожнім колом радіуса r фігури F називається замкнута множина точок $S_r(x_0, y_0)=\{(x,y): dist((x,y), (x_0, y_0))\leq r\}$ така, що $S_r(x_0, y_0)$ належить F, $r \geq 0$.

Максимальним порожнім колом називається порожній коло, який не міститься ні в якому іншому порожньому колі. Скелетом фігури називається безліч центрів всіх її максимальних порожніх кіл. Алгоритми побудови скелетів дискретних фігур в бінарному зображенні можна розділити на 4 основні групи:

- Алгоритми, засновані на обчислення відстані точок до кордону фігури (англ. distance transformation);
- Алгоритми, засновані на обчислення діаграми Вороного;
- Алгоритми ітеративного стоншування (англ. Thinning);
- Алгоритми, засновані на безперервному поданні фігури.

Перша група алгоритмів працює наступним чином. для бінарного зображення генерується карта відстаней, яка зберігає відстані від кожній групі граничної точки зображення до найближчій граничної точки. Скелет фігури визначається локальними екстремумами карти відстаней (рис. 3.11)

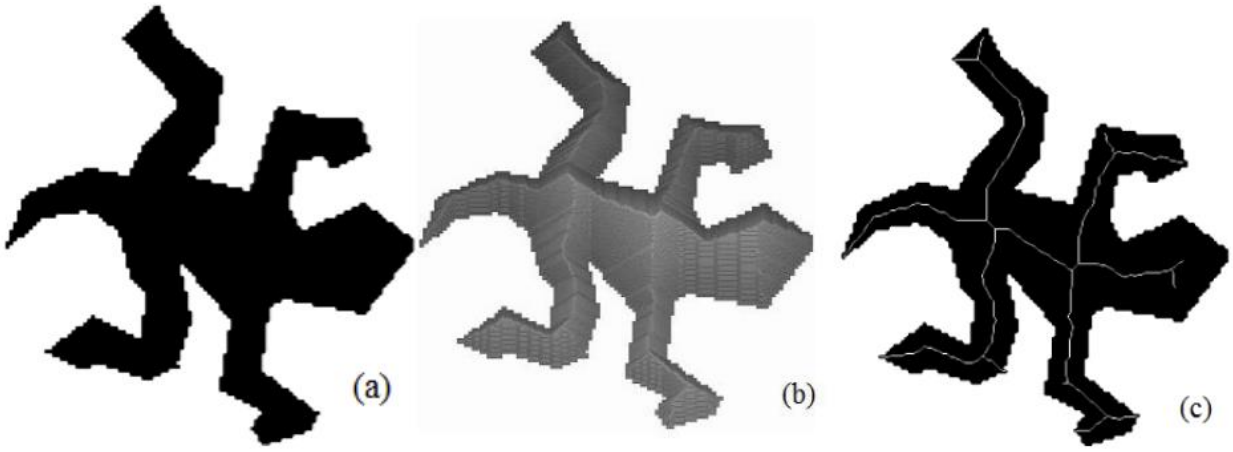


Рисунок 3.11 — (a) Бінарне зображення; (b) Карта відстаней у вигляді тривимірного об'єкту, де третя координата — це відстань точки до найближчій граничної точки; (c) Скелет фігури.

Для заданої кінцевої множини об'єктів діаграма Вороного – це розбиття метричної площині на області, в яких знаходяться ці об'єкти, таким чином, що кожному об'єкту відповідає область площині, утворена точками, які знаходяться ближче до цього об'єкта, ніж до будь-якого іншого. Об'єктом в бінарному зображенні може виступати двовимірна точка або лінійний відрізок. Для знаходження скелета фігури на безлічі точкових відрізків, що апроксимують кордон фігури, будується діаграма Вороного, з якої далі виділяється скелет (рисунок. 3.12).

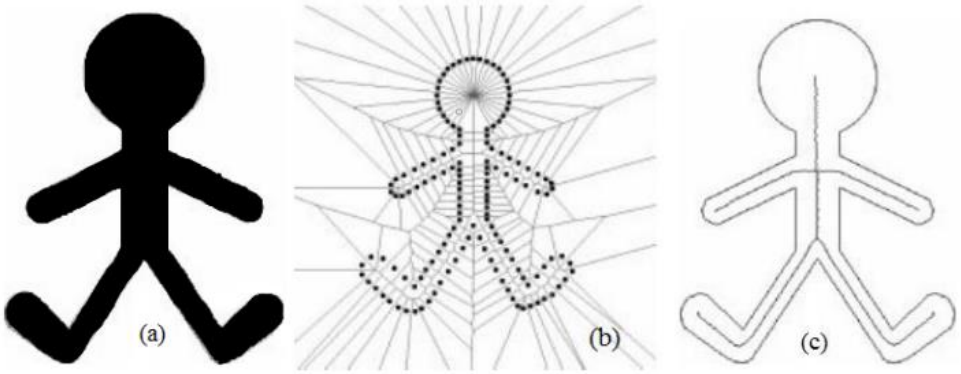


Рисунок 3.12 — (a) Бінарне зображення; (b) Діаграма Вороного для множини точок, що апроксимують кордон фігури; (c) Скелет фігури.

Знаходження скелета фігури за допомогою стоншування — це ітеративна редукція (ерозія) фігури, що нагадує поширення вогню в поле, коли вогонь одночасно починається по всіх краях і рівномірно рухається до центру поля. Скелет фігури, за цією аналогією, є множиною точок, в яких відбувається зустріч кількох вогненних фронтів (рис. 3.13).

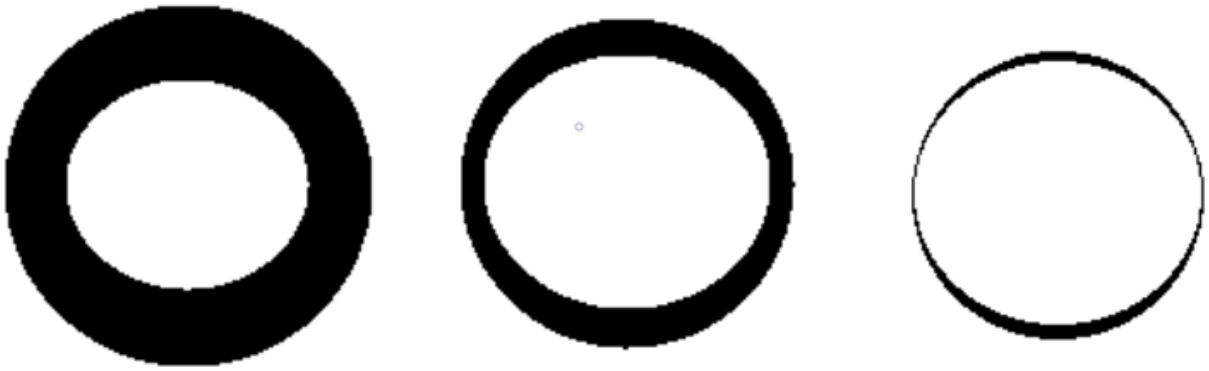


Рисунок 3.13 — Побудова скелета фігури за допомогою стоншування. зліва-направо показані початкова фігура і фігура після кількох ітерацій

Четверта група алгоритмів, використовуваних для створення скелетного уявлення дискретної фігури, що апроксимує дискретну кордон фігури безперервної, після чого створюється геометричний скелет безперервної фігури. У більшості випадків, апроксимація кордону здійснюється за допомогою багатокутника, а скелет, що складається з прямих ліній і сегментів парабол або кривих Безьє першого і другого порядку, шукається за допомогою побудови узагальненого діаграма Вороного для багатокутної фігури (рис. 3.14).

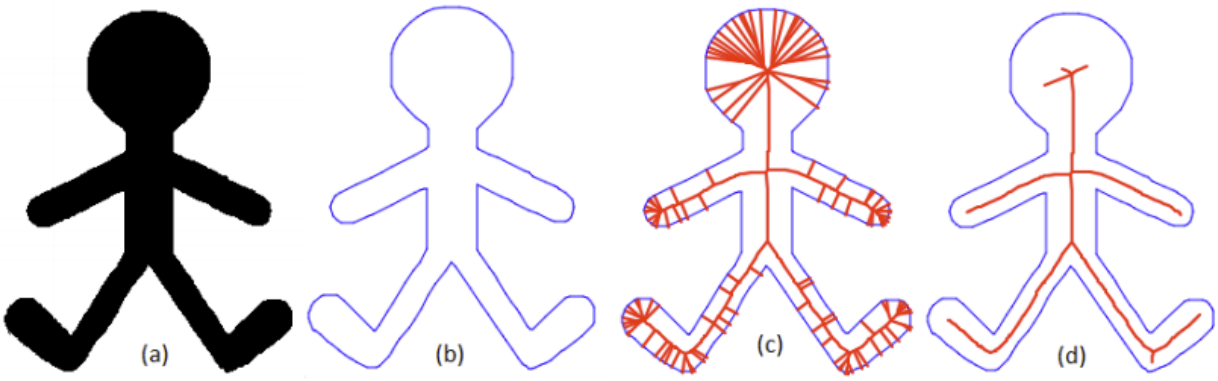


Рисунок 3.14 — Побудова скелета фігури за допомогою апроксимації дискретної фігури безперервної; (a) Дискретна фігура. (b) Апроксимована безперервна межа; (c) Безперервний скелет; (d) Безперервний скелет після «стрижки».

Алгоритми, засновані на обчисленні відстані точок до кордону фігури і ітеративного стоншування, дозволяють отримати скелет зображення в вигляді нового бінарного зображення, що обмежує можливості класифікації зображень і вимагає зберігання великих даних в базі еталонних конфігурацій руки. Отримати безперервний скелет фігури можна за допомогою діаграм Вороного або апроксимацією дискретної фігури безперервної і обчисленням скелета безперервної фігури. обидва методи дозволяють отримати скелет фігури у вигляді плоского графа, але перший метод більш чутливий до нерівностей кордону фігури,. Зауважимо, що, незважаючи на те, що всі розглянуті алгоритми призначені для обчислення середньої осі фігури, при обробці одного зображення різними алгоритмами, на виході можуть виходити значно відрізняють один від одного скелети (рис. 3.13.c і рис. 3.15.c). наприклад, існують різні способи апроксимації кордону дискретної фігури за допомогою безперервного многокутника, звідки випливає, що алгоритми четвертої групи можуть на виході видати різні скелети для однієї дискретної фігури. Іншу особливість геометричних скелетів можна спостерігати на малюнку 3.15.c, де видно дуги, створені через нерівності кордону дискретної фігури.

Незначні зміни кордону фігури можуть привести до істотних змін скелета фігури. Це означає, що завдання скелетизації є некоректно поставленим завданням. Дані завдання вирішуються методом регуляризації (англ. Skeleton pruning) – пошуком наближених рішень, стійких до малих змін форми фігури. У разі безперервних скелетів, результатом регуляризації зазвичай є підграф початкового скелета, який називають базовим скелетом фігури, а процес видалення дуг, що не входять в базовий кістяк, називають «стрижкою» скелета. позначимо через V_G об'єднання всіх максимальних порожніх кіл фігури F, центри яких лежать на скелеті G. V_G називається силуетом скелета G.

Базовим скелетом фігури F називається такий мінімальний укорочений підграф G′ її скелета G, Для силуету $V_{G'}$ якого виконана умова $H(F, V_{G'}) \leq \varepsilon$, де ε — задана позитивна величина, $H(F, V_{G'})$ — хаусдорфова відстань між фігурою F і силуетом $V_{G'}$.

На практиці для стрижки скелета використовують і інші методи, як наприклад видалення дуг, в центрі яких радіуси порожніх кіл менше заданої величини. Цю величину зазвичай називають регуляризуючим параметром. На малюнках 2.15.d показаний скелет фігури після стрижки, з регулюючим параметром 1. Можна зауважити, що після стрижки в скелеті видаляються всі дуги, які були створені через нерівностей кордону фігури.

Як зазначалося вище, дальностним зображенням характерний значний шум на кордонах фігури (рис. 3.5.b, 3.16.a), у зв'язку з чим, регуляризації скелета з малими параметрами стрижки недостатньо для отримання скелета руки, інваріантного до даного шуму. На рисунку 3.15 показані бінарне зображення руки, початковий скелет і скелети, отримані після стрижки з параметрами 1 і 4. Оптимальне налаштування стрижки для бінарних зображень, отриманих на основі дальностних зображень руки, отриманий емпірично, дорівнює значенню 5.

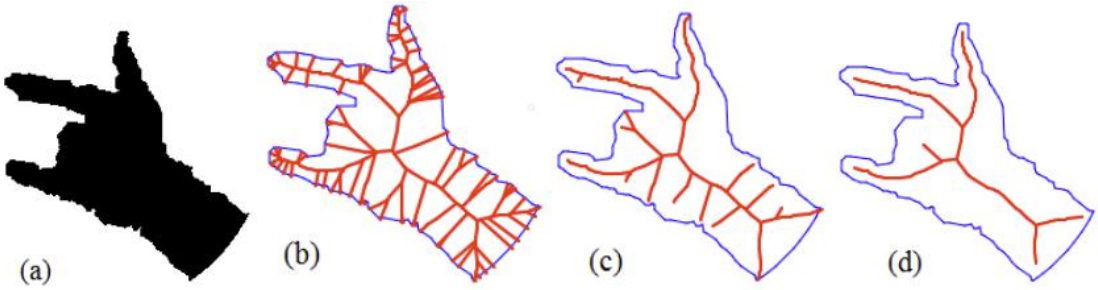


Рисунок 3.15 — (a) Бінарне зображення руки; (b) Скелет руки; (c) Скелет руки після стрижки з регулюючим параметром 1; (d) Скелет руки після стрижки з регулюючим параметром 4.

Стрижка скелета, звичайно, не є єдиним методом отримання скелета, інваріантним до малих змін кордону фігури. отримати такий скелет можна також за допомогою початкового згладжування контуру фігури.

Нехай заданий список точок що належать контуру фігури, упорядкованих за годинниковою стрілкою. Алгоритм згладжування точок контуру здійснюється за допомогою кривих Безьє.

Результат застосування алгоритму на довільній кривою показаний на рисунку 3.16.

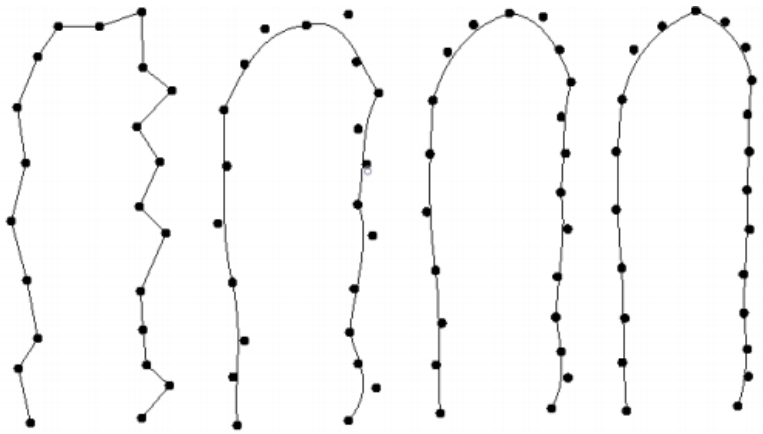


Рисунок 3.16 — Згладжування контуру за допомогою кривої Безьє після 1, 2 і 3 ітерацій.

Даний підхід можна застосувати для скелетизації фігури за допомогою діаграм Вороного. Використовуючи алгоритми Suzuki і Abe для бінарного зображення, знайдемо граничні точки руки. Виконаємо фільтрацію отриманих точок контуру так, щоб будь-які дві точки не знаходились ближче заданого фіксованого відстані.. Для отриманих точок створимо діаграму Вороного і потім видалимо зайві дуги, отримуючи тим самим скелет фігури. Результат застосування запропонованого алгоритму для бінарного зображення руки наведено на рисунку 3.17. Можна помітити, що в отриманому скелеті руки відсутні дуги, створювані внаслідок нерівностей контуру фігури руки.

Таким чином, пропонується перетворити дальностне зображення в бінарне, а потім застосувати один з наступних методів скелетизації дальностного зображення руки:

- Метод скелетизації за допомогою пошуку граничних точок і створення діаграми Вороного для цих точок;

- Метод скелетизації шляхом подання відповідного дискретної фігури в вигляді безперервної фігури і обчислення скелета цієї фігури.

Обидва методи дозволяють в реальному часі отримувати скелет фігури руки, інваріантний до шумів в контурах фігури.

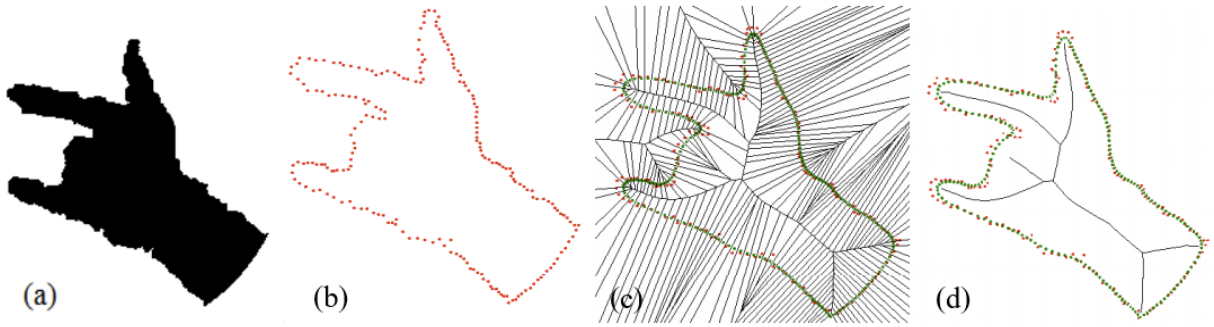


Рисунок 3.17 — (a) Бінарне зображення руки; (b) Відібрані точки контуру; (c) Діаграма Вороного для згладжених точок; (d) Скелет руки.

Єдиним значущим недоліком даного підходу є втрата первинною інформацією про жести руки, під час перетворення дальностного зображення в

бінарне. На рисунку 3.18 показані дальностне і бінарне зображення руки зі скелетом фігури руки.

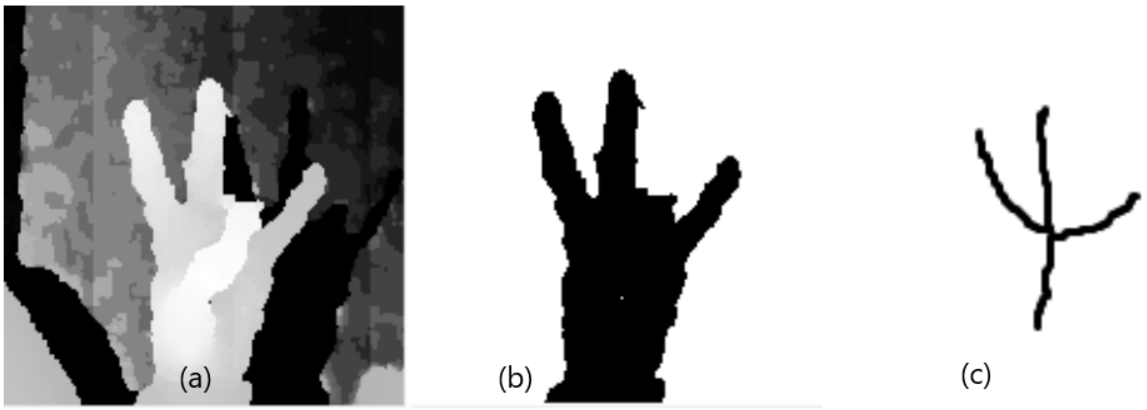


Рисунок 3.18 — (a) Дальностне зображення руки; (b) Бінарне зображення руки; (c) Скелет руки.

Більшість з наведених алгоритмів дозволяють отримати досить якісні скелети з дальностних зображень руки, але їх всіх об'єднує один недолік — висока обчислювальна складність.

Таким чином, незважаючи на те, що криволінійні скелети надають досить багато інформації про статичному жесті руки, обчислювальна складність їх отримання є дуже високою, що перешкоджає використанню цих алгоритмів в додатках реального часу.

4 РОЗРОБКА ТА АДАПТАЦІЯ МЕТОДІВ ЛЮДИННО-МАШИННОЇ ВЗАЄМОДІЇ ЗА ДОПОМОГОЮ ЖЕСТІВ

4.1 Обрані технології та бібліотеки програмної мови

У якості базового язика програмування у рамках цієї роботи було обрано Python. Python — це мова програмування загального призначення, яка часто застосовується в сценаріях. Зазвичай його визначають як об’єктно-орієнтовану мову сценаріїв - визначення, яке поєднує підтримку ООП із загальною орієнтацією на роль сценарію. Фактично, люди часто використовують слово «скрипт» замість «програма» для опису файлу коду Python. Він обладає рядом ключових ознак за які його часто вибирають для написання комплексних програм у сфері аналізу даних та машинного навчання.

- Якість програмного забезпечення. Основна увага Python робиться на читабельність, узгодженість та якість програмного забезпечення загалом відрізняє його від інших інструментів у світі сценаріїв. Однорідність коду Python полегшує розуміння, навіть якщо ви цього не написали. Крім того, Python має глибоку підтримку для більш просунутих механізми повторного використання програмного забезпечення, такі як об'єктно-орієнтоване програмування (ООП).

- Продуктивність розробника. Python багато разів підвищує продуктивність розробників, окрім скомпільованого або статичного типизовані мови, такі як C, C ++ та Java. Код Python зазвичай становить одну третину одна п’ята за розміром еквівалентного коду C ++ або Java. Це означає, що набирати менше, менше налагоджувати і менше підтримувати після факту. Програми Python також запускаються негайно, без тривалих кроків компіляції та посилань, необхідних для деяких інших інструментів, подальше підвищення швидкості програміста.

- Переносимість програми. Більшість програм Python працюють без змін на всіх основних комп'ютерних платформах.

- Портування. Наприклад, код Python між Linux та Windows, як правило, лише питання копіювання коду сценарію між машинами. Більше того, Python

пропонує безліч варіантів кодування переносних графічних користувальницьких інтерфейсів, програм доступу до баз даних, веб-систем тощо. Навіть інтерфейси операційної системи, включаючи програму запуски та обробка каталогів, настільки портативні в Python, наскільки це можливо бути.

- Підтримка бібліотек. Python має велику колекцію попередньо вбудованих та портативних функціональних можливостей, відомих як стандартна бібліотека. Ця бібліотека підтримує масив завдань програмування на рівні програми, від відповідності шаблону тексту до мережевих сценаріїв. В додаток, Python можна розширити як власними бібліотеками, так і величезною колекцією програмне забезпечення для підтримки сторонніх програм. Сторонній домен Python пропонує інструменти для побудови веб-сайтів, числового програмування, доступу до послідовних портів, розробки ігор та багато іншого. Наприклад, у наступних підрозділах буде описано бібліотеки OpenCV, NumPy, Keras

- Інтеграція компонентів. Скрипти Python можуть легко спілкуватися з іншими частинами програми, використовуючи різноманітність механізмів інтеграції. Такі інтеграції дозволяють використовувати Python як інструмент налаштування та розширення товару. Сьогодні код Python може викликати C та бібліотеки C ++, які можна викликати з програм C і C ++, можуть інтегруватися з Java та .NET-компоненти можуть спілкуватися через такі фреймворки, як COM, можуть взаємодіє з пристроями через послідовні порти і може взаємодіяти через мережі з такими інтерфейсами, як SOAP, XML-RPC та CORBA. Це не самостійний інструмент.

4.1.2 OpenCV

Бібліотека комп'ютерного зору і машинного навчання з відкритим вихідним кодом. У неї входять понад 2500 алгоритмів, в яких є як класичні, так і сучасні алгоритми для комп'ютерного зору і машинного навчання. Ця бібліотека має інтерфейси на різних мовах, серед яких є Python (використовується в рамках цієї роботи), Java, C ++ і Matlab.

OpenCV має модульну структуру, що означає, що пакет включає кілька спільних або статичних бібліотек. Доступні наступні модулі:

- Основна функціональність (core) — компактний модуль, що визначає основні структури даних, включаючи щільний багатовимірний масив Mat і основні функції, що використовуються всіма іншими модулями.

- Обробка зображень (imgproc) — модуль обробки зображень, що включає лінійну та нелінійну фільтрацію зображень, геометричні перетворення зображень (зміна розміру, перспективне перекошування, загальне переназначення на основі таблиці), перетворення кольорового простору, гістограми тощо.

- Відеоаналіз (video) — модуль відеоаналізу, що включає оцінку руху, віднімання фону та алгоритми відстеження об’єктів.

- Калібрування камери та 3D-реконструкція (calib3d) — основні алгоритми геометрії декількох видів, калібрування одинарної та стереокамери, оцінка пози об'єкта, алгоритми стереовідповідності та елементи 3D-реконструкції.

- 2D Features Framework (features2d) — основні детектори, дескриптори та збіги дескрипторів.

- Виявлення об’єктів (objdetect) — виявлення об’єктів та екземплярів заздалегідь визначених класів (наприклад, обличчя, очі, кружки, людей, машини тощо).

- Графічний інтерфейс високого рівня (highgui) — простий у використанні інтерфейс для простих можливостей інтерфейсу.

- Video I / O (videoio) — простий у використанні інтерфейс для зйомки відео та відеокодеків.

- OpenCV-Python — це бібліотека прив'язок Python, призначена для вирішення проблем із комп'ютерним зором.

- OpenCV-Python використовує Numpy, що є високооптимізованою бібліотекою для числових операцій із синтаксисом у стилі MATLAB. Усі структури масивів OpenCV перетворюються в масиви Numpy і з них. Це також полегшує інтеграцію з іншими бібліотеками, які використовують Numpy, такими як SciPy та Matplotlib.

За допомогою бібліотеки можна перетворювати різні перетворенням як над статичним зображенням, так і над деякою послідовністю кадрів, наприклад відео та використовувати отримані данні для машинного навчання за допомогою модуля objdetect. Приклад такої операції наведено на рисунку 4.1.

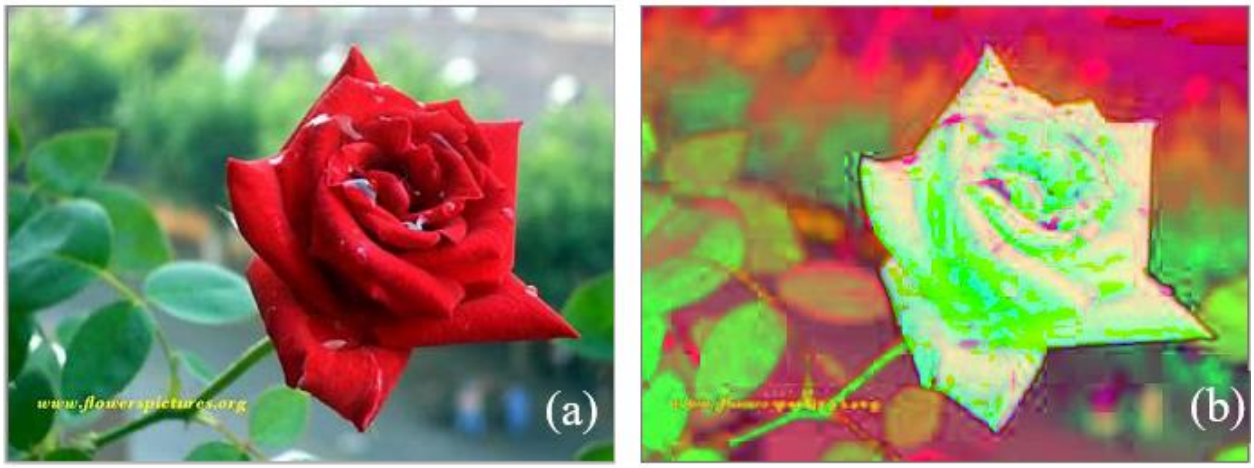


Рисунок 4.1 — результат застосування функції cv2.cvtColor Бібліотеки OpenCV

4.1.3 Keras

Незважаючи на те, що глибокі нейронні мережі — це велика мода, складність основних фреймворків стала перешкодою для їх використання для розробників, нових у машинному навчанні. Було кілька пропозицій щодо вдосконалених та спрощених API високого рівня для побудови моделей нейронних мереж, які, як правило, здаються схожими на відстань, але при детальному вивченні демонструють відмінності.

Keras – один з провідних API високого рівня нейронних мереж. Він написаний на Python і підтримує кілька внутрішніх механізмів обчислень нейронних мереж. Keras був створений для зручності користування, модульності, простоти розширення та роботи з Python. API був «розроблений для людей, а не для машин», і «дотримується найкращих практик зменшення когнітивного навантаження».

Нейронні рівні, функції витрат, оптимізатори, схеми ініціалізації, функції активації та схеми регуляризації — все це окремі модулі, які ви можете об'єднати для створення нових моделей. Нові модулі просто додавати, як нові класи та функції. Моделі визначаються кодом Python, а не окремими файлами конфігурації моделі. Блок-схема такої конфігурації представлена на рисунку 4.2.

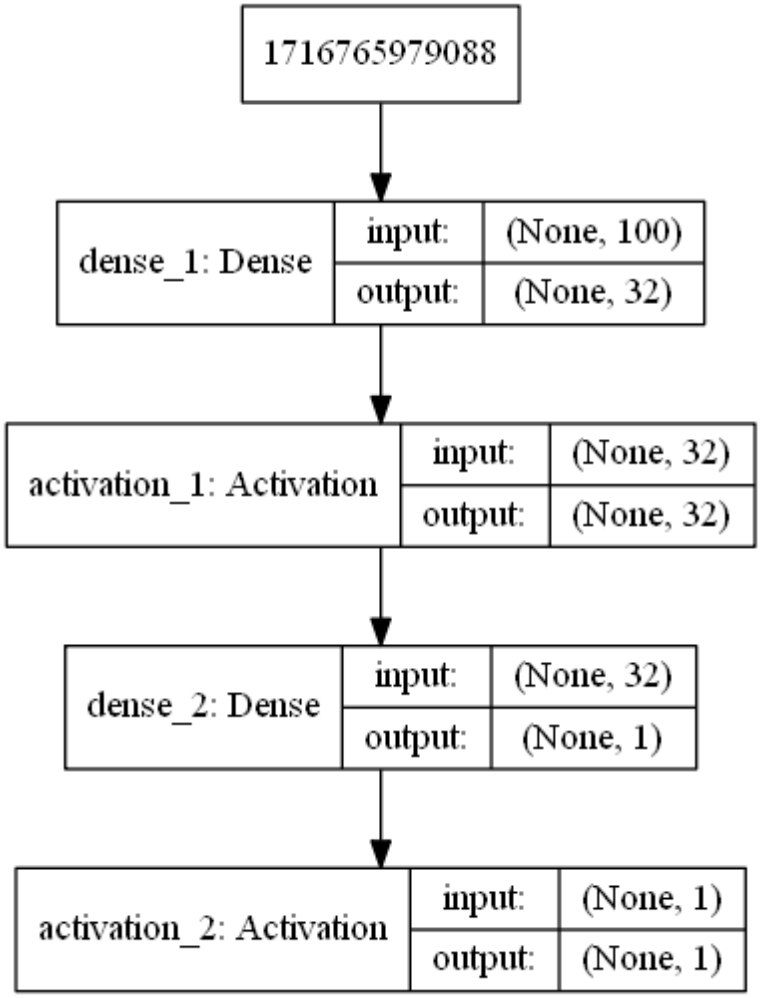


Рисунок 4.2 — вигляд моделі Keras, яка представлена у вигляді об’єктів Python

4.1.3 Numpy

NumPy (Numerical Python) — це бібліотека Python з відкритим кодом, яка використовується майже в усіх галузях науки та техніки. Це універсальний стандарт роботи з числовими даними в Python, і він лежить в основі наукових екосистем Python та PyData. Користувачі NumPy охоплюють усіх — від початківців до досвідчених дослідників, які проводять ультрасучасні наукові та промислові дослідження та розробки.

Бібліотека NumPy містить багатовимірні масиви та матричні структури даних (більше інформації про це ви знайдете у наступних розділах). Він забезпечує

ndarray, однорідний n-вимірний об'єкт масиву, з методами ефективної роботи з ним. NumPy можна використовувати для виконання різноманітних математичних операцій над масивами. Він додає до Python потужні структури даних, які гарантують ефективні обчислення за допомогою масивів і матриць, а також забезпечує величезну бібліотеку високорівневих математичних функцій, які працюють з цими масивами та матрицями.

Хоча список Python може містити різні типи даних в межах одного списку, всі елементи масиву NumPy повинні бути однорідними. Математичні операції, які передбачається виконувати над масивами, були б неможливі, якби масиви не були однорідними. Масиви NumPy швидші та компактніші за списки Python. Масив споживає менше пам'яті і є набагато зручнішим у використанні. NumPy використовує набагато менше пам'яті для зберігання даних, і він забезпечує механізм визначення типів даних, що дозволяє ще більше оптимізувати код.

Масив – це центральна структура даних бібліотеки NumPy. Це сітка значень, що містить інформацію про вихідні дані, як знайти елемент та як інтерпретувати елемент. Він має сітку елементів, які можна індексувати різними способами. Усі елементи одного типу, іменовані як масив dtype (тип даних). Масив може бути проіндексований набором цілих неотрицальних чисел, булевими значеннями, іншим масивом або цілими числами. Ранг масиву - це кількість розмірів. Форма масиву являє собою набір цілих чисел, що задають розмір масиву вздовж кожного виміру. Одним із способів ініціалізації масивів NumPy є вкладені списки Python:

a = np.array([[1 , 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])

Ми можемо отримати доступ до елементів масиву за допомогою квадратних дужок. Коли ви отримуєте доступ до елементів, пам’ятайте, що індексація в NumPy починається з 0. Це означає, що якщо ви хочете отримати доступ до першого елемента у своєму масиві, ви отримаєте доступ до елемента „0”.

Масив, як правило, є контейнером фіксованого розміру з елементами одного типу та розміру. Кількість розмірів та елементів у масиві визначається його формою. Форма масиву — це сукупність невід’ємних цілих чисел, які визначають

розміри кожного виміру. У NumPy розміри називаються осями. Це означає, що якщо у вас є 2D-масив, який виглядає так: [[0., 0., 0.], [1., 1., 1.]]

4.2 Згорткова нейронна мережа (Convolutional Neural Network)

Найкращі результати в області розпізнавання осіб показала Convolutional Neural Network або Згорткова Нейронна Мережа, яка є логічним розвитком ідей таких архітектур як когнітрон і неокогнітрон. Успіх обумовлений можливістю обліку двовимірної топології зображення, на відміну від багатошарового персептрона.

Згорткові нейронні мережі забезпечують часткову стійкість до змін масштабу, зсувів, поворотам, зміні ракурсу і іншим спотворень. Згорткові нейронні мережі об'єднують три архітектурних ідеї, для забезпечення інваріантності до зміни масштабу, повороту зрушення і просторовим спотворень:

- локальні рецепторні поля (забезпечують локальну двовимірну зв'язність нейронів);

- загальні вагові коефіцієнти синапсів (забезпечують детектування деяких рис в будь-якому місці зображення і зменшують загальне число вагових коефіцієнтів);

- ієрархічна організація з просторовими підвибірками.

ЗНМ складається з різних видів шарів: згорткові (convolutional) шари, субдискретизуючі (subsampling/pooling, підвибірка) шари і шари «звичайної» нейронної мережі - персептрона, відповідно до рисунку 4.3.

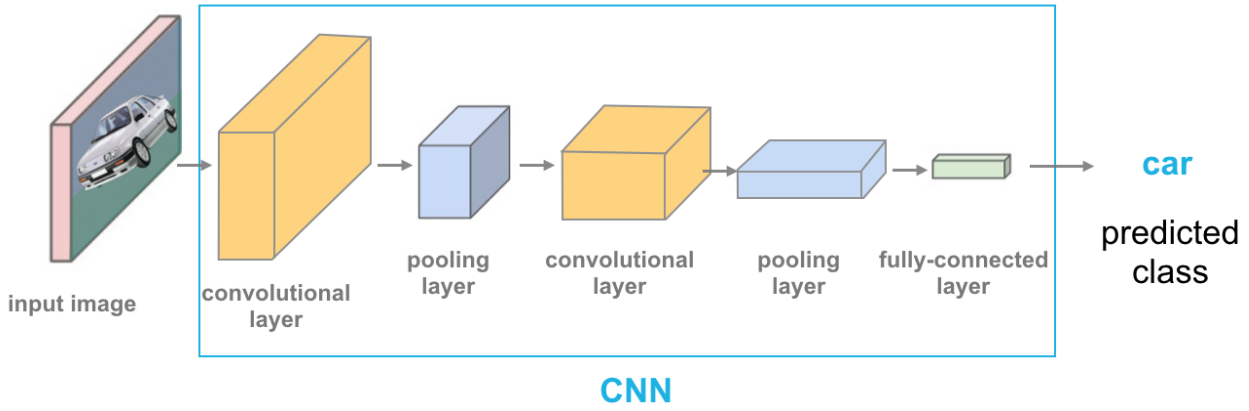


Рисунок 4.3 — Схема шарів Згорткової Нейронної Мережі

Згорткові мережі є вдалою серединою між біологічно правдоподібними мережами і звичайним багатошаровим персептроном. На сьогоднішній день кращі результати в розпізнаванні зображень отримують з їх допомогою. В середньому точність розпізнавання таких мереж перевершує звичайні ІНС на 11-17%. ЗНМ - це ключова технологія Deep Learning.

Основною причиною успіху СНС стало концепція загальних ваг. Незважаючи на великий розмір, ці мережі мають невелику кількість параметрів, що настроюються в порівнянні з їх предком — неокогнітроні. Є варіанти ЗНМ (Tiled Convolutional Neural Network), схожі на неокогнітрон, в таких мережах відбувається, часткова відмова від пов'язаних ваг, але алгоритм навчання залишається тим же і ґрунтується на зворотному поширенні помилки. СНС можуть швидко працювати на послідовній машині і швидко навчатися за рахунок чистого розпаралелювання процесу згортки по кожній карті, а також зворотної згортки при поширенні помилки по мережі.

4.3 Програмна реалізація додатку по розпізнаванню статичних жестів

Для демонстрації розпізнавання статичних жестів будуть використані кілька простих але значущих жестів. Ці жести є простими конфігураціями руки в нерухомому стані.

Розроблений метод розпізнавання включає етапи навчання і розпізнавання. Навчання може здійснюватися двома способами:

- на вхід програмі подається набір зображень для навчання

- оператор в графічному інтерфейсі програми вибирає пункт "навчання за прикладом" і показує відповідний жест. Показаний жест захоплюється камерою. У зображенні шукається рука людини, яка згодом витягується. Побудований жест зберігається в базі еталонних конфігурацій. Інтерфейс розробленої програми показаний на рсиунку 4.4. Зауважимо, що для одного жесту може зберігатися кілька еталонних конфігурацій.

Рисунок 4.4 — Інтерфейс розробленої програми розпізнавання жестів

В програмі реалізовано кілька режимів роботи. В першому – можна обучати класифікатор за допомогою зображень руки або робити ці зображення за допомогою камери. Архітектурою для навчання була обрана Згорткова Нейронна Мережа (ЗНМ) яка була описана раніше. Вона задається програмно на мові Python, та має вигляд:

```
model = Sequential()
model.add(Conv2D(nb_filters, (nb_conv, nb_conv),
                        padding='valid',
                        input_shape=(img_channels, img_rows, img_cols)))
convout1 = Activation('relu')
model.add(convout1)
model.add(Conv2D(nb_filters, (nb_conv, nb_conv)))
convout2 = Activation('relu')
model.add(convout2)
model.add(MaxPooling2D(pool_size=(nb_pool, nb_pool)))
model.add(Dropout(0.5))

model.add(Flatten())
model.add(Dense(128))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(nb_classes))
model.add(Activation('softmax'))
```

Рисунок 4.5 – Програмне завдання шарів нейронної мережі

У результаті виконання коду бібліотекою Keras отримуємо опис нейронної мережи з 12 шарами. Після виконання навчання з прикладами жестів (використалося 3211 фотографій) була отримана нейронна мережа з приблизно 39 мільйонами параметрів. Приклад зображень наведено на рисунку 4.6.

Рисунок 4.6 — зразки зображень для навчання у програмі розпізнання жестів

В другому – програма зчитує файл конфігурації класифікатора та працює у режимі розпізнання. Для демонстрації буде вказано два жести – “Стоп” (англ. Stop) та “Гаразд” (англ. Ok) – на рисунках 4.3 та 4.4.

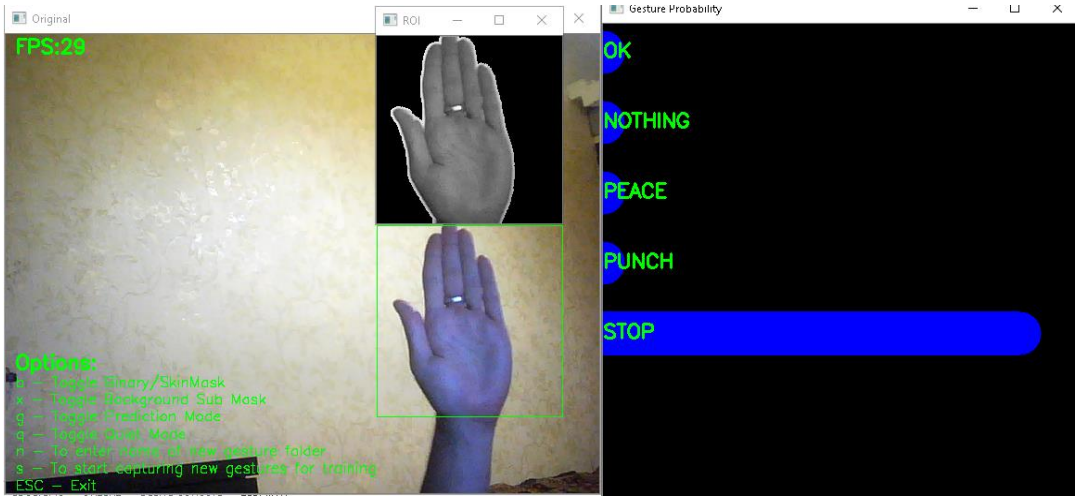


Рисунок 4.3 — режим роботи: Розпізнання, жест що був розпізнаний - Stop

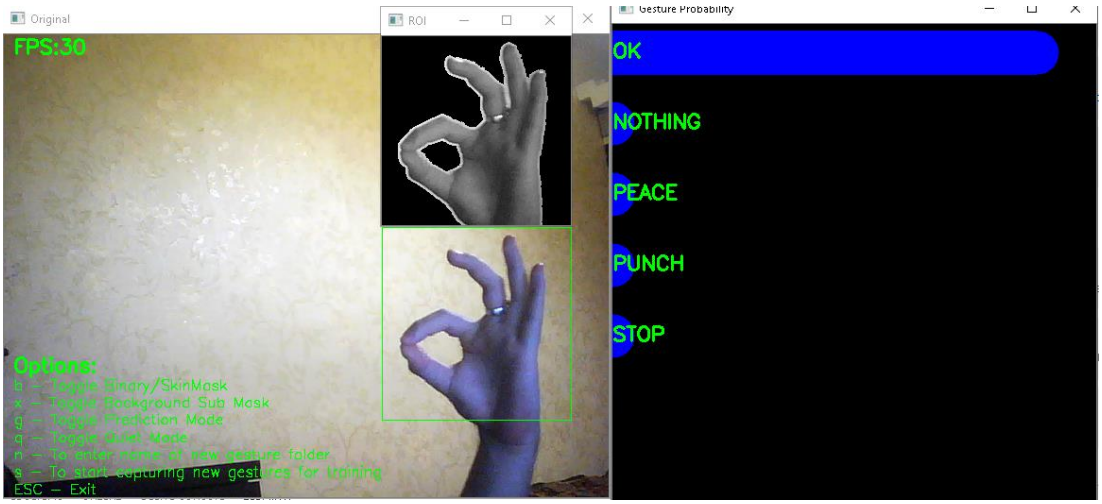


Рисунок 4.4 — режим роботи: Розпізнання, жест що був розпізнаний – Ok

ВИСНОВКИ

У магістерській роботі вирішена задача розпізнавання статичних жестів руки, які можуть бути використані для безконтактної взаємодії людини з комп'ютером. актуальність завдання обумовлена її практичною значущістю, великою кількістю проведених досліджень в цій області, і тим, що якість відомих алгоритмів розпізнавання жестів рук і пальців, як динамічних, так і статичних, з використанням кольорових відеокамер і тривимірних сенсорів все ще залишається недостатнім для побудови практичних систем людино-машинного взаємодії. Отримані наступні основні результати:

Виконана необхідна формалізація, розроблено та досліджено комплекс методів і алгоритмів для попередньої і спеціальної обробки дальностних зображень, в тому числі:

- однопрохідний алгоритм вилучення зображення руки з дальностного зображення за відомою позиції долоні зі складністю O(n), де n — кількість точок вихідного зображення;

- метод обчислення позицій кінчиків пальців і точок з'єднання пальців з долонею руки, на основі аналізу контуру фігури руки зі складністю O(n), де n — кількість точок зображення руки;

- метод скелетизації дальностного зображення зі складністю O(nlog n), що дозволяє в реальному часі обчислити тривимірний скелет фігури, який представляє собою плоский граф, де n — число вершин багатокутника що апроксимує контури фігури в дальностному зображенні;

Сформульовано і доведено необхідні теореми і затвердження, в тому числі доведено, що:

- множини точок $\{(x, y) \mid d(x, y) > 0\}$, отриманих алгоритмом вилучення зображення руки, є єдиною дискретної фігурою в витягнутому дальностном зображенні;

- при перетворенні витягнутого дальностного зображення руки в бінарне зображення, безліч чорних крапок отриманих в бінарному зображенні є дискретною фігурою і збігається з дискретної фігурою руки в дальностном зображенні;

- для запропонованого заходу оцінки відстані між двома скелетами виконуються всі аксіоми метрики за винятком нерівності трикутника

Розроблена програма система розпізнання статичних жестів руки на основі Згорткової Нейронної Мережі використовуючи мову python. Програма має можливість навчання жестам та має приблизну точність у 85%.

ПЕРЕЛІК ПОСИЛАНЬ

1. Karam M., Schraefel M. C. A Taxonomy of Gestures in Human Computer Interactions: Technical Report: : University of Southampton, 2005

2. Wexelblat A. An Approach to Natural Gesture in Virtual Environments // ACM Transactions on Computer-Human Interaction (TOCHI). New York, NY, USA, 1995. sep. Vol. 2, no. 3. 179–200 с.

3. Фролов А. А., Рощин В. Ю. Интерфейс мозг-компьютер. реальность и перспективы // Лекции по нейроинформатике: По материалам Школы-семинара «Современные проблемы нейроинформатики». X Всероссийская научно-техническая конференция «Нейроинформатика-2007». Москва: Московский инженернофизический институт, 2008. jan. 82–117 с.

4. Effects of Virtual Reality Display Types on the Brain Computer Interface System / H. Cho, K. Park, Y. Kim et al. // Universal Access in Human-Computer Interaction. Ambient Interaction / Ed. by C. Stephanidis. Springer Berlin Heidelberg, 2007. Vol. 4555 of Lecture Notes in Computer Science. 633–639 с.

5. Incorporating Feedback from Multiple Sensory Modalities Enhances Brain–Machine Interface Control / A. J. Suminski, D. C. Tkach, A. H. Fagg et al. // The Journal of Neuroscience. 2010. Vol. 30, no. 50. 777 – 787 с.

6. Dix A., Finlay J., Abowd G.D., Beale R. Human-Computer Interaction. — Third Edition, Pearson Education Limited: 2004. — 857 с.

7. Bader T., Beyerer J. Natural Gaze Behavior as Input Modality for Human-Computer Interaction // Eye Gaze in Intelligent User Interfaces / Ed. by Y. I. Nakano, C. Conati, T. Bader. Springer London, 2013. 161– 183 с.

8. Leap Motion Controller. URL: https://www.leapmotion.com/ (дата обращения: 22.10.2013)

9. Jacob R. J. K. Eye Movement-Based Human-Computer Interaction Techniques: Toward Non-Command Interfaces // In advances in humancomputer interaction. Ablex Publishing Co, 1993. 151–190 с.

10. Steichen B., Carenini G., Conati C. User-adaptive Information Visualization: Using Eye Gaze Data to Infer Visualization Tasks and User Cognitive Abilities // Proceedings of the 2013 International Conference on Intelligent User Interfaces. IUI ’13. New York, NY, USA: ACM, 2013. 317–328 с.

11. Шапиро Л., Стокман Дж. Компьютерное зрение; пер. С англ. — М.: Бином. Лаборатория знаний, 2009. — с.752

12. Kinect Rehabilitation with Biofeedback. URL: http://www.virtual-realityrehabilitation.com

13. Controller-free gaming with Xbox 360 + Kinect. URL: http://www.xbox.com/enUS/kinect

14. Samsung Smart TV 2013 with Smart Interaction. URL: http://www.samsung.com/us/2013-smart-tv (дата обращения: 23.11.2020)

15. Freeman W.T., Anderson D.B., Beardsley P., Dodge C.N., Roth M., Weissman C.D., Yerazunis W.S., Kage H., Kyuma K., Miyake Y., Tanaka K.I. Computer vision for interactive computer graphics // Computer Graphics and Applications, IEEE , vol.18, no.3, 1998. — 42 – 53 с.

16. Interactive Virtual Aerobics Trainer. URL: http://www.cse.ohiostate.edu/~jwdavis/CVL/Research/VirtualAerobics/aerobics.html

17. Bobick A., Intille S., Davis J., Baird F., Pinhanez C., Campbell L., Ivanov Y., Schutte A., Wilson A. The Kidsroom: A Perceptually-Based Interactive and Immersive Story Environment // Presence: Teleoperators and Virtual Environments, 8(4), 1999. — 367-391 с.

18. Freeman W.T., Roth M. Orientation Histograms for Hand Gesture Recognition // In International Workshop on Automatic Face and Gesture Recognition, 1994. — 296 – 301 с.

19. Zhou H., Lin D.J., Huang T.S. Static hand gesture recognition based on local orientation histogram feature distribution model // in Proc. Conference on Computer Vision and Pattern Recognition Workshop, vol. 10, 2004. —161–169 с.

20. Wang R.Y., Popovi´c J. Real-time hand-tracking with a color glove // ACM Trans. Graph., Volume 28, Number 3, ACM: 2009. — 63 – 65 с.

21. Theobalt C., Albrecht I., Haber J., Magnor M., Seidel. Pitching a baseball – tracking highspeed motion with multi-exposure images // ACM Transactions on Graphics 23, 3, 2004. — 540–547 с.

22. Dorner B. Chasing the Colour Glove: Visual Hand Tracking // Master's thesis, Simon Fraser University, 1994

23. Pansare J.R., Gawande S.H., Ingle M. Real-Time Static Hand Gesture Recognition for American Sign Language (ASL) in Complex Background // Journal of Signal and Information Processing, Volume 3, Number 3, 2012. — 364-367 с.

24. Shaowei C, Jiro T. Hand gesture for taking self-portrait // Proceedings of the 14th international conference on Human-computer interaction: interaction techniques and environments - Part 2, Springer-Verlag: 2011. — 238-247 с.

25. Smelyakov K., Chupryna A., Kolisnyk M., Ponomarenko O. Search by Image Engine Using Local Feature Detectors // 2020 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), 30 April 2020, Vilnius, Lithuania. – P. 1-4.

26. Smelyakov K., Datsenko A., Skrypka V., Akhundov A. Efficiency of Image Reduction Algorithms with Small-Sized and Linear Details // 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), 8-11 Oct. 2019, Kyiv, Ukraine. – P. 745-750.

27. Microsoft Kinect for Windows. URL: http://www.microsoft.com/enus/kinectforwindows/ (дата обращения: 22.10.2013)