

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»  
Навчально-науковий інститут прикладного системного аналізу  
Кафедра системного проектування**

До захисту допущено:

Завідувач кафедри

\_\_\_\_\_ Вадим МУХІН

«\_\_» \_\_\_\_\_ 2023 р.

## **Магістерська дисертація**

**на здобуття ступеня магістра**

**за освітньо-науковою програмою**

**“Інтелектуальні сервіс-орієнтовані розподілені обчислювання”**

**зі спеціальності 122 "Комп'ютерні науки"**

**на тему: «Реконструкція 3D моделі предмета у руці людини на основі  
даних RGB-D камери та апроксимація її геометрії просторовими  
примітивами»**

Виконав (-ла):

студент (-ка) II курсу, групи ДА-11мн

Вовк Ірина Сергіївна \_\_\_\_\_

Керівник:

к.т.н., доцент

Харченко Костянтин Васильович \_\_\_\_\_

Консультант з Розробка стартап проекту:

к.т.н., доцент

Харченко Костянтин Васильович \_\_\_\_\_

Рецензент:

д.т.н. Аушева Н.М. \_\_\_\_\_

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць  
інших авторів без відповідних  
посилань.

Студент (-ка) \_\_\_\_\_

Київ – 2023

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Навчально-науковий інститут прикладного системного аналізу**  
**Кафедра системного проектування**

Рівень вищої освіти – другий (магістерський)

Спеціальність – 122 "Комп'ютерні науки"

Освітньо-професійна програма – "Інтелектуальні сервіс-орієнтовані розподілені обчислення"

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Вадим МУХІН  
\_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**  
**на магістерську дисертацію студенту**  
**Вовк Ірині Сергіївні**

1. Тема дисертації «Реконструкція 3D моделі предмета у руці людини на основі даних RGB-D камери та апроксимація її геометрії просторовими примітивами», науковий керівник дисертації Харченко Костянтин Васильович, к.т.н., доцент, затверджені наказом по університету від №1343-с від 28.03.23
2. Термін подання студентом дисертації – 23 травня 2023 р.
3. Об'єкт дослідження “Алгоритм реконструкції та апроксимації 3D об'єктів”
4. Вихідні дані: Алгоритм для реконструкція 3D моделі предмета у руці людини на основі даних RGB-D камери та апроксимація її геометрії просторовими примітивами
5. Перелік завдань, які потрібно розробити
  1. Огляд існуючих методів для реконструкція 3D моделі предмета та апроксимація її геометрії просторовими примітивами
  2. Розробка алгоритму реконструкція 3D моделі предмета у руці людини на основі даних RGB-D камери та апроксимація її геометрії просторовими примітивами
  3. Тестування розробленого алгоритму на предметах різної геометричної форми

## 6. Орієнтовний перелік графічного (ілюстративного) матеріалу

1. Електронна презентація

## 7. Орієнтовний перелік публікацій

1. Алгоритм розрахунку точок для захвату об'єкта маніпулятором на основі апроксимації 3D моделі об'єкта просторовими примітивами. / Харченко К.В., Вовк І.С.// Системні науки та інформатика: збірник доповідей I науково-практичної конференції «Системні науки та інформатика», 22–29 листопада 2022 року, Київ. – К., НН ІПСА КПІ ім. Ігоря Сікорського, 2022. – с. 375-376.

## 8. Консультанти розділів дисертації<sup>1\*</sup>

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

## 9. Дата видачі завдання – 06 лютого 2023 р.

### Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Огляд літератури за темою магістерської дисертації	6 грудня 2023	
2.	Аналіз існуючих методів реконструкції 3D моделі об'єкта та аналіз методів апроксимації об'єкта просторовими примітивами.	12 грудня 2023	
3.	Розробка алгоритма реконструкції 3D моделі об'єкта та апроксимації його геометрії просторовими примітивами.	15 березня 2023	
4.	Імплементация алгоритма	5 квітня 2023	
5.	Тестування алгоритма на об'єктах різної геометричної форми.	15 квітня 2023	
6.	Дослідження параметрів алгоритма	25 квітня 2023	
7.	Написання тексту дипломної роботи	15 травня 2023	
8.	Захитм дипломної роботи	23 травня 2023	

Студент

Ірина Сергіївна ВОВК

Науковий керівник

Костянтин Васильович ХАРЧЕНКО

## РЕФЕРАТ

магістерської дисертації Вовк Ірини Сергіївни на тему

«Реконструкція 3D моделі предмета у руці людини на основі даних RGB-D камери та апроксимація її геометрії просторовими примітивами»

Робота виконана на 96 сторінках, містить 39 ілюстрацій, 24 таблиці. При підготовці використовувалась література з 23 джерел.

**Актуальність теми.** З часом потреба у взаємодії людини з комп'ютерними системами зростає, а тим самим зростає і потреба у простих та зручних способах такої взаємодії. У даній магістерській дипломній роботі пропонується алгоритм створення 3D моделі об'єкта під час взаємодії з ним та апроксимації геометрії цього об'єкта просторовими примітивами. Цей алгоритм є актуальним для застосування у багатьох сферах: від робототехніки до систем віртуальної та доповненої реальності.

**Мета та задачі дослідження.** Метою даної магістерської дипломної роботи є розробка алгоритму реконструкції 3D моделі предмета у руці людини на основі даних RGB-D камери та апроксимація її геометрії просторовими примітивами.

**Об'єкт досліджень.** Алгоритми реконструкції 3D моделі об'єкта та апроксимації його геометрії просторовими примітивами.

**Предмет досліджень** - Створення алгоритма реконструкції 3D моделі об'єкта та апроксимації його геометрії просторовими примітивами. З використання RGB-D даних, отриманих у процесі маніпуляції людиною об'єктом у руках.

**Методи досліджень.** Для розробки алгоритму, представленого у даній магістерській дипломній роботі було використано алгоритми комп'ютерного зору, методи оптимізації.

**Наукова новизна.** Розроблений у даній магістерській дипломній роботі алгоритм є новим. Він забезпечує сканування предмета з використанням мінімальної кількості обладнання та виключає потребу у виконанні складних маніпуляцій з об'єктом. Представлений алгоритм є детермінованим, що виключає потребу збору даних та навчання. Також, запропонований алгоритм апроксимації 3D моделі об'єкта просторовими примітивами є легко масштабованим.

Потенційні застосування та практична цінність результатів дипломної роботи:

1. Автоматизація та робототехніка
2. Системи доповненої та віртуальної реальності
3. Аугментація предметів та 3D друк

### **Публікації**

1. Алгоритм розрахунку точок для захвату об'єкта маніпулятором на основі апроксимації 3D моделі об'єкта просторовими примітивами. / Харченко К.В., Вовк І.С.// Системні науки та інформатика: збірник доповідей I науково-практичної конференції «Системні науки та інформатика», 22–29 листопада 2022 року, Київ. – К., НН ІПСА КПІ ім. Ігоря Сікорського, 2022. – с. 375-376.

**Ключові слова:** Реконструкція, апроксимація, 3D модель, маніпулювання об'єктом у руках, просторові примітиви.

# ABSTRACT

Vovk Iryna Serhiivna master's thesis on the topic

«In hand object reconstruction and spatial primitives approximation, based on RGB-D camera data»

The work is completed on 96 pages, contains 39 illustrations, 24 tables. The literature from 24 sources was used in the thesis preparation.

**Relevance of the topic.** Over time, the need for human interaction with computer systems is growing, and thus the need for simple and convenient ways of such interaction is growing. This master's thesis proposes an algorithm for creating a 3D model of an object during interaction with it and approximating the geometry of this object with spatial primitives. This algorithm is relevant for use in many areas: from robotics to virtual and augmented reality systems.

**The purpose and objectives of the study.** The purpose of this master's thesis is to develop an algorithm for reconstructing a 3D model of an object in a human hand based on RGB-D camera data and approximating its geometry with spatial primitives.

**Object of research.** Algorithms for reconstructing a 3D model of an object and approximating its geometry with spatial primitives.

**The subject of research** is the creation of an algorithm for reconstructing a 3D model of an object and approximating its geometry with spatial primitives. Using RGB-D data obtained in the process of manipulating an object in the hands of a person.

**Research methods.** Computer vision algorithms and optimization methods were used to develop the algorithm presented in this master's thesis.

**Scientific novelty.** The algorithm developed in this master's thesis is new. It provides scanning of an object using a minimum amount of equipment and eliminates the need to perform complex manipulations with the object. The presented algorithm is deterministic, which eliminates the need for dataset collection and training. Also, the proposed algorithm for approximating a 3D object model with spatial primitives is easily scalable.

Potential applications and practical value of the thesis results:

1. Automation and robotics
2. Augmented and virtual reality systems

### 3. Object augmentation and 3D printing

#### **Publications.**

Algorithm for calculating points for capturing an object by a manipulator based on approximation of a 3D model of an object by spatial primitives / Kharchenko K.V., Vovk I.S. // System sciences and informatics: collection of reports of the I scientific and practical conference "System sciences and informatics", November 22-29, 2022, Kyiv - K., Igor Sikorsky Kyiv Polytechnic Institute, 2022. - pp. 375-376.

**Keywords:** Reconstruction, approximation, 3D model, manipulation of an object in hands, spatial primitives.

## ЗМІСТ

ВСТУП . . . . .	11
1 ЗАДАЧА РЕКОНСТРУКЦІЇ 3D МОДЕЛІ ПРЕДМЕТА У РУЦІ ЛЮДИНИ НА ОСНОВІ ДАНИХ RGB-D КАМЕРИ ТА АПРОКСИМАЦІЯ ЇЇ ГЕОМЕТРІЇ ПРОСТОРОВИМИ ПРИМІТИВАМИ . . . . .	13
1.1 Постановка задачі реконструкції 3D моделі предмета у руці людини на основі даних RGB-D камери та апроксимація її геометрії просторовими примітивами . . . . .	13
1.2 Огляд існуючих рішень для вирішення задачі реконструкції 3D моделі предмета у руці людини та апроксимація її геометрії просторовими примітивами . . . . .	14
2 ЗБІР ДАНИХ ЗА ДОПОМОГОЮ RGB-D КАМЕРИ . . . . .	17
2.1 Опис використаної камери та формату даних . . . . .	17
3 СЕГМЕНТАЦІЯ ПРЕДМЕТА У РУЦІ . . . . .	20
3.1 Сегментація предметів на фоні . . . . .	20
3.2 Сегментація руки . . . . .	21
3.2.1 Виділення скелетону руки на 2D зображенні . . . . .	21
3.2.2 Розрахунок нормалей у неупорядкованій хмарі точок . . . . .	23
3.2.3 Алгоритм пошуку сусідніх точок . . . . .	25
3.2.4 Використання методу аналізу головних компонент для розрахунку нормалей . . . . .	26
3.2.5 Надсегментація супер вокселями . . . . .	27
3.2.6 Виділення сегментів у неогранізованій хмарі точок, на основі локальних випуклостей . . . . .	31
3.2.7 Побудова графа зв'язності для визначення локальних випуклостей . . . . .	33
3.2.8 Сегментація руки у неорганізованій хмарі точок . . . . .	35
3.2.9 Уточнення сегмента 3D моделі об'єкта . . . . .	37

4	РЕКОНСТРУКЦІЯ 3D МОДЕЛІ ОБ'ЄКТА . . . . .	39
4.1	Проекція сегмента 3D моделі на 2D площину камери . . . . .	39
4.2	Виділення ключових точок на 2D зображенні сегмента предмета . . . . .	41
4.3	Пошук відповідних пар ключових точок . . . . .	44
4.4	Виділення ключових точок сегмента в 3D . . . . .	46
4.5	Суміщення ключових точок в 3D просторі . . . . .	47
4.6	Уточнення 3D моделі об'єкта . . . . .	50
5	АПРОКСИМАЦІЯ 3D МОДЕЛІ ОБ'ЄКТА ПРОСТОРОВИМИ ПРИМІТИВАМИ . . . . .	53
5.1	Вибір множини просторових примітивів, для апроксимації 3D моделі об'єкта . . . . .	53
5.2	Алгоритм апроксимації 3D моделі об'єкта просторовими примітивами . . . . .	55
5.3	Кластеризація моделі 3D об'єкта . . . . .	57
6	ОПИС ПРОГРАМНОГО ПРОДУКТУ . . . . .	59
6.1	Опис використаних технічних засобів . . . . .	59
6.2	Опис результатів роботи програмного продукту . . . . .	62
7	РЕЗУЛЬТАТИ РОБОТИ АЛГОРИТМА . . . . .	65
8	ПРАКТИЧНЕ ЗАСТОСУВАННЯ РЕЗУЛЬТАТІВ РЕКОНСТРУКЦІЇ ТА АПРОКСИМАЦІЇ ОБ'ЄКТА ПРОСТОРОВИМИ ПРИМІТИВАМИ . . . . .	68
8.1	Використання результату 3D реконструкції об'єкта . . . . .	68
8.2	Використання результату апроксимації 3D об'єкта просторовими примітивами . . . . .	70
9	ПАРАМЕТРИ, ЯКІ ВПЛИВАЮТЬ НА ЯКІСТЬ РЕКОНСТРУКЦІЇ 3D МОДЕЛІ ОБ'ЄКТА . . . . .	73
9.1	Дослідження максимального кута повороту предмета, достатнього для реконструкції 3D моделі об'єкта . . . . .	73

10 РОЗРОБКА СТАРТАП ПРОЕКТУ “РЕКОНСТРУКЦІЯ 3D МОДЕЛІ ПРЕДМЕТА У РУЦІ ЛЮДИНИ НА ОСНОВІ ДАНИХ RGB-D КАМЕРИ ТА АПРОКСИМАЦІЯ ЇЇ ГЕОМЕТРІЇ ПРОСТОРОВИМИ ПРИМІТИВАМИ” . . . . .	77
10.1 Опис ідеї проекту . . . . .	77
10.2 Технологічний аудит ідеї . . . . .	79
10.2.1 Аналіз ринкових можливостей . . . . .	80
10.3 Розробка ринкової стратегії проекту . . . . .	86
10.3.1 Розробка маркетингової програми . . . . .	89
ВИСНОВКИ . . . . .	92
ПЕРЕЛІК ПОСИЛАНЬ . . . . .	94

## ВСТУП

В останні роки в галузі комп'ютерного зору стався неймовірний прогрес. Це сталось значною мірою завдяки поширенню доступних і універсальних камер, що маєть вбудовані сенсори для вимірювання глибини, а також розробці потужних обчислювальних фреймворків та алгоритмів. Постійний попит на досягнення в галузі автоматизації, робототехніки та комп'ютерного зору призвів до зростання інтересу до реконструкції 3D-моделей об'єктів, а також спрощеного їх представлення. Можливість точного моделювання та представлення об'єктів у тривимірному просторі має широкий спектр застосувань, від промислової автоматизації до віртуальної та доповненої реальності, і сфери розваг. Ця магістерська дипломна робота присвячена реконструкції 3D-моделі об'єкта в руці людини під час маніпуляцій та подальшій апроксимації його геометрії за допомогою просторових примітивів.

Однією з основних задач реконструкції 3D-об'єктів є отримання точних і повних даних, що в даному випадку ускладнюється коли в процесі зобру даних об'єктом маніпулює людина. Об'єкти обертаються, переміщуються та іншим чином трансформуються в руках, тому стає все важче отримати послідовну та вичерпну інформацію про їхню геометрію. Ця проблема ще більше ускладнюється тим, що об'єкти можуть бути частково або повністю закриті рукою людини, що вносить додаткову невизначеність і складність.

У контексті цих проблем центральною метою цієї магістерської роботи є розробка надійного та ефективного методу реконструкції 3D-моделі об'єкта, яким маніпулює людина в руці, використовуючи дані, отримані з RGB-D камери. RGB-D камери надають інформацію про колір і глибину. Це дозволяє отримувати повноцінні 3D-дані, які можна потім обробляти для створення детальних моделей об'єктів. Крім того, в цій роботі досліджується апроксимація реконструйованої 3D-моделі просторовими примітивами, такими як площини, сфери, циліндри та кубоїди. Просторові примітиви забезпечують спрощене представлення складної геометрії, що робить їх гарним вибором для різноманітних застосунків. Апроксимуючи 3D-модель за допомогою просторових примітивів, отримане представ-

лення може бути більш компактним, обчислювально ефективним та інтуїтивно зрозумілим для людського сприйняття. Це полегшує інтеграцію з іншими системами, такими як засоби автоматизованого проектування (САПР), або, наприклад, алгоритми керування роботами.

Таким чином, магістерська дисертація робить внесок в область 3D реконструкції об'єктів і спрощення представлення їх геометрії, пропонуючи новий підхід, який вирішує проблеми, пов'язані з реконструкцією об'єктів, що знаходяться в руці людини. Результати цього дослідження потенційно можуть вплинути на іришення різноманітних задач, таких як роботизоване захоплення та маніпулювання роботизовани маніпулятором, розпізнавання об'єктів, віртуальна та доповнена реальність та багато інших. Знання, отримані в результаті цього дослідження, також можуть слугувати основою для майбутніх досліджень, вивчення нових способів покращення та розширення можливостей 3D-реконструкції та методів просторової апроксимації.

# 1 ЗАДАЧА РЕКОНСТРУКЦІЇ 3D МОДЕЛІ ПРЕДМЕТА У РУЦІ ЛЮДИНИ НА ОСНОВІ ДАНИХ RGB-D КАМЕРИ ТА АПРОКСИМАЦІЯ ЇЇ ГЕОМЕТРІЇ ПРОСТОРОВИМИ ПРИМІТИВАМИ

1.1 Постановка задачі реконструкції 3D моделі предмета у руці людини на основі даних RGB-D камери та апроксимація її геометрії просторовими примітивами

Метою даного дослідження є розробка алгоритму реконструкції 3D-моделей об'єктів, якими маніпулює людина перед RGB-D камерою, та апроксимації їх форми за допомогою просторових примітивів. Цю задачу можна розділити на дві основні підзадачі:

*Реконструкція об'єкта в руці:* Реконструювати 3D-модель об'єкта, яким маніпулює людина перед RGB-D камерою, використовуючи отримані RGB-зображення та карти глибини.

$F$  - скінченна множина елементів  $f_i$ , де кожен  $f_i$  є сукупністю RGB-зображення та карти глибини, представленою у вигляді хмари точок. Для кожного  $f_i \in F$  необхідно знайти  $s_i \subset f_i$ , шляхом відкидання точок, що не належать до об'єкту, яким маніпулюють.

Для кожної пари  $s_i$  та  $s_{i+1}$  необхідно знайти таку матрицю трансформації  $T_i$  при застосуванні якої до елемента  $s_{i+1}$  отримуємо сегмент  $s_{i+1}$  суміщений до елемента  $s_i$ . Таким чином  $S = \bigcup_{i=1}^{|F|} s_i$  - 3D модель об'єкта, яким маніпулюють.

*Апроксимація просторовими примітивами:* Виконати апроксимацію реконструйованої 3D-моделі за допомогою просторових примітивів, таких як площини, сфери, циліндри або кубоїди, щоб створити спрощене представлення форми об'єкта.

Нехай  $P$  - сімейство підмножин  $R^3$ , де кожна підмножина представляє собою опуклий просторовий примітив. Необхідно знайти таку скінченну підмножину  $\hat{P} \subset P$ , що є розв'язком задачі мінімізації  $\operatorname{argmin}_{\hat{P}} (\|\bigcup_{A \in \hat{P}} A - S\|)$ .

## 1.2 Огляд існуючих рішень для вирішення задачі реконструкції 3D моделі предмета у руці людини та апроксимація її геометрії просторовими примітивами

У цьому розділі представлено огляд існуючих рішень, які вирішують проблему реконструкції 3D-моделі об'єкта в руці людини. Цей огляд літератури має на меті сформулювати комплексне розуміння поточного стану досліджень даної задачі. Розглянувши сильні та слабкі сторони цих підходів, ми зможемо краще оцінити важливість запропонованого у цій магістерській роботі рішення та його внесок у цю галузь досліджень.

У статті "Сканування об'єктів вручну за допомогою сегментації RGB-D відео" (In-hand Object Scanning via RGB-D Video Segmentation) автори Фан Ван та Кріс Хаузер представили новий підхід до сканування 3D-об'єктів за допомогою RGB-D камери [1]. Основна увага в роботі приділяється вирішенню проблем реконструкції 3D-моделі об'єкта під час маніпуляцій з ним в руці. Підхід, представлений у статті, використовує переваги процесу маніпуляції для захоплення поверхні об'єкта під різними кутами, забезпечуючи більш повне і точне представлення. Запропонований метод спочатку виконує попередню обробку RGB-D даних для покращення їхньої якості, після чого відбувається процес сегментації рук з використанням моделі кольору шкіри для відокремлення руки від об'єкта у вхідних даних. Потім алгоритм використовує обмеження часової узгодженості для уточнення сегментації об'єкта, гарантуючи, що сегментовані області є узгодженими на декількох кадрах. Після цього для суміщення поверхонь об'єктів на різних кадрах застосовуються методи реєстрації поверхонь, які поєднують підходи на основі ознак та ітеративного методу найближчої точки (ICP).

Метод має суттєвий недолік, що заключається у використанні моделі кольору шкіри для сегментації рук на фреймі. Такий підхід не буде працювати за умов поганого освітлення, та має обмеження за кольором шкіри руки людини, що маніпулює об'єктом. Запропонований у даній магістерській роботі алгоритм не має таких обмежень. Він не використовує інформацію про колір шкіри для

сегментації рук, а спирається на пошук скелетона руки на зображенні.

Іншою цікавою роботою є стаття Яни Хассон та Гюля Варола "Навчання спільної реконструкції рук та об'єктів, якими маніпулюють"(Learning joint reconstruction of hands and manipulated objects) представляє підхід на основі глибокого навчання для сумісного відтворення 3D-моделі людської руки та об'єкта, яким вона маніпулює, за допомогою RGB-зображень [2]. Автори ставлять за мету вирішити проблеми, пов'язані з оклюзією, а також складною взаємодією між рукою та об'єктом, використовуючи методи глибокого навчання.

Запропонований метод ґрунтується на архітектурі глибокого навчання, яка включає два ключові компоненти: алгоритм оцінки положення руки-об'єкта та алгоритм реконструкції поверхні мешу. Алгоритм оцінки положення руки-об'єкта, побудований на основі згорткової нейронної мережі (CNN), прогнозує тривимірні положення руки та об'єкта на основі вхідного RGB-зображення. Алгоритм реконструкції поверхні мешу, побудований на основі графової згорткової мережі (GCN), уточнює початкові оцінки положення і генерує остаточні 3D-меш-моделі для руки та об'єкта [4]. Для навчання моделі автори створили синтетичний набір даних за допомогою симулятора взаємодії руки з об'єктом. Цей набір даних містить достовірні 3D-пози руки та об'єкта, а також відповідні їм RGB-зображення. Автори також демонструють, що їхній метод можна точно налаштувати, використовуючи реальні RGB-зображення з анотованими позами людини та об'єкта.

Запропонований, у описаній вище роботі, алгоритм має вагому перевагу - може обробляти вхідні RGB-зображення, не вимагаючи даних про глибину, Це підвищує його універсальність і застосовність у реальних сценаріях.

Однак у запропонованого методу є деякі обмеження. По-перше, використання синтетичного набору даних для навчання може не повністю відобразити складність реальних взаємодій між людиною та об'єктом, а процес анотації реальних зображень вимагає збору великого датасету та затрати значної кількості часу на розмітку. Також такий підхід є обмежений набором положень руки, на яких почалась мережа. Алгоритм, що пропонується у даній дипломній роботі не

потребує збору датасету та навчання взагалі, оскільки є детермінованим. Він не обмежений у положенні руки та предмета у ній.

## 2 ЗБІР ДАНИХ ЗА ДОПОМОГОЮ RGB-D КАМЕРИ

### 2.1 Опис використаної камери та формату даних

Для задачі реконструкції 3D моделі в руці людини та апроксимація її геометрії просторовими примітивами було використано камеру Intel RealSense D435. Зображено на Рис.2.1.



Рисунок. 2.1 – Intel RealSense D435 основні сенсори

Intel RealSense D435 - це камера з датчиком глибини, призначена для використання в системах комп'ютерного зору [22]. Камера оснащена інфрачервоним проектором і двома інфрачервоними камерами, які збирають дані про глибину за допомогою технології структурованого світла. Вона також має RGB-камеру з високою роздільною здатністю для зйомки кольорових зображень. Основні технічні характеристики наведено в Табл.2.1

Таблиця 2.1 – Перелік технічних характеристик Intel RealSense D435

Характеристика	Референтні значення
Мінімальна відстань захвату даних	0.3 м
Максимальна відстань захвату даних	3 м
FOV камери глибини	87° × 58°
FOV RGB глибини	69° × 42°
Точність вимірів глибини	<2% на 2 м
Максимальна роздільна здатність RGB камери	1920 × 1080
Максимальна роздільна здатність камери глибини	1280 × 720

Для збору даних камера була встановлена статично на штативі. Дані були зібрані шляхом маніпуляції( в даному випадку комбінації поворотів, нахилів переміщень) предмета у руці людини. На Рис.2.2 зображено приклад RGB зображення одного з об'єктів датасету під час маніпулювання. На Рис.2.3 наведено відповідну хмару точок.



Рисунок. 2.2 – Приклад RGB зображення, отриманого у процесі збору датасета.

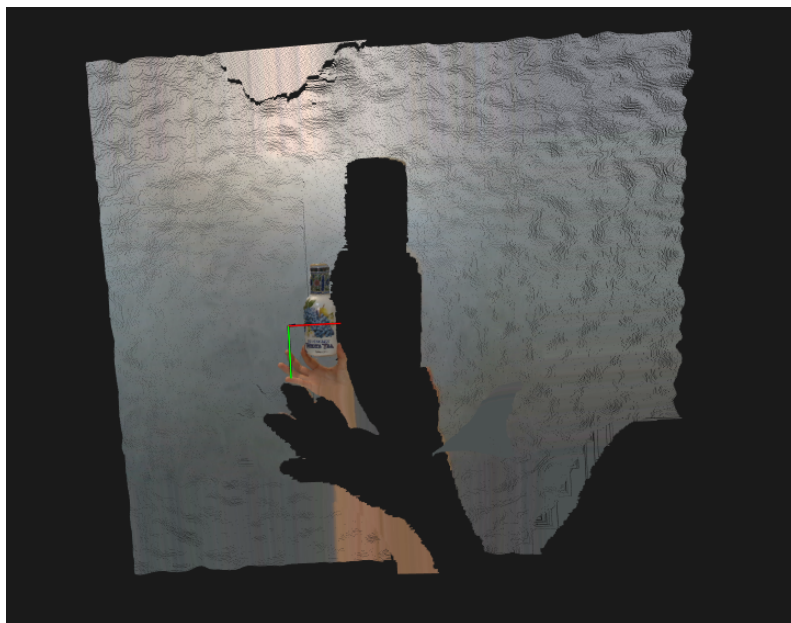


Рисунок. 2.3 – Хмара точок, отримана у процесі збору датасета. Відповідає RGB зображення на Рис. 2.2

Зібраний датасет містить набір 8-бітних кольорових зображень з роздільною здатністю  $640 \times 480$  та відповідні їм кольорові хмари точок, утворені за допомогою 8-бітного кольорового зображення, та 16-бітного зображення, отриманого з сенсора глибини.

### 3 СЕГМЕНТАЦІЯ ПРЕДМЕТА У РУЦІ

Для того, щоб виконати 3D реконструкцію предмета, яким маніпулювали потрібно відділити його точки від точок руки та інших предметів, що потрапили у кадр, тобто виконати сегментацію фону, руки та безпосередньо самого предмета. Розглянемо кожен з цих етапів більш детально.

#### 3.1 Сегментація предметів на фоні

У процесі збору датасету і поле зору камери потрапляє не тільки рука і предмет в ній, а й інші частини інтер'єру: меблі позаду, стеля, стіни, тощо. Такі елементи хмари точок не несуть жодної інформації для даного алгоритму та мають бути відкинутими на почтку роботи з даними для зменшення їх об'єму, а отже і часових затрат аналіз [5]. Для того, щоб відкинути об'єкти, що знаходяться на занадто великій або занадто малій відстані від камери використовується метод відсікання площиною. Він проілюстрований на Рис. 3.1.

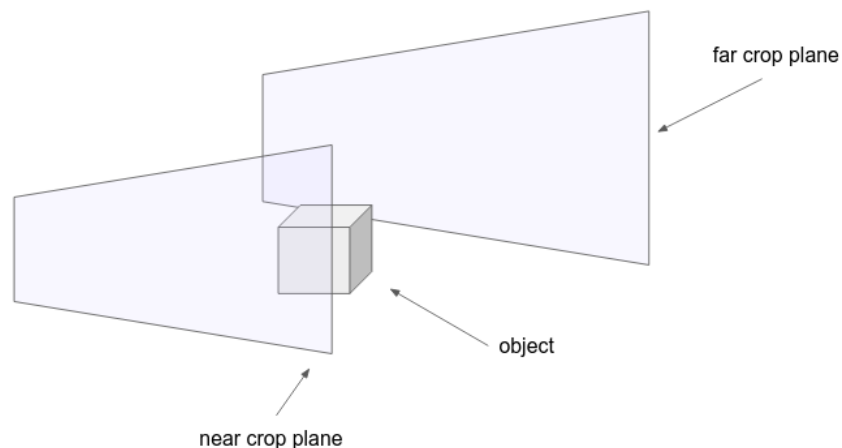


Рисунок. 3.1 – Схематичне зображення методу відсікання площиною

Результат роботи цього алгоритму, застосовно до хмари точок, що була зображена на 2.3 показано на Рис. 3.2.

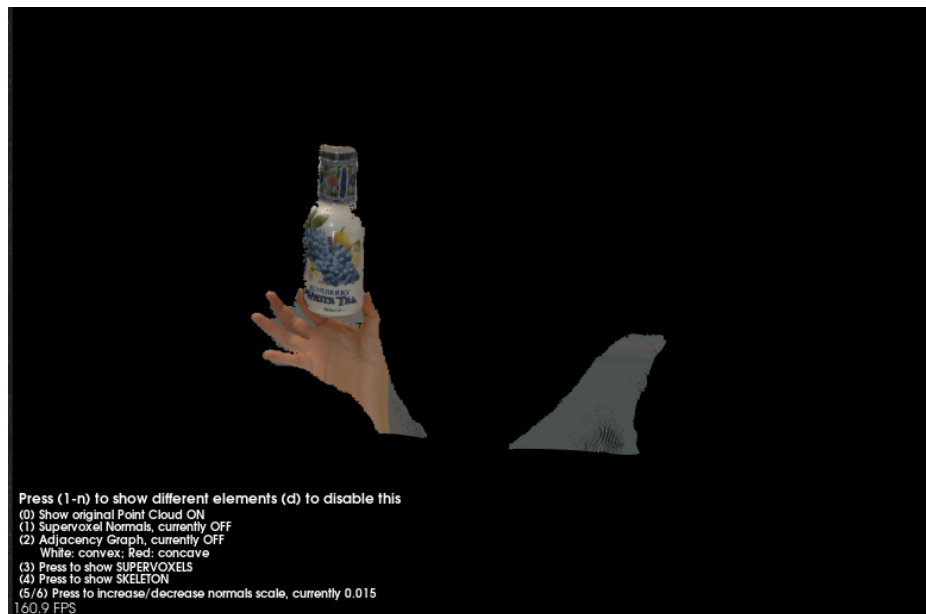


Рисунок. 3.2 – Результат роботи алгоритму відсікання площини

Таким чином, на даному етапі ми отримали хмару точок, що містить руку та об'єкт, яким маніпулювали.

## 3.2 Сегментація руки

Алгоритм сегментації руки включає в себе наступні етапи: виділення скелетону руки на 2D зображенні, розрахунок нормалей у в хмарі точок, надсегментація хмари точок сепурвокселями, сегментація хмари точок на основі локальних опуклостей. Далі у цьому розділі розглянемо більш детально кожен з етапів.

### 3.2.1 Виділення скелетону руки на 2D зображенні

Для виділення скелетону було використано MediaPipe hands.

MediaPipe Hands - це система на основі машинного навчання для виявлення та відстеження ключових точок рук у реальному часі за допомогою камери. Розроблена компанією Google, вона є частиною більш широкого фреймворку MediaPipe, який надає розробникам набір готових компонентів для створення додатків на основі машинного навчання [21].

MediaPipe Hands використовує глибоку нейронну мережу для виявлення та відстеження ключових точок на руках (зображені на Рис. 3.3). Система здатна

визначати 21 ключову точку на кожній руці, включаючи кінчики кожного пальця, основу великого пальця та центр долоні. Ці орієнтири можуть бути використані для оцінки положення, орієнтації та руху рук.

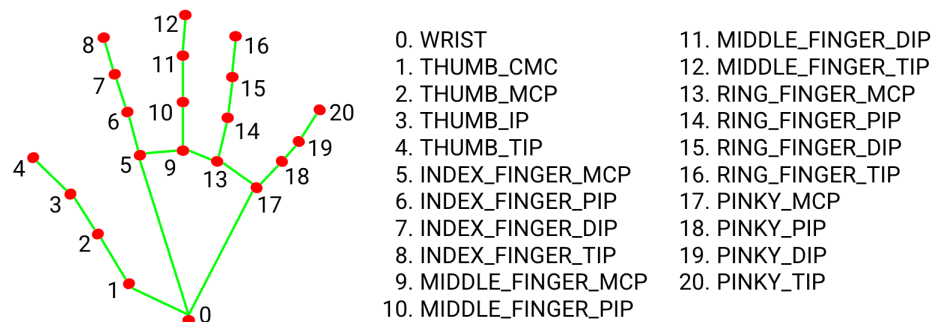


Рисунок. 3.3 – Ключові точки руки у моделі MediaPipe hands

Система визначає ключові точки в 2D, а також їх  $Z$  координату (значення глибини).

Пайплайн MediaPipe hands поєднує декілька моделей:

- Модель детекції долоні. Single shot модель, що працює з повнорозмірним зображенням і потрібна для детекції усієї руки на зображенні, включаючи закриті об'єктами та самозакритті руки. Ця модель повертає орієнтований bounding box у якому надалі буде проводитись пошук ключових точок.
- Модель детекції ключових точок руки. Працює з регіоном зображення, визначеним попередньою моделлю. Потрібна для визначення 21 ключової точки руки. Модель була навчена на 30 тис. анотованих вручну зображень рук за 21 ключовою точкою та  $Z$  координатою. Також для охоплення більшої кількості положень та форм руки до навчальної вибірки було додано синтетично згенеровані дані, відрендерені у високій роздільній здатності.

Такий підхід є схожим до детекції обличчя на 2D зображенні, представленому у продукті MediaPipe Face Mesh.

Модель MediaPipe hands є оптимізованою для роботи з потоком даних. Ключові точки руки на кожному наступному кадрі оновлюються, використовуючи інформацію з попереднього кадру. Це дає змогу не використовувати модель для детекції долоні кожного разу, тільки у випадках, коли на попередніх фреймах ключові точки не були присутніми. Такий підхід дає змогу виконувати детекцію

ключових точок за час, прийнятний для додатків, що працюють в реальному часі. Приклад детекції ключових точок руки за допомогою MediaPipe Hands наведено на Рис. 3.4.



Рисунк. 3.4 – Приклад роботи MediaPipe Hand на одному з зображень, із зібраного датасету.

### 3.2.2 Розрахунок нормалей у невпорядкованій хмарі точок

Задача визначення нормалі до геометричної поверхні, зазвичай, зводиться до визначення напрямку нормалі в певній точці цієї поверхні. Тобто пошуку вектора, що є перпендикулярним до поверхні в цій точці. Однак, оскільки, отримані, на етапі збору датасету, дані являють собою множину точок в просторі, що не лежать на одній площині, а навпаки є хаотично розташованими, нормалі можуть бути розрахованими одним з наступних методів:

1. Отримати базову поверхню з отриманого набору даних, використовуючи методи побудови поверхневої сітки (мешу), а потім обчислити поверхневі нормалі для кожної з площин цієї сітки;
2. Розрахувати апроксимовані нормалі, безпосередньо в кожній точці з набору даних;

В даній роботі використано останній з наведених вище підходів. Такий вибір обумовлено низкою обчислювальною складністю алгоритму апроксимації нормалей, що в свою чергу означає що він буде працювати швидше [8].

Таким чином, задача розрахунку нормалей зводиться до аналізу власних векторів і власних значень коваріаційної матриці, утворених з точок, що є найближчими сусідами до точки в якій розраховується нормаль. Тобто для кожної точки  $p_i$  складається коваріаційна матриця  $C$  наступним чином:

$$C = \frac{1}{k} \sum_{i=1}^k (p_i - \hat{p}) \cdot (p_i - \hat{p})^T, C \cdot \vec{v}_j = \lambda_j \cdot \vec{v}_j, j \in \{1, 2, 3\} \quad (3.1)$$

Де  $k$  – кількість точок, що є сусідами точки  $p_i$ ,

$\hat{p}$  – точка в просторі, що є центроїдом множини точок, утвореної точкою  $p_i$  та її сусідніми точками.

$\lambda_j$  –  $j$ -те власне значення коваріаційної матриці

$\vec{v}_j$  –  $j$ -тий власний вектор

Нормалью до точки  $p_i$  буде такий власний вектор, що відповідає найменшому власному числу. Такий підхід розрахунку нормалей має назву метод головних компонент (PCA). Приклад розрахунку нормалей для однієї з хмар точок наведено на Рис. 3.5.

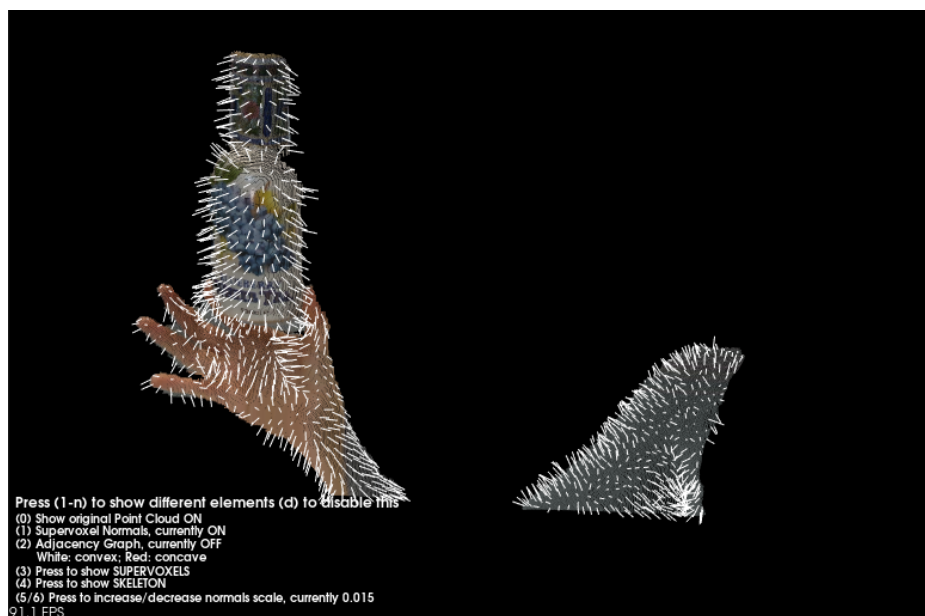


Рисунок. 3.5 – Приклад розрахунку нормалей в одній з хмар точок

Далі розглянемо детальніше алгоритм пошуку сусідніх точок та алгоритм аналізу головних компонент.

### 3.2.3 Алгоритм пошуку сусідніх точок

Для ефективного пошуку сусідніх точок до заданої в невпорядкованій хмарі точок, було використано структуру даних KD-tree. Пошук в такій структурі даних виконується за  $O(\log(n))$  в кращому випадку, та  $O(n)$  - в гіршому.

KD-tree (k-вимірне дерево) - структура даних, що використовується для розбиття простору для організації точок у k-вимірному просторі. Ключова ідея KD-дерева полягає в рекурсивному розбитті точок на підмножини на основі їхніх координат уздовж різних осей і побудові бінарного дерева, в якому кожен вузол представляє підмножину точок, а кожен листовий вузол - одну точку. Це дозволяє ефективно обрізати простір пошуку і зменшити часову складність пошуку з лінійної до логарифмічної [11]. Приклад побудови KD-дерева наведено на Рис.3.6

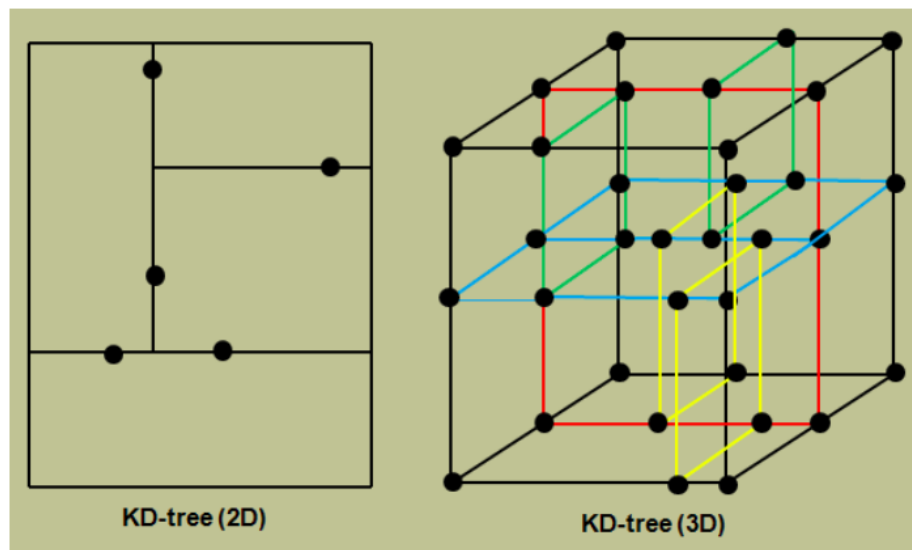


Рисунок. 3.6 – Приклад побудови KD-дерева у 2D та 3D

Процес пошуку найближчих сусідів за допомогою KD-дерева можна узагальнити наступним чином. Маючи хмару точок  $P$  і точку запиту  $q$ , ми спочатку будемо KD-дерево з точок  $P$ . Потім визначаємо метрику відстані між точками, у даному випадку, ми оперуємо точками в тривимірному просторі, тому, можемо взяти за метрику евклідову відстань, і починаємо з кореневого вузла KD-дерева.

Ми рухаємося вниз по дереву, щоб знайти вершину-листок, яка містить точку, найближчу до  $q$ , і оновлюємо найкращу відстань і найкращого сусіда, знайдені на цьому шляху. Потім ми повертаємось вгору по дереву, щоб перевірити, чи немає ближчих точок, які могли бути пропущені під час пошуку, і оновлюємо найкращу відстань і найкращого сусіда відповідно. Нарешті, ми повертаємо знайденого найкращого сусіда як найближчого сусіда  $q$  у хмарі точок.

KD-дерево має декілька переваг над іншими методами пошуку найближчого сусіда, такими як пошук грубою силою та хешування. KD-дерево ефективно для просторів низької розмірності і має логарифмічну часову складність пошуку найближчого сусіда. KD-tree також простий у реалізації і може обробляти великі набори даних з помірним використанням пам'яті.

#### 3.2.4 Використання методу аналізу головних компонент для розрахунку нормалей

Аналіз головних компонент (PCA) – це математичний метод, який використовується для зменшення складності багатовимірних даних шляхом виявлення закономірностей і кореляцій між змінними. Це метод перетворення набору даних, що складається з багатьох змінних, на менший набір змінних, який все ще містить більшу частину інформації вихідного набору даних.

PCA передбачає обчислення коваріаційної матриці вхідних даних, яка описує спільну мінливість змінних у даних. Потім ця матриця розкладається на власні вектори та власні значення, які представляють напрямки максимальної дисперсії та кількість дисперсії, що пояснюється кожним напрямком, відповідно. Власні вектори утворюють новий базис для даних, і дані можуть бути спроектовані на цей новий базис, щоб отримати представлення даних у нижчій розмірності.

Для задачі розрахунку нормалей у хмарі точок, зменшення розмірності даних не стоїть на меті. Достатньо знайти власний вектор, що відповідає найменшому власному числу для підмножини точок, що є сусідніми для заданої.

Таким чином, можемо остаточно сформулювати алгоритм, для пошуку нормалей у неорганізованій хмарі точок.

Дано неорганізовану хмару точок  $P = \{p_1, \dots, p_n\}$ , де кожна  $p_i = (x_i, y_i, z_i)$  є 3D точкою. Знайти  $n_i = (n_{x_i}, n_{y_i}, n_{z_i})$ .

1. Для кожної точки  $p_i$ , знайти точки, що знаходяться в околиці  $r$ , тобто, визначити її  $k$  найближчих сусідів, використовуючи KD-дерево. Позначимо цю підмножину  $P_k$ .
2. Розрахувати матрицю коваріації  $C_k$  для множини  $P_k$ .
3. Виконати декомпозицію матриці  $C_k$ , щоб отримати власні значення та власні вектори.
4. Відсортувати власні вектори у порядку спадання відповідних їм власних значень. Власний вектор з найменшим власним значенням, буде відповідати напрямку найменшої варіації, тобто буде нормаллю для точки  $p_i$ .

### 3.2.5 Надсегментація супер вокселями

Надсегментація супервокселів - це метод, який використовується в обробці хмари точок і 3D-комп'ютерному зорі для сегментації хмари точок на компактні, приблизно сферичні області, які називаються супервокселями [6].

Супервоксель визначається як група сусідніх точок у хмарі точок, які мають схожі геометричні та/або колірні властивості. Групуючи точки зі схожими властивостями у супервокселі, ми можемо зменшити складність хмари точок і отримати більш змістовне представлення сцени.

Надсегментація супервокселів складається з трьох основних кроків:

1. *Вокселізація*. Спочатку хмара точок вокселюється, що передбачає поділ 3D-простору на невеликі кубічні області, які називаються вокселями. Кожній точці в хмарі точок присвоюється воксель, до якого вона належить.
2. *Генерація супервокселів*. Супервокселі генеруються шляхом групування сусідніх вокселів, які мають схожі геометричні та/або колірні властивості. Зазвичай це робиться шляхом обчислення локального дескриптора для кожного вокселя, наприклад, гістограми кольору або нормалі поверхні, і використання цих дескрипторів для кластеризації вокселів у супервокселі. Цей процес кластеризації можна виконати за допомогою таких методів, як кластеризація за середнім зсувом або ієрархічна кластеризація.

3. *Уточнення.* Згенеровані супервокселі можуть містити зашумлені або неправильно класифіковані точки, тому їх уточнюють шляхом забезпечення просторової та часової когерентності. Це передбачає групування сусідніх супервокселів, які належать до одного об'єкта або сцени, та об'єднання супервокселів, які перекриваються або мають схожі властивості.

Надсегментація супервокселів широко використовується в багатьох додатках комп'ютерного зору, таких як розпізнавання об'єктів, розуміння сцен і 3D-реконструкція. Вона дозволяє зменшити складність хмари точок, виділити значущу інформацію на рівні об'єктів і забезпечити ефективну обробку та аналіз даних.

Надсегментація супервокселів має кілька переваг над іншими методами сегментації в обробці хмар точок і 3D-комп'ютерному зорі:

- *Компактність.* Супервокселі - це компактні і приблизно сферичні області, які групують разом сусідні точки зі схожими властивостями, що дозволяє більш ефективно обробляти і зменшує складність хмари точок.

- *Змістовне представлення.* Групуючи точки в супервокселі на основі їх геометричних та/або кольорних властивостей, ми можемо отримати більш змістовне представлення сцени, що може допомогти в розпізнаванні об'єктів, розумінні сцени та 3D-реконструкції.

- *Локальна узгодженість.* Супервокселі генеруються на основі локальної геометрії та кольору точок, що забезпечує локальну узгодженість сегментації. Це означає, що супервокселі фіксують локальні властивості сцени і можуть допомогти зберегти дрібні деталі, які можуть бути втрачені в інших методах сегментації.

- *Часова когерентність.* Надсегментацію супервокселів можна розширити, щоб зафіксувати часову когерентність на декількох кадрах, що забезпечує послідовну сегментацію в часі і може допомогти в таких завданнях, як аналіз і відстеження руху.

- *Гнучкість.* Надсегментація супервокселів може бути адаптована до різних типів даних хмари точок, таких як LiDAR, RGB-D або стерео, і може бути налаштована відповідно до конкретних вимог програми.

У даній дипломній роботі використано підхід, описаний в роботі Jeremie Papou “Voxel Cloud Connectivity Segmentation - Supervoxels for Point Clouds”. В даному випадку, алгоритм надсегментації супервокселів є ітеративним та будується на основі кластеризації К-середніх. Це забезпечує високу просторову зв’язність, що в свою чергу гарантує дуже важливу для даної роботи властивість: супервокселі не можуть перетинати кордони, які є роз’єднаними у 3D просторі, навіть, якщо вони з’єднані в рамках однієї проекційної площини.

Супервокселі є кластерами у 39-мірному просторі, представленим, як

$$F = [x, y, z, L, a, b, FPFH_{1...33}] \quad (3.2)$$

Де  $x, y, z$  – просторові координати ,

$L, a, b$  – колір в CIE Lab просторі

$FPFH_{1...33}$  - 33 елементи

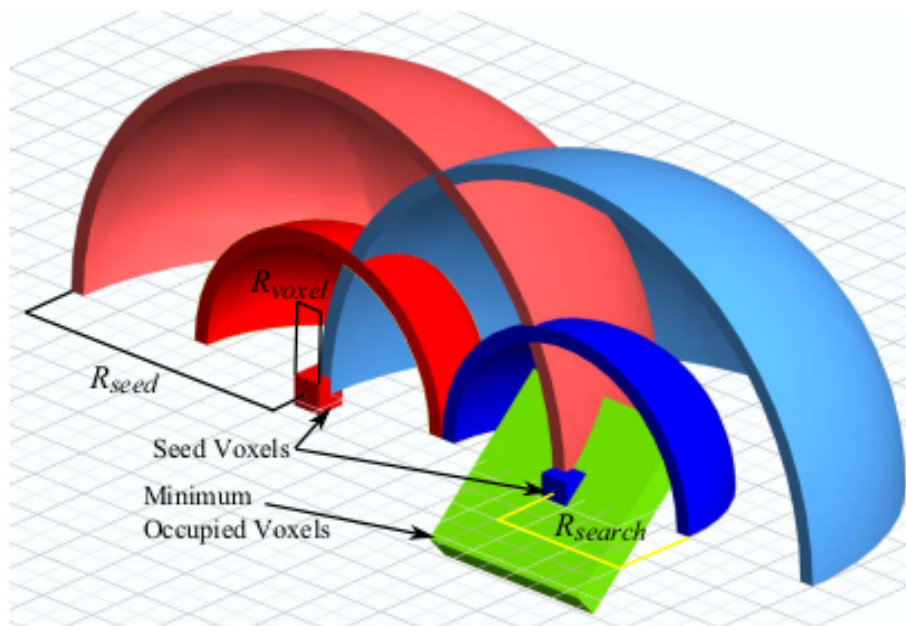


Рисунок. 3.7 – Приклад вокселізації простору для виконання надсегментації супервокселями

Fast Point Feature Histograms - алгоритм для виділення локальних ключових признаков, запропонований Rusu et al. Він обчислює гістограму геометричних характеристик для кожної точки в хмарі точок на основі відносного положення сусідніх точок і нормалей поверхні [19]. FPFH є модифікацією дескриптора Point

Feature Histogram (PFH), що покращує його ефективність та стійкість до шуму та викидів. FPFH має високу дискримінативність, обчислювальну ефективність і може бути адаптований до різних типів даних хмари точок.

Для виконання надсегментації потрібно визначити мінімальний радіус вокселя  $R_{voxel}$  та розбити простір на вокселізовану сітку з роздільною здатністю  $R_{seed}$  що є значною більшою ніж  $R_{voxel}$ . Приклад такого розбиття показано на Рис. 3.7.

Таким чином, метрика в цьому 39-мірному просторі, відповідно до Jeremie Rapin є наступною.

$$D = \sqrt{\frac{\lambda D_c^2}{m^2} + \frac{\mu D_s^2}{3R_{seed}^2} + \epsilon D_{HiK}^2} \quad (3.3)$$

Де  $D_c$  – евклідова відстань у просторі кольору CIE Lab , нормалізована константою  $m$

$D_{HiK}$  – відстань у просторі FPFH

$D_s$  – евклідова відстань у 3D просторі

$R_{seed}$  – роздільна здатність у вокселізованому просторі

$FPFH_{1...33}$  – 33 елементи

$\lambda$  – ваговий коефіцієнт впливу кольору

$\mu$  – ваговий коефіцієнт впливу відстані

$\epsilon$  – ваговий коефіцієнт впливу геометричної подібності кластерів

Приклад роботи алгоритму надсегментації супер вокселями, застосовно до однієї з хмар точок, зображено на Рис. 3.8

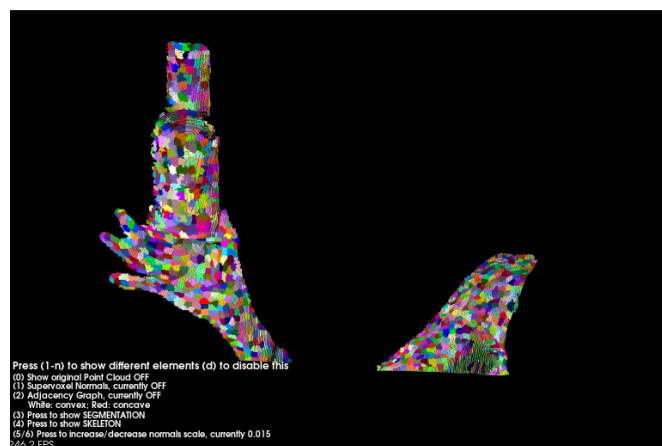


Рисунок. 3.8 – Приклад роботи алгоритму надсегментації супер вокселями

### 3.2.6 Виділення сегментів у неогранізованій хмарі точок, на основі локальних випуклостей

Для задачі виділення сегментів у неогранізованій хмарі точок, на основі локальних випуклостей було використано роботу Simon Christoph Stein “Object Partitioning using Local Convexity” [7].

Розбиття об’єктів за локальною опуклістю (Object Partitioning using Local Convexity, OPCL) - це алгоритм, який використовується для сегментації хмари точок на сегменти на основі їхньої локальної опуклості. Алгоритм був запропонований у 2016 році і показав свою ефективність для сегментації хмари точок у різних додатках комп’ютерного зору, таких як розпізнавання об’єктів, розуміння сцен і 3D-реконструкція.

Алгоритм OPCL складається з трьох основних кроків:

1. *Оцінка локальної опуклості.* OPCL оцінює локальну опуклість для кожної точки в хмарі точок, методом підбору площини до сусідніх точок і обчислюючи кут між нормаллю до площини і вектором, що з’єднує точку з її сусідами. Значення локальної опуклості відображає ступінь опуклості точки по відношенню до її сусідів. Оцінена локальна опуклість є корисною характеристикою для розрізнення об’єктів у хмарі точок. Більш детально процес класифікації опуклих і вгнутих сегментів буде описано у наступному пункті.

2. *Розширення сегмента.* OPCL використовує оцінені значення локальної опуклості для виконання процесу розширення сегменту, який групує сусідні точки зі схожими значеннями опуклості. Починаючи з початкової точки, алгоритм ітеративно додає сусідні точки з подібними значеннями опуклості, доки різниця у значеннях опуклості між сусідніми сегментами не перевищить певний поріг. Процес розширення продовжується до тих пір, поки всі точки хмари точок не будуть включені до якого з регіонів.

3. *Уточнення об’єкта.* Після початкового етапу розширення сегмента, OPCL виконує уточнення об’єкта шляхом злиття сегментів з малою площею, які знаходяться близько один до одного і мають схоже значення опуклості. Це

робиться шляхом вимірювання відстані між малими об'єктами та їх об'єднання, якщо вони знаходяться не далі, ніж заданий поріг відстані.

Алгоритм розбиття об'єктів за локальною опуклістю має кілька переваг над іншими методами сегментації, зокрема:

1. *Стійкість до шуму та викидів.* Оцінка локальної опуклості є стійкою до шуму та викидів, оскільки базується на відносному розташуванні сусідніх точок.
2. *Висока точність.* Локальна опуклість надає корисну функцію для розрізнення об'єктів у хмарі точок, що призводить до високої точності сегментації об'єктів.
3. *Низькі обчислювальні витрати.* Алгоритм OPLC є обчислювально ефективним, оскільки він вимагає лише оцінки локальної опуклості та зростання області.
4. *Гнучкість.* OPLC може бути адаптований до різних типів даних хмари точок і має широкі можливості налаштування параметрів алгоритму до конкретних даних.

Приклад виділення сегментів у неогранізованій хмарі точок, на основі локальних випуклостей зображено на Рис. 3.9

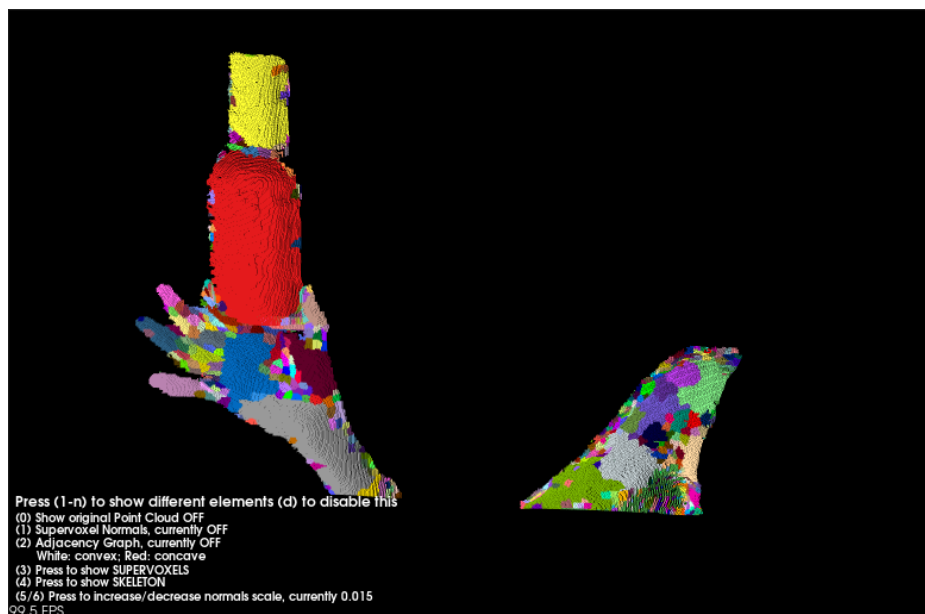


Рисунок. 3.9 – Приклад виділення сегментів у неогранізованій хмарі точок, на основі локальних випуклостей

### 3.2.7 Побудова графа зв'язності для визначення локальних випуклостей

Після етапу надсегментації супервокселями, було отримано набір супервокселів, що є представленим наступним чином:

$$p_i = (x_i, n_i, N_i) \quad (3.4)$$

Де  $x_i$  – центроїд супервокселя,

$n_i$  – нормаль,

$N_i$  – список суміжних до  $i$ -го супервокселів.

Таким чином, супервокселі утворюють граф, ребра якого є границями супервокселів [10].

Для визначення випуклих та вгнутих сегментів потрібно розглянути кожне з ребер супервокселя, та класифікувати його.

Розглянемо пару сусідніх супервокселів з центроїдами в точках  $x_i$  та  $x_j$ , та нормаллями  $n_i$  та  $n_j$  відповідно. Зв'язок між ними класифікується як випуклий чи вогнутий спираючись на величину кута між їх нормаллями та вектором, що з'єднує їх центроїди.

Нехай  $d$  - вектор, що з'єднує центроїди  $x_i$  та  $x_j$ :  $d = x_i - x_j$ . Кут нахилу нормалей до вектора  $d$  може бути розрахований використовуючи скалярний добуток наступним чином:

$$a \cdot b = |a| \cdot |b| \cdot \cos(\alpha) \rightarrow \alpha = \frac{\text{acos}(a \cdot b)}{|a| \cdot |b|} \quad (3.5)$$

Розглянемо схему, зображено на Рис. 3.10. З'єднання можна вважати опуклим, якщо  $\alpha_1 \leq \alpha_2$ , і вгнутих - у іншому випадку.

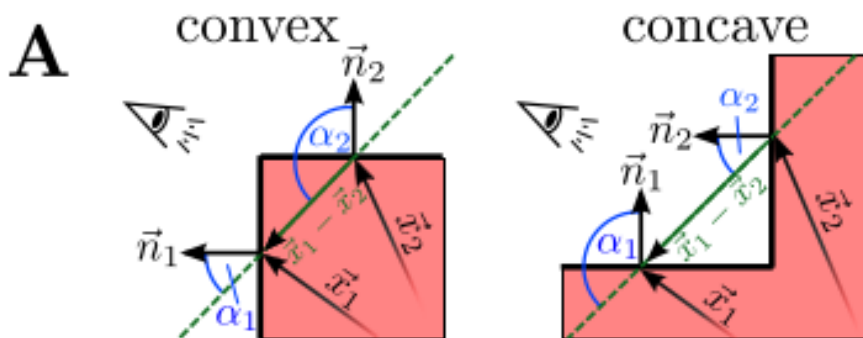


Рисунок. 3.10 – Ілюстрація геометрії, що використовується для розрахунку опуклості та вгнутості

$$\alpha_1 \leq \alpha_2 \Leftrightarrow \cos(\alpha_1) - \cos(\alpha_2) \geq 0 \Leftrightarrow \vec{n}_1 \cdot \hat{d} - \vec{n}_2 \cdot \hat{d} \geq 0 \quad (3.6)$$

і, відповідно

$$\alpha_1 > \alpha_2 \Leftrightarrow \vec{n}_1 \cdot \hat{d} - \vec{n}_2 \cdot \hat{d} < 0 \quad (3.7)$$

Потрібно враховувати, що дані, отримані з RGB-D часто є зашумленими. Через це деякі поверхні можуть мати низьку кривизну. Для того, що працювати з такими сегментами введемо порогове значення  $B_{threshold}$ . Це значення регулюватиме максимальну допустиму кривизну сегменту, щоб вважати його опуклим.

$$\beta = \angle(\vec{n}_1, \vec{n}_2) = |\alpha_1 - \alpha_2| = \cos^{-1}(\vec{n}_1 \cdot \vec{n}_2) < \beta_{threshold} \quad (3.8)$$

В даній роботі не стоїть на меті виконати сегментацію хмари точок з високою точністю, тому  $B_{threshold}$  також використовується для ігнорування вгнутих поверхонь малої площі. Об'єднавши дві умови, що описані вище, можемо остаточно сформулювати критерій опуклості з'єднання:

$$CC_b(\vec{p}_i, \vec{p}_j) = \begin{cases} true, (\vec{n}_1 - \vec{n}_2) \cdot \hat{d} > 0 \vee (\beta < \beta_{threshold}) \\ false \end{cases} \quad (3.9)$$

На Рис. 3.11 зображено приклад побудови графа зв'язності для однієї з хмар точок.

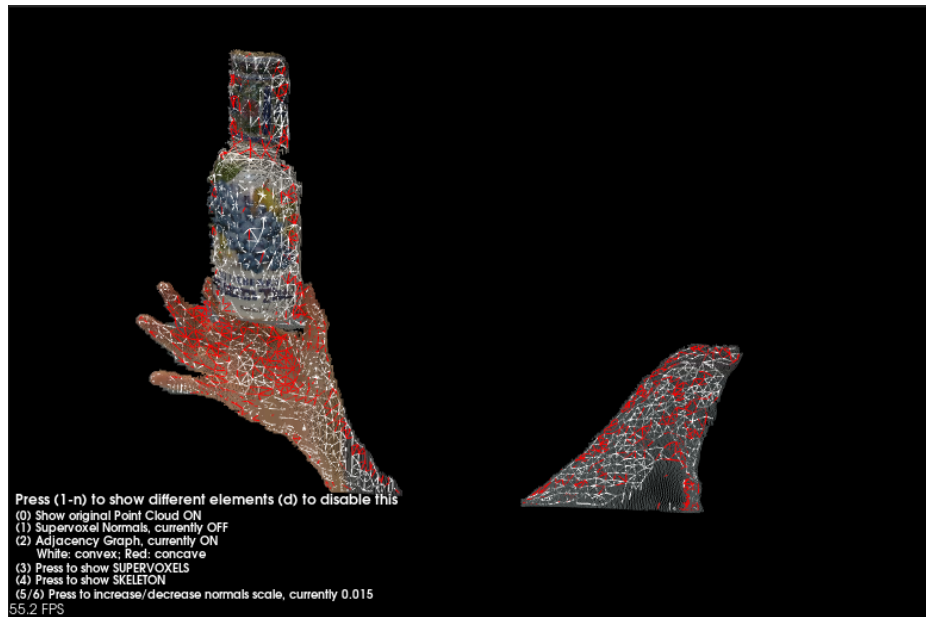


Рисунок. 3.11 – Приклад побудови графа зв’язності. Червоним кольором позначено ребра між сегменти які є вгнутими. Білим кольором позначено ребра між сегментами, які є опуклими.

### 3.2.8 Сегментація руки у неорганізованій хмарі точок

Дано множину опуклих та увігнутих сегментів  $S$ , на які розбито хмару точок, множину ключових точок руки  $P$ , що були виділені фреймворком MediaPipe Hands. Схематично зображено на Рис. 3.3. У розділі 3.2.1.

Сегменти множини  $S$ , котрі перетинаються відрізками з множини  $L$  вважаються такими, що належать до множини сегментів руки  $S_h$  і не беруть участі у подальшій реконструкції моделі об’єкта. Множина відрізків  $L$  задається через з’єднання відповідних точок з множини точок  $P$ , тобто для пари точок  $p_1, p_2 \in P$  що мають бути з’єдані, маємо відрізок  $l \in L$ , де  $l = p_1 + (1 - \alpha) \cdot p_2$ ,  $\alpha \in [0; 1]$ , що схематично зображено на Рис. 3.12.

Сегмент  $s_j \in S$  вважається таким, що належить до множи сегментів руки  $S_h$  з допустимою похибкою  $\delta > 0$ , якщо  $\exists l_i \in L$  який задовільняє умовам системи нерівностей:

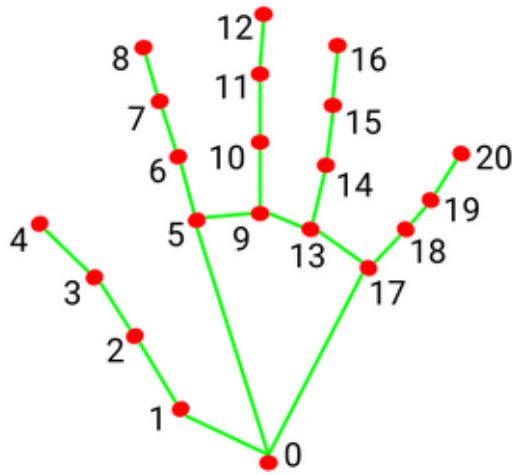


Рисунок. 3.12 – Візуалізація ліній, утвореними з'єднанням ключових точок руки

$$\left\{ \begin{array}{l} \min_x(l_i) + \delta < \max_x(s_j) \\ \min_x(l_i) + \delta \geq \max_x(s_j) \\ \min_y(l_i) + \delta < \max_y(s_j) \\ \min_y(l_i) + \delta \geq \max_y(s_j) \\ \min_z(l_i) + \delta < \max_z(s_j) \\ \min_z(l_i) + \delta \geq \max_z(s_j) \end{array} \right. \quad (3.10)$$

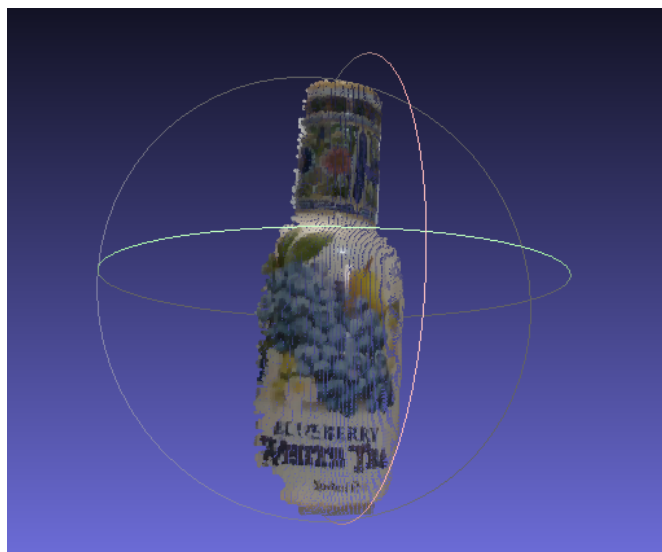


Рисунок. 3.13 – Приклад роботи алгоритму сегментації руки. Зображена множина точок  $S_h$

Після перевірки кожного з елементів множини  $S$  на відповідність цій умові отримуємо, множину  $S_h$ , що містить лише сегменти руки. Приклад роботи алгоритму сегментації руки, застосовно до однієї з хмар точок, зображено на Рис. 3.13.

### 3.2.9 Уточнення сегмента 3D моделі об'єкта

Через похибки в алгоритмі сегментації та шум, що присутній в даних, отриманих з RGB-D камери, окрім точок, що належать предмету в результуючій хмарі точок можуть бути присутні інші невеликі сегменти, що не відносяться до об'єкта, що ми розглядаємо. Окрім цього, потрібно враховувати, що на етапі сегментації точок фону в хмарі точок, могли залишитись інші сегменти, що знаходяться в межах допустимої відстані (нами обраний проміжок 0.3-2.0м) від камери, але не відносяться до цільового об'єкта. Такі сегменти, також не беруть участь у подальшій реконструкції моделі об'єкта.

Задамо множину сегментів цільового об'єкта  $S_{obj} = \frac{S}{S_h}$ .

Задамо множину точок  $C \subset R^3$ , де кожен елемент  $c_i \in C$  - центроїда сегмента  $s_i \in S_{obj}$ .

Враховуючи, що дані про об'єкт були отримані у процесі маніпуляції ним у руці, можемо стверджувати, що в більшості випадків поряд з об'єктом знаходився великий палець руки, тому ми можемо видалити з множини  $S_{obj}$  сегменти, що знаходяться від нього надто далеко. Введемо порогове значення  $H_{thresh}$ , що буде радіусом сфери, з центром в ключовій точці з порядковим номером 3. Всі точки, що не лежать в середині цієї сфери будемо вважати такими, що не належать об'єкту. Сегмент  $s_i \in S_{obj}$  не бере подальшої участі у реконструкції 3D моделі цільового об'єкта, якщо його центроїда  $c_i = (c_{ix}, c_{iy}, c_{iz})$  не належить сфері з радіусом  $H_{thresh}$  і центром в точці  $p_3 = (p_{3x}, p_{3y}, p_{3z}) \in P$ . Тобто, задовільняють умовам нерівності:

$$(c_{ix} - p_{3x})^2 + (c_{iy} - p_{3y})^2 + (c_{iz} - p_{3z})^2 > H_{thresh}^2 \quad (3.11)$$

Алгоритм фільтрування сегментів сзематично зображено на Рис. 3.14

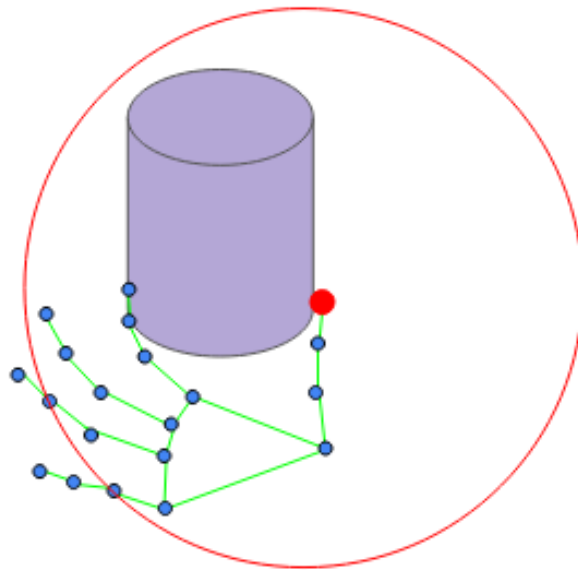


Рисунок. 3.14 – Схематичне зображення алгоритму фільтрації сегментів

Таким чином, можливо відфільтрувати більшість шуму, присутнього через похибки в алгоритмі сегментації та шум, що присутній в даних, отриманих з RGB-D камери. Проте, такий варіант фільтрації не є ідеальним. Сегменти, що були помилково включені до множини  $S_{obj}$  будуть відкинуті пізніше, на етапі реконструкції 3D моделі об'єкта.

## 4 РЕКОНСТРУКЦІЯ 3D МОДЕЛІ ОБ'ЄКТА

3D-реєстрація є дуже важливим компонентом у процесі реконструкції 3D-моделі об'єкта, оскільки вона передбачає суміщення та інтеграцію декількох сегментів або зображень об'єкта для створення цілісного і точного представлення оригінального об'єкта. Ця процедура є незамінною для об'єднання даних з різних точок спостереження, усунення оклюзій та заповнення прогалін у накопиченій інформації. Від ефективності процесу реєстрації значною мірою залежить якість кінцевої 3D-моделі, тому дуже важливо вибрати відповідні методи і прийоми, адаптовані до конкретної задачі і набору даних.

Існує багато методів 3D-реєстрації, від методів, заснованих на ознаках, що базуються на зіставленні характерних особливостей або ключових точок між сегментами об'єкта, до глобальних стратегій оптимізації, які мінімізують заздалегідь визначену функцію вартості, наприклад, відстань між відповідними точками або ступінь перекриття між сегментами. Ітераційні алгоритми найближчої точки (ICP) та їхні варіації є одними з найпоширеніших методів у цій галузі, оскільки вони забезпечують баланс між точністю та обчислювальною складністю [5].

У даній роботі процес реконструкції 3D моделі є додатково ускладненим тим, що об'єкт при зборі даних частково перекриває рука. Це тягне за собою втрату інформації. Тому у даній роботі використано не лише просторові алгоритми суміщення сегментів, а й класичні підходи комп'ютерного зору, що працюють на 2D площині.

### 4.1 Проекція сегмента 3D моделі на 2D площину камери

Для того аби використовувати алгоритми комп'ютерного зору, що працюють з 2D зображеннями, необхідно розрахувати проекцію 3D сегмента зображення на 2D площину камери.

Процедура проектування 3D-сегмента, знятого RGBD-камерою, на площину 2D-зображення передбачає використання внутрішніх параметрів камери(були отримані за допомогою API, що надає Intel для своїх пристроїв) , які визначають

взаємозв'язок між 3D-світом і 2D-зображенням. Внутрішні параметри зазвичай складаються з фокусних відстаней  $(f_x, f_y)$  по осях  $x$  і  $y$ , оптичного центру  $(c_x, c_y)$  і, можливо, інших коефіцієнтів спотворення об'єктива. Ці параметри можуть бути представлені матрицею камери, яка використовується для перетворення 3D-координат у 2D-координати пікселів.

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \quad (4.1)$$

Маючи 3D точку  $(X, Y, Z)$  в системі координат камери, процес проєкції починається з ділення координат  $X$  і  $Y$  на координату  $Z$ , що фактично нормалізує точку. Цей крок відповідає за ефект перспективи, коли об'єкти, розташовані ближче до камери, здаються більшими на зображенні. Потім нормалізована точка  $(\frac{X}{Z}, \frac{Y}{Z})$  множиться на фокусні відстані  $(f_x, f_y)$  і додається до оптичного центру  $(c_x, c_y)$ . Цей крок масштабує точку до відповідного розміру на площині зображення і переводить її в правильне положення відносно оптичного центру.

Математично це можна виразити наступним чином:

$$u = f_x * \left(\frac{X}{Z}\right) + c_x \quad (4.2)$$

$$v = f_y * \left(\frac{Y}{Z}\right) + c_y \quad (4.3)$$

Тут  $(u, v)$  представляє 2D координати пікселів, що відповідають 3D точці  $(X, Y, Z)$ . Застосовуючи це перетворення до всіх точок 3D-сегмента, можна отримати 2D-зображення, яке вирівнюється з RGB-зображенням, захопленим камерою.

Використовуючи цей підхід до кожного кадру, знятого камерою, примочу як до RGB зображень, так і до карт глибини формується новий датасет, що містить наступні елементи для кожного кадру:

- Хмара точок сегмента

- RGB зображення сегмента
- Карта глибини сегмента

## 4.2 Виділення ключових точок на 2D зображенні сегмента предмета

Масштабно-інваріантне перетворення ознак (Scale-Invariant Feature Transform, SIFT) - це широко використовуваний алгоритм для виявлення та отримання характерних ознак з RGB-зображень, які є інваріантними до масштабу, повороту і частково інваріантними до зміни освітлення та точки спостереження. Ці стійкі ознаки відіграють вирішальну роль у різних задачах комп'ютерного зору, таких як розпізнавання об'єктів, зшивання зображень і 3D-реконструкція, де надійне зіставлення та оцінка відповідності є надзвичайно важливими.

Алгоритм SIFT працює в декілька етапів для виявлення та опису ключових точок на зображенні [14]. Перший етап передбачає побудову масштабно-просторового представлення зображення, що досягається шляхом застосування гауссових фільтрів з різними стандартними відхиленнями до вихідного зображення в декількох масштабах. Цей крок гарантує, що виявлені особливості залишаються масштабно-інваріантними і можуть бути ідентифіковані на різних рівнях деталізації. Далі потенційні ключові точки виявляються шляхом визначення локальних екстремумів на зображеннях різниці гаусів (DoG), які отримуються шляхом віднімання гаусівських розмитих зображень на сусідніх масштабах. Цей крок ефективно виявляє краплеподібні структури на зображенні, які, ймовірно, є характерними і стабільними за різних умов.

Після того, як потенційні ключові точки знайдено, вони уточнюються шляхом усунення низькоконтрастних або краєподібних ключових точок, які можуть бути недостатньо стабільними або виразними. Цей крок підвищує надійність виявлених ознак і зменшує ймовірність помилкових збігів. На наступному етапі кожній ключовій точці призначається домінуюча орієнтація на основі інформації

ції про локальний градієнт, що робить ознаки інваріантними до повороту. Ця орієнтація використовується для обчислення дескриптора для кожної ключової точки, який являє собою гістограму градієнтів у локальній околиці, зважену за їхніми величинами та просторовим розташуванням. Отриманий дескриптор є 128-вимірним вектором, який ефективно відображає локальний вигляд ключової точки і може бути використаний для зіставлення та оцінки відповідності. Приклад використання алгоритму виділення ключових точок на одному із зображень датасету зображено на Рис. 4.1.

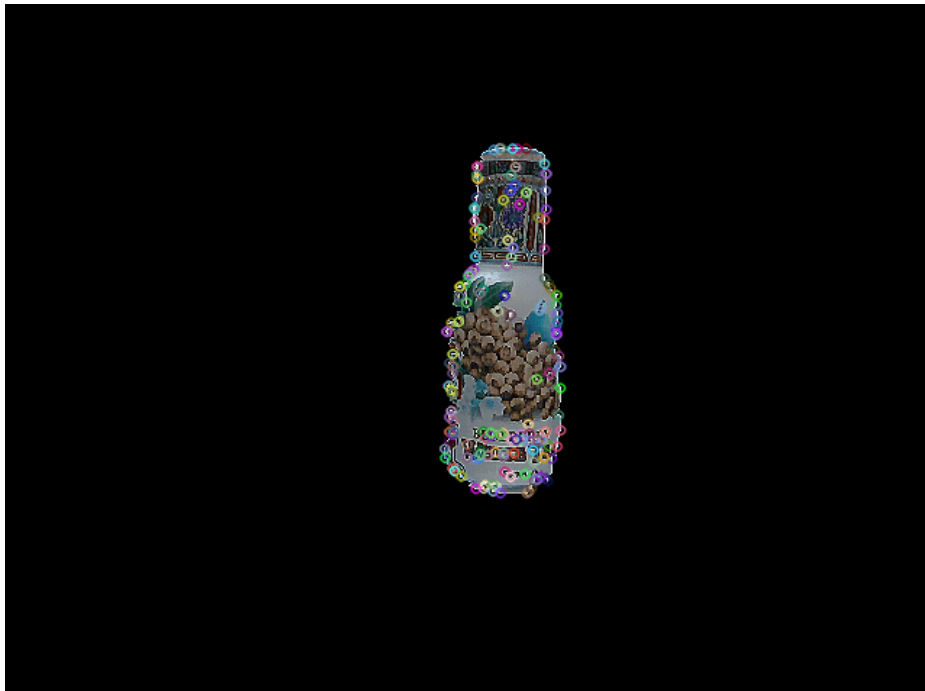


Рисунок. 4.1 – Ключові точки, виділені за допомогою алгоритму SIFT.

Переваги використання SIFT для виявлення особливостей на RGB-зображеннях:

1. *Масштабна інваріантність.* SIFT розроблено для виявлення об'єктів у різних масштабах, що робить його придатним для зіставлення зображень з різними рівнями масштабування або роздільної здатності. Це є надзвичайно важливою особливістю для реконструкції у даній задачі, оскільки об'єкт, яким маніпулюють перед камерою, може віддалятися та наближатися до неї у процесі маніпуляції.

2. *Інваріантність обертання.* SIFT призначає домінуючу орієнтацію для кожного об'єкта, що робить алгоритм стійким до повороту зображення, що корисно

при зіставленні зображень, знятих під різними кутами або в різній орієнтації. Це є однією з найважливіших властивостей дескриптора, що може бути використаний в даній задачі, адже дані про об'єкт отримується у процесі маніпуляції ним, що на пряму є набором поворотів предмета у руках.

3. *Інваріантність до освітлення.* SIFT є частково інваріантним до змін освітлення завдяки тому, що він покладається на інформацію про локальний градієнт, а не на необроблену інтенсивність пікселів. Ця характеристика є корисною при обробці зображень, знятих за різних умов освітлення.

4. *Розрізнення.* Ознаки SIFT мають високу розрізнявальну здатність, що призводить до зменшення кількості помилкових збігів і покращує загальну продуктивність різних завдань комп'ютерного зору, які покладаються на зіставлення ознак, таких як розпізнавання об'єктів, зшивання зображень і 3D-реконструкція.

5. *Надійність.* Багатоетапний процес виявлення та опису ознак у SIFT, включаючи усунення низькоконтрастних і краєподібних ключових точок, робить алгоритм стійким до шуму та інших артефактів зображення.

Недоліки використання SIFT для виявлення особливостей на RGB-зображеннях:

1. *Обчислювальна складність.* SIFT є обчислювально дорогим через його багатоетапний процес і побудову масштабно-просторового представлення. Ця складність може зробити алгоритм повільним для додатків у реальному часі або при обробці великих зображень. У даній роботі не ставилось за мету створити додаток, що працюватиме у реальному часі, тому часова складність SIFT, у даному випадку, є цілком прийнятною.

2. *Запатентований.* SIFT є запатентованим алгоритмом, що може накладати ліцензійні обмеження на його використання в комерційних додатках або програмному забезпеченні. Проте, SIFT може бути використаний в науковій та академічній роботі.

3. *Обмежена інваріантність.* Хоча SIFT частково інваріантний до змін освітлення та точки зору, він може не працювати належним чином в екстремальних умовах, таких як значні зміни точки зору, нежорсткі деформації або різкі перепади освітленості.

4. *Чутливість до розмиття.* Функції SIFT можуть бути чутливими до розмиття зображення, що може призвести до втрати характерних ключових точок і зниження ефективності зіставлення. У даній роботі розмиті зображення були відкинуті на детекції скелетона руки на RGB зображенні.

5. *Незастосовність до нетекстурованих областей.* SIFT покладається на наявність текстури або градієнтів на зображенні для виявлення об'єктів. В областях з малою кількістю текстури або взагалі без неї алгоритм може не виявити жодних значущих ключових точок.

Таким чином, алгоритм SIFT забезпечує надійний метод виявлення та опису відмінних рис на RGB-зображеннях, які є інваріантними до масштабу, повороту і частково інваріантними до змін освітлення та точки зору. Ці особливості є безцінними для широкого спектру застосувань комп'ютерного зору, які вимагають надійного і стабільного зіставлення та оцінки відповідності ознак.

### 4.3 Пошук відповідних пар ключових точок

Після отримання ключових точок з двох сусідніх зображень, та їх дескрипторів, необхідно визначити відповідності між ними. Для цього може бути використаний пошук найкращої відповідності шляхом повного перебору, такий метод гарантовано знайде кращу відповідність, але він є дуже затратний з точки зору часу роботи.

У даній роботі було використано метод, що базується на пошуку найближчих сусідів до запитованої точки - FLANN [13]. Швидка бібліотека наближених найближчих сусідів (FLANN) пропонує ефективне рішення для зіставлення ключових точок шляхом швидкої ідентифікації k-найближчих сусідів (kNN) у просторах високої розмірності. Ключова перевага FLANN полягає у здатності значно прискорити процес пошуку без втрати якості результату завдяки використанню вдосконалених структур даних та алгоритмів індексації замість традиційних методів грубого перебору.

Отримавши дескриптори, на попередньому етапі, FLANN може бути використаний для пошуку k найближчих сусідів для кожного дескриптора в наборі

даних. FLANN пропонує різноманітні алгоритми індексації, такі як k-d дерева або ієрархічні k-середні, які можна адаптувати до конкретної задачі та набору даних. Алгоритм індексування та інші параметри можуть бути встановлені вручну на основі попередніх знань або автоматично визначені за допомогою вбудованої функції оптимізації параметрів.

Під час kNN-пошуку об'єкт на основі FLANN отримує дескриптори з обох зображень на вхід і потім повертає список відповідностей, кожен з яких містить індекси k-найближчих сусідів для кожного дескриптора із запитуваних. Зазвичай, k задається рівним 2 для визначення двох найбільш підходящих збігів, що полегшує застосування тесту на співвідношення для фільтрації неоднозначних збігів. У даній роботі було використано тест-співвідношення Лоу. Тест співвідношення, запропонований Девідом Лоу, передбачає оцінку відстані між найкращою та другою за значенням відповідністю для кожної ключової точки. Якщо співвідношення між цими відстанями опускається нижче заздалегідь визначеного порогу, відповідність вважається надійною і зберігається. В іншому випадку вона вважається неоднозначною і відкидається. Цей тест підвищує надійність процесу зіставлення і мінімізує появу помилкових відповідностей. На Рис.4.2 зображено пару послідовних фреймів з виділеними ключовими точками, відповідні точки з'єднані прямими.

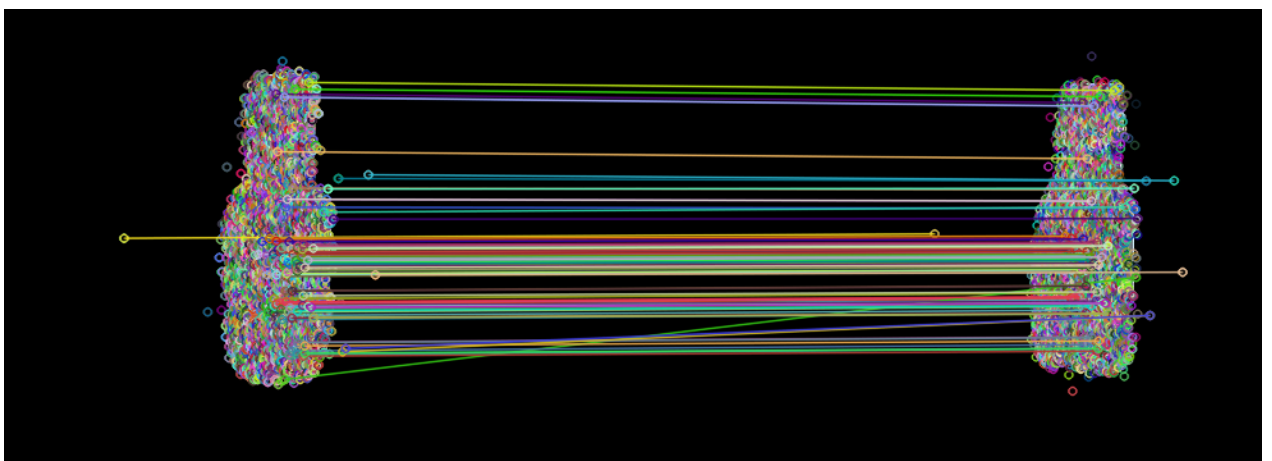


Рисунок. 4.2 – Два послідовних фрейми з виділеними ключовими точками, відповідні точки з'єднані прямими.

#### 4.4 Виділення ключових точок сегмента в 3D

На даному етапі отримано відповідні пари ключових точок, виділених на двох сусідніх RGB зображеннях. Ключові точки лежать у площині зображення, тобто відома їх  $(u_i, v_i)$  координата. Проте, крім цієї інформації, наявна інформація про глибину, тобто  $z$  координата кожної з ключових точок. Скористаємось цією інформацією для отримання 3D хмари ключових точок. Цю процедуру можна виконати шляхом зворотного, до описаного в розділі 4.1, перетворення, тобто математично це можна виразити наступним чином:

$$X = \frac{(u - c_x) * z}{f_x} \quad (4.4)$$

$$Y = \frac{(v - c_y) * z}{f_y} \quad (4.5)$$

Приклад виділених ключових точок в 3D зображено на Рис.4.3.



Рисунок. 4.3 – Ключові точки в 3D

Таким чином, на даному етапі маємо сформовані 2 множини  $K_i, k_{i_m} \in K_i$  – множина ключових точок для  $i$ -го фрейма.  $K_{i+1}, k_{i+1_m} \in K_{i+1}$  – множина ключових точок для  $i+1$ -го фрейма. Ці множини, по суті, є хмарами точок з низькою щільністю, такі хмари точок називають розрідженими.

#### 4.5 Суміщення ключових точок в 3D просторі

Маючи набір хмар точок  $\{P_1, P_2, \dots, P_n\}$ , процес реєстрації має на меті знайти послідовність жорстких перетворень  $T_1, T_2, \dots, T_n$ , таким чином, щоб трансформовані хмари точок  $\{T_1(P_1), T_2(P_2), \dots, T_n(P_n)\}$  були оптимально суміщені. Жорстке перетворення  $T_i$  можна представити як комбінацію матриці обертання  $R_i$  та вектора трансляції  $t_i$ :

$$T_i(P_i) = R_i * P_i + t_i \quad (4.6)$$

Мета полягає в тому, щоб мінімізувати загальну розбіжність між відповідними точками в трансформованих хмарах точок, яка зазвичай вимірюється за допомогою метрики відстані, наприклад, евклідової відстані:

$$D(P_i, P_j) = \|T_i(P_i) - T_j(P_j)\|. \quad (4.7)$$

Для простори пояснень розглянемо процес суміщення однієї пари хмар точок. Для задачі суміщення двох розріджених хмар точок, використовувався алгоритм ICP. Алгоритм Ітеративної Найближчої Точки (ICP) - це широко використовувана техніка для суміщення та реєстрації розріджених хмар точок. Основною метою алгоритму ICP є мінімізація відстані між двома хмарами точок шляхом ітеративного коригування їхнього взаємного розташування, щоб досягти найкращого можливого суміщення [16], [17].

Для розріджених хмар точок алгоритм ICP є особливо актуальним, оскільки точки даних можуть не щільно покривати підстилаючі поверхні, що потенційно може призвести до менш точних відповідностей. Алгоритм ICP починається зі знаходження початкової оцінки перетворення між двома хмарами

точок, яка може бути отримана з попередніх знань або наближених методів, таких як аналіз головних компонент (PCA), детально описано в розділі 3.2.4.

Після встановлення початкової оцінки алгоритм проходить серію ітерацій, виконуючи наступні кроки на кожній ітерації:

1. *Оцінка відповідності.* Для кожної точки вихідної хмари точок визначається найближча точка цільової хмари точок, як правило, за допомогою алгоритму пошуку найближчого сусіда, такого як k-d дерева або FLANN. У даній роботі використовувались k-d дерева. Детально описано в розділі 4.3.

2. *Відкидання викидів.* Застосовується стратегія фільтрації потенційних викидів або неправильних відповідностей. Поширені методи включають порогові значення на основі відстані, консенсус випадкової вибірки (RANSAC).

3. *Оцінка перетворення.* Обчислюється оптимальне стійке перетворення ("ротация" і "трансляція"), яке мінімізує відстань між відповідними точками, часто з використанням таких методів, як розкладання за сингулярними значеннями (SVD) або методів на основі кватерніонів. У даній роботі було використано SVD.

4. *Оновлення перетворення.* Обчислене перетворення застосовується до вихідної хмари точок, оновлюючи її положення відносно цільової хмари точок.

Ці кроки повторюються ітеративно, доки не буде досягнуто критерію збіжності, наприклад, заздалегідь визначеної кількості ітерацій, порогу зміни трансформації або мінімальної похибки у відстанях відповідності. Приклад суміщення ключових точок за допомогою ICP показано на Рис. 4.4.

Переваги використання ICP:

- *Простота.* Алгоритм ICP відносно простий у реалізації та розумінні, з чітким ітеративним процесом суміщення хмар точок.
- *Адаптивність.* Алгоритм ICP може бути адаптований і налаштований для вирішення конкретних завдань, наприклад, включення різних методів оцінки відповідності, методів відкидання викидів або підходів до оцінки трансформації для підвищення його продуктивності.

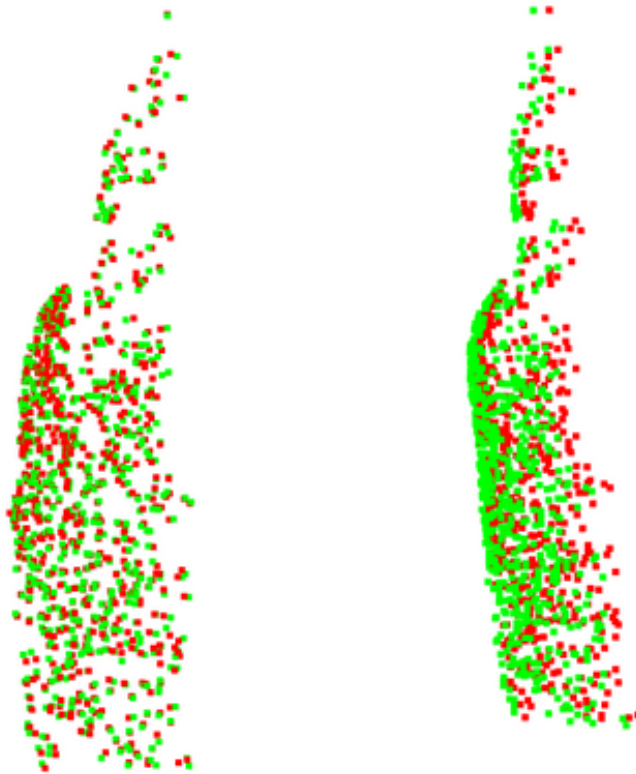


Рисунок. 4.4 – а) Ключові точки двох послідовних фреймів. б) Результат суміщення ключових точок двох послідовних фреймів

- *Збіжність*. ICP зазвичай збігається до локально оптимального рішення, забезпечуючи оптимальне суміщення між хмарами точок за умови відповідної початкової оцінки.
- *Сумісність*. Алгоритм ICP можна легко інтегрувати з іншими бібліотеками комп'ютерного зору або обробки хмар точок, такими як OpenCV, PCL, як це було зроблено в даній роботі.

#### Недоліки використання ICP:

- *Локальні мінімуми*. Алгоритм ICP схильний потрапляти в пастку локальних мінімумів, що може призвести до неоптимального суміщення. Якість остаточного суміщення сильно залежить від початкової оцінки трансформації між хмарами точок.
- *Чутливість до викидів*. ICP може бути чутливим до викидів або шуму в хмарах точок, що може негативно вплинути на оцінку відповідності та якість

вирівнювання. Хоча методи відкидання викидів можуть пом'якшити цю проблему, їх не завжди може бути достатньо. Ця особливість алгоритму була врахована на етапі виділення сегментів об'єкта, шляхом поспроцесингу, що було описано в розділі 4.5.

– *Незастосовність до нежорстких перетворень.* Стандартний алгоритм ICP призначений для жорстких перетворень (обертання і трансляція) і може не застосовуватися безпосередньо до сценаріїв, що включають нежорсткі перетворення, такі як деформації або масштабування. Це не відноситься до задачі, поставленої в даній магістерській роботі. Просто ця особливість є важливою та вартує згадки в переліку недоліків алгоритму.

– *Залежить від початкової оцінки.* Успіх алгоритму ICP значною мірою залежить від достатньо точної початкової оцінки трансформації між хмарами точок. Погані початкові оцінки можуть призвести до повільної збіжності або, в деяких випадках, до неможливості досягти задовільного суміщення. Цей недолік нівелюється тим, що у процесі реконструкції фрейми обробляються у порядку зйомки. Тому між 'сусідніми' сегментами об'єкта трансформація є невеликою.

ICP алгоритм застосовується для кожної послідовної пари ключових точок. Для кожної пари  $p_i$  та  $p_{i+1}$  необхідно знайти таку матрицю трансформації  $T_i$  при застосуванні якої до елемента  $p_{i+1}$  отримуємо сегмент  $p_{i+1}$  суміщений до елемента  $p_i$ .

## 4.6 Уточнення 3D моделі об'єкта

Після суміщення всіх сегментів, була отримана 3D модель об'єкта, яким маніпулювали у руках. Проте, вона не є ідеальною, тобто може містити деяку кількість викидів, що не були відкинуті на попередніх етапах процесингу.

Видалення викидів під час процесу реконструкції 3D-моделі має важливе значення з кількох причин. По-перше, викиди можуть негативно вплинути на точність реконструйованої моделі, призводячи до помилкової геометрії або артефактів, які не відображають справжню структуру об'єкта. По-друге, викиди

можуть негативно вплинути на ефективність алгоритму апроксимації 3D моделі просторовими примітивами. Загалом, видалення викидів має вирішальне значення для отримання якісних і надійних 3D-моделей.

Двома поширеними методами видалення викидів є статистичне видалення викидів (SOR) та видалення викидів за радіусом (ROR), обидва з яких аналізують локальну околицю кожної точки, щоб виявити та усунути потенційні викиди [18].

Видалення статистичних викидів ґрунтується на принципі аналізу розподілу відстаней між точками в межах локальної околиці. Для кожної точки  $p_i$  у хмарі точок алгоритм обчислює середню відстань  $\mu_i$  та стандартне відхилення  $\theta_i$  її  $k$  найближчих сусідів:

$$\mu_i = \frac{1}{k} * \sum \|p_i - p_j\|, j \in N_{p_i} \quad (4.8)$$

$$\sigma_i = \sqrt{\left(\frac{1}{k}\right) * \sum (\|p_i - p_j\| - \mu_i)^2}, j \in N_{p_i} \quad (4.9)$$

Точка вважається викидом, якщо її середня відстань до сусідів перевищує поріг, визначений як добуток глобальної середньої відстані  $\mu$  на масштабний коефіцієнт  $\alpha$ .

Викид, якщо:  $\mu_i > \mu + \alpha * \sigma$

Видалення викидів за радіусом, з іншого боку, базується на підрахунку кількості сусідніх точок в межах сфери фіксованого радіуса з центром навколо кожної точки в хмарі точок. Для заданої точки  $p_i$  алгоритм обчислює кількість сусідніх точок  $N_i$  в радіусі  $r$ :

$$N_i = |p_j | \|p_i - p_j\| \leq r |, j \in N_{p_i} \quad (4.10)$$

Точка вважається викидом, якщо кількість сусідів  $N_i$  в радіусі  $r$  менша за заданий мінімальний поріг  $min_{neighbors}$ .

Викид, якщо  $N_i < min_{neighbors}$ .

Як статистичне видалення викидів, так і видалення викидів за радіусом є ефективними для усунення різних типів викидів. Алгоритми SOR та ROR були

застосовані в до 3D моделі в комбінації - послідовно. Результат після видалення викидів зображено на Рис. 4.5.



Рисунок. 4.5 – Приклад 3D моделі об'єкта після видалення викидів

## 5 АПРОКСИМАЦІЯ 3D МОДЕЛІ ОБ'ЄКТА ПРОСТОРОВИМИ ПРИМІТИВАМИ

Апроксимація 3D-моделей об'єктів просторовими примітивами є ефективною технікою для спрощення складної геометрії, зменшення обчислювальних вимог та сприяння ефективному зберіганню та обробці моделей. Цей підхід передбачає представлення структури об'єкта за допомогою обмеженого набору простих, чітко визначених геометричних фігур, які відображають найсуттєвіші особливості оригінальної моделі, зберігаючи при цьому компактність і зручність представлення.

Зменшення складності представлення дозволяє більш ефективно зберігати, передавати та обробляти дані, що може бути особливо корисним у таких додатках, як комп'ютерна графіка, віртуальна та доповнена реальність і робототехніка. Крім того, використання чітко визначених геометричних форм полегшує інтуїтивно зрозумілу і просту маніпуляцію, аналіз та інтерпретацію моделей, що дозволяє вирішувати широкий спектр завдань, від виявлення зіткнень до розпізнавання об'єктів і розуміння сцени.

### 5.1 Вибір множини просторових примітивів, для апроксимації 3D моделі об'єкта

У даній роботі для апроксимації 3D моделі об'єкта було обрано наступні просторові примітиви:

- Площина
- Сфера
- Циліндр
- Кубоїд

Апроксимація 3D-моделей об'єктів описаними вище просторовими примітивами є широко використовуваною технікою з кількох ключових причин. Ці примітиви обирають через їхню простоту, зображальну силу та обчислювальну

ефективність, що робить їх добре придатними для широкого спектру застосувань та сценаріїв.

1. *Простота.* Ці примітиви є фундаментальними геометричними фігурами, які можна легко визначити, зрозуміти та маніпулювати ними. Вони забезпечують інтуїтивну основу для представлення більш складних структур, які можна розкласти на комбінації цих простих форм.

2. *Репрезентативна сила.* Площини, сфери, циліндри та кубоїди - це універсальні форми, які можуть ефективно апроксимувати широкий спектр реальних об'єктів та сцен. Вони можуть відображати як плоскі, так і криволінійні поверхні, а також різноманітні симетрії та закономірності, що робить їх застосовними у багатьох контекстах та сферах.

3. *Обчислювальна ефективність.* Використання цих просторових примітивів може значно зменшити складність 3D-моделей об'єктів, що призводить до більш ефективного зберігання, передачі та обробки. Крім того, їхні чітко визначені геометричні властивості дозволяють легко і швидко обчислювати різні операції, такі як перевірка перетинів, обчислення відстаней та оновлення трансформацій.

4. *Сумісність з існуючими алгоритмами.* Багато існуючих алгоритмів сегментації, підгонки та оптимізації призначені для роботи з цими специфічними просторовими примітивами. В результаті, включення їх в процес апроксимації забезпечує сумісність з широким спектром інструментів і методів, доступних для обробки хмар точок і сіток.

Апроксимуючи 3D-моделі об'єктів площинами, сферами, циліндрами та кубоїдами, можна досягти балансу між простотою, потужністю представлення та обчислювальною ефективністю. Ці просторові примітиви забезпечують надійну та універсальну основу для спрощення складної геометрії, зберігаючи при цьому основні характеристики оригінальних моделей, що дозволяє більш ефективно та раціонально обробляти, аналізувати та інтерпретувати 3D-дані.

## 5.2 Алгоритм апроксимації 3D моделі об'єкта просторовими примітивами

У цьому розділі буде розглянуто алгоритм апроксимації 3D-моделі об'єкта просторовими примітивами. Основна увага буде приділена проектуванню та реалізації алгоритма для забезпечення точного та ефективного відображення геометрії об'єкта. Крім того, в цьому розділі будуть розглянуті ключові компоненти та етапи алгоритму апроксимації, включаючи вибір та параметризацію просторових примітивів, попередню обробку даних. Розглянемо алгоритм апроксимації покроково.

Алгоритм апроксимації 3D моделі об'єкта включає в себе наступні кроки:

1. Кластеризація множини точок моделі. Апроксимація примітивами є ітеративним процесом і триває доки кількість множини точок моделі більше ніж заздалегідь визначена константа  $thresh_{pts}$ . Детально алгоритм кластеризації буде описано в розділі 5.2.

2. Відбір примітивів на основі RANSAC. Наступним кроком є вибір найкращих примітивів для кожного кластера за допомогою алгоритму консенсусу випадкової вибірки (RANSAC) [18]. RANSAC - це надійна техніка оцінювання, яка може ефективно обробляти зашумлені дані або дані з викидами. Він ітеративно вибирає випадкові підмножини точок з кожного кластера, підбирає примітиви-кандидати на ці підмножини і оцінює якість підбору за кількістю інвайлерів, тобто точок, які близькі до підбраного примітиву. Примітив з найбільшою кількістю інлайерів вважається найкращим представником кластера. Процес повторюється для заданої кількості ітерацій, і примітив з найвищим загальним консенсусом обирається як остаточне наближення.

3. Точки для яких було знайдено наближення (інлайери) видаляються з множини, що буде розглядатись на наступній ітерації.

Алгоритм апроксимації 3D моделі об'єкта схематично зображено на Рис. 5.1



Рисунок. 5.1 – Алгоритм апроксимації 3D моделі об'єкта

Як зазначалось раніше, набір примітивів для апроксимації включає в себе: площину, циліндр, кубоїд, сферу. Вибір найбільш підходящого примітиву до кожного з кластерів виконується за допомогою алгоритму RANSAC, що по суті оптимізує значення параметрів, що описують модель кожного з примітивів. Наведені в Табл 5.1.

Таблиця 5.1 – Просторові примітиви

Примітив	Параметри
Площина	$a, b, c, d$ - коефіцієнти рівняння площини
Циліндр	$c_{axis}$ - вісь циліндра $c_{centr}$ - центроїд циліндра $h$ - висота циліндра $r$ - радіус циліндра
Сфера	$c_{centr}$ - центроїд сфери $r$ - радіус сфери
Кубоїд	Кубоїд це комбінація трьох взаємно перпендикулярних площин $a_1, b_1, c_1, d_1$ - коефіцієнти рівняння першої площини $a_2, b_2, c_2, d_2$ - коефіцієнти рівняння другої площини $a_3, b_3, c_3, d_3$ - коефіцієнти рівняння першої площини

Таким чином, даний етап є фінальним, після його завершення ми отримуємо апроксимацію 3D моделі просторовими примітивами.

### 5.3 Кластеризація моделі 3D об'єкта

Кластеризація 3D-моделі у вигляді хмари точок за допомогою алгоритму Density-Based Spatial Clustering of Applications with Noise (DBSCAN) - це потужний підхід для виявлення та групування точок зі схожими просторовими характеристиками. DBSCAN працює, аналізуючи розподіл щільності точок у хмарі точок і формуючи кластери на основі їхньої близькості та зв'язності.

Алгоритм DBSCAN потребує двох вхідних параметрів: порогової відстані  $\epsilon$  та мінімальної кількості точок  $points_{min-samples}$ , необхідної для формування щільної області. Основний принцип алгоритму полягає в групуванні точок, які близько розташовані в просторі і мають високу щільність сусідів. Процес кластеризації можна описати наступним чином.

Для кожної точки  $p_i$  у хмарі точок обчислюється околиця  $N_{p_i}$ , що містить всі точки  $p_j$  в межах порогової відстані  $\epsilon$ .

$$N_{p_i} = \{p_j \mid \|p_i - p_j\| \leq \epsilon, j \neq i\} \quad (5.1)$$

Точка  $p_i$  вважається опорною, якщо розмір її околиці  $N_{p_i}$  більше або дорівнює  $points_{min-samples}$ .

Точки, які не є основними точками, але знаходяться в околиці  $\epsilon$  основної точки, називаються граничними точками. Точки, які не належать до жодної з цих категорій, вважаються точками шуму.

Алгоритм працює шляхом ітеративного розширення кластерів, починаючи з опорних точок. Для кожної невідвіданої опорної точки створюється новий кластер, до якого додаються всі пов'язані опорні точки в околиці  $\epsilon$ . Потім граничні точки приєднуються до найближчого кластера опорних точок. Процес продовжується до тих пір, поки всі основні точки не будуть відвідані і розподілені по кластерах, в той час як точки шуму залишаються некластеризованими.

Використовуючи алгоритм DBSCAN для кластеризації 3D хмари точок, можна ефективно ідентифікувати і групувати просторово когерентні області моделі. Це може полегшити подальші кроки обробки, такі як сегментація, примі-

тивна підгонка або видалення викидів Заснована на щільності природа DBSCAN дозволяє обробляти кластери різної форми і розміру, будучи при цьому стійкою до шуму, що робить її добре придатною для обробки складних даних 3D-моделей.

## 6 ОПИС ПРОГРАМНОГО ПРОДУКТУ

У даній магістерській роботі було реалізовано програмний продукт для реконструкції 3D моделі об'єкта та апроксимації його моделі просторовими примітивами. Що працюватиме з даними, отриманими у процесі маніпуляції ним людиною у руці перед RGB-D камерою. У цьому розділі буде описано інструменти, бібліотеки, фреймворки, що були використані, а також безпосередньо сам програмний продукт.

### 6.1 Опис використаних технічних засобів

У процесі реалізації алгоритму, для реконструкції 3D моделі предмета у руці людини на основі даних RGB-D камери та апроксимації її геометрії просторовими примітивами було використано низку інструментів та бібліотек. Ці утиліти включають мови програмування, такі як C++ та Python, а також бібліотеки та фреймворки, такі як OpenCV, NumPy, PCL, Open3D та MediaPipe Hand. У цьому розділі буде зроблено короткий огляд кожної утиліти.

C++ - це мова програмування загального призначення, яка пропонує високий рівень ефективності, що робить її оптимальним вибором для обчислювально інтенсивних задач, таких як реконструкція 3D об'єктів. Її багатий набір функцій, включаючи підтримку об'єктно-орієнтованого, процедурного та узагальненого програмування, дозволяє створювати гнучкий код, який легко підтримувати. C++ стандартизована Міжнародною організацією зі стандартизації (ISO) і має кілька реалізацій, доступних під різними ліцензіями, наприклад, GNU Compiler Collection (GCC) під Загальною публічною ліцензією GNU (GPL).

З іншого боку, Python - це інтерпретована, високорівнева мова програмування загального призначення, що відзначається читабельністю та простотою використання. Її велика стандартна бібліотека, а також наявність численних сторонніх бібліотек роблять Python популярним вибором для швидкої

розробки та створення прототипів. Python випускається під ліцензією Python Software Foundation License, яка є дозвільною ліцензією з відкритим вихідним кодом, що дозволяє модифікацію, розповсюдження та використання як у відкритих, так і в пропрієтарних проектах.

OpenCV (Open Source Computer Vision Library) - це потужна бібліотека комп'ютерного зору та машинного навчання, призначена для обробки зображень і відео в режимі реального часу. Вона надає повний набір функцій і алгоритмів для таких завдань, як виділення ознак, виявлення об'єктів і перетворення зображень. OpenCV написана на C++ і має прив'язки для Python, що робить її доступною для широкого кола розробників. Він випускається під вільною ліцензією Apache License 2.0, що дозволяє використовувати, модифікувати та розповсюджувати його як у комерційних, так і в некомерційних проектах.

NumPy - це фундаментальна бібліотека для наукових досліджень на мові Python. Вона пропонує широкий набір математичних функцій, операцій лінійної алгебри та можливостей випадкових чисел, а також підтримку N-вимірних масивів і трансляції. Ефективна реалізація та простота використання NumPy роблять його незамінним для проектів, що включають чисельні розрахунки, наприклад, реконструкцію 3D-об'єктів. NumPy випускається під ліцензією BSD 3-Clause License, що дозволяє використовувати його як у відкритих, так і у пропрієтарних проектах.

Бібліотека хмарних зображень (Point Cloud Library, PCL) - це окремий масштабний проект з відкритим вихідним кодом для обробки 2D/3D зображень та хмар точок. PCL пропонує широкий спектр функцій, включаючи фільтрацію, сегментацію, реєстрацію та реконструкцію поверхонь. Він особливо корисний для обробки великих наборів даних хмари точок, створених під час реконструкції 3D-об'єктів. PCL написаний на C++ і випускається під ліцензією BSD 3-Clause License.

Open3D - це сучасна та універсальна бібліотека з відкритим вихідним кодом, призначена для обробки та маніпулювання 3D-даними. Вона надає повний набір інструментів і функцій для таких завдань, як обробка хмар точок, генерація та редагування сіток, реєстрація та рендеринг. Розроблена на C++ з прив'язкою до Python, Open3D дозволяє користувачам отримати вигоду від високої продуктивності, зберігаючи при цьому простоту використання і доступність. Open3D випускається під ліцензією MIT, яка надає користувачам свободу використовувати, модифікувати та розповсюджувати бібліотеку як в академічних, так і в комерційних цілях. Ця вільна ліцензія гарантує, що бібліотека Open3D залишається популярним вибором для розробників і дослідників, які працюють в різних областях обробки тривимірних даних.

Нарешті, MediaPipe Hand - це кросплатформенний фреймворк для побудови мультимодальних прикладних конвеєрів машинного навчання. У контексті цього проекту він використовується для відстеження рухів рук та скелетону, що є ключовим для відокремлення сегментів руки від сегментів об'єктів, які знаходяться в руці. MediaPipe Hand забезпечує надійне відстеження рук у реальному часі та розпізнавання скелетону руки, яке можна легко інтегрувати в різні додатки. Він випускається під дозвоільною ліцензією Apache License 2.0.

Таким чином, поєднання C++, Python, OpenCV, NumPy, PCL, Open3D та MediaPipe Hand пропонує повний набір інструментів та бібліотек для реалізації алгоритму реконструкції об'єктів у руці та апроксимації її просторовими примітивами. Ці утиліти, доступні під вільними ліцензіями з відкритим вихідним кодом, полегшують процес розробки і забезпечують сумісність з широким спектром платформ і варіантів використання. Використовуючи сильні сторони кожного інструменту, результуюча реалізація може забезпечити точну та ефективну реконструкцію 3D-об'єктів на основі даних RGB-D камер, з потенційним застосуванням у таких галузях, як робототехніка, комп'ютерний зір, віртуальна та доповнена реальність і виробництво.

## 6.2 Опис результатів роботи програмного продукту

У рамках магістерської дипломної роботи було імплементовано алгоритм, призначений для реконструкції 3D-моделі об'єкта, яким маніпулює людина у руці перед RGB-D камерою, та апроксимації об'єкта за допомогою просторових примітивів. Програма використовує алгоритми та методи комп'ютерного зору для обробки вхідних даних, які складаються з RGB-зображень та карт глибини, і генерує два результати: реконструйовану 3D-модель об'єкта та апроксимацію об'єкта за допомогою просторових примітивів.

**Вхідні дані** Вхідні дані отримуються, коли людина маніпулює об'єктом перед RGB-D камерою. Камера фіксує два типи даних: RGB-зображення та карти глибини. Отже, вхідними даними програмного продукту є набір RGB зображень, та відповідних їм карт глибини.

**Вихідні дані** Програма обробляє вхідні дані і генерує два основних результати: реконструйовану 3D модель об'єкта та апроксимацію об'єкта просторовими примітивами. Результат складається з представлення хмари точок(моделі і сегментації) і JSON-файлу, що містить список просторових примітивів і пов'язаних з ними параметрів.

Приклад результату роботи програми зображено на Рис.6.1, Рис.6.2 та Рис. 6.3

```
results.json  
[  
  cylinder: {  
    "center": [-0.02865524, 0.65149359, 0.79318898]  
    "axis": [0.12807818, -0.78895736, -0.60095114]  
    "radius": 0.5836843112508792  
    "height": 1.9205741517975876  
  },  
  cylinder: {  
    "center": [0.17509272, -0.78568256, -0.23518511]  
    "axis": [0.07049652, 0.815933, 0.57383236]  
    "radius": 0.2960851173612676  
    "height": 1.634135437269485  
  }  
]
```

Рисунок. 6.1 – Приклад реконструйованої моделі об'єкта. Представлено у вигляді хмари точок



Рисунок. 6.2 – Приклад апроксимації об'єкта просторовими примітивами. Представлено у вигляді хмари точок



Рисунок. 6.3 – Приклад JSON файлу з переліком примітивів, якими апроксимовано об’єкт, їх параметрів та положення у просторі.

Цей опис висвітлює ключові особливості та функціональність програми, яка призначена для реконструкції 3D-моделі об’єкта та апроксимації його форми за допомогою просторових примітивів на основі даних, знятих RGB-D камерою.

## 7 РЕЗУЛЬТАТИ РОБОТИ АЛГОРИТМА

У даній магістерській роботі ставилось на меті розробка алгоритму для реконструкції 3D моделі об'єкта, яким маніпулює людина в руці, та апроксимація його просторовими примітивами. Результати роботи алгоритма наведено нижче. Для кожного з предметів датасету міститься RGB зображення предмета, реконструйована 3D модель, у третій, сегментація 3D моделі, у відповідності до його апроксимації просторовими примітивами. Результати роботи алгоритму наведено на Рис. 7.1 - 7.6.



Рисунок. 7.1 – Результати роботи алгоритму реконструкції та апроксимації для об'єкта 1



Рисунок. 7.2 – Результати роботи алгоритму реконструкції та апроксимації для об'єкта 2



Рисунок. 7.3 – Результати роботи алгоритму реконструкції та апроксимації для об'єкта 3

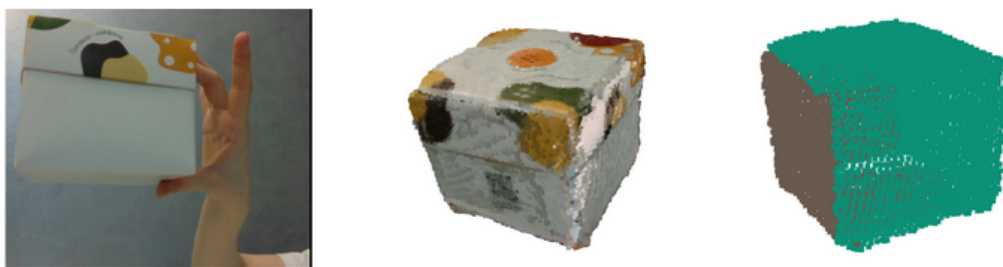


Рисунок. 7.4 – Результати роботи алгоритму реконструкції та апроксимації для об'єкта 4



Рисунок. 7.5 – Результати роботи алгоритму реконструкції та апроксимації для об'єкта 5

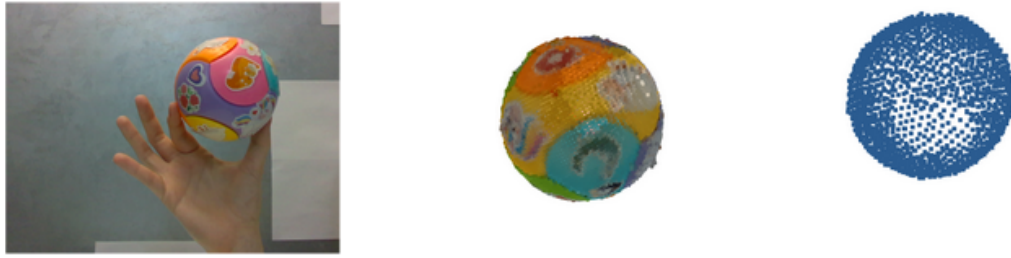


Рисунок. 7.6 – Результати роботи алгоритму реконструкції та апроксимації для об'єкта 6

Наведені результати є правильними, спираючись на візуальну оцінку. Реконструйована модель відповідає фізичному предмету.

Алгоритм може працювати з великою кількістю різноманітних предметів, не дивлячись на їх форму. Не залежить від форми руки та кольору шкіри, людини яка виконує маніпуляцію об'єктом. Перелік просторових примітивів може бути легко розширений, алгоритм є добре масштабованим. Оскільки алгоритм є детермінованим, він не потребує попереднього навчання, збору та розмітки датасету.

Проте, запропонований алгоритм має деякі обмеження:

- *Текстурованість об'єкта.* Об'єкт має бути текстурованим, або мати малюнок. Це обмеження обумовлене процесом реконструкції 3D моделі, який базується на виділенні ключових точок на сегменті RGB зображення. На поверхні однотонних об'єктів таких точок буде недостатньо.
- *Розмір об'єкта.* Об'єкти малого розміру, з тонкими деталями погано захоплюються сенсором глибини. Таке обмеження обумовлене специфікою обраного у даній роботі сенсора.
- *Матеріал об'єкта.* Скляні, відблискуючі металеві, прозорі об'єкти не можуть бути відскановані з використанням запропонованого у даній роботі алгоритму. Об'єкти з таких матеріалів погано взаємодіють з ToF сенсором. Це обмеження теж обумовлене специфікацією обраного сенсора.

## 8 ПРАКТИЧНЕ ЗАСТОСУВАННЯ РЕЗУЛЬТАТІВ РЕКОНСТРУКЦІЇ ТА АПРОКСИМАЦІЇ ОБ'ЄКТА ПРОСТОРОВИМИ ПРИМІТИВАМИ

Реконструкція 3D-моделі об'єкта, отримана шляхом маніпуляції предметом у руках людини перед RGBD-камерою, у поєднанні з його апроксимацією просторовими примітивами, має широкий спектр застосувань у різних галузях. Ця технологія забезпечує ефективне представлення та аналіз об'єктів, що може бути використана в таких галузях, як робототехніка, автоматизоване проектування, збереження культурної спадщини та інше. Спрощені моделі, отримані в результаті цього процесу, пропонують покращену обчислювальну ефективність, розширені можливості візуалізації та більш ефективне управління даними, що робить їх корисними у численних випадках використання.

### 8.1 Використання результату 3D реконструкції об'єкта

3D модель фізичного об'єкта має великий перелік сфер застосування. У цьому пункті наведено декілька з них, та приведено приклади використання 3D моделі об'єкта, отриманої у результаті маніпуляції ним людиною у руках.

#### *Робототехніка та автоматизація*

Маніпулювання об'єктами в руках перед RGBD-камерою дає змогу розробляти удосконалені роботизовані системи, які можуть ефективніше взаємодіяти з навколишнім середовищем. Отримуючи детальні 3D-моделі об'єктів під час маніпуляцій, роботи можуть покращити своє розуміння властивостей і геометрії об'єктів, що призводить до кращого захоплення та маніпуляційних можливостей. Наприклад, на складальних лініях роботи, оснащені RGBD-камерами, можуть точніше ідентифікувати та піднімати об'єкти, зменшуючи ймовірність помилок та оптимізуючи виробничий процес.

### *Мистецтво та збереження культурної спадщини*

У сфері збереження мистецтва та культурної спадщини використання 3D-реконструкції об'єкта, отриманого в результаті маніпуляції об'єкта у руках, може полегшити цифрове архівування та аналіз артефактів. Тримаючи та обертаючи артефакт перед RGBD-камерою, реставратори можуть створювати 3D-моделі з високою роздільною здатністю, які фіксують деталі поверхні та геометрію об'єкта. Ці цифрові зображення можна використовувати для подальшого аналізу, планування реставрації або створення віртуальних виставок для ширшого доступу громадськості. Крім того, 3D-моделі можуть слугувати цінною документацією для майбутніх поколінь, зберігаючи культурну спадщину в цифровому форматі.

### *Освіта та навчання*

Інтеграція 3D-реконструкції об'єктів може значно покращити освіту і навчання в різних галузях. Створюючи 3D-моделі об'єктів під час маніпуляцій, викладачі можуть створювати для студентів імерсивний навчальний досвід, дозволяючи їм візуалізувати та взаємодіяти з цифровими представленнями об'єктів реального світу. Наприклад, на курсах інженерії та дизайну студенти можуть вивчати складні деталі механічних компонентів або архітектурних споруд за допомогою 3D-моделей, що сприяє глибшому розумінню предмета.

### *Віртуальна і доповнена реальність*

Технології віртуальної і доповненої реальності (VR/AR) можуть отримати значну користь від поєднання 3D-реконструкції об'єктів, отриманих під час маніпуляції ними у руках людини. Захоплюючи 3D-моделі об'єктів під час маніпуляцій, розробники можуть створювати більш реалістичні та захоплюючі VR/AR-досвіди. Користувачі можуть відсканувати реальний об'єкт, а потім взаємодіяти з його віртуальною копією, що призводить до більш захопливого та природного досвіду.

### *Розваги та кінематограф*

Індустрія розваг також охопила технологію 3D-реконструкції об'єктів, особливо у кіно та візуальних ефектах (VFX). Створюючи 3D-моделі об'єктів, кінематографісти можуть створювати більш реалістичні та візуально приголомшливі сцени.

### *Пошук предмета*

Отримавши 3D модель предмета, значно спрощується процес його пошуку та розпізнавання у складних сценах. Такий підхід має потенціал до підвищення для точності та ефективної ідентифікації об'єктів у складних сценах, оскільки оперує не лише одним зображення шуканого об'єкта, а множиною зображень. Це підвищує шанс успішної ідентифікації об'єкта на сцені у будь-якому його положенні.

Отже, 3D-реконструкція об'єктів має широкий спектр застосування в різних галузях, від виробництва і контролю якості до індустрії розваг і доповненої реальності. Можливість створювати точні 3D-моделі об'єктів і середовищ суттєво вплинула на ці галузі, що призвело до розвитку технологій, вдосконалення процесів і створення більш захоплюючих вражень. А технологія, що дозволяє створювати 3D модель об'єкта, використовуючи всього одну RGBD камеру може значно спростити та полегшити цей процес.

## 8.2 Використання результату апроксимації 3D об'єкта просторовими примітивами

Апроксимація тривимірних моделей об'єктів просторовими примітивами останніми роками привертає значну увагу завдяки різноманітним перевагам, таким як спрощення складної геометрії, зменшення обчислювальної складності, а також покращення зберігання та керування даними. Цей підхід передбачає представлення 3D-об'єктів як комбінацій примітивних форм. Далі наведено потен-

цінні можливості застосування апроксимації 3D-моделей об'єктів просторовими примітивами у різних прикладних сферах.

#### *Розваги та кінематограф*

Використання просторових примітивів для представлення 3D-об'єктів є особливо корисним у галузі робототехніки та автоматизації. Апроксимуючи складну геометрію простішими формами, роботи можуть досягти більш ефективного планування шляху, уникнення зіткнень і планування захоплення. Наприклад, у виробничих умовах роботизованій руці може знадобитися піднімати і збирати різні компоненти. Апроксимуючи 3D-моделі цих компонентів за допомогою просторових примітивів, робот може швидко розрахувати оптимальні точки захоплення і траєкторії, скорочуючи загальний час обробки і підвищуючи ефективність.

#### *Комп'ютерне проектування (CAD) та інжиніринг*

Апроксимація просторовими примітивами має місце у застосуванні в САПР та інженерних додатках, де вони дозволяють дизайнерам та інженерам більш ефективно створювати та маніпулювати 3D-моделями. Розбиваючи складні об'єкти на простіші форми, користувачі можуть швидко оцінювати конструкції, виконувати симуляції та оптимізувати функціональність продукту. Наприклад в автомобільній промисловості дизайнери можуть використовувати просторові примітивні апроксимації для спрощення процесу проектування компонентів автомобіля. Спрощуючи геометрію складних деталей, інженери можуть легше проводити аналіз напружень і оптимізувати конструкцію для зменшення ваги та підвищення продуктивності.

#### *Віртуальна реальність (VR) і доповнена реальність (AR)*

Апроксимація просторовими примітивами може підвищити продуктивність систем віртуальної і доповненої реальності за рахунок зниження обчислювальних вимог, пов'язаних з рендерингом і взаємодією зі складними 3D-моделями. Використовуючи простіші представлення, ці системи можуть забезпечити більш

плавний і захоплюючий користувацький досвід.

### *3D-друк і швидке створення прототипування*

У сфері 3D-друку та швидкого прототипування апроксимація моделі просторовими примітивами може допомогти зменшити складність 3D-моделей, що призведе до скорочення часу нарізки та друку. Цей підхід також може допомогти оптимізувати якість друку та зменшити використання матеріалів.

Застосування апроксимації 3D-моделі об'єкта просторовими примітивами пропонує численні переваги в різних галузях і сферах застосування. Спрощуючи складну геометрію до комбінацій базових форм, ці апроксимації уможливають більш ефективні обчислення, управління даними та маніпуляції з об'єктами. Як наслідок, вони особливо цінні в робототехніці та автоматизації, комп'ютерному проектуванні та інженерії, віртуальній та доповненій реальності, 3D-друку а також у швидкому створенні прототипів.

## 9 ПАРАМЕТРИ, ЯКІ ВПЛИВАЮТЬ НА ЯКІСТЬ РЕКОНСТРУКЦІЇ 3D МОДЕЛІ ОБ'ЄКТА

Визначення оптимальної кількості фреймів, необхідних для реконструкції 3D-моделі об'єкта є критично важливим аспектом для отримання якісних результатів. Цей процес передбачає врахування різних факторів, які впливають на динаміку руху та якість отриманих зображень.

Стабільність руки відіграє важливу роль в отриманні чітких зображень під час маніпуляції. Стабільна рухів забезпечує меншу розмитість та більш точну інформацію про глибину, що призводить до кращої реконструкції. Нестабільні маніпуляції можуть призвести до необхідності робити більше знімків, щоб компенсувати втрату якості зображення.

Крім того, слід враховувати складність і геометрію об'єкта. Об'єкти зі складними деталями та складними формами можуть потребувати більшої кількості зображень, щоб точно відобразити всі особливості. І навпаки, прості об'єкти з меншою кількістю деталей можуть бути адекватно реконструйовані за допомогою меншої кількості зображень.

Іншим важливим фактором є швидкість обертання об'єкта. Швидша швидкість обертання може ускладнити отримання достатньої кількості зображень, що перекриваються, для точної 3D-реконструкції. Тому підтримка відповідної швидкості обертання протягом усього процесу маніпуляції має важливе значення для створення якісних 3D-моделей з використанням мінімальної кількості зображень.

### 9.1 Дослідження максимального кута повороту предмета, достатнього для реконструкції 3D моделі об'єкта

У процесі реконструкції 3D-моделі об'єкта обертання об'єкта відіграє вирішальну роль у тому, щоб забезпечити точне захоплення всієї поверхні об'єкта. Максимальна величина повороту об'єкта, достатня для реконструкції 3D-моделі,



Рисунок. 9.1 – RGB зображення предмета

відноситься до мінімального діапазону поворотів, необхідного для покриття всієї поверхні об'єкта, уникаючи при цьому надлишковостей і неоднозначностей.

Для дослідження оптимального кута повороту предмета, розглянемо один з елементів датасету з простою геометрією, зображено на Рис. 9.1

Для визначення максимального кута повороту предмета, достатнього для реконструкції 3D моделі об'єкта, було зібрано окремий невеликий датасет, що налічує 26 фреймів, які отримані шляхом обертання об'єкта навколо його вертикальної осі. Мінімальний кут повороту: 14.97553877 градусів. Результат реконструкції, моделі об'єкта зображено на Рис. 9.2

Візьмемо кожен другий фрейм із зібраного датасету. Тобто кут повороту між двома сусідніми фреймами 30 градусів. Результат реконструкції, моделі об'єкта зображено на Рис. 9.3



Рисунок. 9.2 – Результат реконструкції 3D моделі об'єкта при куті повороту 15.



Рисунок. 9.3 – Результат реконструкції 3D моделі об'єкта при куті повороту 30.

Збільшимо кут повороту до 45 градусів, взявши кожен 3 фрейм. Результат реконструкції, моделі об'єкта зображено на Рис. 9.4



Рисунок. 9.4 – Результат реконструкції 3D моделі об'єкта при куті повороту 45.

На Рис. 9.4 видно, що модель не вдалось зробити реконструкцію 3D моделі об'єкта.

Спираючись на отримані результати, можна зробити висновок, що оптимальний для реконструкції об'єкта з не складною геометрією кут поворота лежить у проміжку від 15 до 30 градусів. Чим складніша геометрія об'єкта та його розміри тим ближче максимальний кут поворота наблизатиметься до нижньої допустимої межі.

## 10 РОЗРОБКА СТАРТАП ПРОЕКТУ “РЕКОНСТРУКЦІЯ 3D МОДЕЛІ ПРЕДМЕТА У РУЦІ ЛЮДИНИ НА ОСНОВІ ДАНИХ RGB-D КАМЕРИ ТА АПРОКСИМАЦІЯ ЇЇ ГЕОМЕТРІЇ ПРОСТОРОВИМИ ПРИМІТИВАМИ”

### 10.1 Опис ідеї проекту

У даній магістерській роботі представлено алгоритм для реконструкції 3D моделі предмета у руці людини на основі даних RGB-D камери та апроксимації геометрії цієї моделі просторовими примітивами. Дані отримується під час маніпуляції предметом у руках людини перед стаціонарно встановленою RGB-D камерою. Алгоритм покликаний спростити процес створення 3D моделі об'єкта, а також створити його зручне представлення за допомогою просторових примітивів. Алгоритм не потребує значної кількості сенсорів та девайсів, лише одну RGB-D камеру.

Для більш детального опису запропонованої ідеї наведено Табл. 10.1

Таблиця 10.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Алгоритм для реконструкції 3D моделі предмета у руці людини на основі даних RGB-D камери та	1. Робототехніка та автоматизація	Отримавши реконструйовану модель об'єкта, або її апроксимацію робот може виконувати автоматизований захват об'єкта.
	2. Мистецтво та збереження культурної спадщини	Оцифровка об'єктів культурної спадщини

Таблиця 10.1 (продовження)

апроксимації геометрії цієї моделі просторовими примітивами	3. Освіта та навчання	Створюючи 3D-моделі об'єктів під час маніпуляцій, викладачі можуть створювати для студентів імерсивний навчальний досвід, дозволяючи їм візуалізувати та взаємодіяти з цифровими представленнями об'єктів реального світу.
	4. Віртуальна і доповнена реальність	Захоплюючи 3D-моделі об'єктів під час маніпуляцій, розробники можуть створювати більш реалістичні та захоплюючі VR/AR-досвіди.

Наступний етап передбачає оцінку сильних і слабких сторін системи для визначення потенціалу впровадження на ринок та конкурентоспроможності. Відповідна інформація наведена в Таблиці 10.2.

Таблиця 10.2 – Сильні, слабкі та нейтральні характеристики проекту

Техніко-економічні характеристики ідеї	Потенційні товари/концепції конкурентів			Недоліки	Нейтральні сторони	Переваги
	Поточний проект	Конкурент 1	Конкурент 2			
Форма виконання	Детермінований алгоритм	Детермінований алгоритм	Детермінований алгоритм			+
Потреба у зборі даних	Ні	Ні	Так			+
Кросплатформеність	Так	Ні	Ні		+	
Універсальність	Так	Ні	Так		+	

Перевагами запропонованого алгоритму є його простота використання - людині достатньо "покрутити" предмет перед камерою, ефективність та відсутність

потреби у великій кількості обладнання та попередньому зборі даних. Алгоритм має велику кількість сфер застосування.

## 10.2 Технологічний аудит ідеї

В даному розділі буде проведено аудит технологій, що використовувались для реалізації запропонованої в дипломній роботі ідеї (таблиця 10.3).

Таблиця 10.3 – Технологічна здійсненість проекту

Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
Алгоритм для реконструкції 3D моделі предмета у руці людини на основі даних RGB-D камери та апроксимації геометрії цієї моделі просторовими примітивами	Python	Наявна	Безкоштовна, доступна
	C++	Наявна	Безкоштовна, доступна
	Open3D	Наявна	Безкоштовна, доступна
	PCL	Наявна	Безкоштовна, доступна
	OpenCV	Наявна	Безкоштовна, доступна
	NumPy	Наявна	Безкоштовна, доступна
	MediaPipe Hand	Наявна	Безкоштовна, доступна

Усі обрані бібліотеки та фреймворки, що були використані для реалізації алгоритму, запропонованого в дипломній роботі є безкоштовними та доступними для використання в наукових цілях.

Таким чином, об'єднання C++, Python, OpenCV, NumPy, PCL, Open3D та MediaPipe Hand забезпечує комплексний інструментарій для реалізації запропонованого в магістерській роботі алгоритму.

### 10.2.1 Аналіз ринкових можливостей

Аналіз ринку має першорядне значення у формуванні стратегічного напрямку розвитку стартапу. Він дозволяє отримати уявлення про галузевий ландшафт, вподобання споживачів та позиціонування конкурентів. Проводячи ретельний аналіз ринку, можна визначити потенційні можливості для зростання, виявити незайняті ніші та розробити ефективні маркетингові стратегії, адаптовані до своєї цільової аудиторії. У Табл. 10.4 наведено попередню характеристику потенційного ринку для стартап-проекту

Таблиця 10.4 – Попередня характеристика потенційного ринку для стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	10
2	Загальний обсяг продаж, грн/ум.од	1000
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу	Висока вартість початкового капіталу
5	Специфічні вимоги до стандартизації та сертифікації	-
6	Середня норма рентабельності в галузі (або по ринку), %	$R = (3000000 * 10) / (1000000 * 12) = 25\%$

Можемо зробити висновок, що проект має досить високу рентабельність.

Далі визначено перелік потенційних груп клієнтів групи клієнтів, їх характеристики, та сформовано орієнтовний перелік вимог для кожної групи, наведено у Табл 10.5.

Таблиця 10.5 – Характеристика потенційних клієнтів стартап-проекту

Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
Алгоритм для реконструкції 3D моделі об'єкта та апроксимації його моделі просторовими примітивами	а) Автоматизація та робототехніка б) Мистецтво та збереження культурної спадщини в) Освіти та наука г) Віртуальна і доповнена реальність	а) спрощення процесу сканування об'єкта б) система, що працює з мінімальною кількістю обладнання	Алгоритм має бути надійним, швидким, точним. Простим у використанні.

Таким чином, основними вимогами до продукту є надійним, швидким, точним. Простота використання.

Розглянемо фактори загроз та можливих реакцій на них у Табл.10.6.

Таблиця 10.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Зростаюча вимогливість покупців	Зростання вимог до ефективності роботи алгоритму та точності	Проведення досліджень для покращення алгоритму

Таблиця 10.6 (продовження)

2	Збільшення витрат на технічну підтримку	Застарілість алгоритму відносно розвитку індустрії	Своєчасне оновлення алгоритму
3	Конкуренція	Вихід на ринок великої компанії	Своєчасне оновлення алгоритму
4	Зменшення кількості замовників	Зменшення зацікавленості замовників у наданих сервісах	Додати нові можливості до програмного продукту, що можуть зацікавити потенційних користувачів

Було визначено основні фактори ризику для стартапу, серед яких найбільш значущим викликом є зміна вимог користувачів та швидка зміни потреб індустрії. Для подолання цих ризиків необхідно своєчасно додавати нові можливості для алгоритму та слідкувати за змінами потреб в індустрії. Розглянемо фактори можливостей в Табл. 10.7.

Таблиця 10.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Зростання попиту на програмний продукт	Зростання попиту на програмний продукт за рахунок простоти та швидкості його використання	Популяризація програмного продукту

Таблиця 10.7 (продовження)

2	Зменшення попиту до продукту конкурентів	Програмні продукти конкурентів почали втрачати свою актуальність у зв'язку з низкою факторів	Врахувати помилки конкурентів. Покращити свій програмний продукт
3	Зменшення витрат на підтримку програмного продукту	Автоматизація процесів підтримки та тестування програмного продукту	Займатись автоматизацією процесів в компанії та підвищувати кваліфікацію співробітників

Можна зробити висновок, що ключовою точкою для розвитку можливостей поширення програмного продукту та утримання цільової аудиторії користувачів є підтримка стабільної роботи продукту та своєчасна реакція на зміни ринку та потреби користувачів.

Проведемо ступеневий аналіз конкуренції на ринку. Наведено в Табл. 10.8.

Таблиця 10.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції - олігополія	Мізерна кількість конкурентів на ринку	Врахування недоліків конкурентів. Покращення технології програмного продукту.

Таблиця 10.8 (продовження)

2. За рівнем конкурентної - інтернаціональна	Більшість компаній-конкурентів іноземні.	Врахування недоліків конкурентів. Покращення технології програмного продукту.
3. За галузевою ознакою - внутрішньогалузева	Сервіси використовуються у вузьких рамках однієї сфери	Спробувати поширити використання програмного продукту на інші сфери
4. Конкуренція за видами товарів: - товарно-видова	Відмінність у програмному продукті, але схожість у функціях	Покращувати показники роботи алгоритму
5. За характером конкурентних переваг - нецінова	Висока собівартість програмного продукту	Збільшення функціоналу програмного продукту
6. За інтенсивністю - не марочна	Відсутність бренду	Побудова бренду компанії. Вихід на ринок з високою впізнаваністю.

Для даного проекту не можна провести аналіз конкуренції за М.Портрером, оскільки у даній роботі наведено новий алгоритм і його конкурентами є інші алгоритми, описані в дослідницьких публікаціях, а не готові програмні продукти якихось компаній. Натомість наведемо фактори конкурентоспроможності для запропонованого в даній науковій роботі алгоритму. Наведено в Табл. 10.9

Таблиця 10.9 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
-------	-------------------------------	---

Таблиця 10.9 (продовження)

1	Науково-технічний потенціал	Чим краще досліджена проблематика - тим краще функціонал програмного продукту
2	Розвиток індустрії	Чим більше уваги до індустрії - тим більше фінансування

Проведемо порівняльний аналіз слабких та сильних сторін, для наведених вище факторів. Табл. 10.10

Таблиця 10.10 – Сильні та слабкі сторони стартап-проекту

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів						
			-3	-2	-1	0	1	2	3
1	Науково-технічний потенціал	20				+			
2	Розвиток індустрії	20				+			

Наступним кроком проведемо SWOT аналіз (Табл. 10.11).

Таблиця 10.11 – SWOT- аналіз стартап-проекту

Сильні сторони: інноваційний алгоритм	Слабкі сторони: відсутність впізнаваності та бренду
Можливості: стрімкий розвиток індустрії	Загрози: конкуренти більш впізнавані і мають більше можливостей, як кадрових, так і фінансових.

За допомогою SWOT - аналізу було виявлено сильні та слабкі сторони стартап-проекту. А саме - новизна запропонованого алгоритму, а слабкою стороною є відсутність бренду та його впізнаваності взагалі. Стрімкий розвиток індустрії надає можливості для розвитку програмного продукту, проте, це є і загрозою, адже існують більші компанії. Вони мають великий штат інженерів та більше технічні, технологічні та фінансові можливості.

Складемо альтернативи впровадження проекту на основі SWOT аналізу. Наведено в Таблиці 10.12.

Таблиця 10.12 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Суттєве покращення алгоритму	70%	2 роки
2	Розвиток бренду та підвищення впізнаваності	30%	1 рік

### 10.3 Розробка ринкової стратегії проекту

Розробка ринкової стратегії проекту починається з визначення визначення цільової аудиторії проекту. Почнемо з визначення цільових груп. Розглянуто в 10.13.

Таблиця 10.13 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Складність входу у сегмент
1	Сфера робототехніки	Готові, якнайшвидше	Попит - високий	Інтенсивна	Висока
2	Доповнена реальність та VR	Готові, якнайшвидше	Попит - середній	Інтенсивна	Висока
3	Освіта та наука	Можливо	Попит - середні	Не інтенсивна	Середня

Було обрано цільові групи: компанії, що займаються робототехнікою та автоматизацією, компанії що займаються аугментацією, а також, можливо освітні заклади. Маючи це представлення, визначимо базову стратегію розвитку стартап-проекту. Наведено в Табл.10.14.

Таблиця 10.14 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкуренто-спроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Запропонований в даній роботі алгоритм є новим. Основною його перевагою є простота використання	Створення інновацій	Основна позиція - новизна та простота, це є ключовою перевагою даного проекту	Стратегія диференціації

Далі буде описано вибір стратегії конкурентної поведінки, наведено в Табл. 10.15.

Таблиця 10.15 – Визначення базової стратегії розвитку

№ п/п	Чи є проект «першопроходцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Так	Шукати нових	Надійність, точність	Оборонна

Визначимо стратегія позиціонування проекту, яка допоможе користувачам ідентифікувати програмний продукт Табл. 10.17.

Таблиця 10.16 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкуренто-спроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Надійність, точність, простота використання	Стратегія диференціації	Основна позиція - новизна	Надійність, точність, простота використання

У даному розділі було визначено стратегію позиціонування програмного продукту, ними є надійність, точність та простота використання алгоритму.

### 10.3.1 Розробка маркетингової програми

Для того щоб виконати розробку маркетингової програми повною мірою, найперше- потрібно визначити маркетингову концепцію товару, який пропонується споживачу. У Табл.10.18 наведено підсумок аналізу конкурентоспроможності товару, що була наведена вище.

Таблиця 10.17 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода для користувачів	Ключові переваги перед конкурентами
1	Створення 3D моделі об'єкта та апроксимації його геометрії з даних, отриманих під час маніпуляції предметом у руках	Простота використання	Дані отримуються під час маніпуляції об'єктом у руках.

Розробимо трирівневу маркетингову модель товару. Табл. 10.18.

Таблиця 10.18 – Визначення ключових переваг концепції потенційного товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Створення 3D моделі об'єкта та апроксимації його геометрії з даних, отриманих під час маніпуляції предметом у руках		
II. Товар у реальному виконанні	Властивості / характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	Надійність	М	Тл Тл Тл
	Точність	М	
	Простота	М	
Спрощує процес створення 3D моделі реальних об'єктів			

Таблиця 10.18 (продовження)

III. Товар із підкріпленням	Модернізація алгоритму
	Підтримка

Визначимо цінові межі за допомогою експертного метода. Табл 10.19

Таблиця 10.19 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	Не визначено	Не визначено	Не визначено	Можлива підписка 10\$/міс

У визначенні цінової політики неможливо спиратись на середній рівень товарів - замінників.

Визначимо оптимальну систему збуту. Табл. 10.20.

Таблиця 10.20 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Пробний період	Фінансування витрат на функціонування каналу збуту	Канал 0 рівня	Проводити збут власними силами

Найбільш ефективним є пропонування системи збуту у вигляді місячної підписки. Останнім кроком є розробка маркетингових комунікацій. Табл. 10.21.

Таблиця 10.21 – Формування системи збуту

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Компанії, що займаються робототехнікою, аугментацією реальності. Наукові заклади.	Презентація на конференціях, медіа	Спрощення процесу створення 3D моделі об'єкта	Привернути увагу до програмного продукту, показати його переваги	Простота процесу сканування

**Висновки:** У даному розділі було розглянуто стартап-проект на тему “Реконструкція 3D моделі предмета у руці людини на основі даних RGB-D камери та апроксимація її геометрії просторовими примітивами”. Було визначено основну ідею роботи, проведено технологічний аудит, в якому розглянуто доцільність використання бібліотек та фреймворків. Було проведено аналіз ринкових можливостей, аналіз ризиків та можливостей, визначено сильні та слабкі сторони. Також був проведений аналіз цільової аудиторії. На основі всіх цих даних була розроблена стратегія виходу на ринок та маркетингова програма.

## ВИСНОВКИ

У представленій магістерській дисертації було розглянуто задачу реконструкції 3D моделі предмета у руці людини на основі даних RGB-D камери та апроксимація її геометрії просторовими примітивами. Враховуючи швидкий розвиток технологій комп'ютерного зору та зростання потреби у зручному способі взаємодії людини з комп'ютерними системами, дана робота є актуальною, адже пропонує швидкий і зручний спосіб отримання 3D моделі об'єкта та її апроксимації з використанням мінімальної кількості обладнання та мінімальною взаємодією з предметом, який потрібно відсканувати.

У ході роботи було розглянуто існуючі підходи до отримання 3D моделі об'єкта. Перевагою запропонованого алгоритму є те, що він пропонує більш адаптивний метод для виділення сегмента предмета у руці людини, що працює з будь-яким кольором шкіри людини та не залежить від умов освітленості приміщення у якому відбувається сканування. Запропонований у магістерській дисертації алгоритм є детермінованим. Це означає, що він не потребує збору великої кількості даних для навчання та працює з більш широкою множиною об'єктів та геометрій.

Алгоритм апроксимації геометрії об'єкта просторовими примітивами є легко масштабованим, тобто його можна розширити більшим переліком просторових примітивів, ніж запропоновано у даній роботі.

Проте, запропонований у даній магістерській дисертації алгоритм має перелік обмежень. Він працює лише з предметами, що не деформуються у процесі взаємодії з ними (твердими тілами). Об'єкт має бути виготовлений з непрозорого, невідблискуючого матеріалу, це обмеження обумовлене технічними можливостями використаного в роботі RGB-D сенсора. Об'єкт має бути текстурованим, або мати на собі малюнок. Це обмеження обумовлюється алгоритмом реконструкції моделі об'єкта.

Сфера застосування представленого алгоритму є досить широкою. Він може бути застосований у сфері робототехніки для автоматизованого захвату предметів маніпулятором. Може бути використаний у сфері віртуальної та доповненої

реальності, для швидкого сканування реальних об'єктів для додавання їх на віртуальну сцену. Алгоритм, також, може бути використаний для створення 3D моделей реальних об'єктів, а спрощене представлення об'єктів за допомогою просторових примітивів спрощує процес відтворення об'єкта на 3D принтері.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Fan Wang, Kris Hauser *In-hand Object Scanning via RGB-D Video Segmentation*. In International Conference on Robotics and Automation, 2019
2. Hasson, Y., Varol, G., Tzionas, D., Kalevatykh, I., Black, M.J., Laptev, I., Schmid, C. *Learning Joint Reconstruction of Hands and Manipulated Objects*. CVPR 2019
3. T. Weise, T. Wismer, B. Leibe, and L. V. Gool *In-hand scanning with online loop closure*. 2009 IEEE 12th International Conference on Computer Vision Workshop
4. V. Vezhnevets, V. Sazonov, and A. Andreeva *A survey on pixel-based skin color detection techniques*. IN PROC. GRAPHICON-2003, pp. 85–92
5. P. J. Besl and N. D. McKay. *method for registration of 3-d shapes*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 239–256, 1992
6. Papon, J., Abramov, A., Schoeler, M., Worgotter, F *Voxel Cloud Connectivity Segmentation—Supervoxels for Point Clouds*. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2027–2034
7. S. C. Stein; M. Schoeler; J. Papon; F. Wörgötter *Object Partitioning Using Local Convexity* 2014 IEEE Conference on Computer Vision and Pattern Recognition
8. Jacob E. Goodman, Joseph O'Rourke and Piotr Indyk, *Handbook of Discrete and Computational Geometry* CRC press 2009
9. Bentley, J. L. *Multidimensional binary search trees used for associative searching*. Communications of the ACM. 509–517pp 1975
10. *Clustering of Pointclouds into Supervoxels* [Електроний ресурс] - [https://pcl.readthedocs.io/projects/tutorials/en/latest/supervoxel\\_clustering.html](https://pcl.readthedocs.io/projects/tutorials/en/latest/supervoxel_clustering.html) Дата доступу – 07.05.2023

11. *How to use a KdTree to search* [Электроний ресурс] - [https://pcl.readthedocs.io/projects/tutorials/en/latest/kdtree\\_search.html](https://pcl.readthedocs.io/projects/tutorials/en/latest/kdtree_search.html) Дата доступа – 07.05.2023
12. Marius Muja and David G. Lowe *Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration* International Conference on Computer Vision Theory and Applications, 2009
13. D. Lowe *Distinctive Image Features from Scale-Invariant Keypoints*. International Journal of Computer Vision 60(2), 91–110 (2004)
14. *Introduction to SIFT (Scale-Invariant Feature Transform)* [Электроний ресурс] - [https://docs.opencv.org/4.x/da/df5/tutorial\\_py\\_sift\\_intro.html](https://docs.opencv.org/4.x/da/df5/tutorial_py_sift_intro.html) Дата доступа – 07.05.2023
15. *Point cloud outlier removal* [Электроний ресурс] - [http://www.open3d.org/docs/latest/tutorial/Advanced/pointcloud\\_outlier\\_removal.html](http://www.open3d.org/docs/latest/tutorial/Advanced/pointcloud_outlier_removal.html) Дата доступа – 07.05.2023
16. Zhengyou Zhang , Kondratova L.P. *Iterative Closest Point (ICP)* Computer Vision pp 433–434Cite, 2016
17. *ICP registration* [Электроний ресурс] - [http://www.open3d.org/docs/release/tutorial/t\\_pipelines/t\\_icp\\_registration.html](http://www.open3d.org/docs/release/tutorial/t_pipelines/t_icp_registration.html) Дата доступа – 07.05.2023
18. Martin A. Fischler, Robert C. Bolles *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography* Communications of the ACM, pp 381–395, 1981
19. R. B. Rusu, N. Blodow; M. Beetz *Fast Point Feature Histograms (FPFH) for 3D registration* IEEE International Conference on Robotics and Automation, 2009
20. *Hand landmarks detection* [Электроний ресурс] - [https://developers.google.com/mediapipe/solutions/vision/hand\\_landmarker](https://developers.google.com/mediapipe/solutions/vision/hand_landmarker) Дата доступа – 07.05.2023

21. *MediaPipe Hands* [Електроний ресурс] - <https://github.com/google/mediapipe/blob/master/docs/solutions/hands.md> Дата доступу – 07.05.2023

22. *Intel Realsense D-435 specification* [Електроний ресурс] - <https://www.intelrealsense.com/depth-camera-d435/> Дата доступу – 07.05.2023

23. Харченко К.В., Вовк. І.С. *Алгоритм розрахунку точок для захвату об'єкта маніпулятором на основі апроксимації 3D моделі об'єкта просторовими примітивами*. Системні науки та інформатика: збірник доповідей І науково-практичної конференції «Системні науки та інформатика», 22–29 листопада 2022 року, Київ. – К., НН ІІСА КПІ ім. Ігоря Сікорського, 2022. – с. 375-376.