

Вінницький Національний Технічний Університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення

Бакалаврська дипломна робота

на тему: Розробка методів і програмних засобів антропометричних
вимірювань з використанням тривимірного моделювання

Виконав: студент IV курсу, групи 2ПІ-206
спеціальності

121 – Інженерія програмного забезпечення
(шифр і назва напрямку підготовки, спеціальності)

Захарчук Максим Дмитрович
(прізвище та ініціали)

Керівник: д.т.н., професор каф. ПЗ Романюк О.Н.
(прізвище та ініціали)

Рецензент: д.т.н., професор каф. ОТ Мартинюк Т.Б.
(прізвище та ініціали)

Допущено до захисту
Зав. кафедри ПЗ О.Н. Романюк
«10» червня 2024 р.

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення
Рівень вищої освіти перший (бакалаврський)
Галузь знань 12 – «Інформаційні технології»
Спеціальність 121 – «Інженерія програмного забезпечення»
Освітньо-професійна програма – «Інженерія програмного забезпечення»



ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
д.т.н., професор
О. Н. Романюк _____
«12» березня 2024 р.

ЗАВДАННЯ НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Захарчуку Максиму Дмитровичу

1. Тема роботи – розробка методів і програмних засобів антропометричних вимірювань з використанням тривимірного моделювання.

Керівник роботи: Романюк Олександр Никифорович, д.т.н., професор, завідувач кафедри ПЗ, затверджені наказом вищого навчального закладу від 11 березня 2024 р. № 80.

2. Строк подання студентом роботи 3 червня 2024.

3. Вихідні дані до роботи: методи антропометричного вимірювання – алгоритм SMPLX та розроблені методи антропометричного вимірювання; моделі – тривимірні моделі тіла людини; графічний режим – TrueColor; вихідні дані для вимірювання – антропометричні виміри та координати суглобів та точок антропометрії; дані про антропометричні параметри.

4. Зміст текстової частини: вступ; аналіз стану антропометричних вимірювань; аналіз аналогів; аналіз методів розв'язання задачі; постановка задачі; розробка методів визначення площі та об'єму на 3Д моделі; розробка метод визначення точок антропометрії; розробка методу вимірювання відстані на тривимірній моделі; розробка алгоритмів роботи програми; розробка програмного забезпечення; розробка інтерфейсу програми; тестування розробленого програмного забезпечення, створення інструкції користувача.

5. Перелік ілюстративного матеріалу: титульний слайд, актуальність дослідження, галузі застосування, використання тривимірного моделювання в антропометрії, роль ваги та площі в медичній практиці, мета дослідження, основні етапи підготовки тривимірної моделі, розробка методів антропометричного вимірювання з використанням тривимірного моделювання,

алгоритми роботи, розробка програмних модулів, інтерфейс, тестування програми.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1-4	Романюк О.Н., д.т.н., професор каф ПЗ	13.03.24	03.06.24

7. Дата видачі завдання 13 березня 2024 року

КАЛЕНДАРНИЙ ПЛАН

№ з\п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз підходів до антропометричних вимірювань з використанням тривимірного моделювання	14.03.24 – 22.03.24	Вик.
2	Розробка методів антропометричних вимірювань з використанням тривимірного моделювання	22.03.24 – 19.04.24	Вик.
3	Розробка алгоритмів роботи програмного забезпечення	19.04.24 – 10.05.24	Вик.
4	Розробка програмного забезпечення	10.05.24 – 24.05.24	Вик.
5	Тестування програмного забезпечення	24.05.24 – 27.05.24	Вик.
6	Оформлення матеріалів до захисту БДР	27.05.24 – 03.06.24	Вик.

Студент М. Д. Захарчук
(підпис) (ініціали та прізвище)

Керівник бакалаврської дипломної роботи О. Н. Романюк
(підпис) (ініціали та прізвище)

АНОТАЦІЯ

УДК 00492

Захарчук М. Д. Розробка методів і програмних засобів антропометричних вимірювань з використанням тривимірного моделювання : бакалаврська дипломна робота зі спеціальності 121 – інженерія програмного забезпечення, освітня програма – інженерія програмного забезпечення. Вінниця: ВНТУ, 2024. 75 с.

На укр. мові. Бібліогр. : 22 назв ; рис. : 39; табл. 2.

У бакалаврській дипломній роботі було розроблено методи та програмні засоби для антропометричного вимірювання з використанням засобів тривимірного моделювання. Проведено аналіз сучасних підходів до антропометричних вимірювань з використанням тривимірного моделювання. Обґрунтовано необхідність розробки методів виконання антропометричних вимірювань з використанням тривимірного моделювання з метою підвищення точності вимірювання антропометричних параметрів людини.

Розроблено методи вимірювання антропометричної відстані на тривимірній моделі тіла людини використовуючи засоби тривимірного моделювання та розроблені формули та методи, що дозволяють з максимальною точністю визначити антропометричні параметри тривимірної моделі людини.

На основі проведених у БДР досліджень розроблено алгоритми та програмні засоби для антропометричних вимірювань.

Проведене тестування підтвердило достовірність теоритичних досліджень методів антропометричних вимірювань та засобів їх реалізації. Розроблені засоби вимірювання антропометричних параметрів можна використати в медичній галузі для визначення особливостей лікування.

Ключові слова: тривимірне моделювання, 3Д модель, антропометрія, вимірювання, програмне забезпечення, 3Д модель людини.

ABSTRACT

UDC 00492

Zakharchuk M. D. Development of methods and software tools for anthropometric measurements using three-dimensional modeling : bachelor's thesis in specialty 121 - Software Engineering, educational program - Software Engineering. Vinnytsia: VNTU, 2024. 75 c.

In Ukrainian. Bibliography: 22 titles; Figures: 39; Table 2.

In the bachelor's thesis, methods and software tools for anthropometric measurement using three-dimensional modeling were developed. The analysis of modern approaches to anthropometric measurements using three-dimensional modeling was carried out. The necessity of developing methods for performing anthropometric measurements using three-dimensional modeling in order to improve the accuracy of measuring human anthropometric parameters is substantiated.

Methods for measuring anthropometric distance on a three-dimensional model of the human body using three-dimensional modeling tools have been developed, and formulas and methods have been developed that allow to determine the anthropometric parameters of a three-dimensional human model with maximum accuracy.

Based on the research conducted at the GDR, algorithms and software tools for anthropometric measurements were developed.

The testing confirmed the reliability of theoretical studies of anthropometric measurement methods and means of their implementation. The developed means of measuring anthropometric parameters can be used in the medical field to determine the features of treatment.

Keywords: three-dimensional modeling, 3D model, anthropometry, measurements, software, 3D human model.

ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ ПІДХОДІВ ДО АНТРОПОМЕТРИЧНИХ ВИМІРЮВАНЬ З ВИКОРИСТАННЯМ ТРИВИМІРНОГО МОДЕЛЮВАННЯ.....	9
1.1 Аналіз стану технологій антропометричних вимірювань з використанням тривимірного моделювання	9
1.2 Аналіз методів розв’язання задачі	11
1.3 Аналіз аналогів.....	14
1.4 Постановка задачі дослідження	20
1.5 Висновки.....	22
2 РОЗРОБКА МЕТОДІВ АНТРОПОМЕТРИЧНИХ ВИМІРЮВАНЬ З ВИКОРИСТАННЯМ ТРИВИМІРНОГО МОДЕЛЮВАННЯ.....	23
2.1 Розробка методів визначення площі та об’єму фігур з використанням тривимірного моделювання	23
2.2 Розробка методу визначення точок антропометрії на основі даних тривимірної моделі.....	28
2.3 Розробка методу вимірювання антропометричних параметрів на тривимірній моделі	31
2.4 Висновки	34
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АНТРОПОМЕТРИЧНИХ ВИМІРЮВАНЬ З ВИКОРИСТАННЯМ ТРИВИМІРНОГО МОДЕЛЮВАННЯ	35
3.1 Розробка алгоритмів і програм антропологічних вимірювань	35
3.2 Розробка модуля визначення основних точок антропометрії на тривимірній моделі	42
3.3 Розробка модуля вимірювання антропометричних параметрів на тривимірній моделі	47
3.4 Розробка модуля інтерфейсу користувача	52
3.5 Висновки	57

4 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	58
4.1 Тестування програмного забезпечення.....	58
4.2 Розробка інструкції користувача	67
4.3 Висновки	69
ВИСНОВКИ.....	70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	72
ДОДАТКИ.....	76
ДОДАТОК А (Обов’язковий). Технічне завдання	77
ДОДАТОК Б (Обов’язковий). Протокол перевірки на плагіат	80
ДОДАТОК В (Обов’язковий). Лістинг програмного забезпечення	81
ДОДАТОК Г (Обов’язковий). Графічна частина	103

ВСТУП

Обґрунтування вибору теми дослідження.

Сучасний етап розвитку технологій вимагає постійного вдосконалення методів аналізу та вимірювань, зокрема в галузі антропометрії. Однією з важливих задач у вимірюваннях масо-вагових параметрів людського тіла є висока точність та надійність отриманих даних [1].

Традиційні методи антропометрії часто обмежені двовірними підходами та недостатньою деталізацією [2]. У цьому контексті, використання 3D-сканерів, здатних реєструвати геометричні особливості тіла в тривимірному просторі, відкриває нові перспективи для точного визначення параметрів, що має значущий потенціал для розвитку як у медичній галузі, так і в суміжних областях.

Тривимірні моделі людини є найреалістичнішими, відображають анатомічну структуру, точно передають рельєфні та кольорові особливості об'єкта, підлягають модифікації для зміни зовнішності. 3D-модель людини є багатофакторним джерелом інформації про пацієнта, дозволяє істотно знизити необхідний обсяг взаємодії з користувачем порівняно з існуючими методами [3].

Точність отриманих даних в антропометрії, особливо на тривимірних моделях, є критичною для забезпечення надійності та об'єктивності результатів вимірювань. У медичній сфері це має особливе значення, оскільки точні дані про антропометричні параметри дозволяють здійснювати більш точну діагностику та планування лікування.

Наприклад, у випадках, коли існує необхідність у визначенні об'єму тіла та розподілу маси, точні дані є ключовим елементом для розробки індивідуальної стратегії лікування. Надмірна маса тіла або ожиріння можуть бути факторами ризику для розвитку різних захворювань, таких як серцево-судинні захворювання, діабет типу 2 та інші. Точне визначення антропометричних параметрів допомагає медичним працівникам ефективніше діагностувати стан пацієнтів, прогнозувати ризики та призначати належне лікування.

Наразі існує невелика кількість програмних засобів, які дозволяють вимірювати антропометричні параметри. Наявні аналоги не мають високої точності вимірювання, негативно впливають на точність діагностування.

Тому розробка програмного забезпечення для антропометричних вимірювань з використанням тривимірного моделювання є актуальною задачею, яка має високу практичну значимість.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалася згідно плану виконання наукових досліджень на кафедрі програмного забезпечення.

Мета та завдання дослідження. Метою дослідження є підвищення точності вимірювання антропометричних параметрів людини за рахунок використання тривимірних моделей, що дозволяють реєструвати усі деталі геометрії тіла людини.

Відповідно до поставленої мети в бакалаврській дипломній роботі потрібно вирішити такі **задачі**:

- аналіз предметної галузі;
- розробка алгоритму обробки та аналізу даних, отриманих із тривимірної моделі тіла людини.
- розробити алгоритм побудови тривимірних моделей тіла людини на основі антропометричних даних та їх тривимірної візуалізації;
- розробити алгоритми вимірювання, які забезпечать високу точність та надійність отриманих даних на основі тривимірних моделей;
- розробка алгоритму визначення позицій суглобів, який буде спроможний виявляти, відстежувати та відображати позиції 16 основних суглобів, необхідних для вимірювань;
- покращення процесу вимірювання антропометричних даних із забезпеченням точності даних.
- тестування розроблених програмних продуктів;

Об'єкт дослідження – процес вимірювання антропометричних параметрів з використанням тривимірного моделювання.

Предмет дослідження – методи та засоби вимірювання антропометричних параметрів з використанням тривимірного моделювання.

Методи дослідження. У процесі дослідження застосовувалися: теорія антропометричних вимірювань; теорія алгоритмів для обробки та аналізу даних із тривимірних моделей та методи тривимірного моделювання для вимірювання антропометричних відстаней та визначення позицій точок антропометрії, комп'ютерне моделювання для аналізу та перевірки отриманих теоретичних положень.

Наукова новизна отриманих результатів:

1. Вперше запропоновано метод визначення об'єму тіла людини з використанням тривимірного моделювання, особливість якого полягає у виконанні горизонтальних зрізів тривимірної моделі заданої товщини, виконання тріангуляції та визначення сумарного об'єму отриманих складових призм, що дозволяє підвищити точність вимірювання.

2. Вперше запропоновано метод визначення площі тіла людини з використанням тривимірного моделювання, особливість якого полягає у формуванні текстурної розгортки поверхні людини на горизонтальну площину з подальшим визначенням сумарної площі отриманих фігур, що дозволяє підвищити точність вимірювання.

3. Вперше запропоновано метод визначення антропометричних відстаней на тривимірній моделі, на основі позицій 16 основних суглобів, особливість якого полягає у використанні нової формули визначення відстані між двома точками на тривимірній моделі тіла людини, у застосуванні горизонтальної кореляційної лінії через вершини трикутників та обрахування відстані використовуючи формули векторів. Це дає можливість автоматизувати процес вимірювання.

Практичне значення отриманих результатів полягає в тому, що на основі теоретичних досліджень розроблено алгоритми та програми для антропометричних вимірювань та їх інтеграції в професійний графічний рушій idx3d.

Особистий внесок здобувача. Усі наукові результати отримано здобувачем самостійно. У працях, опублікованих у співавторстві, студенту належать: [1] – проаналізовано методи використання засобів тривимірного моделювання у медичній галузі; [2] – запропоновано нові методи та формули визначення масо-вагових параметрів з використанням тривимірного моделювання; [3] – проаналізовано алгоритм LSPC; [4] – проаналізовано можливість використання засобів тривимірного моделювання у медицині; [5] – наведено методику визначення генетичних захворювань за тривимірними моделями лица; [6] – виконано аналіз програмне забезпечення для морфінгу зображень; [7] – проаналізовано програмне забезпечення для діагностування захворювань; [8] – наведено формули та алгоритми для визначення векторів нормалей; [9] – проаналізовано особливості застосування вокселів; [10] – проаналізовано особливості форматів цифрових зображень; [11] – наведено методики визначення захворювань за тривимірними моделями; [12] – проаналізовано процес текстурювання тривимірної моделі; [13] – наведено формули та алгоритми рендерингу тривимірних моделей залежно від позиції світла.

Апробація матеріалів бакалаврської дипломної роботи. Основні положення БДР доповідалися та обговорювалися на Міжнародних і Всеукраїнських конференціях: Всеукраїнська науково-практична інтернет-конференції студентів аспірантів та молодих науковців «Молодь в науці: дослідження, проблеми, перспективи (МН-2020)» (м. Вінниця, 2021), Міжнародна науково-практична конференція «Прикладні питання математичного моделювання» (м. Херсон, 2021), XXI Всеукраїнська науково-технічна конференція молодих вчених, аспірантів та студентів (м. Одеса, 2021), XIV міжнародна науково-практична конференція «Інформаційні технології і автоматизація – 2021» (м. Одеса, 2021), Міжнародна науково-практична інтернет конференція «Електронні інформаційні ресурси: створення, використання, доступ» (м. Суми, 2021), VI Міжнародна науково-практична конференція «Сучасна наука: проблеми та перспективи (частина II)» (м. Київ, 2022), IV

Міжнародна науково-практична конференція «Актуальні проблеми сучасної науки та освіти (частина I)» (м. Львів, 2022), XXII Всеукраїнська науково-технічна конференція молодих вчених, аспірантів та студентів «Стан, досягнення та перспективи інформаційних систем і технологій» (м. Одеса, 2022), Міжнародна науково-практична інтернет конференція «Електронні інформаційні ресурси: створення, використання, доступ» (м. Суми, 2022), XXXI міжнародна науково-практична конференція MicroCAD-2023 «Інформаційні технології: наука, техніка, технологія, освіта, здоров'я» (м. Харків, 2023), Міжнародна конференція «Інноваційні дослідження та перспективи розвитку науки і техніки у XXI столітті» (м. Рівне, 2023), VIII Міжнародна науково-практична конференція "АКТУАЛЬНІ ПИТАННЯ НАУКИ, ОСВІТИ І СУСПІЛЬСТВА В СУЧАСНИХ УМОВАХ" (м. Харків, 2022).

Публікації. За темою бакалаврської роботи надруковано 12 праць у матеріалах республіканських і міжнародних конференціях та 1 робота у фаховому виданні. Отримано авторське право на комп'ютерну програму.

Структура та обсяг БДР. Бакалаврська дипломна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел і додатків, у які винесено технічне завдання, протокол перевірки на плагіат, лістинг програмного забезпечення й ілюстративний матеріал до захисту роботи.

1 АНАЛІЗ ПІДХОДІВ ДО АНТРОПОМЕТРИЧНИХ ВИМІРЮВАНЬ З ВИКОРИСТАННЯМ ТРИВИМІРНОГО МОДЕЛЮВАННЯ

1.1 Аналіз стану технологій антропометричних вимірювань з використанням тривимірного моделювання

Сучасний етап розвитку технологій вимагає постійного вдосконалення методів аналізу та вимірювань, зокрема в області антропометрії [3-13].

Антропометрія – це наука, що вивчає фізичні розміри та форми людського тіла. Вона використовується для вимірювання, аналізу та інтерпретації різноманітних параметрів тіла, таких як довжина, ширина, об'єм та інші анатомічні показники.

До основних антропометричних показників входять:

- зріст людини: визначається як вертикальна відстань від нижньої точки ступні до верхньої точки голови;
 - обхват голови: вимірюється навколо найширшої частини голови;
 - обхват талії: визначається як вертикальна відстань навколо найтоншої частини тіла;
 - обхват стегна: вимірюється навколо найширшої частини стегна.
- обхват шиї: визначається як вертикальна відстань навколо найтоншої частини шиї;
- обхват шиї: визначається як вертикальна відстань навколо найтоншої частини шиї, зазвичай в області, де проходить кінчик підборіддя;
 - обхват стегон: вимірюється навколо найширшої частини стегна;
 - обхват зап'ястя: визначається як обхват навколо зап'ястя, зазвичай на рівні суглобу зап'ястя;
 - обхват біцепсу: вимірюється навколо найширшої частини верхньої частини руки, зазвичай на рівні середини між плечовим суглобом і ліктьовим суглобом;
 - обхват грудної клітки: вимірюється на рівні максимального

дихального вдиху;

- довжина рук: визначається як вертикальна відстань від плечового суглоба до кінчика вказівного пальця, коли рука зігнута у лікті під прямим кутом.
- обхват гомілки: визначається навколо найширшої частини гомілки;
- відстань від потилиці до тазового суглобу: вимірюється як вертикальна відстань від потилиці до верхнього краю тазу;
- відстань від таза до колінного суглобу: визначається як вертикальна відстань від верхнього краю тазу до середини колінного суглобу;

Традиційні методи антропометрії, хоча і залишаються важливими, часто обмежені двовимірними підходами та недостатньою деталізацією. Вони включають в себе використання стрічок, лінійок, кліперів та інших інструментів для вимірювання різних параметрів тіла. Однак ці методи мають деякі недоліки, такі як обмежена точність, низька швидкість, можливість помилок вимірювань через людський фактор та важкість вимірювання певних параметрів. У цьому контексті, використання тривимірних сканерів, здатних реєструвати геометричні особливості тіла в тривимірному просторі, стає дедалі більш поширеним.

Одним із ключових аспектів розвитку цих технологій є зростання швидкості та потужності обчислювальних систем. Сучасні комп'ютери та обчислювальні методи дозволяють швидко та ефективно обробляти великі обсяги даних, що є необхідним для створення та аналізу тривимірних моделей людського тіла.

Тривимірні моделі людини надають значні переваги порівняно з традиційними двовимірними методами.

Головною перевагою використання тривимірного моделювання для визначення антропометричних параметрів тіла людини є те, що тривимірні моделі людини набагато реалістичніші та деталізовані, що дозволяє точніше відображати анатомічну структуру тіла. Це дозволяє отримувати більш точні вимірювання та аналізувати різноманітні антропометричні параметри з вищою ступенем деталізації.

Також, тривимірні моделі можуть включати різні типи даних, такі як

геометричні виміри, текстури, колірні дані тощо. Це дозволяє отримувати більш комплексну інформацію про об'єкт та забезпечує можливість використання різноманітних методів аналізу.

Ще одною перевагою використання 3D моделювання в області антропометрії є точність отриманих даних. В медичній галузі точні дані про антропометричні параметри відіграють важливу роль у діагностиці та лікуванні хвороб, пов'язаних із зайвою масою тіла та іншими аспектами здоров'я. Зокрема, точне визначення об'єму тіла, площі тіла та розподілу маси може бути критичним для ефективного лікування пацієнтів.

Отже, аналіз стану технологій антропометричних вимірювань з використанням тривимірного моделювання підтверджує важливість переходу до більш точних та надійних методів вимірювань у медичній галузі та суміжних сферах. Розробка програмного забезпечення для впровадження антропометричних вимірювань з використанням тривимірного моделювання стає необхідністю для покращення точності та надійності отриманих даних.

1.2 Аналіз методів розв'язання задачі

Розробка програмного забезпечення для антропометричних вимірювань з використанням тривимірного моделювання вимагає комплексного підходу до вирішення завдань, пов'язаних з визначенням антропометричних параметрів тіла людини та їх тривимірною візуалізацією. Основною задачею, яку повинен реалізувати програмний застосунок, є можливість автоматично визначати позиції точок антропометрії та виконувати вимірювання антропометричних параметрів. Для досягнення цієї мети існують різні методи, кожен з яких має свої переваги та недоліки.

Один з основних аспектів розробки програмних засобів для антропометричних вимірювань – це визначення 16 основних суглобів на тривимірній моделі людини, що дозволить в подальшому визначити основні точки антропометрії. Для цього можна використовувати алгоритми

комп'ютерного зору, які базуються на використанні камер та датчиків руху для визначення положення суглобів на зображеннях людини. Однак такий підхід має певні недоліки. Він вимагає використання більшої кількості апаратних ресурсів та має низький рівень точності результатів. Тому такий підхід не варто застосовувати.

Також можливим методом визначення позицій точок антропометрії є використання алгоритмів машинного навчання для автоматичного визначення антропометричних точок на тривимірних моделях тіла. Цей підхід дозволяє програмному забезпеченню самостійно навчатися розпізнавати точки на основі навчальних даних, що можуть бути представлені зображеннями або тривимірними моделями. Однак цей метод може вимагати значних обчислювальних ресурсів та великої кількості навчальних даних для досягнення високої точності. Тому такий підхід не варто застосовувати.

Інший підхід полягає в використанні алгоритму SMPLX (Simple Human Pose Lower-body eXtended), який є розширеною версією відомого алгоритму SMPL. Алгоритм SMPLX базується на глибоких нейронних мережах та використовує велику кількість даних для навчання моделі. Він працює з тривимірними моделями тіла, які містять інформацію про геометрію тіла, рух та зв'язки між різними частинами тіла. Застосування алгоритму SMPLX дозволяє отримати докладні та точні відомості про позу людини, включаючи положення ніг, стегон, суглобів та інших ключових точок. Цей алгоритм може бути використаний для визначення основних характеристик людини в реальному часі, що робить його ефективним для розробки програмних засобів антропометричних вимірювань з використанням тривимірного моделювання. Однак цей алгоритм не визначає основні антропометричні точки, які необхідні для визначення антропометричних параметрів. Тому використання цього методу буде ефективним лише при умові, його доопрацювання, а саме визначення основних антропометричних точок.

Також існує метод реєстрації точок, що використовує стандарти антропометрії. Цей метод полягає в тому, що спочатку проводиться розмітка

ключових антропометричних точок на поверхні тіла. Для цього використовуються стандартні анатомічні орієнтири, які визначені в міжнародних стандартах антропометрії, таких як ISO 7250 та ASTM F1930. Після розмітки точок антропометрії вони реєструються на тривимірній моделі відносно інших точок. Однак, даний метод вимагає чіткої пози людини при скануванні та не спроможний працювати із дефектними тривимірними моделями. Тому використання цього методу є малоефективним, так як точність визначення антропометричних параметрів напряму залежить від пози тривимірної моделі.

Найефективнішим підходом є використання методів проведення площин та корелюючих ліній для визначення антропометричних параметрів на тривимірній моделі людини. Цей метод дозволяє точно визначити положення ключових точок антропометрії, шляхом аналізу відомих пропорцій та відстаней між суглобами та точками тіла. Додатково, самостійне визначення антропометричних точок за стандартами антропометрії є важливим етапом у розробці програмних засобів для антропометричних вимірювань. Цей процес передбачає детальний аналіз наукових даних та виконання вимірювань з використанням стандартизованих методик. Такий підхід гарантує високу точність та стандартизацію результатів антропометричного аналізу. Однак, він має недолік – необхідність визначення пози та позицій основних суглобів.

Для визначення відстані між визначеними точками антропометрії на тривимірній моделі тіла людини можна скористатися методом векторів. Цей підхід дозволяє розрахувати відстань між двома точками, використовуючи їх відомі координати в тривимірному просторі. Для цього необхідно спочатку визначити вектори, які представляють відстань між визначеними точками. Це можна зробити шляхом віднімання координат однієї точки від координат іншої. Отримані вектори потім можна використовувати для розрахунку їхньої довжини за допомогою формули для обчислення модуля вектора у тривимірному просторі. Розробка власних формул для цього дозволяє зменшити похибку вимірювань і досягнути максимальної точності у результаті. Під час розробки формул слід враховувати особливості тривимірного простору та геометричну

складову самого тіла. Тіло людини може мати складну форму, яка відрізняється від ідеалізованих геометричних фігур, що часто використовуються в математичних моделях. Тому важливим аспектом є вимірювання відстаней вздовж тривимірної моделі, що дозволить врахувати геометричні особливості тіла людини і забезпечити більш точні результати антропометричних вимірювань.

Розглянувши переваги та недоліки кожного методу, було вирішено використовувати комбінування методів проведення площин та корелюючих ліній зі стандартами антропометрії та застосування алгоритму SMPLX, що забезпечить найточніші та найефективніші результати у визначенні антропометричних параметрів на тривимірній візуалізації. Також було вирішено розробити та запрограмувати власні формули для обрахування антропометричної відстані на тривимірній моделі. Більше того, такий метод розробки дозволить створити унікальну платформу з унікальним функціоналом.

1.3 Аналіз аналогів

Використання технології тривимірного моделювання стає все більш популярним в останні роки, і ця технологія інтегрується в різні додатки і системи, включаючи системи вимірювання антропометричних даних. Розробка програмних засобів для антропометричних вимірювань з використанням тривимірного моделювання забезпечить ефективність та точність аналізу антропометричних даних та допоможе швидше ідентифікувати захворювання та призначати індивідуальне лікування. До найбільш відомих рішень відносять:

- WHO Anthro;
- Anthropometric iTool;
- ZEISS Measuring Software;
- DINED Mannequin;
- Vision Measuring Software iMeasuring

Одним із прикладів використання тривимірного моделювання для визначення антропометричних параметрів є програмний застосунок WHO Anthro (рис. 1.1) – програмний засіб для аналізу антропометричних даних дітей та підлітків, розроблений Всесвітньою організацією охорони здоров'я [14]. Він дозволяє проводити аналіз антропометричних даних за допомогою стандартів, розроблених WHO. Його особливість полягає у тому, що він може побудувати тривимірні моделі на основі вікових та гендерних норм, що дозволяє отримати детальнішу та більш точну інформацію про антропометричні характеристики об'єктів дослідження.

The screenshot displays the 'Individual assessment' window of the WHO Anthro software. The interface is divided into several sections:

- Children:** A list on the left shows selected children: Kwanza, Kofi (1), Lopez, Flora (2), Smith, Jane (3), Lulai (8), and DINA (9). The main area shows details for 'Child: DINA'.
 - First name: DINA
 - Last name: (empty)
 - Sex: Female (selected)
 - Child ID: 9
 - Date of birth: 1/ 1/2011
 - Age: 0mo
 - Mother: NA
 - Father: NA
 - Address: NA
- Visits:** A table showing 3 visits. The selected visit is 1/4/2011.

Date	Observer ID	Weight (kg)	Oedema	Recumbent	Ln/ht (cm)	HC (cm)	MUAC (cm)	TSF (mm)	SSF (mm)	Motor milestones
1/1/2011	Liong-PC...	2.80	No	Yes	48.00	35.00				
1/4/2011	Liong-PC...	2.70	No	Yes	49.00	36.00				
2/1/2011	Liong-PC...	3.40	No	Yes	50.00	38.00				
- Visit: 1/4/2011:** A section showing growth indicators with percentile sliders and z-scores.

Indicator	Percentile	z-score
Weight-for-length	3.8	-1.78
Weight-for-age	11.6	-1.19
Length-for-age	36.3	-0.35
BMI-for-age	4.3	-1.71
HC-for-age	94.0	1.56
MUAC-for-age	NA	NA
TSF-for-age	NA	NA
SSF-for-age	NA	NA

Рисунок 1.1 – Приклад роботи застосунку WHO Anthro

Інший приклад – застосунок Anthropometric iTool (рис. 1.2), є потужним програмним засобом для тривимірного моделювання антропометричних даних [15]. Він надає користувачам можливість створювати деталізовані тривимірні моделі тіла з урахуванням різних параметрів, таких як вік, стать, ріст і вага. Цей програмний інструмент знаходить широке застосування у наукових дослідженнях, медичних дослідженнях та дизайні одягу. Він надає користувачам

можливість створювати деталізовані тривимірні моделі тіла з урахуванням різних параметрів, таких як вік, стать, ріст і вага. Цей інструмент часто використовується у наукових дослідженнях, медичних дослідженнях та дизайні одягу.

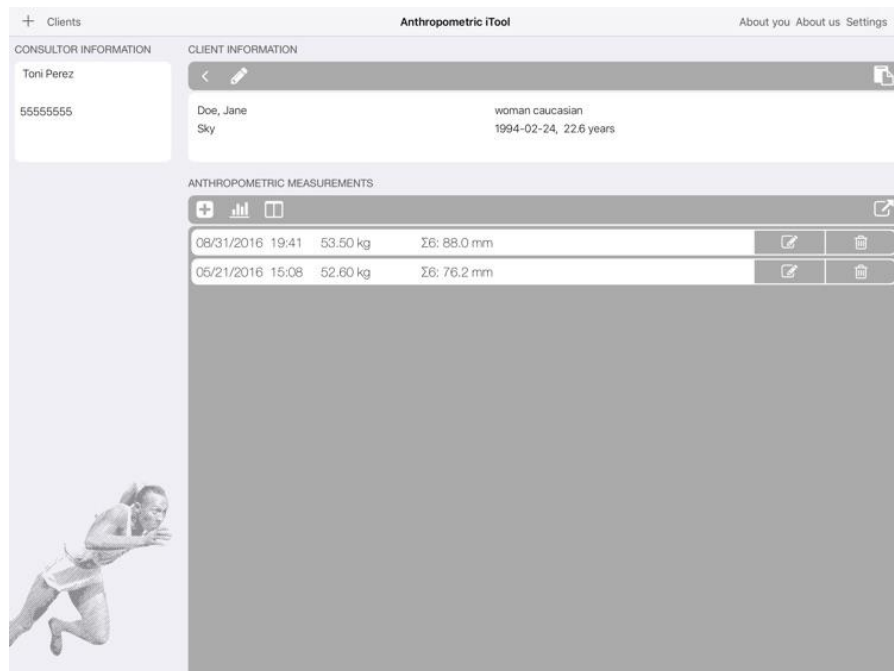


Рисунок 1.2 – Приклад роботи застосунку Anthropometric iTool

ZEISS Measuring Software (рис. 1.3) є іншим програмним засобом, спеціально розробленим для вимірювання антропометричних параметрів за допомогою оптичних систем [16].

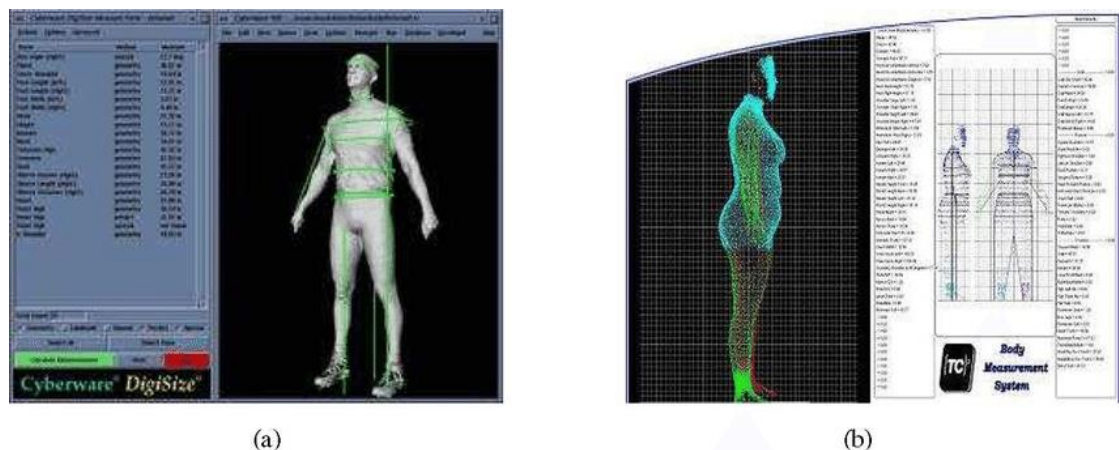


Рисунок 1.3 – Приклад роботи застосунку ZEISS Measuring Software

Цей інструмент використовується для точного визначення розмірів та пропорцій тіла, забезпечуючи високу точність і швидкість вимірювання. Варто зазначити, що дане програмне забезпечення використовує тривимірну графіку для візуалізації результатів вимірювань.

DINED Mannequin (рис. 1.4) є програмним засобом, розробленим для генерації тривимірних моделей людського тіла, спеціально призначених для використання в галузі дизайну одягу та взуття [17]. Цей інструмент забезпечує дизайнерам можливість створювати віртуальні тривимірні моделі, які точно відображають анатомічні особливості людського тіла. Використовуючи DINED Mannequin, дизайнери можуть проводити аналіз взаємодії між одягом або взуттям та фізичною формою людини, а також створювати прототипи з високим ступенем реалізму, що дозволяє ефективно вирішувати завдання дизайну та розробки продуктів.



Рисунок 1.4 – Приклад роботи застосунку DINED Mannequin

Vision Measuring Software iMeasuring (рис. 1.5) є спеціалізованим програмним засобом, розробленим для вимірювання антропометричних

параметрів з використанням систем комп'ютерного зору [18]. Цей інструмент забезпечує автоматизацію процесу вимірювання та аналізу антропометричних даних за допомогою комп'ютерного зору, що в результаті гарантує високу точність і швидкість вимірювання. Інтеграція з системою комп'ютерного зору дозволяє отримувати об'єктивні вимірювання без втручання оператора, що забезпечує значне збільшення продуктивності та ефективності процесу антропометричного аналізу.

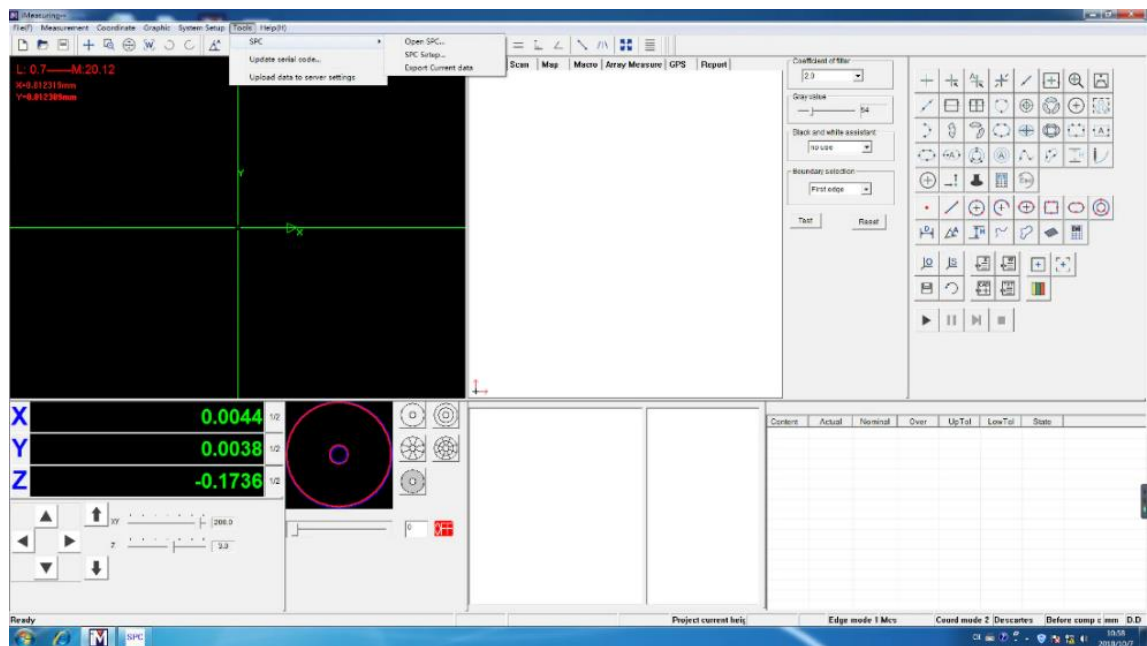


Рисунок 1.5 – Приклад роботи застосунку Vision Measuring Software iMeasuring

Розробка програмних засобів антропометричних вимірювань з використанням тривимірного моделювання вимагає досвіду роботи з тривимірним моделюванням, розробки програмного забезпечення та дизайну інтерфейсу користувача. Це передбачає створення внутрішньої системи, яка буде здатна визначати позицію 16-ти основних суглобів на тривимірній моделі людини, а також виконувати вимірювання основних антропометричних параметрів. Складова інтерфейсу користувача передбачає розробку інтерфейсів, які дозволять користувачу взаємодіяти з розробленими алгоритмами.

Результати порівняння аналогів зведено в табл. 1.1.

Таблиця 1.1 – Порівняльна характеристика аналогів

Критерії	WHO Anthro	Anthropometric iTool	ZEISS Measuring Software	DINED Mannequin	Vision Measuring Software iMeasuring	Власний модуль
Точність вимірювань	+	+	-	-	+	+
Підтримка 3Д моделювання	-	-	+	+	+	+
Автоматизація вимірювання	+	+	-	+	+	+
Підтримка стандартів антропометрії	+	+	+	+	-	+
Швидкість та зручність вимірювання	+	+	-	+	-	+
Загальна оцінка	80%	80%	40%	80%	60%	100%

Відповідно до таблиці порівняльних характеристик розробка власних програмних засобів для антропометричних вимірювань з використанням тривимірного моделювання є доцільною. Отриманий продукт зможе покрити недоліки існуючих рішень та забезпечити розширені можливості для вимірювання антропометричних параметрів з використанням тривимірного моделювання.

Отже, розробка програмних засобів антропометричних вимірювань з використанням тривимірного моделювання відкриває перед медичною галуззю широкі можливості для покращення діагностики, лікування та реабілітації. Завдяки точним та деталізованим тривимірним моделям тіла людини, медичні

фахівці можуть здійснювати більш точне та індивідуалізоване вимірювання параметрів тіла, що є критичним для правильної діагностики та лікування захворювань.

1.4 Постановка задачі дослідження

Для реалізації програмного продукту для антропометричних вимірювань з використанням тривимірного моделювання будуть використані поєднання мов програмування JavaFX та Python, а також буде використано алгоритм SMPLX для роботи із тривимірними моделями. Java – [19] це кросплатформена об'єктно-орієнтована мова програмування, яка використовується для розробки різноманітних програмних додатків, використовуючи віртуальну машину Java (JVM). Python – [20] це інтерпретована, високорівнева мова програмування з простим синтаксисом, що використовується для розробки додатків для тривимірного моделювання, штучного інтелекту та інших задач. SMPLX – [21] це алгоритм для створення та обробки й аналізу тривимірних моделей людського тіла на основі антропометричних даних. Поєднання мов програмування JavaFX та Python, а також імплементація алгоритму SMPLX дозволить створити програмні засоби, що дозволять виконувати вимірювання антропометричних параметрів з використанням тривимірного моделювання.

Розробка програмних засобів для вимірювання антропометричних параметрів з використанням тривимірного моделювання передбачає кілька етапів. По-перше, потрібно створити модуль, який буде відповідати за зручну та інтуїтивно зрозумілу взаємодію із системою. Цей модуль буде відповідати за взаємодію користувача із програмним продуктом та навігацією із функціоналом застосунку. Далі буде розроблений модуль для імпорту, аналізу та обробки отриманих даних, шляхом застосування алгоритму SMPLX для виявлення 16-ти основних суглобів та визначення позицій точок антропометрії на 3Д моделі.

Після виявлення та визначення позицій суглобів та точок антропометрії буде розроблений модуль для автоматичного вимірювання антропометричних

параметрів на тривимірній моделі. Цей модуль використовуватиме особливості тривимірної моделі, процеси тріангуляції, проведення кореляційних ліній та площин між суглобами, для визначення основних антропометричних параметрів. Також буде розроблено та запрограмовано формули для обрахунку відстані, використовуючи формули векторів та модифікуючи їх відповідно до задач дослідження, а саме використовуючи позиції точок антропометрії. Наступним буде модуль для візуалізації обчислених антропометричних параметрів на тривимірній моделі. Цей модуль дозволить користувачам взаємодіяти з 3D-моделями, переглядати позиції точок антропометрії та суглобів.

Розробка цих програмних модулів потребує глибокого розуміння тривимірного моделювання, графічного програмування та програмування на Python та JavaFX. Для реалізації цих модулів будуть використані різноманітні інструменти та бібліотеки програмування, зокрема IntelliJ IDEA, Ploty та PyMesh. Буде проведено тестування програмних модулів для забезпечення точності та швидкості антропометричного вимірювання тривимірних моделей.

Також, ретельно проаналізувавши переваги та недоліки існуючих програмних засобів для антропометричних вимірювань з використанням тривимірного моделювання, було визначено наступні завдання, які необхідно виконати для успішної розробки власного програмного забезпечення:

- розробити алгоритм побудови тривимірних моделей тіла людини на основі антропометричних даних та їх тривимірної візуалізації;
- розробити алгоритми вимірювання, які забезпечать високу точність та надійність отриманих даних на основі тривимірних моделей;
- розробити алгоритм та програмний модуль для визначення позицій суглобів, який буде спроможний виявляти, відстежувати та відображати позиції 16 основних суглобів, необхідних для вимірювань;
- розробити модуль графічного інтерфейсу користувача, який буде зручним у використанні для медичного персоналу та дослідників, забезпечуючи зручність і швидкість взаємодії з програмними засобами;

- реалізувати програмний продукт з використанням засобів тривимірного моделювання;
- автоматизація процесу вимірювання антропометричних параметрів, забезпечуючи високу точність та надійність отриманих результатів;
- провести тестування програмного забезпечення згідно поставлених задач.

Отже, розробка програмних модулів для реалізації програмних засобів антропометричних вимірювань з використанням тривимірного моделювання на JavaFX та Python, а також імплементація алгоритму SMPLX є комплексним завданням, яке включає кілька етапів. Модулі повинні бути ретельно спроектовані, реалізовані та протестовані, щоб забезпечити користувачам зручну та ефективну роботу з програмним забезпеченням для вимірювання антропометричних параметрів.

1.5 Висновки

У першому розділі було розглянуто стан програмних засобів для антропометричних вимірювань з використанням тривимірного моделювання. Були розглянуті програмні засоби такі як WHO Anthro, Anthropometric iTool, ZEISS Measuring Software, DINED Mannequin та Vision Measuring Software iMeasuring. Кожен з цих засобів має свої переваги і недоліки, що були проаналізовані.

Проаналізувавши переваги та недоліки існуючих програмних засобів, було сформульовано завдання для подальшої розробки власного програмного забезпечення для антропометричних вимірювань. Ці завдання включають розробку алгоритмів побудови тривимірних моделей тіла людини, визначення позицій суглобів та обчислення антропометричних параметрів, розробку графічного інтерфейсу. Таким чином було доведено доцільність розробки власного програмного рішення. На основі отриманої інформації було сформовано перелік задач, які необхідно виконати для розробки власних програмних засобів.

2 РОЗРОБКА МЕТОДІВ АНТРОПОМЕТРИЧНИХ ВИМІРЮВАНЬ З ВИКОРИСТАННЯМ ТРИВИМІРНОГО МОДЕЛЮВАННЯ

2.1 Розробка методів визначення площі та об'єму фігур з використанням тривимірного моделювання

Для визначення масо-вагових характеристик людини по її антропометричних параметрах використовують 3D-сканери (боді-сканери) для оцифровки тіла. Це дає можливість створювати в 3D просторі модель тіла людини на повний зріст, а також окремих частин тіла: рук, ніг, голови, тощо. Для цього вони роблять множину кадрів і потім збирають їх разом в одне зображення, на основі якого створюється 3D-модель (рис. 2.1).

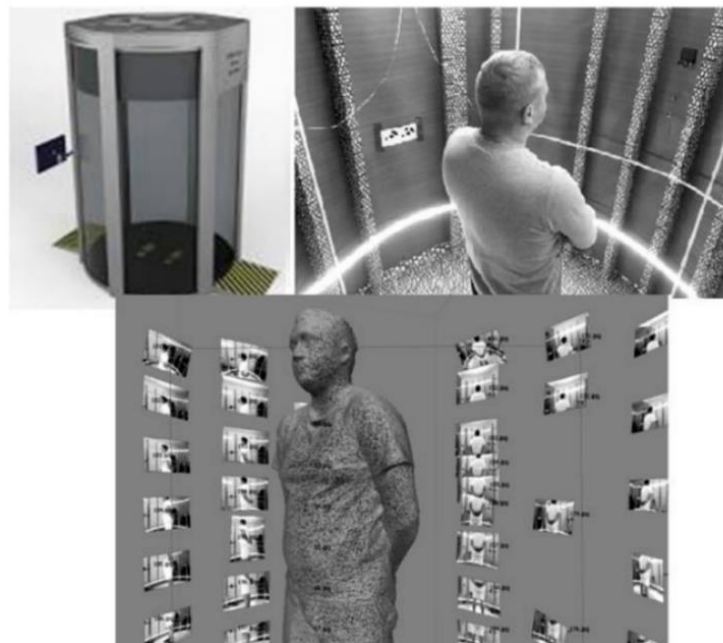


Рисунок 2.1 – Панорамний тривимірний сканер

Для визначення площі тіла людини пропонується розробити тривимірну модель людини за допомогою тривимірних сканерів. Далі виконується мозаїка, яка полягає в заміні поверхні багатокутниками, які примикають один до одного без пропусків і не перекриваються (рис. 2.2). Цей підхід дозволяє апроксимувати

поверхню тіла з високою точністю, розділяючи її на дрібні частини. Кожен багатокутник описується своїми вершинами та гранями, що дозволяє математично обчислити площу кожного з них.

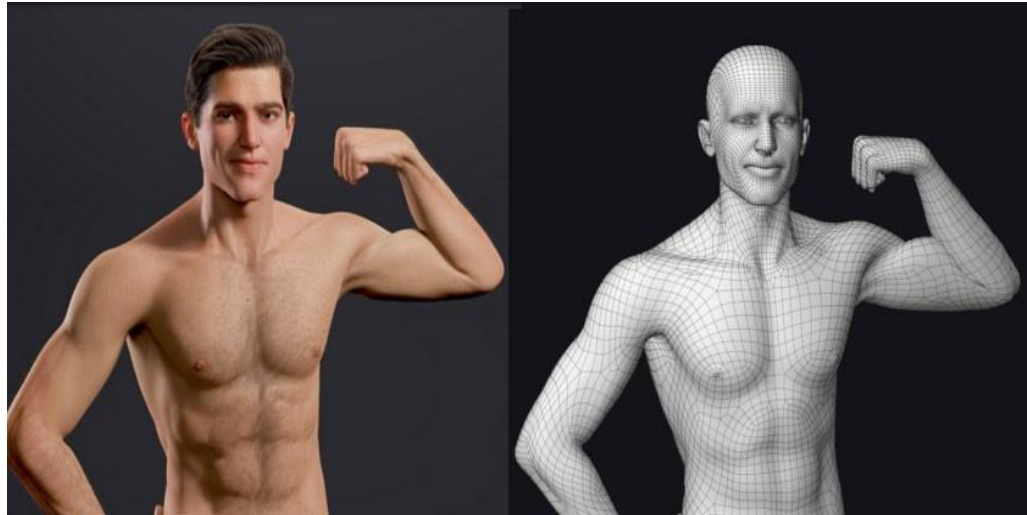


Рисунок 2.2 – Приклад полігональної моделі

У графічних пакетах, при виконанні тріангуляції, модель перетворюється у полігональну форму, де кожна поверхня фігури представлена трикутниками як клітинками. Цей процес тріангуляції є важливим етапом, оскільки перетворює абстрактну тривимірну форму у набір простих геометричних об'єктів. Така полігональна мережа дозволяє подальше обчислення різних параметрів фігури, включаючи площу та об'єм.

Точність визначення площі в значній мірі залежить від густоти полігональної мережі, оскільки більша кількість трикутників дозволяє краще апроксимувати форму фігури. Чим більша кількість трикутників у полігональній моделі, тим більш точно можна визначити площу поверхні. Багатокутну мережу, що складається з полігонів, можна розгорнути на площину, використовуючи різноманітні методи проекції (рис. 2.3). Цей процес дозволяє перетворити тривимірну форму на площину, забезпечуючи можливість подальшого аналізу та обробки даних.

Застосування алгоритмів обробки зображень, таких як сегментація, дозволяє виділити структури тіла на тривимірній моделі. Після сегментації

можна створити точкову хмару, яка представляє собою набір тривимірних точок, що описують структури тіла. Ця точкова хмара може бути використана для подальшого моделювання або аналізу фізичних властивостей фігури.

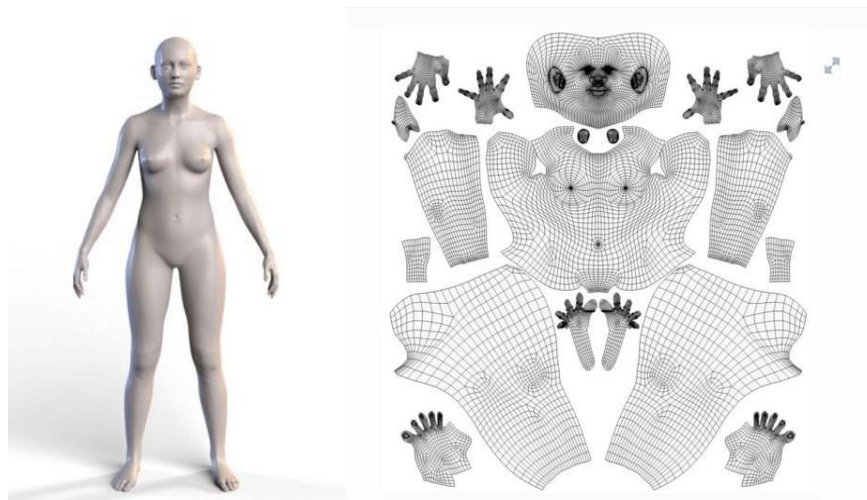


Рисунок 2.3 – Розгорнута полігональна модель на площині

Щоб визначити площу поверхні, потрібно розбити поверхню на складові трикутники та обчислити їхні площі, а потім знайти їхню суму. Цей підхід дозволяє врахувати особливості будови людського тіла, оскільки дозволяє отримати деталізовану інформацію про кожную частину поверхні та уникнути необхідності в диференціюванні між чоловічою та жіночою фігурою. Такий підхід до визначення площі поверхні забезпечує більш точний результат, оскільки враховує різноманітність форм та розмірів людського тіла.

Для точного визначення об'єму тіла людини ефективно використовувати тривимірне моделювання. За допомогою тривимірних моделей можна точно розрахувати об'єм тіла, враховуючи його форму та структуру. Цей підхід надає можливість отримати деталізовану інформацію про об'єм тіла та його складові частини, що допомагає в дослідженнях з медичної діагностики та антропометрії.

Зрізи тривимірної моделі людини можуть бути виконані різними типами площин, такими як поперечні, вертикальні або діагональні, що дозволяє отримати різні перспективи аналізу та оцінки фігури. Кожен тип площин дозволяє вивчити певні аспекти структури тіла (рис. 2.4).

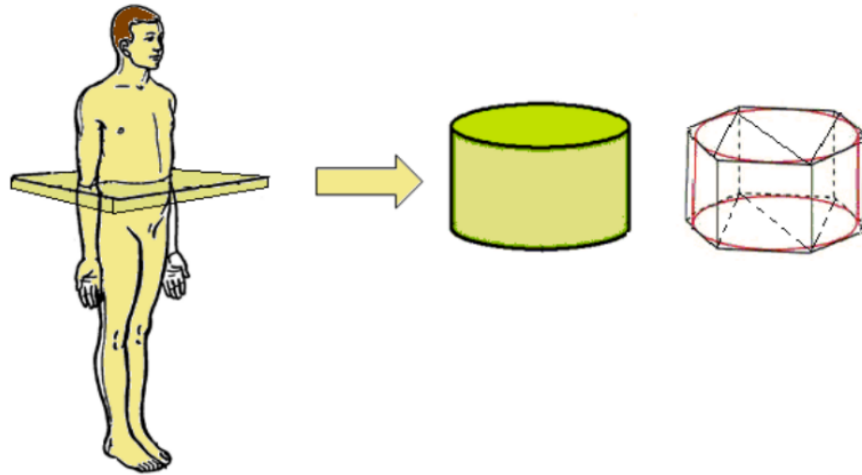


Рисунок 2.4 – Етапи визначення об'єму сегмента

Після розбиття тіла на сегменти, отриманий верхній зріз піддається триангуляції для апроксимації його форми та створення внутрішньої структури. Це важливий етап, оскільки дозволяє представити тіло у вигляді трикутних призм, що полегшує подальші обчислення. Для триангуляції зрізу використовують вертикальні площини, які перпендикулярні до основи відрізка. Це дозволяє розбити поверхню на трикутники, що складають сегмент тіла. Далі для визначення криволінійної площі та об'єму тіла людини використовують метод об'ємних пікселів.

Виділені структури перетворюються в тривимірну точкову хмару, де кожна точка відповідає вокселю з певними координатами x, y, z . До кожної точки додається інформація про інтенсивність та геометричні властивості, такі як координати.

Нехай $I(x, y, z)$ – інтенсивність точки в тривимірному просторі. Визначимо функцію $f(x, y, z)$ так, що $f(x, y, z) = I(x, y, z)$. Це означає, що значення функції відповідає інтенсивності точки.

Для отримання функції $f(x)$, будемо використовувати значення $f(x, y, z)$ на поверхні тіла. Тоді, функцію $f(x)$ можна визначити як:

$$f(x) = \max_{y,z} I(x, y, z) \quad (2.1)$$

Для визначення криволінійної площі одного трикутника розглянемо функцію $f(x)$, яка представляє координати точок поверхні тіла у тривимірному просторі. Для виведення формули для криволінійної площі S , розглянемо елементарний відрізок поверхні між двома сусідніми точками $P(x, f(x))$ і $Q(x + dx, f(x + dx))$, використовуючи теорему Піфагора, довжина цього відрізка буде:

$$dS = \sqrt{dx^2 + [f'(x)dx]^2} \quad (2.2)$$

Виведемо вираз для похідної $f'(x)$. Якщо $f(x) = \max_{y,z} I(x, y, z)$, то $f'(x)$ можна апроксимувати як $\frac{dI}{dx}$ в точках, де $I(x, y, z)$ максимальне. Визначимо криволінійну площу одного трикутника за формулою:

$$Area(T_i) = \int_b^a \sqrt{1 + \left(\frac{df}{dx}\right)^2} dx \quad (2.3)$$

Знайшовши криволінійну площу одного трикутника на сегменті тривимірної моделі, виведемо формулу загальної площі тіла людини:

$$S = \sum_{i=1}^n Area(T_i) \quad (2.4)$$

Для обрахунку об'єму тіла використаємо аналогічний підхід, $A(x)$ - площа поперечного перерізу тіла у площині, паралельній осі x . Розглянемо відрізок цього перерізу між x і $x+dx$. Площа такого сегменту обраховується за формулою Паскаля:

$$dA = \pi[R(x)]^2 \quad (2.5)$$

, де $R(x)$ позначає криволінійну площу усіх сегментів що входять до перерізу. Тоді об'єм тіла людини буде обраховуватись як:

$$V = \pi \int_c^d [R(x)]^2 dx \quad (2.6)$$

Отже, було виконано розробку методів визначення площі та об'єму тривимірних моделей тіла людини з використанням тривимірного моделювання. Було отримано формули та методики, що дозволяють обрахувати площу поверхні тіла людини та об'єм тіла людини з максимальною точністю, використовуючи засоби тривимірного моделювання. Розроблені методи дозволяють врахувати різноманітність форм та розмірів людського тіла незалежно від віку та статі, що сприяє отриманню точних результатів вимірювань.

2.2 Розробка методу визначення точок антропометрії на основі даних тривимірної моделі

Для визначення точок антропометрії на основі тривимірної моделі людини необхідно розробити метод, який апроксимує реальні антропометричні точки на поверхні тривимірної моделі. Для цього, необхідно визначити позиції точок антропометрії відповідно до прийнятих міжнародних стандартів, таких як ISO 7250 та ASTM F1930. Дані стандарти дозволять ідентифікувати положення точок антропометрії на тривимірній моделі незалежно від віку, статі та пози моделі людини.

Програмне забезпечення буде спроможне зчитувати дані з тривимірних моделей форматів .obj та .pkl. Дані формати тривимірних моделей дозволяють виконувати зчитування координат усіх точок, що формують тривимірну модель. Наступним кроком програмне забезпечення буде формувати хмару точок відтворення поверхні тіла людини. Після завершення цього процесу буде виконуватися тріангуляція тривимірної моделі – це метод побудови 3Д моделі на основі базових точок, де базові точки розміщуються таким чином, що вони утворюють взаємопов'язану систему трикутників.

При виконанні тріангуляції для планування поверхні є дуже важливим є принцип розташування трикутників. Найчастіше використовується принцип екстремумів, коли вершини трикутників визначаються найвищою і найнижчою точками в найближчій області, попередньо визначаючи максимальну відстань між точками.

Антропометричні виміри знімаються за допомогою розрізів по периметру тривимірної моделі тіла людини, вимірів проміжних антропометричних точок та кореляційних ліній. Після обробки даних отримують тривимірні дані з трьома рівнями роздільної здатності - найнижча роздільна здатність (~26 000 точок), найвища роздільна здатність (~70 000 точок) та 3D модель тіла людини. У кожному файлі отримуються два типи параметрів: - значення x , y і z 3D-координат (позначені літерою v), - послідовність з'єднання трикутників (позначена літерою f). Координати точок поверхні тіла, отримані при скануванні, використовуються для розробки 3D моделі. Файл з координатними параметрами зберігається і створюються підфайли для визначення координат точок і положення трикутників.

Після цього система циклічно зчитує положення точок, послідовність і відстані трикутників і розміщує точки. З розміщених точок збирається поверхня моделі і створюється індивідуальна тривимірна модель тіла людини.

У програмне забезпечення було додано інформацію про антропометричні точки та міжнародні стандарти проведення антропометричних вимірювань. Дана інформація містить дані про кожну точку антропометрії, з якими суглобами вона пов'язана та як знайти її. Усі точки знаходяться на певній відстані від суглобів, так як суглоби є основними анатомічними орієнтирами на тривимірній моделі. Після завантаження інформації про точки антропометрії, необхідною задачею стає знаходження цих точок шляхом обрахування відстані.

Для знаходження позиції суглобів на тривимірній моделі було вирішено використовувати алгоритми комп'ютерного зору та імплементувати алгоритм SMPLX, що дозволить знаходити позиції та точні тривимірні координати суглобів.

Для знаходження точок антропометрії у тривимірному просторі пропонується виконувати зчитування трикутників послідовно. Кожна вершина трикутника має відомі координати x , y та z . Для кожного трикутника обчислюються необхідні дані (довжина ребер, кути). Наступним кроком, координати перетворюються у хмару точок. Знаючи координати вершин можна обрахувати довжини сторін трикутника, висоту, косінуси кутів.

Для апроксимації точок антропометрії на поверхні тривимірної моделі використовується метод найближчих точок. Нехай P_i – антропометрична точка, яку необхідно визначити, Q_j – точка на поверхні тривимірної моделі. Тоді відстань між P_i та Q_j обчислюється за допомогою евклідової відстані:

$$d(P_i, Q_j) = \sqrt{(x_{P_i} - x_{Q_i})^2 + (y_{P_i} - y_{Q_i})^2 + (z_{P_i} - z_{Q_i})^2} \quad (2.7)$$

Наступним кроком, виразимо координати P_i – точки антропометрії, при умові, що Q_j – це відома точка суглобу на тривимірній моделі, а $d(P_i, Q_j)$ – відома відстань, отримана з міжнародних стандартів антропометрії. Виведемо формулу, що буде знаходити координати точки P_i . Для цього використаємо формули для визначення трьох координат $(x_{P_i}, y_{P_i}, z_{P_i})$.

$$x_{P_i} = x_{Q_i} + \sqrt{d(P_i, Q_j)^2 - (y_{Q_i})^2 - (z_{Q_i})^2} \quad (2.8)$$

$$y_{P_i} = x_{Q_i} + \sqrt{d(P_i, Q_j)^2 - (x_{Q_i})^2 - (z_{Q_i})^2} \quad (2.9)$$

$$z_{P_i} = x_{Q_i} + \sqrt{d(P_i, Q_j)^2 - (x_{Q_i})^2 - (y_{Q_i})^2} \quad (2.10)$$

Отже, було виконано розробку методів визначення точок антропометрії на тривимірних моделях тіла людини з використанням тривимірного моделювання.

Було отримано формули та методики, що дозволяють визначити позиції та координати точок антропометрії, використовуючи засоби тривимірного моделювання. Розроблені методи базуються на інформації міжнародних стандартів та дозволяють визначити основні точки, що необхідні для антропометричних вимірювань.

2.3 Розробка методу вимірювання антропометричних параметрів на тривимірній моделі

Визначивши основні формули для обрахунку площі та об'єму тіла, а також метод визначення точок антропометрії на тривимірній моделі, необхідно розробити метод обрахунку відстані між двома точками на тривимірній моделі. Для цього необхідно використати узагальнену систематизацію параметрів тривимірних моделей.

Зазвичай 3D-поза кодується як набір спільних координат (наприклад, файли *trc*) або орієнтацій (наприклад, файли *bvh*, *asf/amc* файли). Популярні 3D моделі людського тіла *MakeHuman*, *SCAPE*, *BlendSCAPE* та *SMPL* мають схожу параметризацію моделі $\{T, S, O\}$, де T початкова модель у "канонічній формі" та "канонічній позі", S визначає деформацію форми та O визначає позу. Параметризація тривимірної моделі базується на скелетному каркасі з J суглобів.

Тому для визначення відстані між двома точками необхідно визначити J основних суглобів та обчислити параметризацію утвореної тривимірної моделі. Обчислена параметризація дозволить визначити точність вимірювання та мінімальну кількість суглобів, які необхідні для вимірювання.

Для аналізу основних компонентів тривимірної моделі використаємо метод PCA, де головні компоненти представляють найважливіші осі варіації в тривимірній моделі. У просторі PCA будь-яка форма може бути реконструйована шляхом лінійного додавання головних напрямків до середньої форми (нульової форми):

$$T + B(\beta) = T + \sum_{n=1}^{|\beta|} \beta_n S_n \quad (2.11)$$

Для вимірювання антропометричних відстаней на тривимірній моделі будемо використовувати 16-суглобну скелетну модель (рис. 2.5), так як відстань у тривимірній моделі пов'язана лише з конфігурацією основних суглобів тіла.

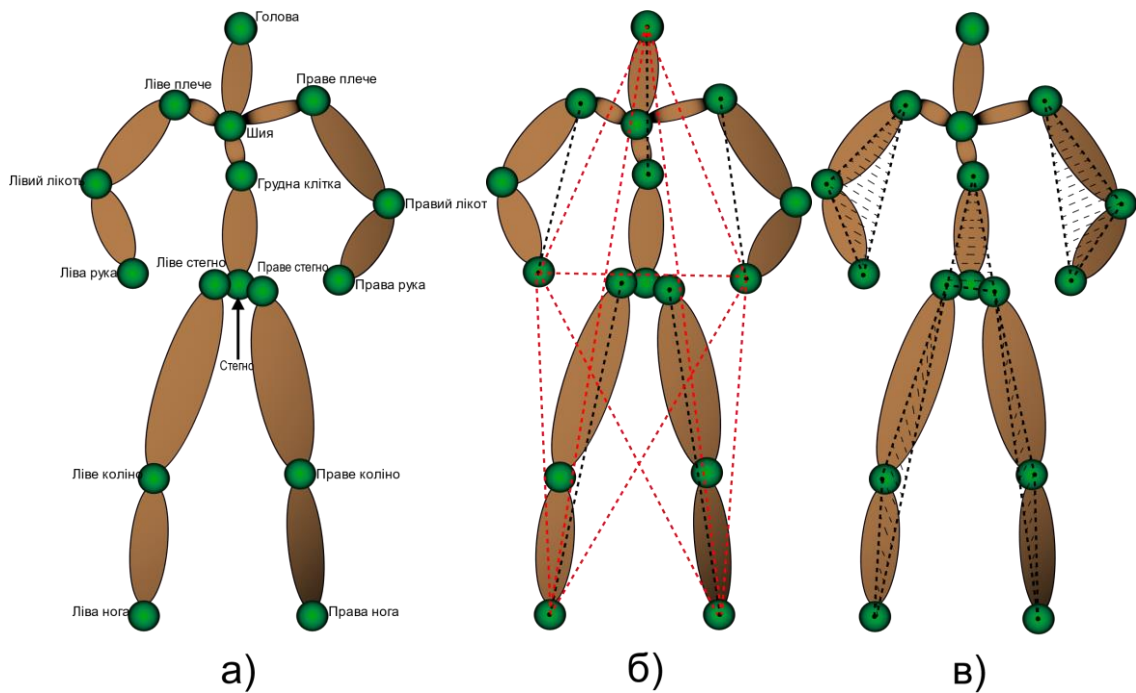


Рисунок 2.5 – Модель скелета

а) суглоби та кінцівки, б) корелюючі лінії в) площини

Варто зазначити, що позиції 16 основних суглобів визначаються автоматично при створенні тривимірної моделі. Для обрахунку відстані між суглобами використаємо формулу Евклідової відстані:

$$f_{JJ_d}(J_1, J_2) = \|\overrightarrow{J_1 J_2}\| \quad (2.12)$$

$$f_{JJ_o}(J_1, J_2) = \text{unit}(\overrightarrow{J_1 J_2}) \quad (2.13)$$

, де J_1, J_2 – відповідні суглоби, $\text{unit}()$ - масштабує вектор до одиничної довжини.

Для обрахунків відстані між лінією суглобів $f_{JJ_d}(J_1, J_2)$ та окремим суглобом J використовується розширена формула Хелена:

$$f_{JL_d}(J, L_{J_1 \rightarrow J_2}) = 2 * \frac{S_{\Delta JJ_1 J_2}}{f_{JJ_d}(J_1, J_2)}, \quad (2.14)$$

де $S_{\Delta JJ_1 J_2}$ – площа трикутника утвореного позиціями суглобів.

Для обрахунку відстані між суглобом J та площиною $P_{J_1 \rightarrow J_2 \rightarrow J_3}$, використаємо формулу перехресного добутку двох тривимірних векторів:

$$f_{JP_d}(J, P_{J_1 \rightarrow J_2 \rightarrow J_3}) = f_{JJ_o}(J_1, J_2) * \text{unit}(f_{JJ_o}(J_1, J_2) * f_{JJ_o}(J_1, J_3)) \quad (2.15)$$

На рисунку 2.6 показано візуалізація наведених формул.

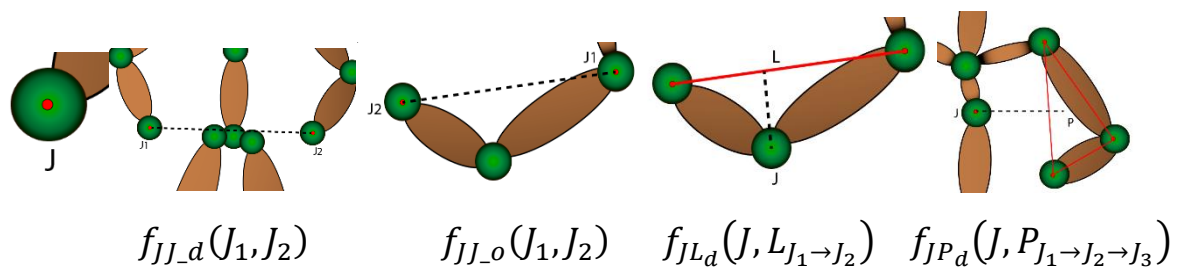


Рисунок 2.6 – Візуалізація вимірювань

Перетворимо формули антропометричних метрик для тривимірних моделей, для цього визначимо набір даних, що буде представляти собою матрицю точок $\{x_1, x_2 \dots x_{n-1}, x_n\}$ отриману з тривимірної моделі. Щоб запобігти надмірному відхиленню, слід подбати про те, щоб метрика відстані не відходила надто далеко від початкової метрики відстані. Для цього введено регуляризатор.

Нехай d_{ij} – це відстань L_2 між x_i та x_j . Відхилення від початкового співвідношення даних вимірюється за допомогою формули Махаланобіса:

$$\sum_{i,j} |d_{ij} - d_{ij}^*| = \sum_{i,j} (x_i - x_j)^T (x_i - x_j) - T * r * I \quad (2.16)$$

, де r – регуляризатор, I - стовпчикова ортогональна матриця, що складається з набору даних суглобів через які проходить шукана лінія відстані.

Тоді для знаходження відстані між двома точками на тривимірній моделі використовується наступна формула:

$$d_{ij} = \sum_{k=0}^m d_E^2(c_i^k, c_j^k) + \lambda \sum_{k=0}^m d_E^2(\dot{c}_i^k, \dot{c}_j^k) \quad (2.17)$$

де c_i^k, c_j^k – тривимірні координати 3D моделі, $\dot{c}_i^k, \dot{c}_j^k$ – тривимірні координати 3D моделі відносно позицій суглобів, d_E^2 – евклідова відстань між заданою точкою і найближчим суглобом.

2.4 Висновки

У другому розділі було проведено розробку методів визначення площі та об'єму тривимірної моделі тіла людини, методу визначення точок антропометрії на основі даних 3д моделі та методу вимірювання антропометричних параметрів на тривимірній моделі. Усі методи використовують засоби тривимірного моделювання.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АНТРОПОМЕТРИЧНИХ ВИМІРЮВАНЬ З ВИКОРИСТАННЯМ ТРИВИМІРНОГО МОДЕЛЮВАННЯ

3.1 Розробка алгоритмів і програм антропологічних вимірювань

Для реалізації програмного продукту для антропометричних вимірювань з використанням тривимірного моделювання будуть використані поєднання мов програмування JavaFX та Python, а також буде використано алгоритм SMPLX для роботи із тривимірними моделями.

При розробці програмного забезпечення для антропометричних вимірювань з використанням тривимірного моделювання важливо обрати найбільш оптимальні технології для розробки кожного модуля програмних засобів. JavaFX є потужним інструментом для розробки графічного інтерфейсу користувача. Використання JavaFX забезпечить можливість створення зручного інтерфейсу для взаємодії з користувачем та відображення тривимірних моделей людського тіла. Python має широкий набір бібліотек, які дозволяють ефективно працювати з тривимірними об'єктами, виконувати аналіз антропометричних даних та візуалізувати результати вимірювань.

SceneBuilder – це інтегрована програма, яка надає графічний інтерфейс для швидкої та простої розробки інтерфейсів користувача для додатків, що базуються на JavaFX. Він був створений компанією Oracle та є частиною JavaFX платформи. SceneBuilder дозволяє розробникам створювати інтерфейси користувача шляхом перетягування та розміщення візуальних елементів, таких як кнопки, тексти, таблиці та інші, на графічному полотні, без прямого написання коду.

SMPLX – це алгоритм для обробки тривимірних моделей людського тіла. Він був розроблений з метою визначення позицій суглобів та точок антропометрії на тривимірних моделях з високою точністю та ефективністю. SMPLX надає широкий спектр функцій, включаючи точне відтворення

анатомічних характеристик людського тіла у тривимірному просторі для різноманітних застосувань, зокрема в областях медицини та біометрії.

Plotly – це бібліотека для створення інтерактивних графіків та візуалізації даних в програмному середовищі Python. Вона надає розробникам широкий спектр інструментів для створення різноманітних видів графіків, включаючи лінійні графіки, різні види діаграм та тривимірні графіки. Можливості Plotly дозволяють створювати інтерактивні графіки з можливістю масштабування, навігації та взаємодії з даними. За допомогою вбудованих функцій та параметрів, Plotly дозволяє створювати інтерактивні 3D-графіки, що демонструють різні аспекти тривимірних даних, такі як форма, розмір та розташування об'єктів у просторі. Це робить її корисним інструментом для візуалізації результатів тривимірного моделювання та аналізу даних, що використовуються у програмах для антропометричних вимірювань.

PyMesh – це бібліотека для обробки та роботи з тривимірними моделями у середовищі мови програмування Python. Вона надає широкий спектр функцій для роботи з тривимірними об'єктами, включаючи завантаження, збереження, обробку та аналіз геометричних даних. Бібліотека дозволяє виконувати різноманітні операції з тривимірними моделями, такі як обчислення об'єму, площі поверхні, відокремлення компонентів, робота з тривимірними координатами. Вона також надає засоби для візуалізації тривимірних моделей та їхніх характеристик, що робить її ідеальним інструментом для використання в задачах з тривимірним моделюванням у наукових дослідженнях та розробці програмного забезпечення.

Поєднання цих технологій дозволяє розробити програмні засоби для антропометричних вимірювань з використанням тривимірного моделювання. Python та бібліотеки PyMesh і Plotly забезпечують ефективну кодову базу для взаємодії та маніпуляцій з тривимірною моделлю тіла людини, в той час як JavaFx та SceneBuilder надають розширені інструменти для створення зручного інтерфейсу користувача. Імплементация алгоритму SMPLX забезпечує швидке та точне визначення позицій суглобів, що необхідне для визначення точок

антропометрії.

Для кращого розуміння роботи модулів програмних засобів антропометричних вимірювань з використанням тривимірного моделювання було вирішено розробити модель роботи системи на прикладі UML діаграми діяльності (рис. 3.1). Згідно вказаної діаграми користувач для початку взаємодії із програмним застосунком повинен імпортувати тривимірну модель у форматі .pkl, у разі іншого формату тривимірної моделі програма видаватиме помилку. Наступним кроком буде перевірка моделі за алгоритмом SMPLX на наявність позицій основних суглобів. Якщо позиції певних суглобів не вдасться визначити (через фізичні особливості тривимірної моделі), програма повідомить про це у робочому середовищі і виведе перелік антропометричних параметрів, які неможливо буде отримати. У разі отримання позицій усіх основних суглобів, програмний застосунок надає інформацію про імпортовану модель. Після успішного імпортування тривимірної моделі, програмний застосунок автоматично проводить аналіз позицій суглобів, проведення кореляційних ліній та визначення основних точок антропометрії за алгоритмом SMPLX та використовуючи стандарти антропометрії. Далі, користувач може почати вимірювання антропометричних показників імпортованої тривимірної моделі, внаслідок чого програмний застосунок автоматично обрахує антропометричні відстані та сформує перелік антропометричних показників. Після завершення вимірювання, користувач може переглянути отримані антропометричні параметри у віртуальному робочому середовищі, переглянути візуалізацію антропометричних параметрів на 3д моделі, з можливістю взаємодіяти з нею (змінювати кут огляду, масштаб, відображення певних елементів легенди) та експортувати отримані результати вимірювання у форматі .xlsx.

Розробка діаграми діяльності була виконана у програмному забезпеченні Visio [22].

Для розробки програмного застосунку, для вимірювання антропометричних параметрів на тривимірній моделі необхідно розробити загальний алгоритм, згідно якого буде працювати програмний модуль (рис. 3.2).

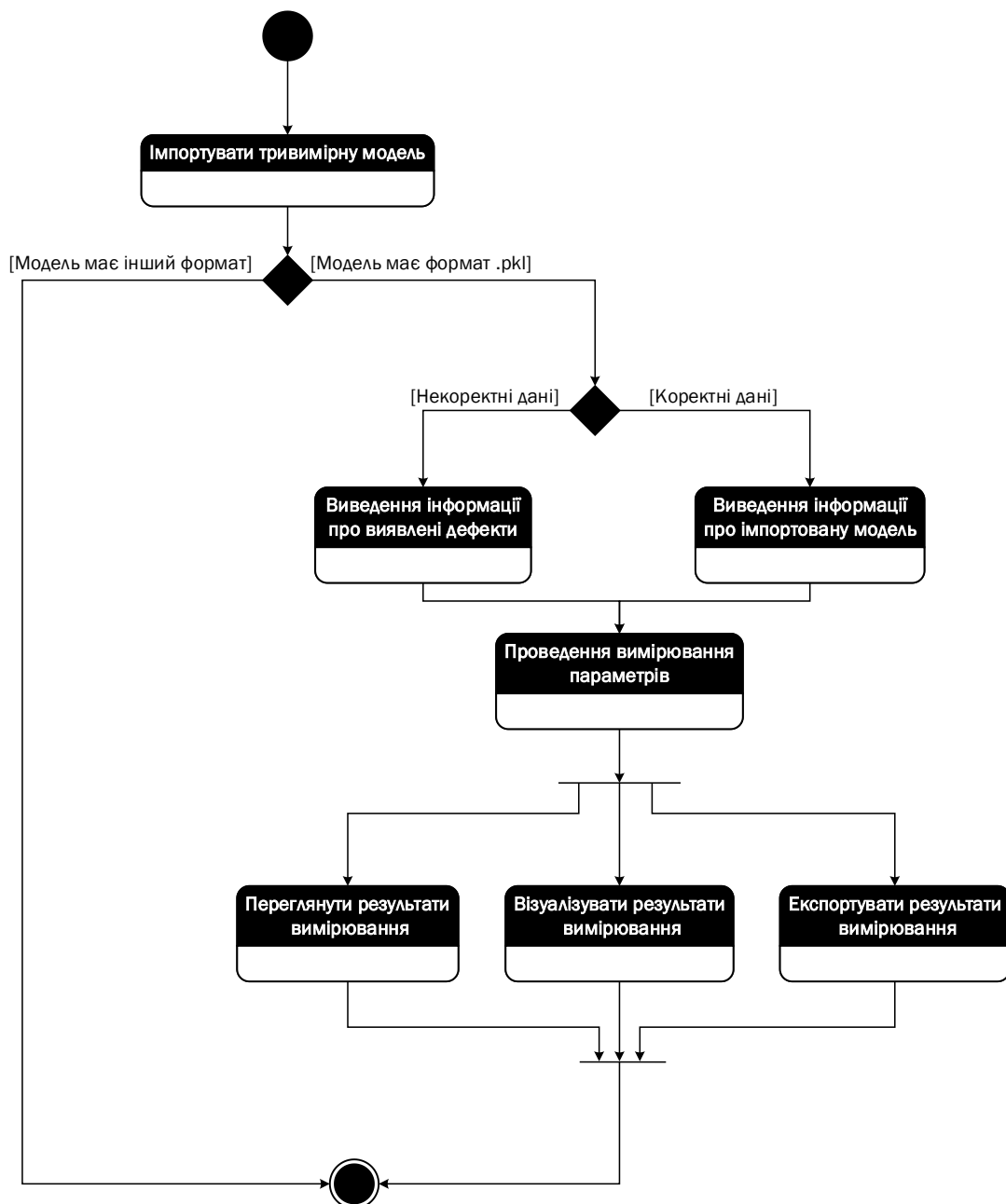


Рисунок 3.1 – Діаграма діяльності програмних засобів

Згідно даного алгоритму, першим ділом, відбувається завантаження робочого середовища програми, яке надає користувачу доступ до функціоналу розробленого програмного застосунку. Далі, користувачу необхідно імпортувати тривимірну модель тіла людини у форматі .pkl. Обравши 3D модель, програмний застосунок запускає модуль перевірки імпортованої тривимірної моделі на наявність дефектів. При наявності дефектів тривимірної моделі або невірному форматі імпортованої моделі програмний застосунок надасть користувачу

інформацію про виявлені дефекти або виведе інформацію про невірний формат 3D моделі.

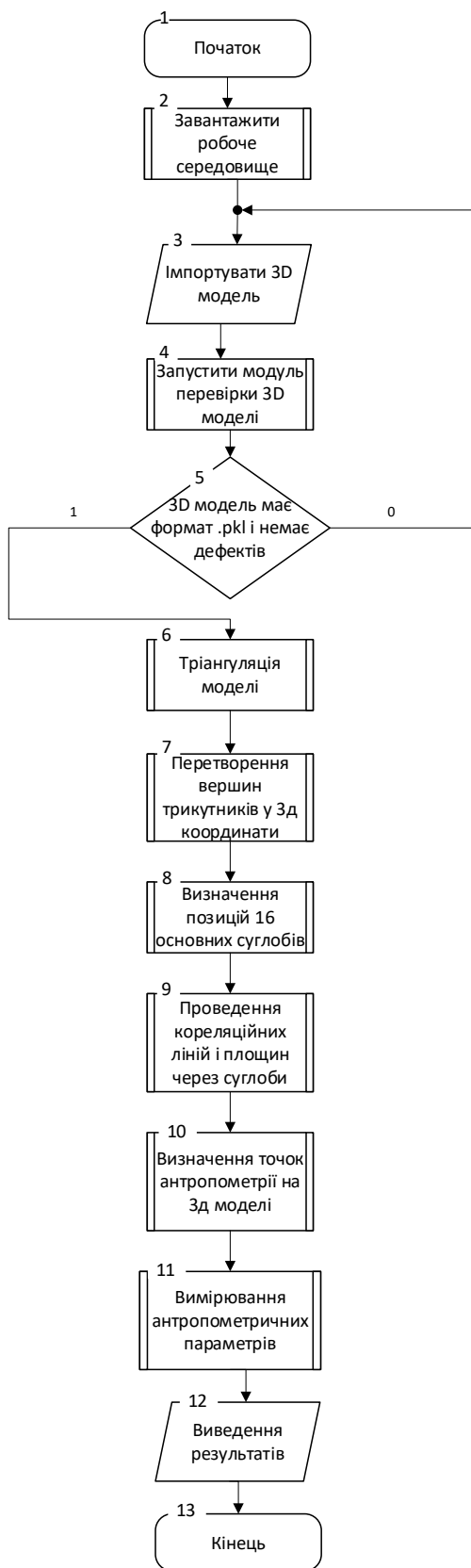


Рисунок 3.2 – Алгоритм роботи програмного застосунку

При відсутності дефектів програмний засіб виконає модуль тріангуляції, для розбиття 3Д моделі тіла людини на множину трикутників. Після чого виконується модуль перетворення вершин отриманих трикутників на множину тривимірних координат, що необхідно для обрахування відстані між точками і суглобами на 3Д моделі. Наступним кроком буде відбуватись модуль визначення 16 основних суглобів, що реалізує в собі алгоритм SMPLX. Після визначення позицій суглобів через них будуть проводитися кореляційні лінії, для визначення антропометричних параметрів, та площини, для розміщення точок антропометрії. Після визначення цих точок відбувається вимірювання антропометричних параметрів шляхом вимірювання відстані між двома точками або суглобами на тривимірній моделі, застосовуючи формули векторів.

Для кращого розуміння роботи модулів генерації тривимірної хмари точок та визначення позицій суглобів і проведення площин, було розроблено додатковий алгоритм (рис. 3.3). Згідного даного алгоритму першим етапом є завантаження 3Д моделі тіла людини і її подальша перевірка на наявність дефектів і вірність формату. Після чого виконується автоматична тріангуляція моделі. Далі виконується цикл, який проходить через усі вершини створених трикутників і перетворює їх у масив тривимірних точок, що мають координати x , y , z . Після чого виконується алгоритм SMPLX, для визначення позицій суглобів на 3Д моделі. Далі відбувається проведення кореляційних ліній та площин. Для цього виконується цикл перевірки усіх суглобів, у якому виконується визначення чи є суглоб під позицією j батьківським для суглоба під позицією $j-1$. Наприклад при розгляданні колінного суглоба, батьківським суглобом для нього буде тазовий суглоб.

Після визначення спорідненості суглобів відбувається проведення кореляційної лінії (якщо суглоби мають спорідненість), у іншому випадку проводиться площина. Далі для визначення позицій точок антропометрії та антропометричних вимірів відбувається завантаження стандартів антропометрії (розміщення основних антропометричних точок вимірювання). Після ідентифікації даних точок вони позначаються на кореляційних лініях та

площинах. Далі відбувається вимірювання антропометричних відстаней та виведення результатів вимірювань.

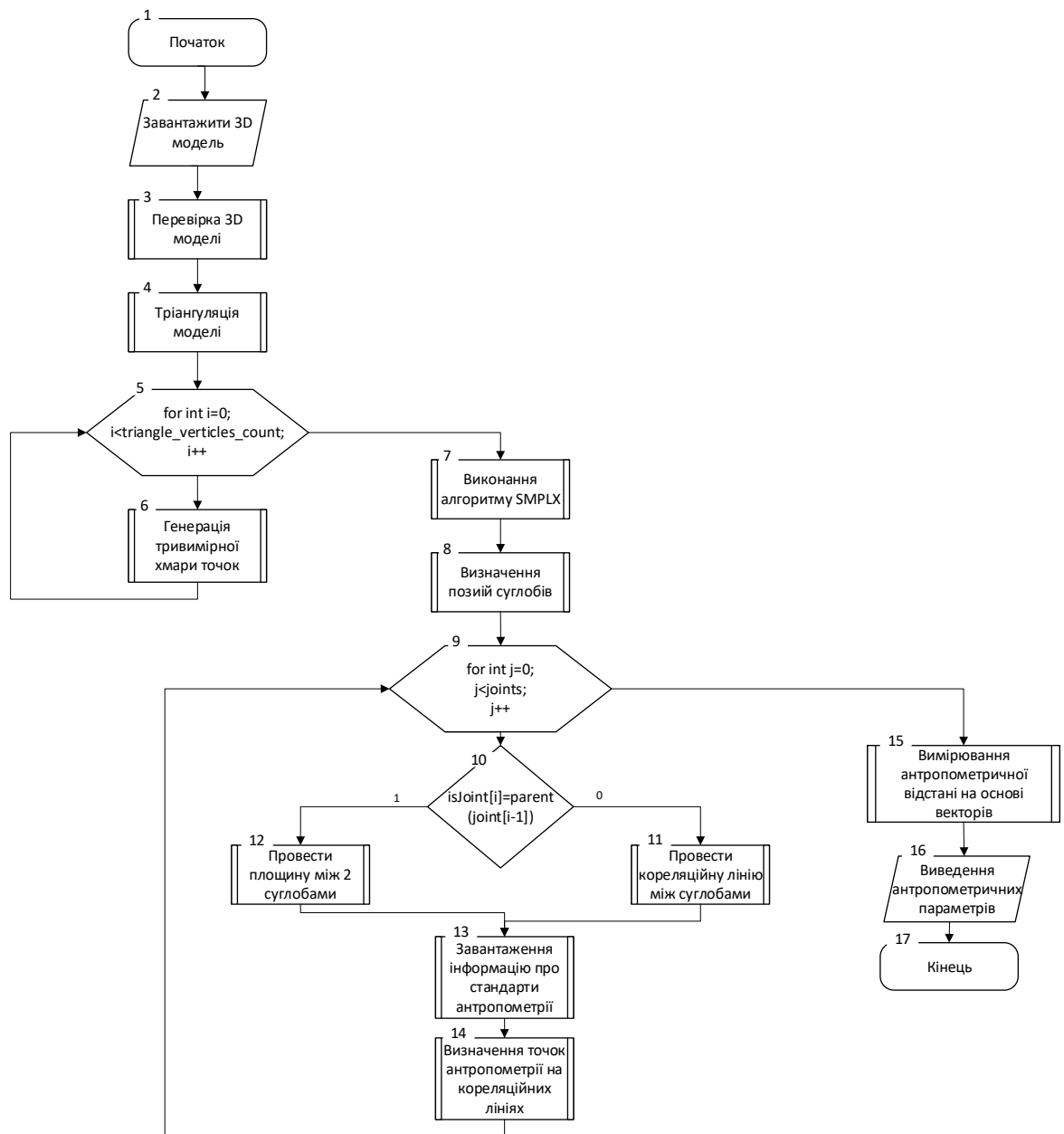


Рисунок 3.3 – Алгоритм визначення точок антропометрії

Отже, було обґрунтовано вибір технологій та інструментів, що використовувались для розробки програмних модулів. Також було розроблено основний алгоритм роботи програмних засобів та алгоритм визначення позицій точок антропометрії. Також було розроблену модель роботи програмного продукту за допомогою UML діаграм.

3.2 Розробка модуля визначення основних точок антропометрії на тривимірній моделі

Основні точки антропометрії є ключовими точками на тілі людини, які використовуються для вимірювання та аналізу різних антропометричних параметрів. Ці точки включають такі позначення, як зв'язки кісток, суглоби та інші анатомічні орієнтири.

Для розробки модуля визначення основних точок антропометрії на тривимірній моделі буде використано алгоритми комп'ютерного зору та обробки зображень. При розробці модуля важливо враховувати ряд технічних аспектів. Варто зауважити, що для визначення точок антропометрії, необхідно визначити позиції суглобів за допомогою алгоритму SMPLX. Даний алгоритм визначає позиції суглобів і переводить їх у тривимірні координати. Тому для визначення антропометричних точок було вирішено створити перелік суглобів, які відіграють ключову роль у визначенні точок антропометрії та забезпечують точність вимірюваних даних. Даний перелік суглобів було визначено у класі Measerments (рис. 3.4). Перелік суглобів було обрано відповідно до вищеповисаних міжнародних стандартів антропометрії. Всього було обрано 56 суглобів, для досягнення найвищої точності у визначенні точок антропометрії. Також клас містить перелік антропометричних вимірювань, що складається з точок антропометрії, що містять відомості про необхідні вимірювання, ґрунтуючись на міжнародних стандартах антропометрії.

Measerments
enum smplx_joints Enum anthropometric_measerments
SetMeaser(array[] joints) setTypeMeaser(Measer measer) AnthropometricMeaserments(ArrayList<Joints> joints)

Рисунок 3.4 – Діаграма класу, що визначає перелік суглобів необхідних для визначення точок антропометрії

Даний клас містить перелік основних суглобів та має два методи, що дозволяють визначити основні точки антропометрії.

Після визначення суглобів необхідною дією є визначення антропометричних вимірів. Для цього було визначено перелік основних антропометричних показників відповідно до ISO 8559:2017 "Антропометрія людського тіла". Далі даний перелік був реалізований у програмному застосунку з ідентифікацією позицій точок антропометрії. Для цього у класі Measerments було реалізовано функцію setMeaser(), яка формує антропометричний вимір відповідно до переліку вхідних суглобів (рис. 3.5).

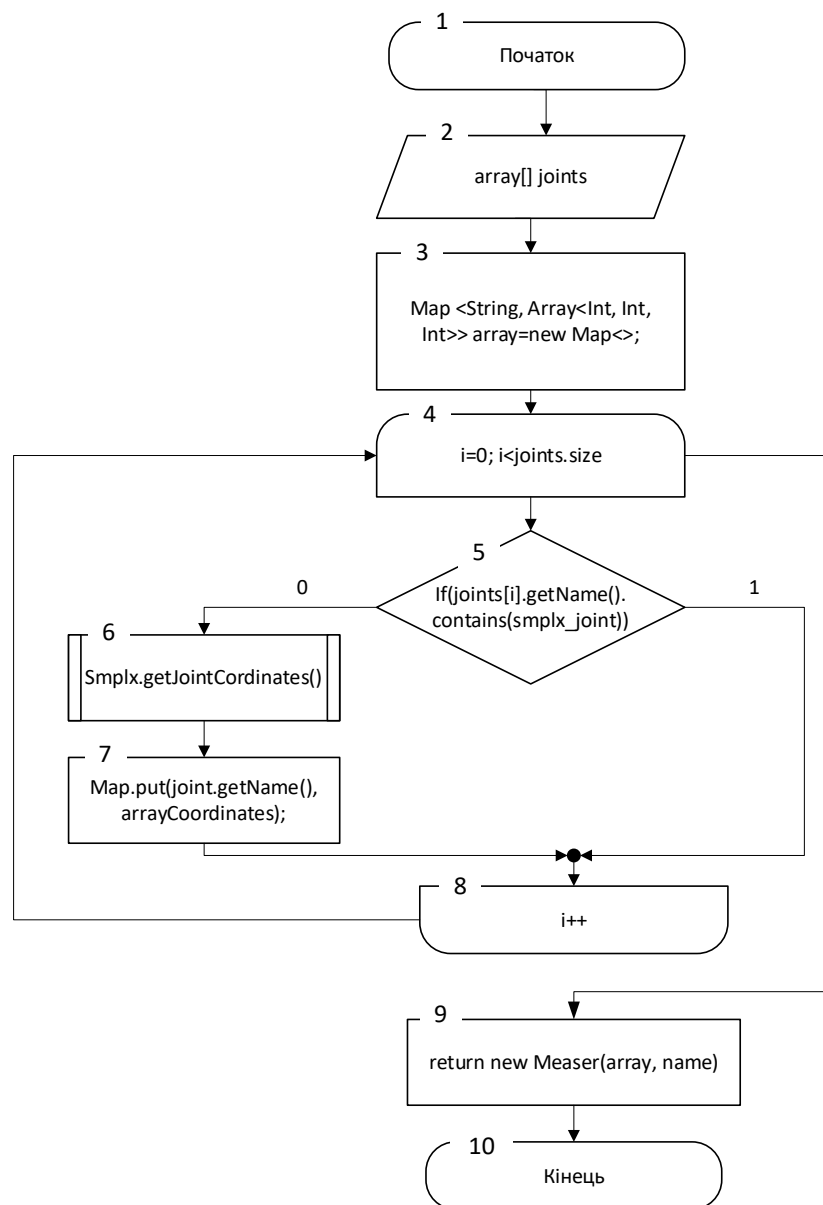


Рисунок 3.5 – Функція setMeaser()

Дана функція, крім отримання вимірювань та координат суглобів, також забезпечує можливість вказати додаткові параметри для аналізу, наприклад, точність вимірювань або тип використовуваного алгоритму. Після отримання переліку суглобів у вигляді масиву, функція створює контейнер, який може зберігати не лише назви суглобів, але й додаткову інформацію про них, таку як тип суглобу або область, до якої він належить. Для ефективності та точності аналізу кожного суглобу використовується алгоритм SMPLX, який забезпечує високу ступінь точності та надійність визначення позицій суглобів на тривимірній моделі. Після виконання аналізу кожного суглобу його координати записуються у відповідний масив. Результатом роботи даної функції є об'єкт класу Measer, що представляє собою повний набір вимірювань, включаючи координати суглобів, відповідно до заданих параметрів та вимог користувача.

Наступним кроком є визначення точок антропометрії, відповідно до міжнародних стандартів вимірювання. Для цього буде створюватись об'єкт класу Measer у функції AnthropometricMeaserments(ArrayList<Joints> joints). Дана функція буде створювати антропометричні вимірювання, що не передбачають використання суглобів, наприклад визначення точки вимірювання показника обхвату стегна, що знаходиться на відстані $1/3$ від суглобу коліна до тазового суглоба. Функція приймає перелік суглобів необхідних для вимірювання та ідентифікує їх назви, після чого відбувається звернення до реалізованої бази даних антропометричних вимірів та перевіряється чи міститься антропометричні точки, що пов'язані з отриманими суглобами. При наявності точки антропометрії, БД повертає суглоб від якого буде вестись вимірювання, відстань на якій знаходиться точка антропометрії, відповідно до міжнародних стандартів. Після чого відбувається вимірювання цієї відстані та обрахування координат нової точки. Наступним кроком є створення нової точки та об'єкту класу Measer. В подальшому дана точка також може бути використана для обрахування відстані, наприклад між суглобом та знайденою точкою антропометрії.

Реалізація даної функції наведена на рисунку 3.6.

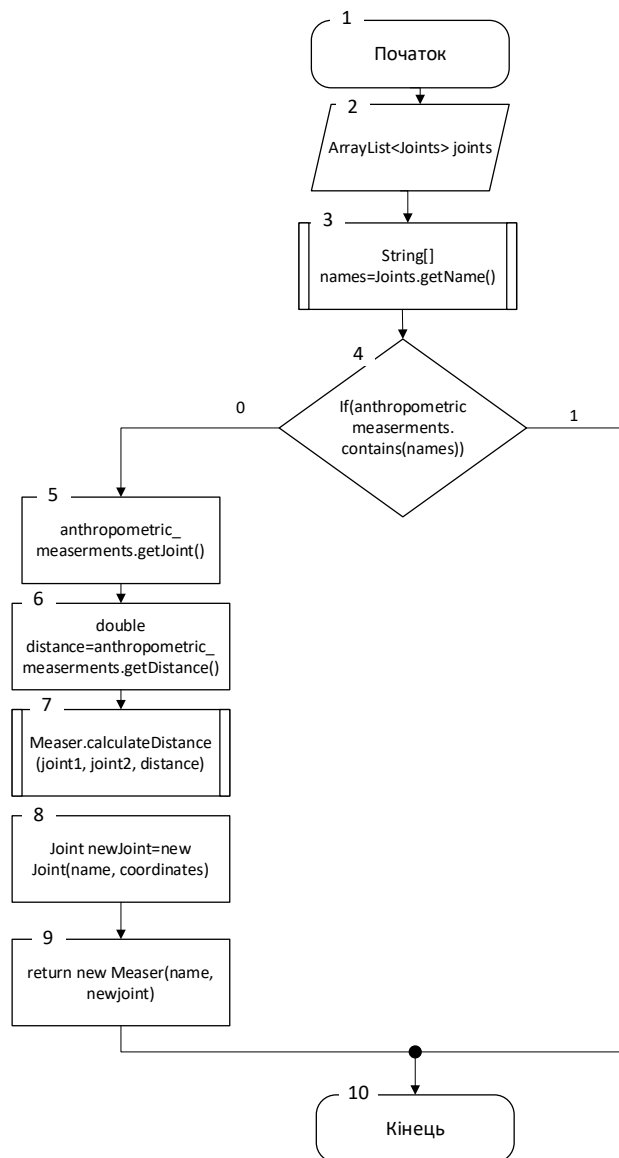


Рисунок 3.6 – Функція AnthropometricMeasermets(ArrayList<Joints> joints)

Наступним кроком є визначення необхідних антропометричних показників, а саме розподіл показників на необхідність визначення висоти та довжини між площинами чи суглобами, а також визначення обхвату. Для цього було створено функцію у якому ідентифікують необхідні виміри. А саме відбувається ідентифікація процесу вимірювання довжини чи обхвату. Дана функція дозволяє розподілити вимірювання на 2 категорії, а саме: вимірювання криволінійної довжини та вимірювання обхвату. Дане розмежування ґрунтується на необхідності застосування різних формул для обчислення антропометричних параметрів. Реалізацію даного функціоналу наведено у

функції setTypeMeaser() (рис. 3.7).

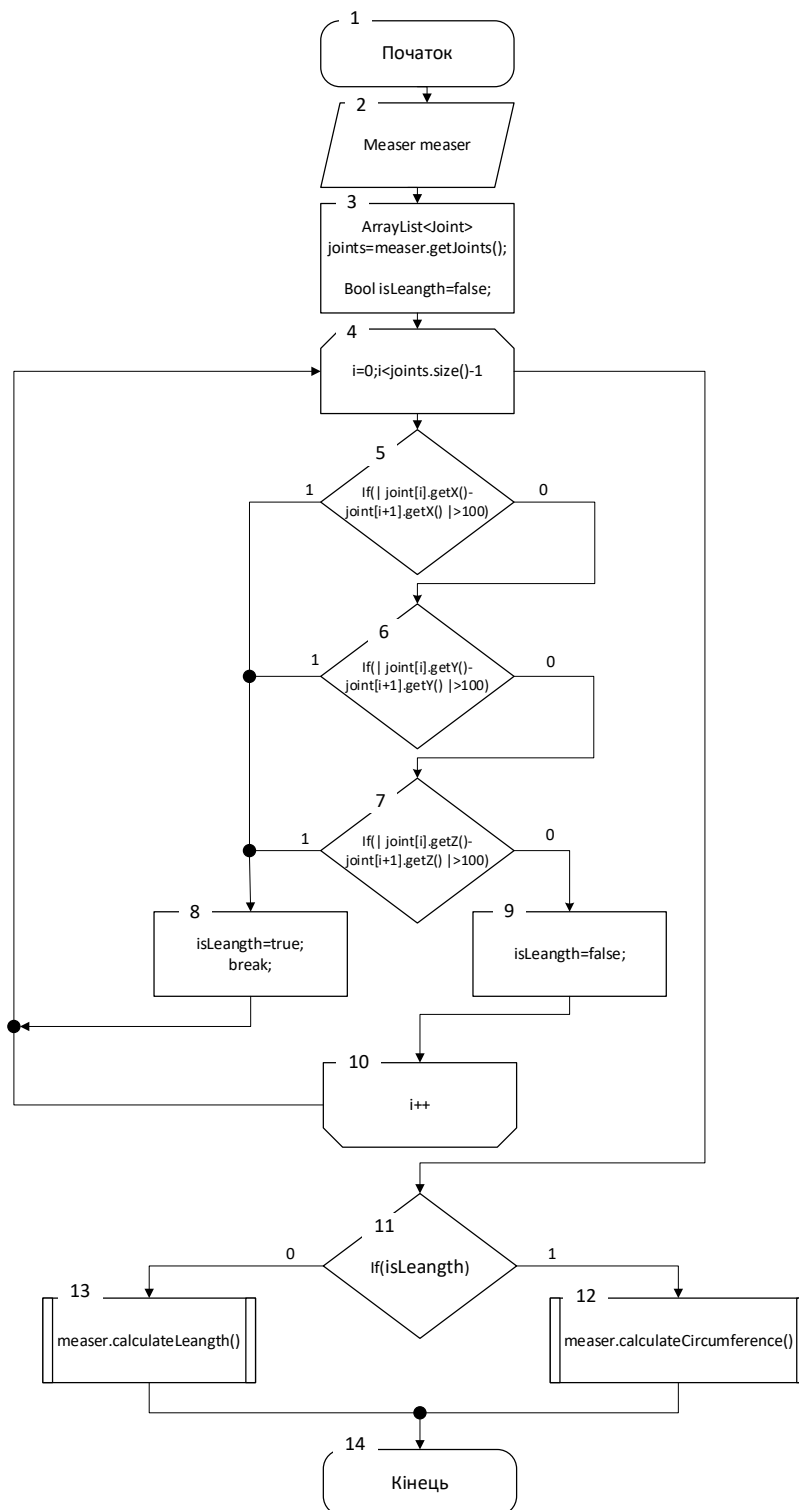


Рисунок 3.7 – Функція setTypeMeaser()

Дана функція забезпечує розподіл антропометричних вимірювань на типи, а саме які формули для отримання антропометричної відстані будуть

використовуватись. Для цього функція отримує у якості параметра об'єкт класу Measer та отримує з нього масив точок, що формують антропометричні відстані. Наступним кроком є перевірка відхилень за осями x,y та z точок антропометрії. При умові відхилення хоча б одної осі на 100 одиниць буде застосовуватись вимірювання криволінійної довжини, інакше буде використано вимірювання обхвату.

3.3 Розробка модуля вимірювання антропометричних параметрів на тривимірній моделі

Головною задачею модуля вимірювання антропометричних параметрів є точне вимірювання основних антропометричних показників на тривимірній моделі тіла людини. Для реалізації цієї задачі було створено клас Measurer, що приймає в себе в якості параметра тривимірну модель тіла людини. При створенні і запуску класу автоматично виконується тріангуляція та тривимірна хмара точок, визначаються позиції суглобів та точки антропометрії (рис. 3.8).

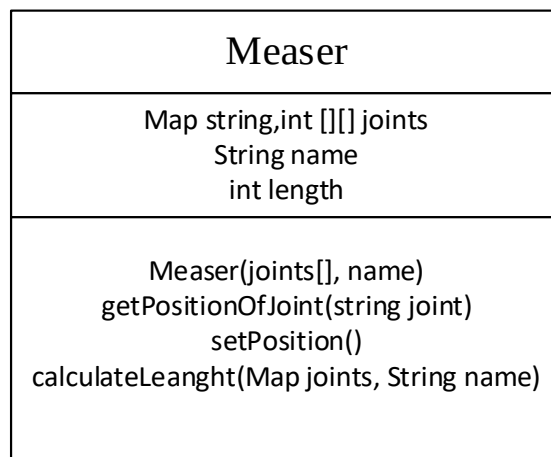


Рисунок 3.8 – Клас Measurer

Крім того було розроблено окрему функцію для аналізу тривимірної моделі, а саме перевірки тривимірної моделі тіла людини на наявність дефектів чи знаходження невідповідностей чи аномалій (рис. 3.9) тривимірної моделі.

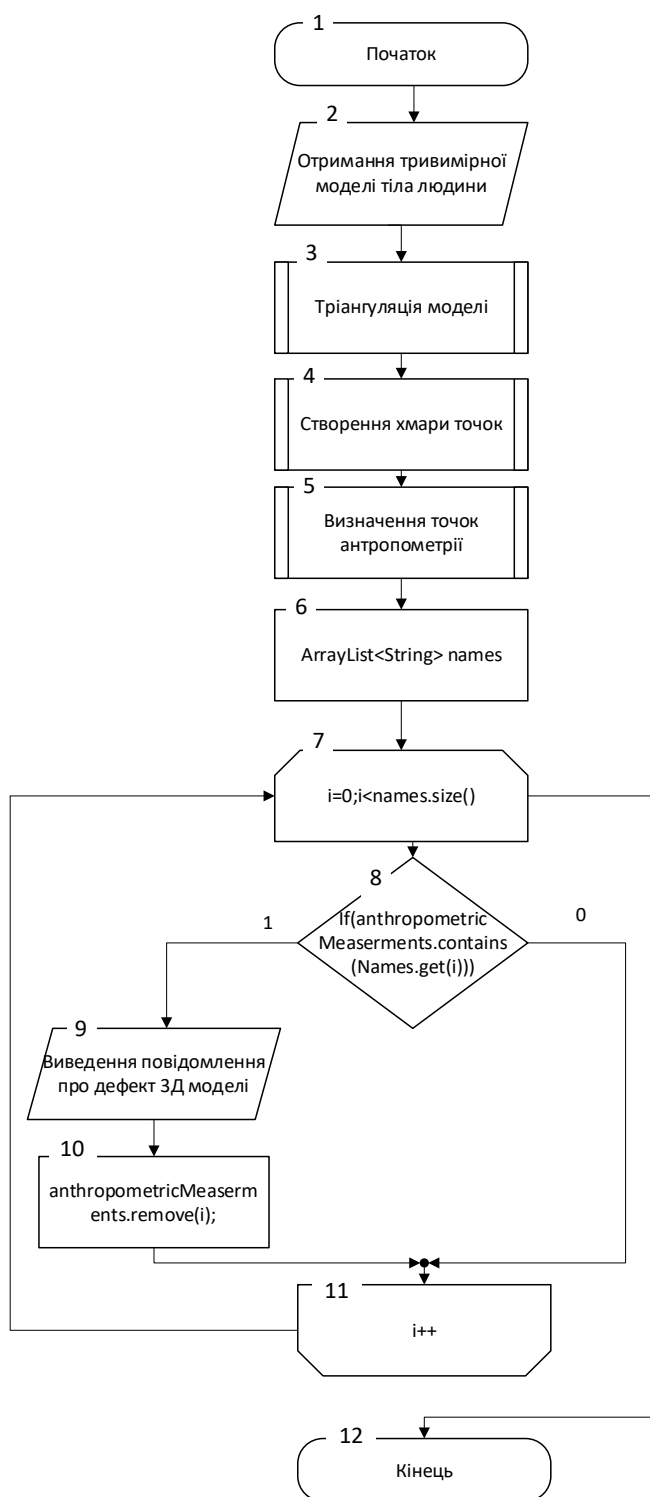


Рисунок 3.9 – Функція аналізу тривимірної моделі

Наступним кроком є налагодження алгоритмів вимірювання відстаней антропометричних параметрів. Для цього було розроблено дві формули, що базуються на формулах знаходження відстані векторів. Дані формули дозволяють виміряти відстань між двома точками на тривимірній моделі та

параметри обхвату.

На рисунку 3.10 відображено формулу знаходження довжини антропометричних параметрів тіла людини. Дана функція отримує тривимірні координати необхідних точок чи суглобів, що були раніше визначені, та обраховує відстань між двома тримірними точками. Для вимірювання параметру функція приймає в себе інформацію про необхідний вимір, перевіряє чи доступне вимірювання даного параметру та виконує вимірювання між отриманими координатами.

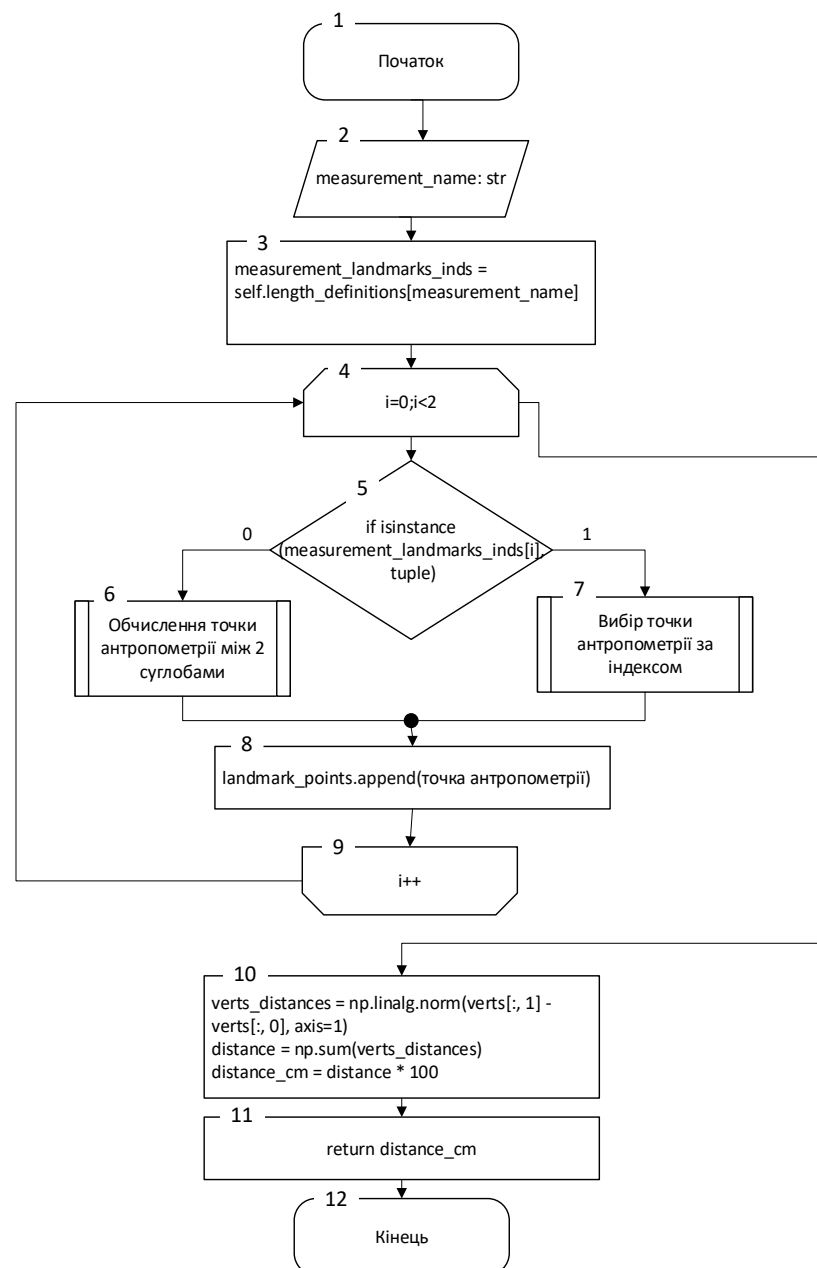


Рисунок 3.10 – Функція вимірювання параметрів довжини

Обрахування відстані відбувається шляхом знаходження суми довжин ліній, що прокладені між вершинами трикутників, що були отримані внаслідок тріангуляції та виконання модуля формування тривимірної хмари точок.

Функція обрахування параметрів обхвату на тривимірній моделі тіла людини відображена на рисунку 3.11.

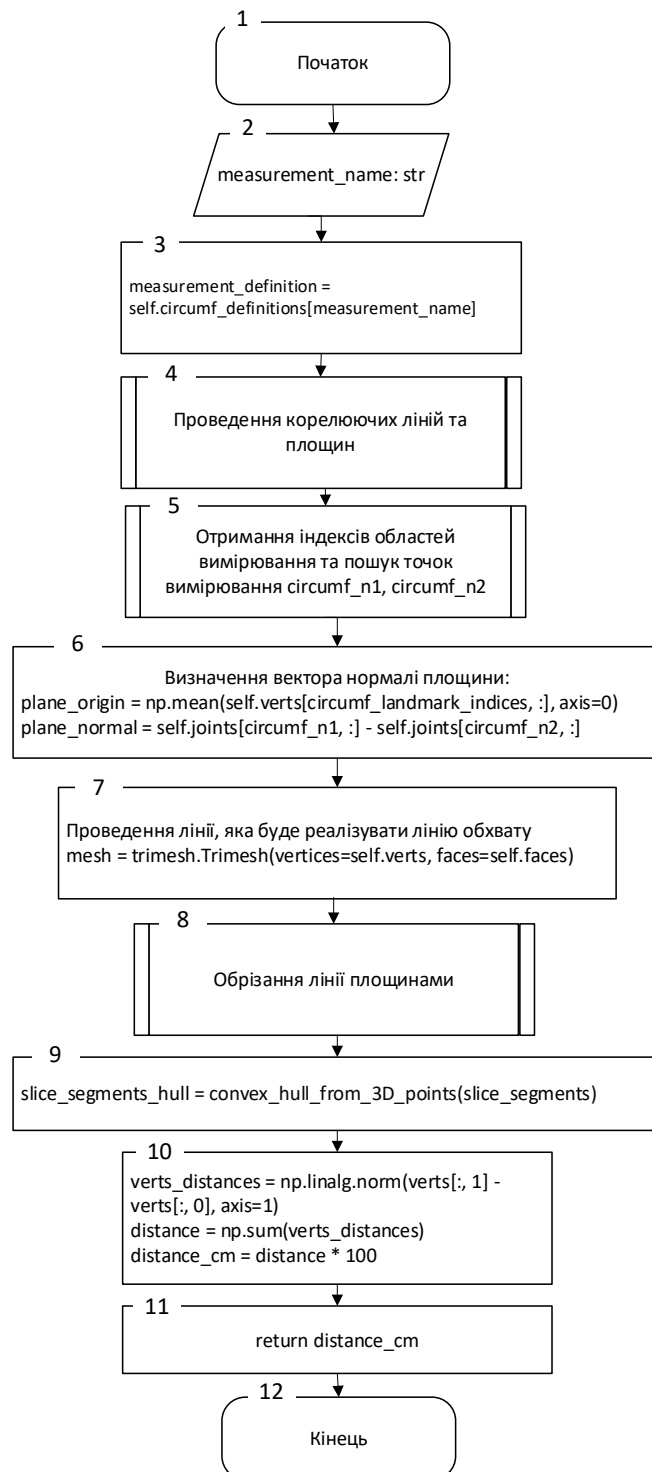


Рисунок 3.11 – Функція обрахування параметрів обхвату

Ця функція використовується для точного визначення обхвату антропометричних параметрів, таких як обхват талії, обхват стегон тощо. Першим кроком функції є отримання позиції необхідної точки на тривимірній моделі, яка відповідає конкретному антропометричному параметру. Ця позиція визначається відповідно до стандартів антропометрії та базується на анатомічних орієнтирах та позиціях суглобів. Далі функція проводить горизонтальну лінію через вершини трикутників, що утворюють модель, відповідно до позиції суглобу, пов'язаного з конкретним антропометричним параметром. Далі відбувається обчислення довжини отриманої горизонтальної лінії, яка відображає обхват конкретного антропометричного параметра.

На рисунку 3.12 представлена функція, яка призначена для вимірювання висоти людини.

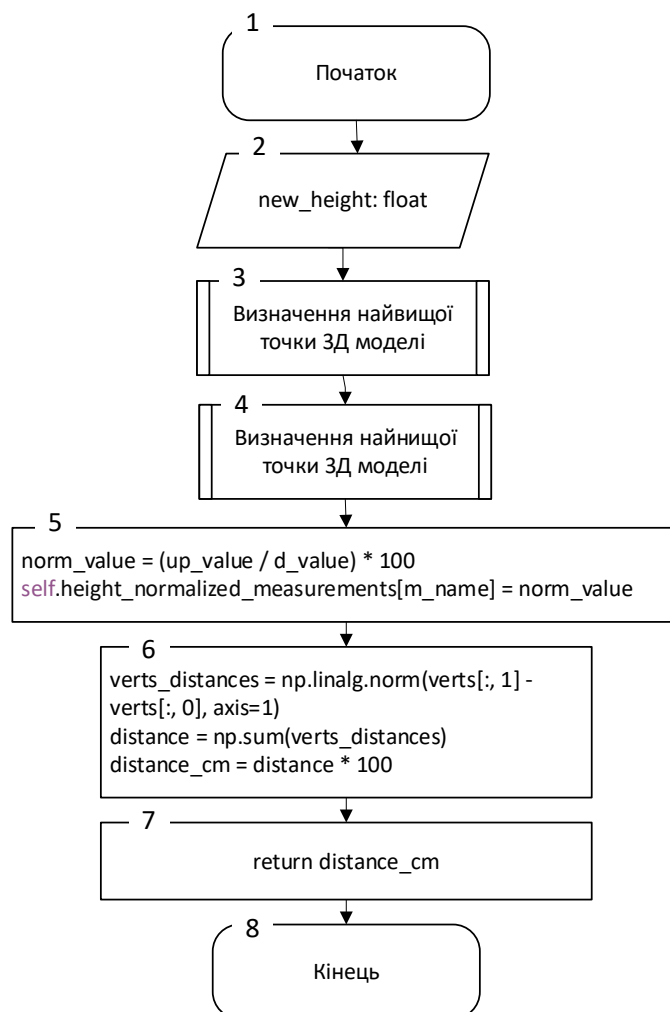


Рисунок 3.12 – Функція обрахування параметру висоти людини

Ця функція опирається на інформацію про позиції суглобів ніг та голови, яку вона отримує у вигляді вхідних даних. Першим кроком функції є визначення найвищої та найнижчої точок у хмарі точок, що репрезентують тривимірну модель людини. Для цього враховується позиція суглобів ніг та голови. Далі проводиться побудова площини між суглобами ніг та найнижчою точкою. Ця площина централізується відносно позиції найвищої точки тривимірної моделі. Після цього за допомогою горизонтальної лінії, що проходить через найвищу точку та центральну точку побудованої площини, визначається висота людини. Це досягається за допомогою використання формул суми тривимірних векторів, які дозволяють знайти відстань між цими двома точками у тривимірному просторі.

3.4 Розробка модуля інтерфейсу користувача

При розробці інтерфейсу програмного забезпечення було приділено особливу увагу його зрозумілості та зручності для користувача. Основним кольором інтерфейсу обрано зелений, оскільки даний колір асоціюється з медициною. Допоміжними кольорами обрано білий та чорний, оскільки вони контрастують один одному. При відкритті програмного додатку, користувача зустрічає екран завантаження із назвою системи (рис. 3.13).

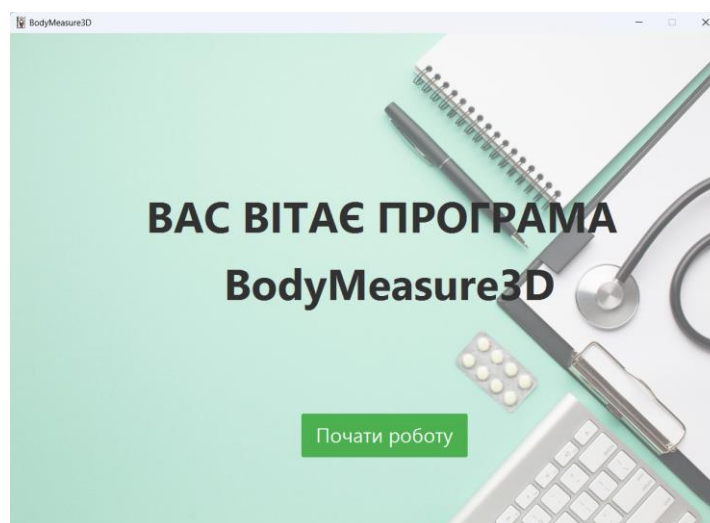


Рисунок 3.13 – Екран завантаження

Оскільки програмні засоби передбачають взаємодію із тривимірними моделями та антропометричними параметрами, було вирішено створити віртуальне робоче середовище із чітким розмежуванням між антропометричними параметрами та тривимірною візуалізацією. Також було створено меню програмного застосунку (рис. 3.14), для полегшення взаємодії із розробленим функціоналом.

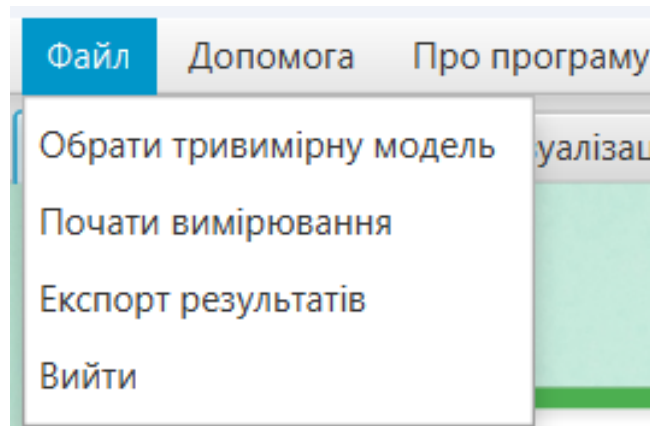


Рисунок 3.14 – Меню програмного застосунку

На рисунку 3.15 відображено початковий стан робочого середовища, що допомагає покращити користувацьку імерсивність та полегшити взаємодію із застосунком.

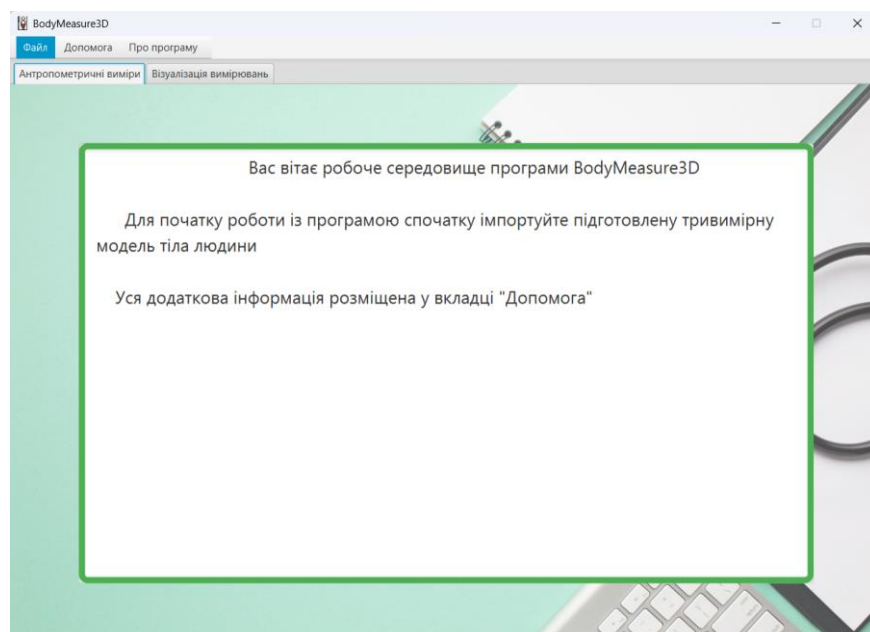


Рисунок 3.15 – Початковий стан робочого середовища застосунку

Отримавши доступ до моделі, користувач може активувати автоматизоване вимірювання антропометричних показників, натиснувши відповідний пункт меню в програмі. Це запускає процес, під час якого програма або середовище здійснює взаємодію з імпортованою моделлю. Після натискання на відповідний пункт меню стан віртуального робочого середовища змінюється, щоб відобразити результати обчислень.

На рисунку 3.16 можна побачити, як віртуальне середовище змінюється після початку автоматизованого вимірювання. Новий стан включає відображення обчислених антропометричних параметрів для конкретної тривимірної моделі. Важливо зазначити, що цей процес дозволяє користувачеві отримати швидкий доступ до даних про антропометрію без необхідності в ручному вимірюванні або обробці результатів. Це спрощує процес аналізу та дозволяє швидше отримати необхідні дані для подальшого дослідження чи використання.

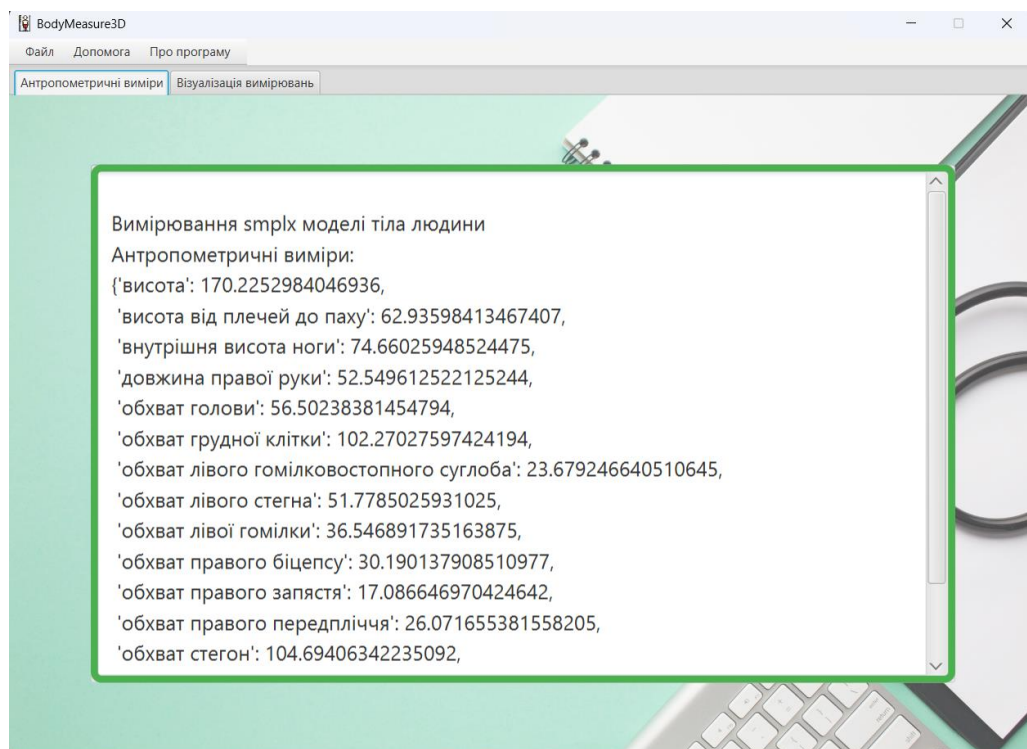


Рисунок 3.16 – Стан робочого середовища після вимірювання

Після вимірювання антропометричних показників, користувач може

перейти на екран візуалізації (рис. 3.17), на якому відображена тривимірна модель людини із позначеними антропометричними параметрами, позиціями суглобів і точок антропометрії, а також описом візуалізації, з яким можна взаємодіяти.

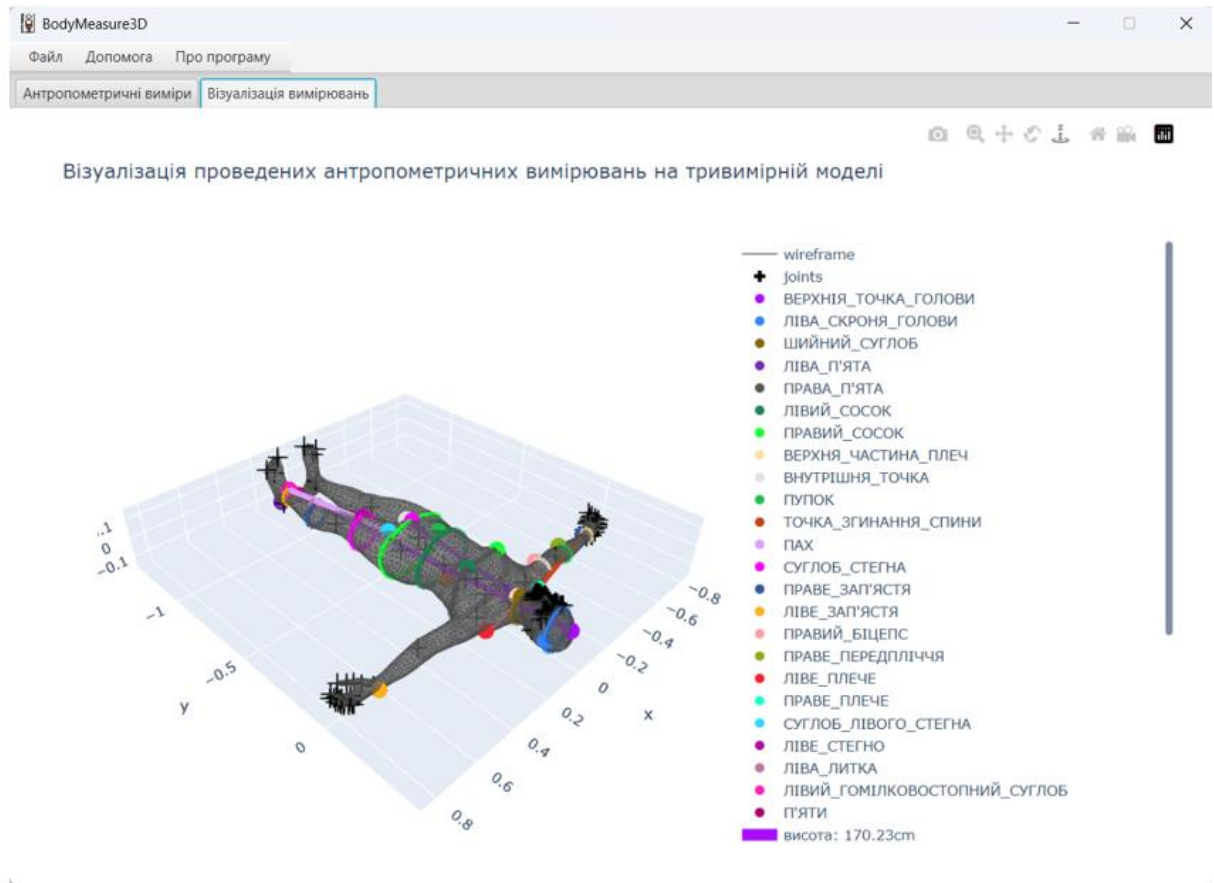


Рисунок 3.17 – Вікно візуалізації результатів вимірювання

Перейшовши на екран довідки (рис.3.18), користувач може ознайомитися із основним функціоналом програмного застосунку, а також переглянути додаткову інформацію про програмний застосунок.

На екрані «Про програму» (рис. 3.19) користувач може побачити інформацію про програмний застосунок, його основне призначення, доступний функціонал застосунку. Також користувач на даному екрані може знайти інформацію про розробника програмного застосунку, а також інформацію про версію програмного застосунку.

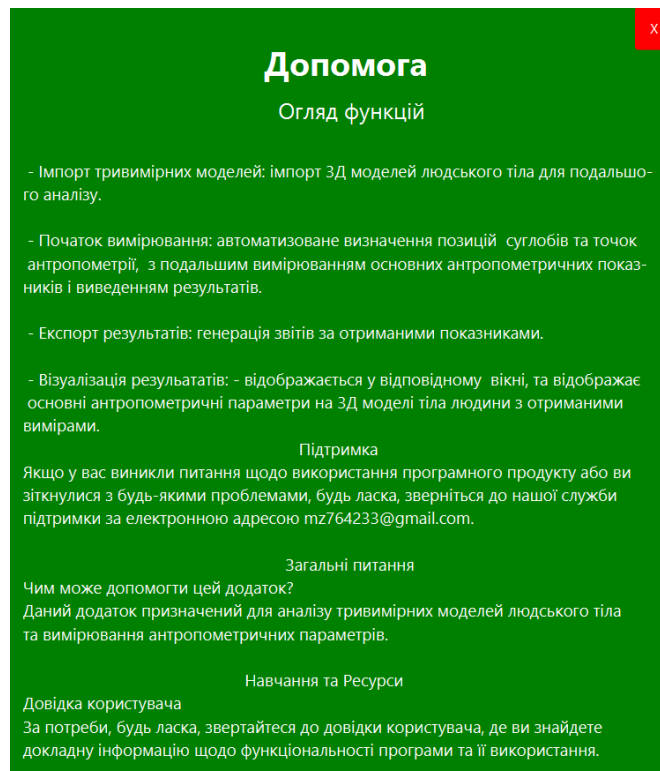


Рисунок 3.18 – Екран «Довідка»

Також на екрані «Про програму» розміщено логотип програмного застосунку.



Рисунок 3.19 – Екран «Про програму»

Розробка графічного інтерфейсу програмного застосунку була проведена у середовищі розробки IntelliJ IDEA з використанням технології SceneBuilder21 та мови програмування JavaFX.

3.5 Висновки

У третьому розділі було проведено розробку алгоритмів програмного забезпечення. Також було наведено засоби розробки програмного застосунку. У розділі також було описано розробку основних програмних модулів для визначення позицій точок антропометрії, вимірювання антропометричних параметрів на тривимірній моделі тіла людини та інтерфейсу користувача.

4 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Тестування програмного забезпечення

Тестування програмного забезпечення – це процес перевірки програмного продукту з метою виявлення помилок, дефектів або неполадок та перевірки відповідності вимогам замовника. Основна мета тестування – забезпечити якість та надійність програмного продукту перед його випуском у виробництво або початком експлуатації.

Для початку тестування необхідно встановити тестову версію програми на комп'ютер з операційною системою Windows або MacOS, яка буде використовуватись під час тестування. Перед початком тестування також важливо забезпечити належну документацію, включаючи специфікації вимог та план тестування. Проведення тестування включає кілька етапів та видів тестів.

Першим етапом є функціональне тестування, яке спрямоване на перевірку правильності роботи програми. Це включає в себе тести на введення та виведення даних, тести на реакцію програми на негативні вхідні дані, а також тести на відповідність бізнес логіці програми.

Другим етапом є тестування відмов, спрямоване на виявлення дефектів та неполадок у програмному забезпеченні. Проводиться аналіз програмного забезпечення з точки зору його здатності відновлювати роботу після виникнення помилок або відмов. Такі сценарії тестування можуть включати штучне введення неправильних даних, непередбачувані вхідні параметри або спроби взаємодії з програмою в умовах, які відповідають екстремальним ситуаціям або навіть випадкам, коли програмне забезпечення працює у відсутності зв'язку з іншими компонентами або системою.

Після завершення функціонального тестування та тестування відмов, проводиться додаткове тестування для визначення продуктивності програмного забезпечення. Цей етап включає тести швидкодії, які спрямовані на оцінку швидкості виконання певних операцій або завдань програмою. Під час цього

тестування вимірюються час відповіді програми на різноманітні запити або дії користувача.

Окремим етапом тестування може бути тестування сумісності, яке спрямоване на перевірку взаємодії програмного забезпечення з різними операційними системами, версіями програм та апаратними платформами. Під час цього тестування перевіряється, чи працює програма на різних конфігураціях та середовищах, а також чи виникають конфлікти або проблеми під час взаємодії з іншими програмами.

Під час першого етапу тестування програмного забезпечення перевіряється реакція програми на невірний формат тривимірної моделі. Якщо користувач спробує імпортувати тривимірну модель тіла людини у форматі, що не визначається як .rkl, програма має виявити цю помилку та повідомити користувача про невідповідність формату. Далі, програма пропонує користувачеві імпортувати іншу тривимірну модель у правильному форматі .rkl. На рисунку 4.1 показано інтерфейс програми з повідомленням про невірний формат імпортованих даних та запропонованою альтернативою для імпортування правильного формату тривимірної моделі.

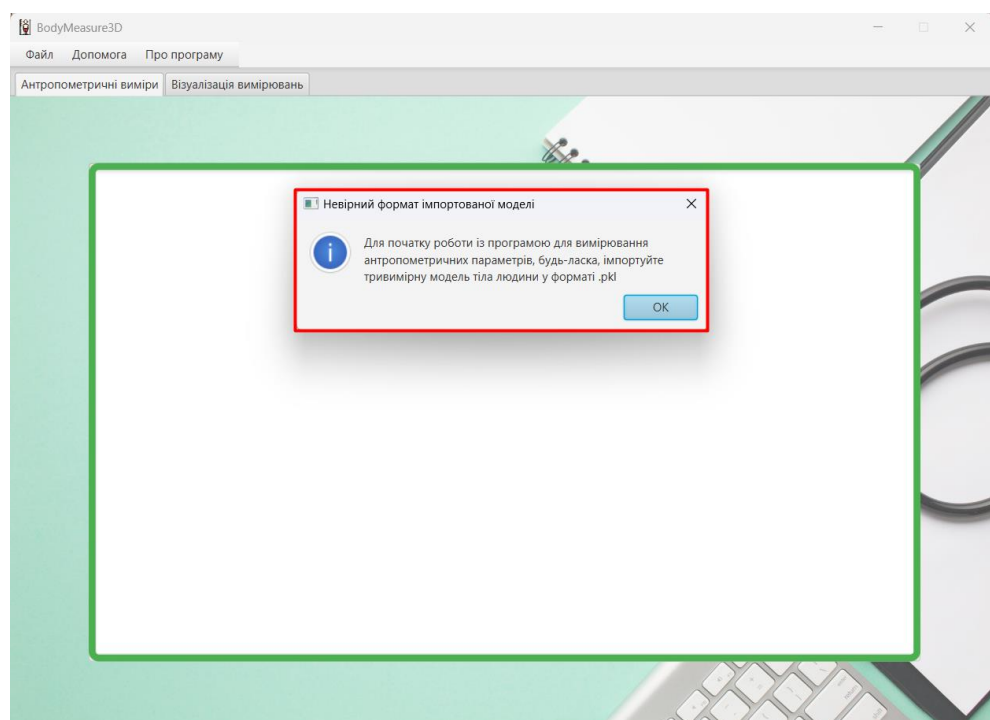


Рисунок 4.1 – Імпортування 3Д моделі іншого формату

Після проведення тестування програмного забезпечення на імпортування тривимірних моделей іншого формату, за винятком .pkl, було прийнято рішення провести додаткове тестування, спрямоване на перевірку реакції програмного забезпечення на відсутність вхідних даних. Це було зроблено з метою визначення, як програма реагує на таку ситуацію та чи забезпечує вона користувача відповідними повідомленнями або інструкціями. У результаті цього тестування (рис. 4.2) було виявлено, що програмне забезпечення обробляє такий сценарій використання користувачем та надає інформацію про можливий шлях рішення.

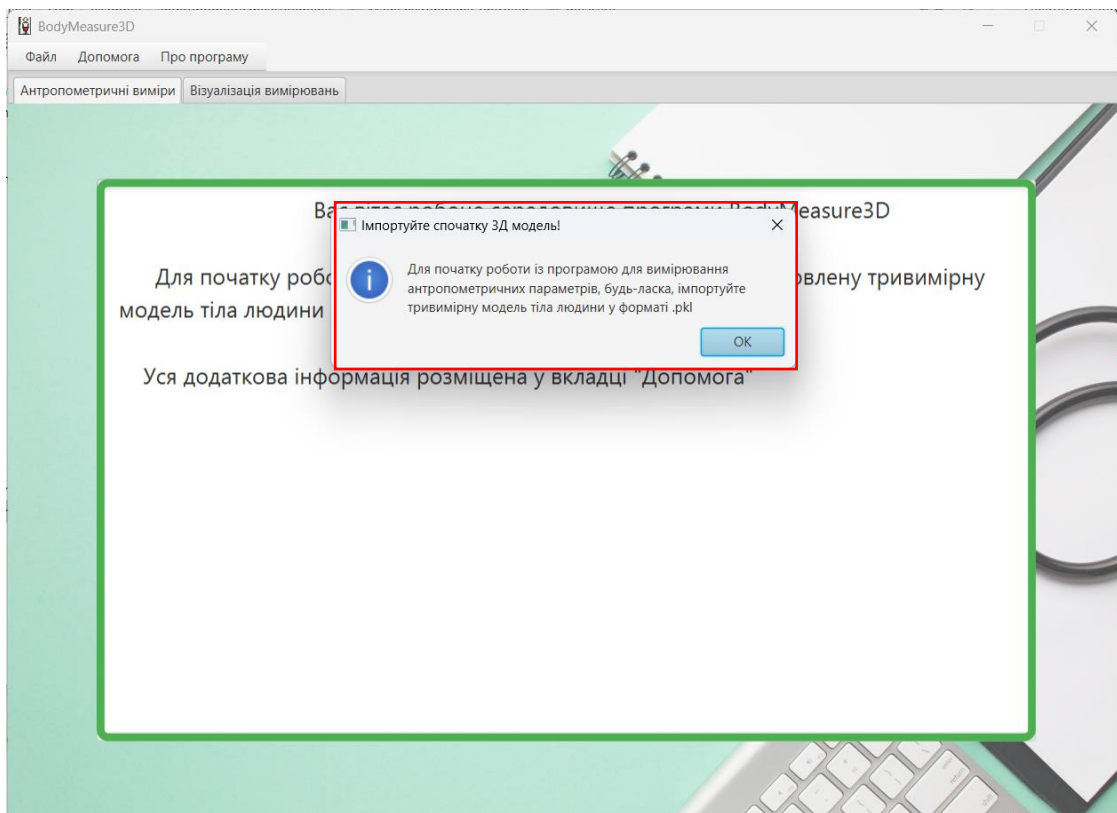


Рисунок 4.2 – Спроба вимірювання параметрів без вхідних даних

Після успішної перевірки функціональності програмного забезпечення на правильність обробки вхідних даних, наступним кроком є проведення тестування на реакцію програми на імпорт пошкодженої тривимірної моделі тіла людини. В цьому випадку, використовується тривимірна модель з відсутньою правою рукою. На рисунку 4.3 наведена реакція системи під час обробки

пошкодженій тривимірній моделі. Результати тестування свідчать про те, що програмне забезпечення успішно обробляє такі випадки та може виміряти антропометричні параметри тривимірних моделей навіть у разі виявлення дефектів. Детальний аналіз результатів тестування показав, що програмне забезпечення коректно виводить інформацію про модель, а також ідентифікує знайдені дефекти. Серед цих дефектів виділяються відсутні суглоби, які є основою для вимірювання антропометричних параметрів, і які через особливості 3D моделі тіла людини стають недоступними для вимірювання.

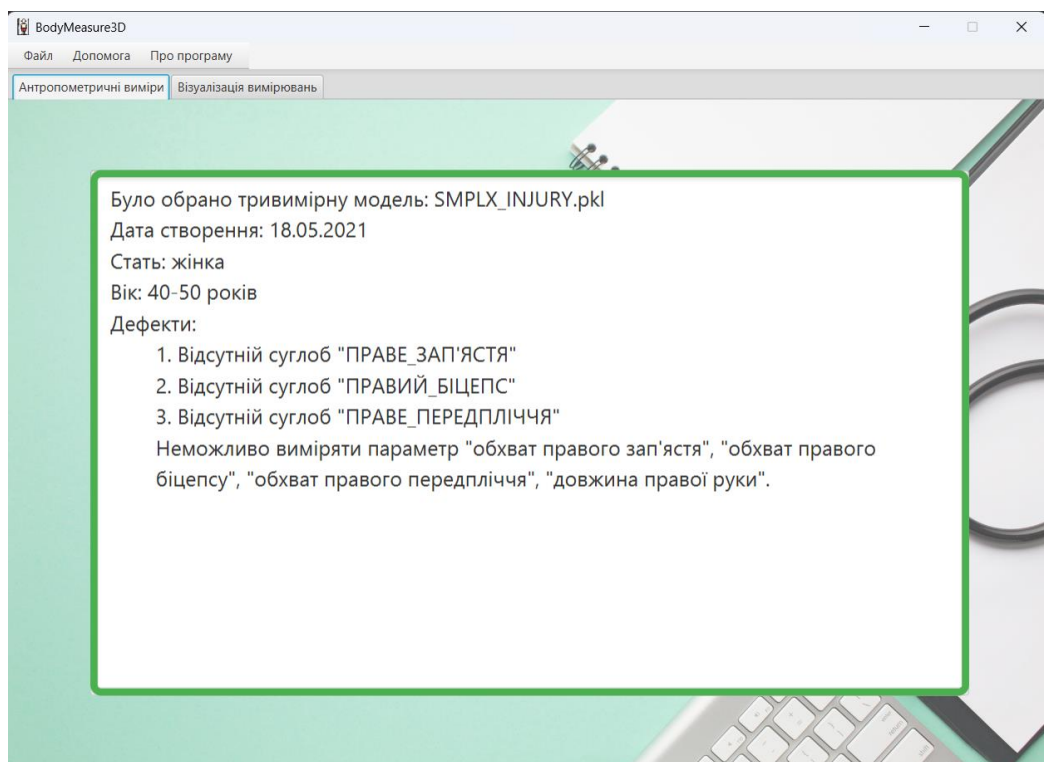


Рисунок 4.3 – Результат імпортування 3D моделі з дефектами

Після перевірки імпортування у робоче середовище програмного забезпечення даних, які були не дійсними або пошкодженими, перейдемо до імпортування звичайної тривимірної моделі тіла людини. Цей процес дозволить переконатися, що програмне забезпечення працює коректно з валідними даними та може ефективно обробляти їх. Після завершення імпортування проаналізуємо результати, які програмне забезпечення виведе щодо імпортованої тривимірної моделі у головне робоче середовище. Результати тестування представлені на

рисунку 4.4.

У рамках тестування програмного забезпечення також було проведено аналіз його сумісності з різними версіями операційних систем та мінімальною конфігурацією апаратного забезпечення. Зокрема, програмний продукт був перевірений на сумісність з наступними операційними системами: Windows 10, Windows 8.1 та MacOS. Під час тестування використовувалися як високопродуктивні комп'ютери, так і пристрої з обмеженими технічними можливостями для визначення роботи програмного забезпечення на різних конфігураціях апаратного забезпечення. Результати тестування показали успішну роботу програмного забезпечення як на потужних, так і на менш потужних пристроях, що вказує на його високу оптимізацію та ефективність роботи.

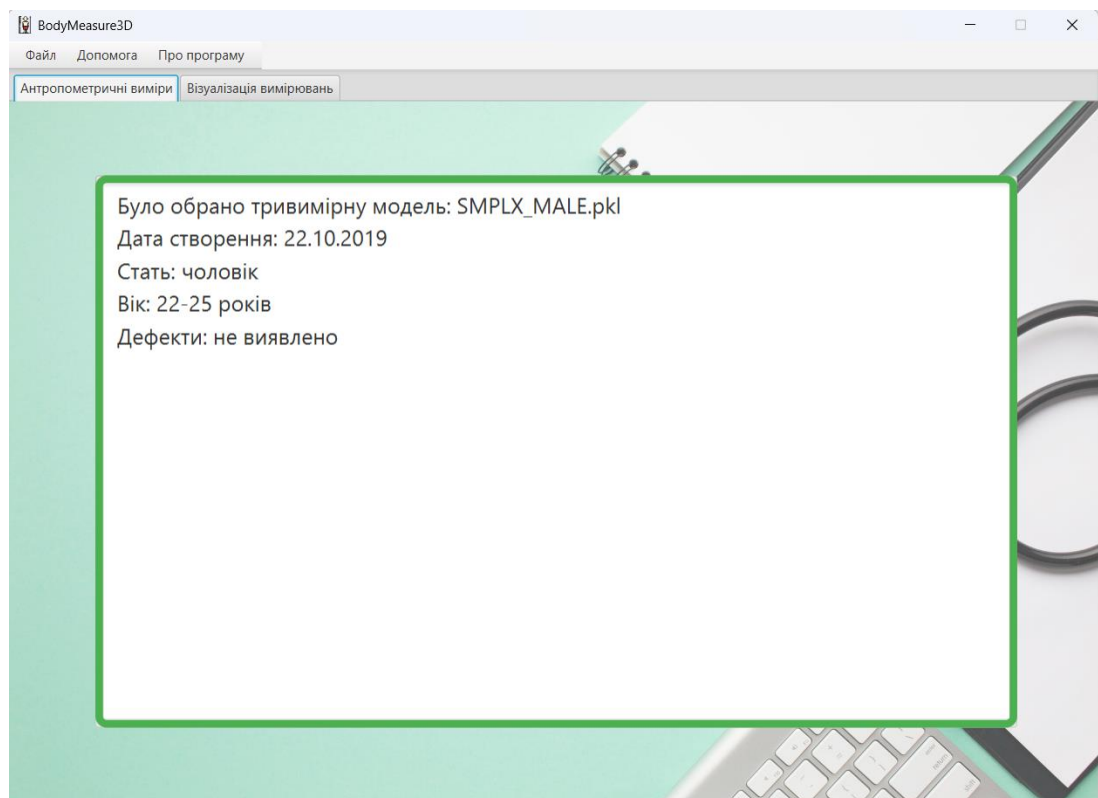


Рисунок 4.4 – Результат успішного імпортування чоловічої 3Д моделі

У додаток до тестування на тривимірній моделі тіла чоловіка, програмне забезпечення було також протестовано на іншій тривимірній моделі, а саме на

3D-моделі жінки. Ця модель не мала анатомічних дефектів та була представлена відповідно до стандартних параметрів жіночого тіла. Результати цього тестування представлені на рисунку 4.5. За цими результатами можна побачити, що програмне забезпечення здатне до вимірювання антропометричних параметрів на різних вхідних даних та вхідних тривимірних моделях, включаючи моделі тіл обох статей. Такі результати свідчать про високу функціональність та адаптивність програмного забезпечення для вимірювання антропометричних параметрів на різних типах тривимірних моделей.

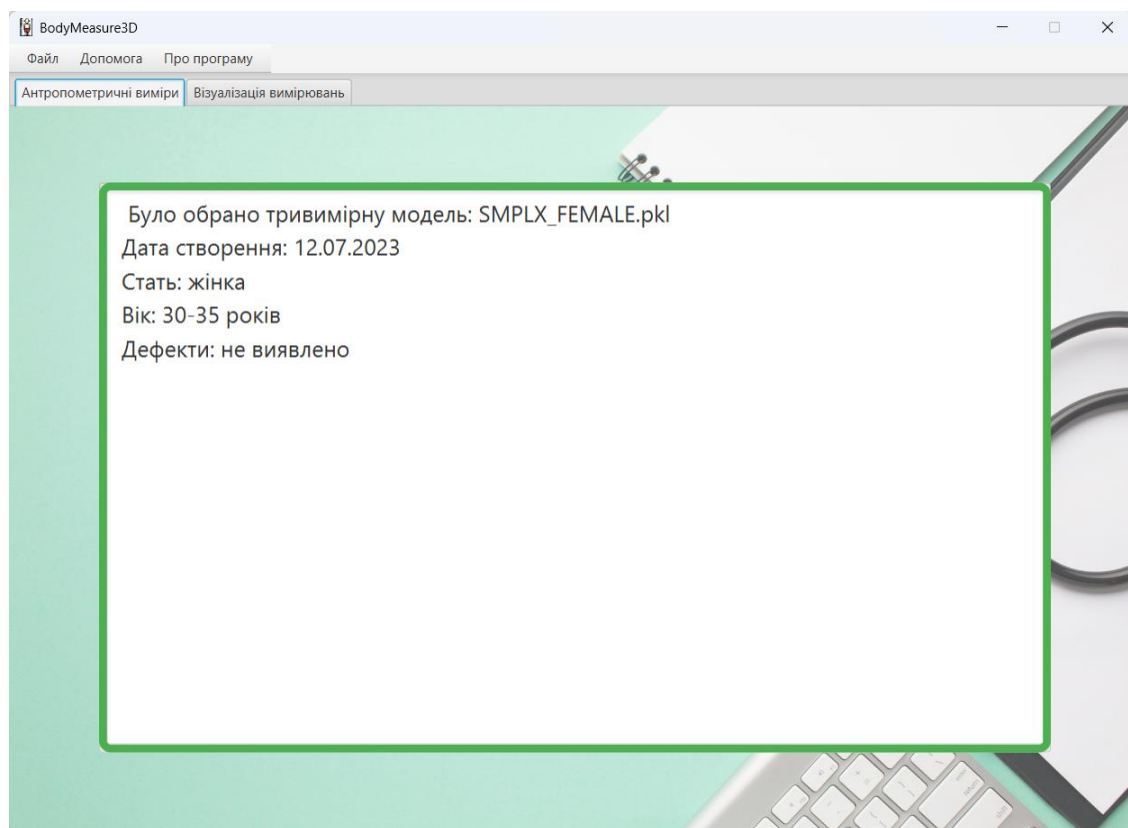


Рисунок 4.5 – Результат успішного імпортування жіночої 3Д моделі

Після успішного завершення тестування імпорту, аналізу та обробки вхідних даних програмним забезпеченням, було проведено тестування функціоналу вимірювання та відображення результатів антропометричних параметрів тривимірного тіла чоловіка і жінки окремо.

Для цього необхідно було імпортувати одну з тривимірних моделей у робоче середовище програмного забезпечення та провести вимірювання,

натиснувши на відповідну кнопку у меню "Файл" -> "Почати вимірювання".

На рисунку 4.6 зображено результати вимірювання антропометричних параметрів тривимірного тіла чоловіка у робочому середовищі програмного забезпечення.

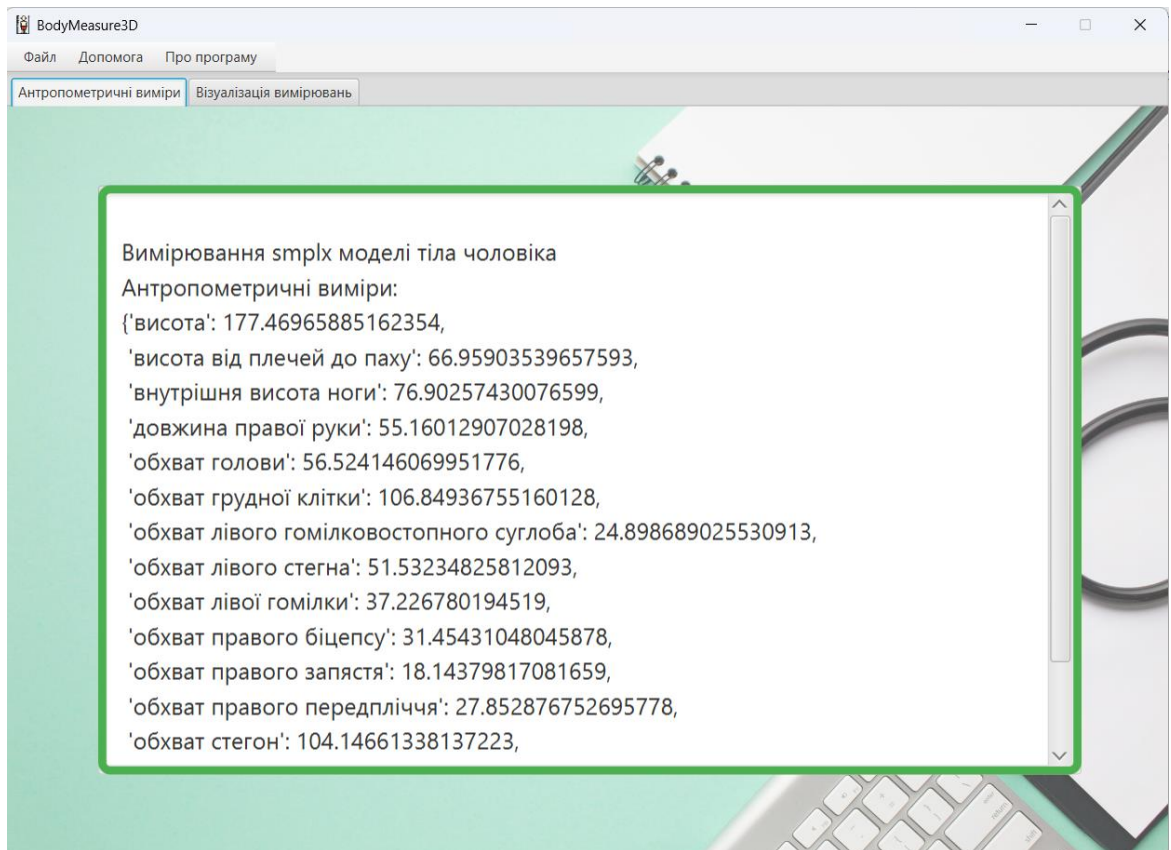


Рисунок 4.6 – Результати вимірювання 3Д тіла чоловіка у робочому середовищі

Крім того, на рисунку 4.7 представлені результати вимірювання антропометричних параметрів тривимірного тіла жінки у робочому середовищі програмного забезпечення. Як можна побачити з результатів вимірювання двох різних тривимірних моделей, програмне забезпечення демонструє здатність точно вимірювати та зручно відображати основні антропометричні параметри у робочому середовищі.

Додатково, з результатів тестування випливає, що програмне забезпечення здатне ефективно та точно вимірювати антропометричні параметри незалежно

від статі, що підтверджує його універсальність та високу продуктивність.

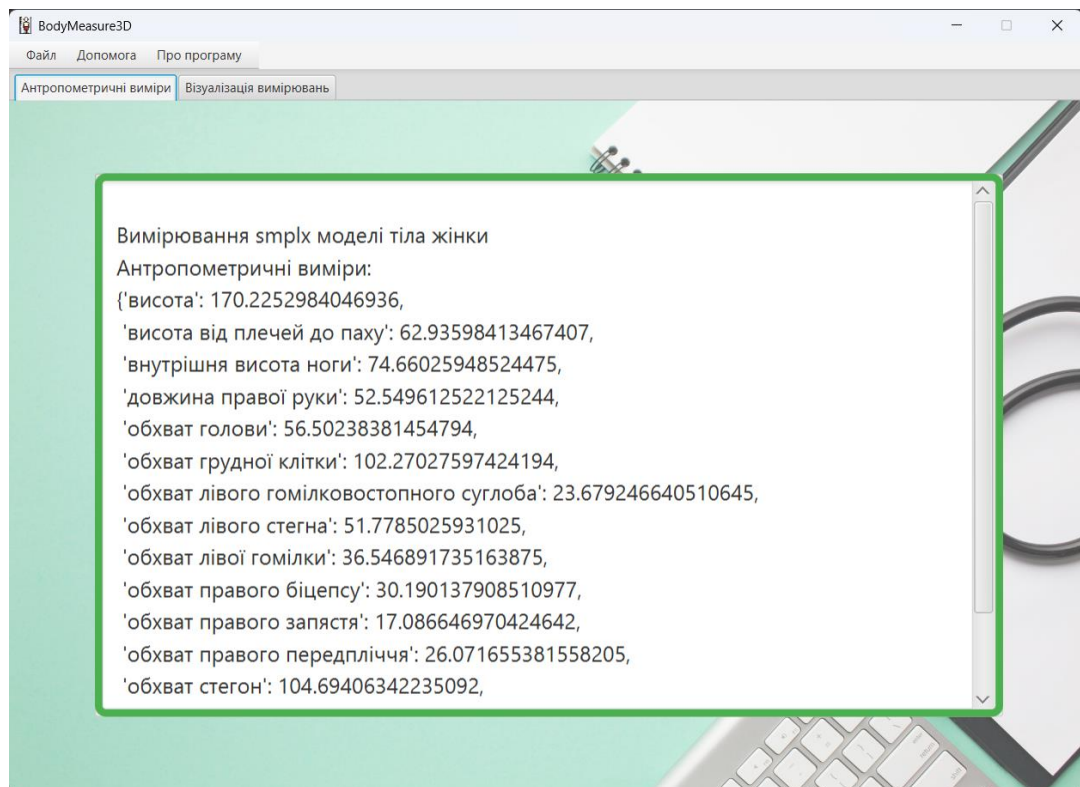
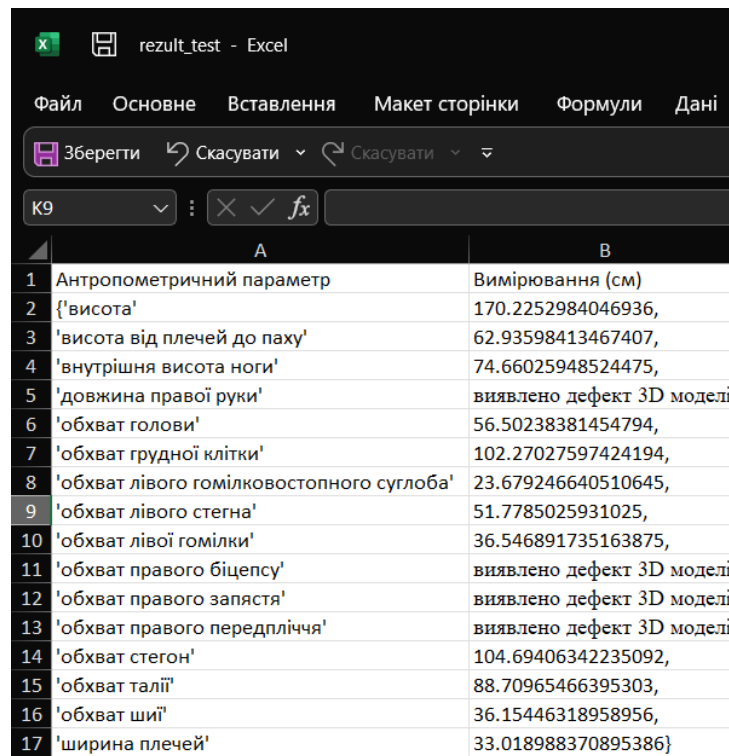


Рисунок 4.7 – Результати вимірювання 3Д тіла жінки у робочому середовищі

Під час тестування програмного забезпечення було проведено аналіз процесу експорту антропометричних параметрів у формат .xlsx. Для цього використовувалася тривимірна модель з відомими дефектами, яка піддавалася вимірюванню. Результати експорту були записані у вказаний формат для подальшого аналізу та перевірки. На рисунку 4.8 представлені результати тестування, які свідчать про те, що всі необхідні параметри відображаються коректно та зручно для користувача.

Це дозволяє ефективно використовувати отримані дані для подальшого аналізу та обробки. Крім того, у випадку виявлення дефектів у тривимірній моделі, такі параметри, які неможливо виміряти, позначаються значенням «виявлено дефект 3D моделі». Це дозволяє користувачеві чітко розрізняти дефекти та відсутність даних, що є важливим для подальшого аналізу та інтерпретації результатів вимірювань.



	А	В
1	Антропометричний параметр	Вимірювання (см)
2	'висота'	170.2252984046936,
3	'висота від плечей до паху'	62.93598413467407,
4	'внутрішня висота ноги'	74.66025948524475,
5	'довжина правої руки'	виявлено дефект 3D моделі
6	'обхват голови'	56.50238381454794,
7	'обхват грудної клітки'	102.27027597424194,
8	'обхват лівого гомілковостопного суглоба'	23.679246640510645,
9	'обхват лівого стегна'	51.7785025931025,
10	'обхват лівої гомілки'	36.546891735163875,
11	'обхват правого біцепсу'	виявлено дефект 3D моделі
12	'обхват правого зап'ястя'	виявлено дефект 3D моделі
13	'обхват правого передпліччя'	виявлено дефект 3D моделі
14	'обхват стегон'	104.69406342235092,
15	'обхват талії'	88.70965466395303,
16	'обхват ший'	36.15446318958956,
17	'ширина плечей'	33.018988370895386}

Рисунок 4.8 – Результат тестування експорту даних

Проведений аналіз часу, потрібного для автоматизованого вимірювання антропометричних даних тіла людини та їхньої візуалізації у тривимірному просторі, дозволив отримати важливі висновки щодо ефективності та швидкості роботи системи. Виявлено, що система може відповідати на взаємодію з користувачем з швидкістю, що не перевищує 0.5 секунди. Це свідчить про те, що користувачі отримують миттєвий зворотний зв'язок під час використання інтерфейсу програмного забезпечення. Такий швидкий відгук дозволяє забезпечити зручність та ефективність використання системи, а також покращує загальне враження користувача від взаємодії з програмним забезпеченням.

При проведенні автоматизованого вимірювання антропометричних даних та їхньої візуалізації у тривимірному просторі, час, необхідний для цих операцій, складає не більше ніж 5.5 секунд. Це свідчить про високу ефективність та швидкість обробки даних програмним забезпеченням під час виконання складних операцій з тривимірним моделюванням. Такий швидкий обчислювальний процес значно полегшує роботу з програмою та дозволяє оперативно отримувати результати аналізу антропометричних даних.

4.2 Розробка інструкції користувача

Розробка інструкції користувача є ключовим етапом в процесі створення програмного забезпечення для вимірювання антропометричних параметрів. Цей розділ присвячений опису процесу розробки інструкції користувача для забезпечення зручності та ефективності користування програмним забезпеченням.

Програмне забезпечення призначене для автоматизованого вимірювання антропометричних параметрів за допомогою тривимірного моделювання тіла людини. Воно дозволяє швидко та точно вимірювати різні антропометричні параметри та виявляти дефекти на поверхні тіла.

Огляд можливостей:

- імпорт 3d моделі: програма підтримує імпорт тривимірних моделей тіла людини у різних форматах, таких як .obj або .pkl;
- аналіз 3d моделі: користувач може виконувати аналіз тривимірної моделі для виявлення дефектів, таких як відступність кінцівок або неправильна форма;
- вимірювання антропометричних параметрів: програма надає можливість вимірювати різні антропометричні параметри, такі як довжина, висота, обхват та інші;
- перегляд результатів в текстовому форматі: користувач може переглядати результати вимірювань у текстовому форматі для подальшого аналізу чи зберігання;
- перегляд результатів в тривимірному просторі: програма надає можливість візуалізувати антропометричні вимірювання на тривимірній моделі тіла людини для кращого розуміння результатів;
- експорт результатів: користувач може експортувати результати вимірювань у різних форматах для подальшого використання або обробки в інших програмах.

Інструкція користувача передбачає визначення технічних вимог для

запуску програмного продукту. Деталі щодо мінімальної та рекомендованої конфігурації розміщено в таблиці 4.1.

Таблиця 4.1 – Вимоги до апаратного забезпечення

Вимоги до конфігурування	Рекомендовано	Мінімально
Процесор	Intel Core i7	Intel Core i3
Оперативна пам'ять	16 ГБ RAM	4 ГБ RAM
Вільний простір на диску	15 ГБ	4 ГБ
Відеокарта	NVIDIA GeForce GTX 1660 Ti або аналогічна	NVIDIA GeForce GTX 1050
Операційна система	Windows 10 або macOS Big Sur	Windows 7 або macOS Mojave
Монітор	Роздільна здатність не менше 1920x1080 пікселів, діагональ 21" або більше	Роздільна здатність не менше 1280x720 пікселів, діагональ 15" або більше

Програмне забезпечення має безкоштовну ліцензію та відкритий код. Процес інсталяції виконується автоматично.

Після інсталяції та запуску програмного забезпечення, користувач переходить на вітальний екран програмного застосунку та очікує на завантаження робочого середовища, після чого користувач може використовувати реалізований функціонал програмного забезпечення. Першим кроком взаємодії із програмними засобами передбачається імпортування тривимірної моделі тіла людини у форматі .pkl. Після чого запускається модуль перевірки тривимірної моделі на наявність дефектів, при наявності яких застосунок повідомить користувача. Після перевірки моделі користувач побачить повідомлення, з інформацією про неї. Далі користувач може взаємодіяти із пунктами меню «Довідка», «Про програму» або розпочати

вимірювання, натиснувши відповідну кнопку. Після чого запускається модуль визначення точок антропометрії та модуль вимірювання антропометричних параметрів на тривимірній моделі. Після завершення виконання цих модулів у робочому середовищі виведеться інформація про виміряні антропометричні параметри. Також користувач може перейти на вкладку «Візуалізувати параметри», де він може переглядати виміряні параметри на тривимірній моделі тіла людини та взаємодіяти з нею (рис. 4.9). Після завершення вимірювання користувач може почати нове, завантаживши іншу тривимірну модель, або експортувати отримані результати у форматі .xlsx.

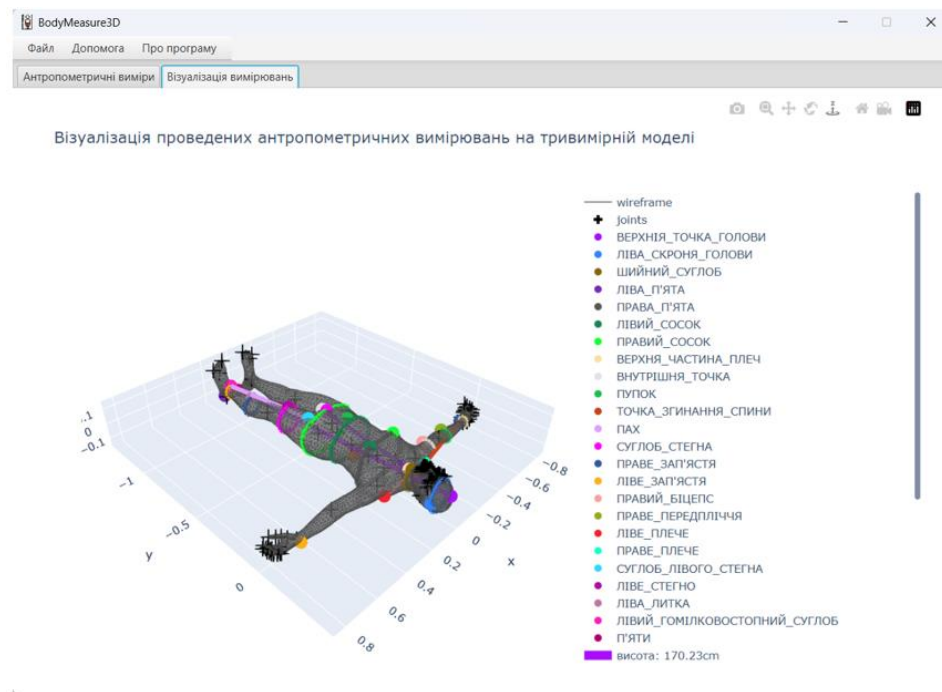


Рисунок 4.9 – Екран «Візуалізація вимірювань»

4.3 Висновки

У четвертому розділі було проведено тестування програмних модулів програмних засобів антропометричних вимірювань з використанням тривимірного моделювання. Було перевірено реакцію програмних засобів на помилкові ситуації, вхідні дані різних форматів. Також було розроблено інструкцію користувача по початковій експлуатації програмного продукту.

ВИСНОВКИ

У бакалаврській дипломній роботі було розроблено методи та програмні засоби для антропометричного вимірювання з використанням засобів тривимірного моделювання.

Проведено аналіз сучасних підходів до антропометричних вимірювань з використанням тривимірного моделювання. Обґрунтовано необхідність розробки методів виконання антропометричних вимірювань з використанням тривимірного моделювання з метою підвищення точності вимірювання антропометричних параметрів людини.

Вперше запропоновано метод визначення об'єму тіла людини з використанням тривимірного моделювання, особливість якого полягає у виконанні горизонтальних зрізів тривимірної моделі заданої товщини, виконання тріангуляції та визначення сумарного об'єму отриманих складових призм, що дозволяє підвищити точність вимірювання.

Вперше запропоновано метод визначення площі тіла людини з використанням тривимірного моделювання, особливість якого полягає у формуванні текстурної розгортки поверхні людини на горизонтальну площину з подальшим визначенням сумарної площі отриманих фігур, що дозволяє підвищити точність вимірювання.

Вперше запропоновано метод визначення антропометричних відстаней на тривимірній моделі, на основі позицій 16 основних суглобів, особливість якого полягає у використанні нової формули визначення відстані між двома точками на тривимірній моделі тіла людини, у застосуванні горизонтальної кореляційної лінії через вершини трикутників та обрахування відстані використовуючи формули векторів. Це дає можливість автоматизувати процес вимірювання.

Практичне значення отриманих результатів полягає в тому, що на основі теоретичних досліджень розроблено алгоритми та програми для антропометричних вимірювань та їх інтеграції в професійний графічний рушій `idx3d`.

Розроблено методи вимірювання антропометричної відстані на тривимірній моделі тіла людини використовуючи засоби тривимірного моделювання та розроблені формули та методи, що дозволяють з максимальною точністю визначити антропометричні параметри тривимірної моделі людини.

На основі проведених у БДР досліджень розроблено алгоритми та програмні засоби для антропометричних вимірювань.

Проведене тестування підтвердило достовірність теоритичних досліджень методів антропометричних вимірювань та засобів їх реалізації. Розроблені засоби вимірювання антропометричних параметрів можна використати в медичній галузі для визначення особливостей лікування.

Ключові слова: тривимірне моделювання, 3Д модель, антропометрія, вимірювання, програмне забезпечення, 3Д модель людини.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Романюк О. Н., Захарчук М. Д., Коробейнікова Т. І. Використання тривимірної графіки у медичній галузі // Матеріали молодіжної науково-практичної інтернет-конференції студентів аспірантів та молодих науковців «Молодь в науці: дослідження, проблеми, перспективи (МН-2020)» : збірник матеріалів. – Вінниця: ВНТУ, 2021.
2. Романюк О. Н., Захарчук М. Д., Снігур А. В., Коваль Л.В., Михайлов П. І., Чехмейструк Р. Ю. Використання тривимірного моделювання для визначення масо-вагових характеристик людини по її антропометричним параметрам/ Прикладні питання математичного моделювання т. 4, № 2.1, 2021.- С.188-199.
3. Романюк О. Н. , Захарчук М. Д., Михайлов П. М., Чехмейструк Р. Ю. Аналіз технології LSPC. Стан, досягнення та перспективи інформаційних систем і технологій / Матеріали XXI Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів. Одеса, 22-23 квітня 2021 р. - Одеса, Видавництво ОНАХТ, 2021 р. – 146-147 с.
4. Романюк О.Н., Кокункін В.Л., Захарчук М.Д., Котлик С.В. Використання морфінгу 3D- зображень облич людей в медицині. Інформаційні технології і автоматизація – 2021/ Матеріали XIV міжнародної науково-практичної Конференції . Одеса, 21-22 жовтня 2021р. -Одеса, Видавництво ОНАХТ, 2021 р. –С. 252-255.
5. Романюк О.Н., Захарчук М. Д., Михайлов П.І., Чехместрук Р.Ю., В.М., Перун І.В. Визначення генетичних захворювань людини за тривимірною моделлю лица. Електронні інформаційні ресурси: створення, використання , доступ. Збірник матеріалів Міжнародної науково-практичної Інтернет конференції, 9-10 листопада 2021 р . –Суми/Вінниця: НІКО/ВНТУ, 2021. – С. 179-184.
6. Романюк О.Н., Захарчук М.Д., Чехместрук Р.Ю, Коробейнікова Т.І, Аналіз програмного забезпечення для морфінгу зображень. Сучасна наука:

проблеми та перспективи (частина II): матеріали VI Міжнародної науково-практичної конференції м. Київ, 12-13 січня 2022 року. –Київ: МЦНіД, 2022. –С. 62-64.

7. Романюк О.Н., Бажан В.М., Захарчук М.Д. , Романюк О.В., Коробейнікова Т. І. Комп'ютерні програми для медичного діагностування за зовнішніми ознаками людини. Сучасна наука: проблеми та перспективи (частина II): матеріали VI Міжнародної науково-практичної конференції м. Київ, 12-13 січня 2022 року. – Київ: МЦНіД, 2022. – с. 64-66.

8. Романюк О.Н.,Захарчук М.Д., Чехместрук Р.Ю., Романюк О.В., Коробейнікова Т.І. Визначення векторів нормалей у довільній точці трикутника. Актуальні проблеми сучасної науки та освіти (частина I): матеріали IV Міжнародної науково-практичної конференції м. Львів, 20-21 січня 2022 року. – Львів: Львівський науковий форум, 2022. – с.64-67.

9. Романюк О. Н, Захарчук М. Д., Коваль Л. Г., Чехместрук Р. Ю., Михайлов П. І Аналіз воксельної технології. Стан, досягнення та перспективи інформаційних систем і технологій / Матеріали XXII Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів. Одеса, 21-22 квітня 2022 р. - Одеса, Видавництво ОНТУ, 2022 р. С.74-75.

10. Майданюк В. П., Захарчук М. Д. Аналіз форматів зображень / Електронні інформаційні ресурси: створення, використання, доступ. Збірник матеріалів Міжнародної науково-практичної Інтернет конференції 28-29 листопада 2022 р. – Суми/Вінниця: НІКО/ВНТУ, 2022. – С. 143-147.

11. Романюк О.Н., Захарчук М.Д., Романюк С.О., Чехместрук Р.Ю., Тітова Л.В. Діагностика генетичних захворювань за тривимірною моделлю обличчя. Інформаційні технології: наука, техніка, технологія, освіта, здоров'я: тези доповідей XXXI міжнародної науково-практичної конференції MicroCAD-2023, 17–20 травня 2023 р. / за ред. проф. Сокола Є.І. — Харків : НТУ «ХПІ». — С. 1134.

12. Романюк О.Н., Захарчук М.Д. Метод процедурного текстурування. Інноваційні дослідження та перспективи розвитку науки і техніки у XXI столітті,

Рівне, 19 жовтня 2023 р. Рівне, Редакційно-видавничий центр Приватного вищого навчального закладу «Міжнародний економіко-гуманітарний університет імені академіка Степана Дем'янчука», 2023 р. ЧЗ. С.175-178.

13. Романюк О., Чехмestрук Р., Станіславенко Є., Вінтонюк В., Захарчук М. Взаємозв'язок між векторами нормалі до поверхні, вектором спостерігача та вектором джерела світла для задач рендерингу. 8-а Міжнародна науково-практична конференція " АКТУАЛЬНІ ПИТАННЯ НАУКИ, ОСВІТИ І СУСПІЛЬСТВА В СУЧАСНИХ УМОВАХ" (26-28 лютого 2022 р.) НВЦ "Sciconf.com.ua", Харків, Україна. 2022. с. 300-303.

14. WHO Anthro Survey Analyser and other tools [Електронний ресурс] – Режим доступу до ресурсу: <https://www.who.int/tools/child-growth-standards/software> (дата звернення: 21.03.2024)

15. Anthropometric iTool [Електронний ресурс] – Режим доступу до ресурсу: <https://anthropometric-itool-ios.soft112.com/> (дата звернення: 21.03.2024)

16. ZEISS Measuring Software [Електронний ресурс] – Режим доступу до ресурсу: <https://www.zeiss.com/metrology/products/software.html> (дата звернення: 21.03.2024)

17. DINED Mannequin [Електронний ресурс] – Режим доступу до ресурсу: <https://dined.io.tudelft.nl/en/mannequin/introduction> (дата звернення: 21.03.2024)

18. Vision Measuring Software iMeasuring [Електронний ресурс] – Режим доступу до ресурсу: <https://dined.io.tudelft.nl/en/mannequin/introduction> (дата звернення: 21.03.2024)

19. Java [Електронний ресурс] – Режим доступу до ресурсу: <https://www.java.com/ua/> (дата звернення: 08.04.2024)

20. Python [Електронний ресурс] – Режим доступу до ресурсу: <https://www.python.org/> (дата звернення: 08.04.2024)

21. SMPLX [Електронний ресурс] – Режим доступу до ресурсу: <https://smpl.is.tue.mpg.de/> (дата звернення: 08.04.2024)

22. Visio [Електронний ресурс] – Режим доступу до ресурсу:

<https://www.microsoft.com/uk-ua/microsoft-365/visio/flowchart-software>
звернення: 08.04.2024)

(дата

ДОДАТКИ

Додаток А – Технічне завдання

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
д.т.н., професор
О. Н. Романюк
«12» березня 2024 р.

Технічне завдання
на бакалаврську дипломну роботу «Розробка методів і програмних
засобів антропометричних вимірювань з використанням тривимірного
моделювання» за спеціальністю 121 – Інженерія програмного забезпечення

Керівник бакалаврської дипломної роботи:
д.т.н., професор О. Н. Романюк
«12» березня 2024 р.

Виконав:
студент гр. 2ПІ-206 М. Д. Захарчук
«12» березня 2024 р.

Вінниця – 2024 року

1. Найменування та галузь застосування

Бакалаврська дипломна робота: «Розробка методів і програмних засобів антропометричних вимірювань з використанням тривимірного моделювання».

Галузь застосування – медична діагностика.

2. Підстава для розробки

Підставою для виконання бакалаврської дипломної роботи (БДР) є індивідуальне завдання на БДР та наказ №80 від «11» березня 2024 р. ректора ВНТУ про закріплення тем БДР.

3. Мета та призначення розробки

Метою дослідження є підвищення точності вимірювання антропометричних параметрів людини за рахунок використанням тривимірних моделей, що дозволяють реєструвати усі деталі геометрії тіла людини.

4. Вихідні дані для проведення НДР

Перелік основних літературних джерел, на основі яких буде виконуватись БДР.

1. Романюк О. Н., Захарчук М. Д., Снігур А. В., Коваль Л.В., Михайлов П. І., Чехмейструк Р. Ю. Використання тривимірного моделювання для визначення масо-вагових характеристик людини по її антропометричним параметрам/ Прикладні питання математичного моделювання т. 4, № 2.1, 2021.- С.188-199.

2. Information Technology in Medical Diagnostics / ed. by W. Wójcik, A. Smolarz. CRC Press, 2017.

5. Технічні вимоги

Вихідні дані до роботи: кольоровий режим, Truescolor, розмір кординат, розмір екрану 2560-1440px, результати вимірювань антропометричних параметрів, з можливістю їх візуалізації на тривимірній моделі тіла людини.

6. Конструктивні вимоги

Графічна та текстова документація повинна відповідати діючим стандартам України.

7. Перелік технічної документації, що пред'являється по закінченню робіт:

1. Пояснювальна записка до БДР.
2. Технічне завдання.
3. Лістинги програми

8. Вимоги до рівня уніфікації та стандартизації

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ

9. Стадії та етапи розробки:

№ з\п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи
1	Аналіз підходів до антропометричних вимірювань з використанням тривимірного моделювання	14.03.24 – 22.03.24
2	Розробка методів антропометричних вимірювань з використанням тривимірного моделювання	22.03.24 – 19.04.24
3	Розробка алгоритмів роботи програмного забезпечення	19.04.24 – 10.05.24
4	Розробка програмного забезпечення	10.05.24 – 24.05.24
5	Тестування програмного забезпечення	24.05.24 – 27.05.24
6	Оформлення матеріалів до захисту БДР	27.05.24 – 03.06.24

10. Порядок контролю та прийняття

Виконання етапів бакалаврської дипломної роботи контролюється керівником згідно з графіком виконання роботи. Прийняття бакалаврської дипломної роботи здійснюється ДЕК, затвердженою зав. кафедрою згідно з графіком.

Додаток Б – Протокол перевірки на плагіат

80

ПРОТОКОЛ ПЕРЕВІРКИ БАКАЛАВРСЬКОЇ ДИПЛОМНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Розробка методів і програмних засобів антропометричних
випробувань з використанням тривимірного моделювання

Тип роботи: БДР

Підрозділ: кафедра програмного забезпечення, ФІТКІ

Науковий керівник: Романюк О.Н.

Оригінальність	94,7%
Схожість	5,3%

Аналіз звіту подібності

- Запозичення, виявлені у роботі, оформленні коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цілісності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховання недобросовісних запозичень.

Особа, відповідальна за перевірку



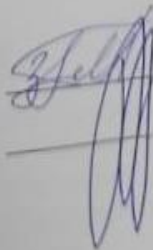
Черноволик Г.О.

Ознайомлені з повним звітом подібності, який був згенерований системою

Unichек

Автор роботи

Керівник роботи



Захарчук М.Д.

Романюк О.Н.

Додаток В – Лістинг програмного забезпечення

//управління графічного інтерфейсу застосунку

```
public class MainEnvironmentController {
    @FXML
    private TextArea anthropometric_info;
    @FXML
    private WebView webview;

    public String modelName="";

    @FXML
    public void exit() {
        System.exit(0);
    }
    @FXML
    public void import3Dmodel(){
        FileChooser fileChooser = new FileChooser();
        fileChooser.getExtensionFilters().add(new FileChooser.ExtensionFilter("PKL Files", "*.pkl"));

        File selectedFile = fileChooser.showOpenDialog(webview.getScene().getWindow());

        if (selectedFile != null) {
            modelName = selectedFile.getName();
        }
        anthropometric_info.clear();
        anthropometric_info.setText("Було обрано тривимірну модель: "+modelName);
    }

    @FXML
    public void exportRezult() {
        FileChooser fileChooser = new FileChooser();
        fileChooser.setTitle("Save Data to Excel");
        fileChooser.getExtensionFilters().add(new FileChooser.ExtensionFilter("Excel Files",
        "*.xlsx"));
        File file = fileChooser.showSaveDialog(null);

        if (file != null) {
            try (Workbook workbook = new XSSFWorkbook()) {
                Sheet sheet = workbook.createSheet("Data");

                String str=anthropometric_info.getText();
                str.replaceAll(";", "");

                String[] lines = str.split("\n");
                int rowNum = 0;
                Row headerRow = sheet.createRow(rowNum++);
                Cell headerCell1 = headerRow.createCell(0);
                Cell headerCell2 = headerRow.createCell(1);
                headerCell1.setCellValue("Антропометричний параметр");
```

```

headerCell2.setCellValue("Вимірювання (см)");
for (String line : lines) {
    String[] parts = line.split(":");
    if (parts.length == 2) {
        Row row = sheet.createRow(rowNum++);
        Cell cell1 = row.createCell(0);
        Cell cell2 = row.createCell(1);
        cell1.setCellValue(parts[0].trim());
        cell2.setCellValue(parts[1].trim());
    }
}

try (FileOutputStream outputStream = new FileOutputStream(file)) {
    workbook.write(outputStream);
    showAlert("Success", "Data exported successfully!");
}
} catch (IOException e) {
    e.printStackTrace();
    showAlert("Error", "Failed to export data: " + e.getMessage());
}
}

private void showAlert(String title, String content) {
    Alert alert = new Alert(Alert.AlertType.INFORMATION);
    alert.setTitle(title);
    alert.setHeaderText(null);
    alert.setContentText(content);
    alert.showAndWait();
}

@FXML
public void showHelp(){
    try{
        Parent root =
FXMLLoader.load(Objects.requireNonNull(getClass().getResource("help.fxml")));
        Scene scene = new Scene(root);

        Stage stage = new Stage();
        stage.setScene(scene);
        stage.setTitle("BodyMeasure3D - Help");

        stage.setFullScreen(false);
        stage.initStyle(StageStyle.UNDECORATED);
        stage.getIcons().add(new Image("D:\\Универ\\4 курс\\2
семестр\\Захарчук\\БДР\\Програма\\anthropometricMeasurements\\src\\main\\resources\\com\\ex
ample\\anthropometricmeasurements\\icon.png"));
        stage.setResizable(false);
        stage.show();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

```

@FXML
public void showAbout(){
    try{
        Parent root =
FXMLLoader.load(Objects.requireNonNull(getClass().getResource("about.fxml")));
        Scene scene = new Scene(root);

        Stage stage = new Stage();
        stage.setScene(scene);
        stage.setTitle("BodyMeasure3D - Help");

        stage.setFullScreen(false);
        stage.getIcons().add(new Image("D:\\Универ\\4 курс\\2
семестр\\Захарчук\\БДР\\Програма\\anthropometricMeasurements\\src\\main\\resources\\com\\ex
ample\\anthropometricmeasurements\\icon.png"));
        stage.initStyle(StageStyle.UNDECORATED);
        stage.setResizable(false);
        stage.show();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
@FXML
public void measure() {
    try {
        String url="http://127.0.0.1:";
        // Вказуємо повний шлях до файлу Python
        String pythonScriptPath = "";
        if (modelName.toLowerCase().contains("male"))
        {
            pythonScriptPath = "C:\\Users\\Dell\\PycharmProjects\\SMPL-
Anthropometry\\measure_male.py";
        }else if (modelName.toLowerCase().contains("female")){
            pythonScriptPath = "C:\\Users\\Dell\\PycharmProjects\\SMPL-
Anthropometry\\measure_female.py";
        }else pythonScriptPath= "C:\\Users\\Dell\\PycharmProjects\\SMPL-
Anthropometry\\measure.py";
        // Формуємо команду для запуску Python-скрипта з аргументами
        String[] command = {"cmd", "/c", "python", pythonScriptPath, "--
measure_neutral_smplx_with_mean_shape"};
        // Встановлюємо робочий каталог на папку, де знаходиться файл Python
        File workingDirectory = new File("C:\\Users\\Dell\\PycharmProjects\\SMPL-
Anthropometry");
        // Запускаємо команду Python з вказаною робочою папкою
        Process process = Runtime.getRuntime().exec(command, null, workingDirectory);
        BufferedReader reader = new BufferedReader(new
InputStreamReader(process.getInputStream(), "CP1251"));
        String line;
        anthropometric_info.clear();
        // Оновлюємо вміст Label у головному потоці JavaFX
        while ((line = reader.readLine()) != null) {
            if (!line.contains("Адреса для візуалізації:")) {

```



```

        anthropometric_info.setText(anthropometric_info.getText() + "\n" + line);
    } else {
        String s=line.substring(line.indexOf("#")+1, line.length());
        int ss=Integer.parseInt(s)+1;
        url+=ss+"/";
        System.out.println(url);
        webview.getEngine().load(url);
    }
}
int exitCode = process.waitFor();
} catch (IOException | InterruptedException e) {
    e.printStackTrace();
}
}
}
//вимірювання антропометричних параметрів

```

```

def set_shape(model, shape_coefs):
    shape_coefs = shape_coefs.to(torch.float32)
    return model(betas=shape_coefs, return_verts=True)

```

```

def create_model(model_type, model_root, gender, num_betas=10, num_thetas=24):
    return smplx.create(model_path=model_root,
        model_type=model_type,
        gender=gender,
        use_face_contour=False,
        num_betas=num_betas,
        ext='pkl')

```

```

class Measurer():

```

```

    def __init__(self):
        self.verts = None
        self.faces = None
        self.joints = None
        self.gender = None

        self.measurements = {}
        self.height_normalized_measurements = {}
        self.labeled_measurements = {}
        self.height_normalized_labeled_measurements = {}
        self.labels2names = {}

```

```

    def from_verts(self):
        pass

```

```

    def from_body_model(self):
        pass

```

```

    def measure(self,
        measurement_names: List[str]

```

```

    ):

for m_name in measurement_names:
    if m_name not in self.all_possible_measurements:
        print(f"Вимірювання {m_name} не розпізнано")
        pass

    if m_name in self.measurements:
        pass

    if self.measurement_types[m_name] == MeasurementType().LENGTH:

        value = self.measure_length(m_name)
        self.measurements[m_name] = value

    elif self.measurement_types[m_name] == MeasurementType().CIRCUMFERENCE:

        value = self.measure_circumference(m_name)
        self.measurements[m_name] = value

    else:
        print(f"Вимірювання {m_name} не розпізнано")

def measure_length(self, measurement_name: str):

    measurement_landmarks_inds = self.length_definitions[measurement_name]

    landmark_points = []
    for i in range(2):
        if isinstance(measurement_landmarks_inds[i], tuple):
            lm = (self.verts[measurement_landmarks_inds[i][0]] +
                  self.verts[measurement_landmarks_inds[i][1]]) / 2
        else:
            lm = self.verts[measurement_landmarks_inds[i]]

        landmark_points.append(lm)

    landmark_points = np.vstack(landmark_points)[None, ...]

    return self._get_dist(landmark_points)

@staticmethod
def _get_dist(verts: np.ndarray) -> float:

    verts_distances = np.linalg.norm(verts[:, 1] - verts[:, 0], axis=1)
    distance = np.sum(verts_distances)
    distance_cm = distance * 100
    return distance_cm

def measure_circumference(self,
                           measurement_name: str,
                           ):

```

```

measurement_definition = self.circumf_definitions[measurement_name]
circumf_landmarks = measurement_definition["LANDMARKS"]
circumf_landmark_indices = [self.landmarks[l_name] for l_name in circumf_landmarks]
circumf_n1, circumf_n2 = self.circumf_definitions[measurement_name]["JOINTS"]
circumf_n1, circumf_n2 = self.joint2ind[circumf_n1], self.joint2ind[circumf_n2]

plane_origin = np.mean(self.verts[circumf_landmark_indices, :], axis=0)
plane_normal = self.joints[circumf_n1, :] - self.joints[circumf_n2, :]

mesh = trimesh.Trimesh(vertices=self.verts, faces=self.faces)

slice_segments, sliced_faces = trimesh.intersections.mesh_plane(mesh,
                                                                    plane_normal=plane_normal,
                                                                    plane_origin=plane_origin,
                                                                    return_faces=True) # (N, 2, 3), (N,)

slice_segments = filter_body_part_slices(slice_segments,
                                          sliced_faces,
                                          measurement_name,
                                          self.circumf_2_bodypart,
                                          self.face_segmentation)

slice_segments_hull = convex_hull_from_3D_points(slice_segments)

return self._get_dist(slice_segments_hull)

def height_normalize_measurements(self, new_height: float):

    if self.measurements != {}:
        old_height = self.measurements["height"]
        for m_name, m_value in self.measurements.items():
            norm_value = (m_value / old_height) * new_height
            self.height_normalized_measurements[m_name] = norm_value

    if self.labeled_measurements != {}:
        for m_name, m_value in self.labeled_measurements.items():
            norm_value = (m_value / old_height) * new_height
            self.height_normalized_labeled_measurements[m_name] = norm_value

def label_measurements(self, set_measurement_labels: Dict[str, str]):

    if self.labeled_measurements != {}:
        print("Виконання нового вимірювання")

    self.labeled_measurements = {}
    self.labels2names = {}

    for set_label, set_name in set_measurement_labels.items():

        if set_name not in self.all_possible_measurements:
            print(f"Вимірювання {set_name} не розміжане.")

```

```

pass

if set_name not in self.measurements.keys():
    self.measure([set_name])

self.labeled_measurements[set_label] = self.measurements[set_name]
self.labels2names[set_label] = set_name

def visualize(self,
    measurement_names: List[str] = [],
    landmark_names: List[str] = [],
    title="Візуалізація проведених антропометричних вимірювань на тривимірній
моделі",
    visualize_body: bool = True,
    visualize_landmarks: bool = True,
    visualize_joints: bool = True,
    visualize_measurements: bool = True):

    if measurement_names == []:
        measurement_names = self.all_possible_measurements

    if landmark_names == []:
        landmark_names = list(self.landmarks.keys())

    vizz = Visualizer(verts=self.verts,
        faces=self.faces,
        joints=self.joints,
        landmarks=self.landmarks,
        measurements=self.measurements,
        measurement_types=self.measurement_types,
        length_definitions=self.length_definitions,
        circumf_definitions=self.circumf_definitions,
        joint2ind=self.joint2ind,
        circumf_2_bodypart=self.circumf_2_bodypart,
        face_segmentation=self.face_segmentation,
        visualize_body=visualize_body,
        visualize_landmarks=visualize_landmarks,
        visualize_joints=visualize_joints,
        visualize_measurements=visualize_measurements,
        title=title
    )

    vizz.visualize(measurement_names=measurement_names,
        landmark_names=landmark_names,
        title=title)

class MeasureSMPLX(Measurer):

    def __init__(self):
        super().__init__()

```

```

self.model_type = "smplx"
self.body_model_root = "data"
self.body_model_path = os.path.join(self.body_model_root,
                                     self.model_type)

self.faces = smplx.SMPLX(self.body_model_path, ext="pkl").faces
face_segmentation_path = os.path.join(self.body_model_path,
                                       f"{self.model_type}_body_parts_2_faces.json")
self.face_segmentation = load_face_segmentation(face_segmentation_path)

self.landmarks = SMPLX_LANDMARK_INDICES
self.measurement_types = MEASUREMENT_TYPES
self.length_definitions = SMPLXMeasurementDefinitions().LENGTHS
self.circumf_definitions = SMPLXMeasurementDefinitions().CIRCUMFERENCES
self.circumf_2_bodypart =
SMPLXMeasurementDefinitions().CIRCUMFERENCE_TO_BODYPARTS
self.all_possible_measurements = SMPLXMeasurementDefinitions().possible_measurements

self.joint2ind = SMPLX_JOINT2IND
self.num_joints = SMPLX_NUM_JOINTS

self.num_points = 10475

def from_verts(self,
               verts: torch.tensor):
    verts = verts.squeeze()
    error_msg = f"verts need to be of dimension ({self.num_points},3)"
    assert verts.shape == torch.Size([self.num_points, 3]), error_msg

    joint_regressor = get_joint_regressor(self.model_type,
                                          self.body_model_root,
                                          gender="NEUTRAL",
                                          num_thetas=self.num_joints)
    joints = torch.matmul(joint_regressor, verts)
    self.joints = joints.numpy()
    self.verts = verts.numpy()

def from_body_model(self,
                   gender: str,
                   shape: torch.tensor):
    model = create_model(model_type=self.model_type,
                        model_root=self.body_model_root,
                        gender=gender,
                        num_betas=10,
                        num_thetas=self.num_joints)
    model_output = set_shape(model, shape)

    self.verts = model_output.vertices.detach().cpu().numpy().squeeze()
    self.joints = model_output.joints.squeeze().detach().cpu().numpy()
    self.gender = gender

```

```

class MeasureBody():
    def __new__(cls, model_type):
        model_type = model_type.lower()
        if model_type == 'simpl':
            return MeasureSMPL()
        elif model_type == 'simplx':
            return MeasureSMPLX()
        else:
            raise NotImplementedError("Тип тривимірної моделі не розпізнано")

```

//визначення точок антропометрії

SMPLX_NUM_JOINTS = 55

SMPLX_IND2JOINT = {

```

0: 'pelvis',
1: 'left_hip',
2: 'right_hip',
3: 'spine1',
4: 'left_knee',
5: 'right_knee',
6: 'spine2',
7: 'left_ankle',
8: 'right_ankle',
9: 'spine3',
10: 'left_foot',
11: 'right_foot',
12: 'neck',
13: 'left_collar',
14: 'right_collar',
15: 'head',
16: 'left_shoulder',
17: 'right_shoulder',
18: 'left_elbow',
19: 'right_elbow',
20: 'left_wrist',
21: 'right_wrist',
22: 'jaw',
23: 'left_eye',
24: 'right_eye',
25: 'left_index1',
26: 'left_index2',
27: 'left_index3',
28: 'left_middle1',
29: 'left_middle2',
30: 'left_middle3',
31: 'left_pinky1',
32: 'left_pinky2',
33: 'left_pinky3',
34: 'left_ring1',
35: 'left_ring2',
36: 'left_ring3',

```

```

37: 'left_thumb1',
38: 'left_thumb2',
39: 'left_thumb3',
40: 'right_index1',
41: 'right_index2',
42: 'right_index3',
43: 'right_middle1',
44: 'right_middle2',
45: 'right_middle3',
46: 'right_pinky1',
47: 'right_pinky2',
48: 'right_pinky3',
49: 'right_ring1',
50: 'right_ring2',
51: 'right_ring3',
52: 'right_thumb1',
53: 'right_thumb2',
54: 'right_thumb3'
}

SMPLX_JOINT2IND = {name:ind for ind,name in SMPLX_IND2JOINT.items()}

def get_joint_regressor(body_model_type, body_model_root, gender="MALE", num_thetas=24):

    model = smplx.create(model_path=body_model_root,
                          model_type=body_model_type,
                          gender=gender,
                          use_face_contour=False,
                          num_betas=10,
                          body_pose=torch.zeros((1, num_thetas-1 * 3)),
                          ext='pkl')
    return model.J_regressor

SMPLX_LANDMARK_INDICES = {"ВЕРХНЯ_ТОЧКА_ГОЛОВИ": 8976,
                          "ЛІВА_СКРОНЯ_ГОЛОВИ": 1980,
                          "ШИЙНИЙ_СУГЛОБ": 8940,
                          "ЛІВА_П'ЯТА": 8847,
                          "ПРАВА_П'ЯТА": 8635,
                          "ЛІВИЙ_СОСОК": 3572,
                          "ПРАВИЙ_СОСОК": 8340,

                          "ВЕРХНЯ_ЧАСТИНА_ПЛЕЧ": 5616,
                          "ВНУТРІШНЯ_ТОЧКА": 5601,
                          "ПУПОК": 5939,
                          "ТОЧКА_ЗГИНАННЯ_СПИНИ": 5941,
                          "ПАХ": 3797,
                          "СУГЛОБ_СТЕГНА": 5949,
                          "ПРАВЕ_ЗАП'ЯСТЯ": 7449,
                          "ЛІВЕ_ЗАП'ЯСТЯ": 4823,
                          "ПРАВИЙ_БЦЕПС": 6788,
                          "ПРАВЕ_ПЕРЕДПЛІЧЧЯ": 7266,
                          "ЛІВЕ_ПЛЕЧЕ": 4442,

```

```

    "ПРАВЕ_ПЛЕЧЕ": 7218,
    "СУГЛОБ_ЛІВОГО_СТЕГНА": 4112,
    "ЛІВЕ_СТЕГНО": 3577,
    "ЛІВА_ЛИТКА": 3732,
    "ЛІВИЙ_ГОМІЛКОВОСТОПНИЙ_СУГЛОБ": 5880
}

```

```

SMPLX_LANDMARK_INDICES["П'ЯТИ"] =
(SMPLX_LANDMARK_INDICES["ЛІВА_П'ЯТА"],
 SMPLX_LANDMARK_INDICES["ПРАВА_П'ЯТА"])

```

```

STANDARD_LABELS = {
    'A': 'обхват голови',
    'B': 'обхват шиї',
    'C': 'висота від плечей до паху',
    'D': 'обхват грудної клітки',
    'E': 'обхват талії',
    'F': 'обхват стегон',
    'G': 'обхват правого зап'ястя',
    'H': 'обхват правого біцепсу',
    'I': 'обхват правого передпліччя',
    'J': 'довжина правої руки',
    'K': 'внутрішня висота ноги',
    'L': 'обхват лівого стегна',
    'M': 'обхват лівої гомілки',
    'N': 'обхват лівого гомілковостопного суглоба',
    'O': 'ширина плечей',
    'P': 'висота'
}

```

```

class MeasurementType():
    CIRCUMFERENCE = "circumference"
    LENGTH = "length"

```

```

MEASUREMENT_TYPES = {
    "висота": MeasurementType.LENGTH,
    "обхват голови": MeasurementType.CIRCUMFERENCE,
    "обхват шиї": MeasurementType.CIRCUMFERENCE,
    "висота від плечей до паху": MeasurementType.LENGTH,
    "обхват грудної клітки": MeasurementType.CIRCUMFERENCE,
    "обхват талії": MeasurementType.CIRCUMFERENCE,
    "обхват стегон": MeasurementType.CIRCUMFERENCE,

    "обхват правого зап'ястя": MeasurementType.CIRCUMFERENCE,
    "обхват правого біцепсу": MeasurementType.CIRCUMFERENCE,
    "обхват правого передпліччя": MeasurementType.CIRCUMFERENCE,
    "довжина правої руки": MeasurementType.LENGTH,
    "внутрішня висота ноги": MeasurementType.LENGTH,
    "обхват лівого стегна": MeasurementType.CIRCUMFERENCE,
    "обхват лівої гомілки": MeasurementType.CIRCUMFERENCE,
}

```



```

    "обхват лівого гомілковостопного суглоба": MeasurementType.CIRCUMFERENCE,
    "ширина плечей": MeasurementType.LENGTH,
}

```

```
class SMPLXMeasurementDefinitions():
```

```
possible_measurements = MEASUREMENT_TYPES.keys()
```

```
LENGTHS = { "висота":
    (SMPLX_LANDMARK_INDICES["ВЕРХНЯ_ТОЧКА_ГОЛОВИ"],
    SMPLX_LANDMARK_INDICES["П'ЯТИ"]
    ),
    "висота від плечей до паху":
    (SMPLX_LANDMARK_INDICES["ВЕРХНЯ_ЧАСТИНА_ПЛЕЧ"],
    SMPLX_LANDMARK_INDICES["ВНУТРІШНЯ_ТОЧКА"]
    ),
    "довжина лівої руки":
    (SMPLX_LANDMARK_INDICES["ЛІВЕ_ПЛЕЧЕ"],
    SMPLX_LANDMARK_INDICES["ЛІВЕ_ЗАП'ЯСТЯ"]
    ),
    "довжина правої руки":
    (SMPLX_LANDMARK_INDICES["ПРАВЕ_ПЛЕЧЕ"],
    SMPLX_LANDMARK_INDICES["ПРАВЕ_ЗАП'ЯСТЯ"]
    ),
    "внутрішня висота ноги":
    (SMPLX_LANDMARK_INDICES["СУГЛОБ_ЛІВОГО_СТЕГНА"],
    SMPLX_LANDMARK_INDICES["ЛІВИЙ_ГОМІЛКОВОСТОПНИЙ_СУГЛОБ"]
    ),
    "ширина плечей":
    (SMPLX_LANDMARK_INDICES["ЛІВЕ_ПЛЕЧЕ"],
    SMPLX_LANDMARK_INDICES["ПРАВЕ_ПЛЕЧЕ"]
    ),
}
```

defined with landmarks and joints

landmarks are defined with indices of the smpl model points

normals are defined with joint names of the smpl model

```
CIRCUMFERENCES = {
    "обхват голови": {"LANDMARKS": ["ЛІВА_СКРОНЯ_ГОЛОВИ"],
                       "JOINTS": ["pelvis", "spine3"]},

    "обхват шиї": {"LANDMARKS": ["ШИЙНИЙ_СУГЛОБ"],
                    "JOINTS": ["spine1", "spine3"]},

    "обхват грудної клітки": {"LANDMARKS": ["ЛІВИЙ_СОСОК", "ПРАВИЙ_СОСОК"],
                               "JOINTS": ["pelvis", "spine3"]},

    "обхват талії": {"LANDMARKS": ["ПУПОК", "ТОЧКА_ЗГИНАННЯ_СПИНИ"],
                     "JOINTS": ["pelvis", "spine3"]},

    "обхват стегон": {"LANDMARKS": ["СУГЛОБ_СТЕГНА"],
                      "JOINTS": ["pelvis", "spine3"]},
```

```

"обхват правого зап'ястя":{"LANDMARKS":["ПРАВЕ_ЗАП'ЯСТЯ"],
                             "JOINTS":["right_wrist","right_elbow"]}, # different from SMPL

"обхват правого біцепсу":{"LANDMARKS":["ПРАВІЙ_БІЦЕПС"],
                             "JOINTS":["right_shoulder","right_elbow"]},

"обхват правого передпліччя":{"LANDMARKS":["ПРАВЕ_ПЕРЕДПЛІЧЧЯ"],
                             "JOINTS":["right_elbow","right_wrist"]},

"обхват лівого стегна":{"LANDMARKS":["ЛІВЕ_СТЕГНО"],
                             "JOINTS":["pelvis","spine3"]},

"обхват лівої гомілки":{"LANDMARKS":["ЛІВА_ЛИТКА"],
                             "JOINTS":["pelvis","spine3"]},

"обхват лівого гомілковостопного
суглоба":{"LANDMARKS":["ЛІВИЙ_ГОМІЛКОВОСТОПНИЙ_СУГЛОБ"],
                             "JOINTS":["pelvis","spine3"]},
    }
CIRCUMFERENCE_TO_BODYPARTS = {
    "обхват голови": "head",
    "обхват шиї": "neck",
    "обхват грудної клітки":["spine1","spine2"],
    "обхват талії":["hips","spine"],
    "обхват стегон":"hips",
    "обхват правого зап'ястя":["rightHand","rightForeArm"],
    "обхват правого біцепсу":"rightArm",
    "обхват правого передпліччя":"rightForeArm",
    "обхват лівого стегна": "leftUpLeg",
    "обхват лівої гомілки": "leftLeg",
    "обхват лівого гомілковостопного суглоба": "leftLeg",
}

class Visualizer():
    def __init__(self,
        verts: np.ndarray,
        faces: np.ndarray,
        joints: np.ndarray,
        landmarks: dict,
        measurements: dict,
        measurement_types: dict,
        length_definitions: dict,
        circumf_definitions: dict,
        joint2ind: dict,
        circumf_2_bodypart: dict,
        face_segmentation: dict,
        visualize_body: bool = True,
        visualize_landmarks: bool = True,
        visualize_joints: bool = True,
        visualize_measurements: bool=True,
        title: str = "Візуалізація проведених антропометричних вмірювань на тривимірній

```

моделі"

):

```

self.verts = verts
self.faces = faces
self.joints = joints
self.landmarks = landmarks
self.measurements = measurements
self.measurement_types = measurement_types
self.length_definitions = length_definitions
self.circumf_definitions = circumf_definitions
self.joint2ind = joint2ind
self.circumf_2_bodypart = circumf_2_bodypart
self.face_segmentation = face_segmentation

self.visualize_body = visualize_body
self.visualize_landmarks = visualize_landmarks
self.visualize_joints = visualize_joints
self.visualize_measurements = visualize_measurements

self.title = title

```

@staticmethod

def create_mesh_plot(verts: np.ndarray, faces: np.ndarray):

```

    mesh_plot = go.Mesh3d(
        x=verts[:,0],
        y=verts[:,1],
        z=verts[:,2],
        color="gray",
        hovertemplate = '<i>Index</i>: %{text}',
        text = [i for i in range(verts.shape[0])],

        i=faces[:,0],
        j=faces[:,1],
        k=faces[:,2],
        opacity=0.6,
        name='body',
    )
    return mesh_plot

```

@staticmethod

def create_joint_plot(joints: np.ndarray):

```

    return go.Scatter3d(x = joints[:,0],
        y = joints[:,1],
        z = joints[:,2],
        mode='markers',
        marker=dict(size=8,
            color="black",
            opacity=1,
            symbol="cross"
        ),

```

```

        name="joints"
    )

@staticmethod
def create_wireframe_plot(verts: np.ndarray, faces: np.ndarray):

    i=faces[:,0]
    j=faces[:,1]
    k=faces[:,2]

    triangles = np.vstack((i,j,k)).T

    x=verts[:,0]
    y=verts[:,1]
    z=verts[:,2]

    vertices = np.vstack((x,y,z)).T
    tri_points = vertices[triangles]

    Xe = []
    Ye = []
    Ze = []
    for T in tri_points:
        Xe.extend([T[k%3][0] for k in range(4)]+[ None])
        Ye.extend([T[k%3][1] for k in range(4)]+[ None])
        Ze.extend([T[k%3][2] for k in range(4)]+[ None])

    wireframe = go.Scatter3d(
        x=Xe,
        y=Ye,
        z=Ze,
        mode='lines',
        name='wireframe',
        line=dict(color= 'rgb(70,70,70)', width=1)
    )
    return wireframe
def create_landmarks_plot(self,
    landmark_names: List[str],
    verts: np.ndarray
    ) -> List[plotly.graph_objs.Scatter3d]:
    plots = []

    landmark_colors = dict(zip(self.landmarks.keys(),
        px.colors.qualitative.Alphabet))

    for lm_name in landmark_names:
        if lm_name not in self.landmarks.keys():
            print(f"Landmark {lm_name} is not defined.")
            pass

        lm_index = self.landmarks[lm_name]
        if isinstance(lm_index,tuple):

```

```

        lm = (verts[lm_index[0]] + verts[lm_index[1]]) / 2
    else:
        lm = verts[lm_index]

    plot = go.Scatter3d(x = [lm[0]],
                        y = [lm[1]],
                        z = [lm[2]],
                        mode='markers',
                        marker=dict(size=8,
                                    color=landmark_colors[lm_name],
                                    opacity=1,
                                    ),
                        name=lm_name
                    )
    plots.append(plot)

return plots

def create_measurement_length_plot(self,
                                   measurement_name: str,
                                   verts: np.ndarray,
                                   color: str
                                   ):
    measurement_landmarks_inds = self.length_definitions[measurement_name]

    segments = {"x": [], "y": [], "z": []}
    for i in range(2):
        if isinstance(measurement_landmarks_inds[i], tuple):
            lm_tnp = (verts[measurement_landmarks_inds[i][0]] +
                      verts[measurement_landmarks_inds[i][1]]) / 2
        else:
            lm_tnp = verts[measurement_landmarks_inds[i]]
        segments["x"].append(lm_tnp[0])
        segments["y"].append(lm_tnp[1])
        segments["z"].append(lm_tnp[2])
    for ax in ["x", "y", "z"]:
        segments[ax].append(None)

    if measurement_name in self.measurements:
        m_viz_name = f"{measurement_name}: {self.measurements[measurement_name]:.2f} cm"
    else:
        m_viz_name = measurement_name

    return go.Scatter3d(x=segments["x"],
                        y=segments["y"],
                        z=segments["z"],
                        marker=dict(
                            size=4,
                            color="rgba(0,0,0,0)",
                        ),
                        line=dict(
                            color=color,

```

```

        width=10),
        name=m_viz_name
    )
def create_measurement_circumference_plot(self,
        measurement_name: str,
        verts: np.ndarray,
        faces: np.ndarray,
        color: str):

    circumf_landmarks = self.circumf_definitions[measurement_name]["LANDMARKS"]
    circumf_landmark_indices = [self.landmarks[l_name] for l_name in circumf_landmarks]
    circumf_n1, circumf_n2 = self.circumf_definitions[measurement_name]["JOINTS"]
    circumf_n1, circumf_n2 = self.joint2ind[circumf_n1], self.joint2ind[circumf_n2]

    plane_origin = np.mean(verts[circumf_landmark_indices,:],axis=0)
    plane_normal = self.joints[circumf_n1,:] - self.joints[circumf_n2,:]

    mesh = trimesh.Trimesh(vertices=verts, faces=faces)

    slice_segments, sliced_faces = trimesh.intersections.mesh_plane(mesh,
        plane_normal=plane_normal,
        plane_origin=plane_origin,
        return_faces=True)

    slice_segments = filter_body_part_slices(slice_segments,
        sliced_faces,
        measurement_name,
        self.circumf_2_bodypart,
        self.face_segmentation)

    slice_segments_hull = convex_hull_from_3D_points(slice_segments)

    draw_segments = {"x":[], "y":[], "z":[]}
    map_ax = {0:"x", 1:"y", 2:"z"}

    for i in range(slice_segments_hull.shape[0]):
        for j in range(3):
            draw_segments[map_ax[j]].append(slice_segments_hull[i,0,j])
            draw_segments[map_ax[j]].append(slice_segments_hull[i,1,j])
            draw_segments[map_ax[j]].append(None)

    if measurement_name in self.measurements:
        m_viz_name = f"{measurement_name}: {self.measurements[measurement_name]:.2f} cm"
    else:
        m_viz_name = measurement_name

    return go.Scatter3d(
        x=draw_segments["x"],
        y=draw_segments["y"],
        z=draw_segments["z"],
        mode="lines",
        line=dict(

```

```

        color=color,
        width=10),
        name=m_viz_name
    )

def visualize(self,
    measurement_names: List[str] = [],
    landmark_names: List[str] = [],
    title="Візуалізація проведених антропометричних вимірювань на тривимірній
моделі"
):
    fig = go.Figure()

    if self.visualize_body:
        mesh_plot = self.create_mesh_plot(self.verts, self.faces)
        fig.add_trace(mesh_plot)

        wireframe_plot = self.create_wireframe_plot(self.verts, self.faces)
        fig.add_trace(wireframe_plot)

    if self.visualize_joints:
        joint_plot = self.create_joint_plot(self.joints)
        fig.add_trace(joint_plot)

    if self.visualize_landmarks:
        landmarks_plot = self.create_landmarks_plot(landmark_names, self.verts)
        fig.add_traces(landmarks_plot)

    measurement_colors = dict(zip(self.measurement_types.keys(),
        px.colors.qualitative.Alphabet))

    if self.visualize_measurements:
        for m_name in measurement_names:
            if m_name not in self.measurement_types.keys():
                pass

            if self.measurement_types[m_name] == MeasurementType().LENGTH:
                measurement_plot =
self.create_measurement_length_plot(measurement_name=m_name,
                                    verts=self.verts,
                                    color=measurement_colors[m_name])
            elif self.measurement_types[m_name] == MeasurementType().CIRCUMFERENCE:
                measurement_plot =
self.create_measurement_circumference_plot(measurement_name=m_name,
                                            verts=self.verts,
                                            faces=self.faces,
                                            color=measurement_colors[m_name])

            fig.add_trace(measurement_plot)

    fig.update_layout(scene_aspectmode='data',
        width=1000, height=700,
        title=title,

```

[illegible]


```

if visualize_body:
    plot_body = go.Mesh3d(
        x=smplx_vertices[:,0],
        y=smplx_vertices[:,1],
        z=smplx_vertices[:,2],
        color = "red",
        i=smplx_faces[:,0],
        j=smplx_faces[:,1],
        k=smplx_faces[:,2],
        name='smplx mesh',
        showscale=True,
        opacity=0.5
    )
    fig.add_trace(plot_body)

fig.update_layout(scene_aspectmode='data',
                    width=1000, height=700,
                    title=title,
                    )
    if show:
        fig.show()
    else:
        return fig

def viz_smpl_face_segmentation(fig=None, show=True, title="SMPL face segmentation"):
    body = smplx.SMPL("data/smpl",ext="pkl")

    with open("data/smpl/smpl_body_parts_2_faces.json","r") as f:
        face_segmentation = json.load(f)

    faces = body.faces
    verts = body.v_template

    colors = px.colors.qualitative.Alphabet + \
        px.colors.qualitative.Dark24
    mapping_bp2ind = dict(zip(face_segmentation.keys(),
                               range(len(face_segmentation.keys()))
                               ))
    face_colors = [0]*faces.shape[0]
    for bp_name,bp_indices in face_segmentation.items():
        bp_label = mapping_bp2ind[bp_name]
        for i in bp_indices:
            face_colors[i] = colors[bp_label]

    if isinstance(fig,type(None)):
        fig = go.Figure()

    fig = viz_face_segmentation(verts,faces,face_colors,title=title,name="smpl",show=False)

    if show:
        fig.show()

```

```

else:
    return fig

def viz_smplx_face_segmentation(fig=None,show=True):

    body = smplx.SMPLX("data/smplx",ext="pkl")

    with open("data/smplx/smplx_body_parts_2_faces.json","r") as f:
        face_segmentation = json.load(f)

    faces = body.faces
    verts = body.v_template

    colors = px.colors.qualitative.Alphabet + \
        px.colors.qualitative.Dark24
    mapping_bp2ind = dict(zip(face_segmentation.keys(),
                               range(len(face_segmentation.keys()))
                               ))
    face_colors = [0]*faces.shape[0]
    for bp_name,bp_indices in face_segmentation.items():
        bp_label = mapping_bp2ind[bp_name]
        for i in bp_indices:
            face_colors[i] = colors[bp_label]

    if isinstance(fig,type(None)):
        fig = go.Figure()

    fig = viz_face_segmentation(verts,faces,face_colors,title=title,name="smpl",show=False)
    if show:
        fig.show()
    else:
        return fig

def viz_smplx_point_segmentation(fig=None,show=True,title="SMPLX point segmentation"):
    model_path = "data/smplx"
    smpl_verts = smplx.SMPLX(model_path,ext="pkl").v_template
    with open("data/smplx/point_segmentation_meshcapade.json","r") as f:
        point_seg = json.load(f)
    fig = viz_point_segmentation(smpl_verts,point_seg,title=title,fig=fig,show=show)

    if show:
        fig.show()
    else:
        return fig

def viz_landmarks(verts,landmark_dict,title="Visualize
landmarks",fig=None,show=True,name="points"):

    if isinstance(fig,type(None)):
        fig = go.Figure()

    plot = go.Scatter3d(x = verts[:,0],
                        y = verts[:,1],

```

```

        z = verts[:,2],
        mode='markers',
        hovertemplate = '<i>Index</i>: %{text}',
        text = [i for i in range(verts.shape[0])],
        marker=dict(size=5,
                    color="black",
                    opacity=0.2,
                    ),
        name=name
    )
    fig.add_trace(plot)
    colors = px.colors.qualitative.Alphabet + \
        px.colors.qualitative.Dark24 + \
        px.colors.qualitative.Alphabet + \
        px.colors.qualitative.Dark24
    for i, (lm_name, lm_ind) in enumerate(landmark_dict.items()):
        plot = go.Scatter3d(x = [verts[lm_ind,0]],
                            y = [verts[lm_ind,1]],
                            z = [verts[lm_ind,2]],
                            mode='markers',
                            marker=dict(size=10,
                                        color=colors[i],
                                        opacity=1,
                                        symbol="cross"
                                        ),
                            name=name+"-"+lm_name
                            )
        fig.add_trace(plot)
    fig.update_layout(scene_aspectmode='data',
                      width=1000, height=700,
                      title=title)
    if show:
        fig.show()
    else:
        return fig

def viz_smplx_landmarks(fig=None,show=True,title="SMPLX landmarks"):

    verts = smplx.SMPLX("data/smplx",ext="pkl").v_template
    landmark_dict = SMPLX_LANDMARK_INDICES
    if isinstance(fig,type(None)):
        fig=go.Figure()
    fig = viz_landmarks(verts,
                        landmark_dict,
                        title="Visualize landmarks",
                        fig=fig,
                        show=show,
                        name="smplx")
    if show:
        fig.show()
    else:
        return fig

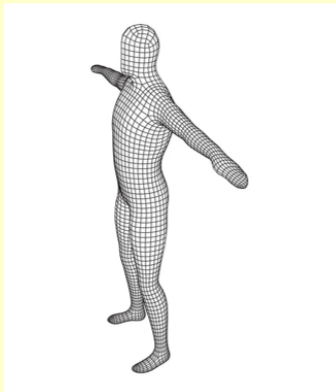
```

Додаток Г – Графічна частина

ГРАФІЧНА ЧАСТИНА

**РОЗРОБКА МЕТОДІВ І ПРОГРАМНИХ ЗАСОБІВ
АНТРОПОМЕТРИЧНИХ ВИМІРЮВАНЬ З ВИКОРИСТАННЯМ
ТРИВИМІРНОГО МОДЕЛЮВАННЯ**

Вінницький Національний Технічний Університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення



БАКАЛАВРСЬКА ДИПЛОМНА РОБОТА

на тему:

«Розробка методів і програмних засобів
антропометричних вимірювань з використанням
тривимірного моделювання»

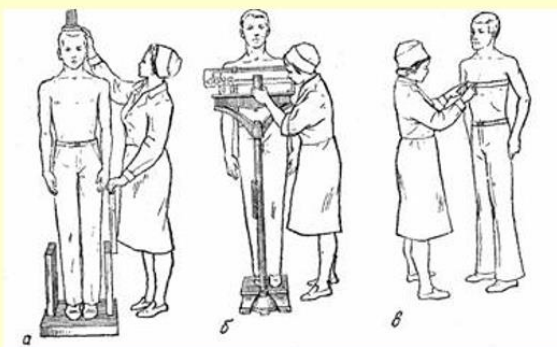
Виконав:
ст. групи 2ПІ-206
Захарчук М.Д.

Науковий керівник:
д.т.н., професор каф. ПЗ
Романюк О.Н.

1

Рисунок Г.1 – Титульний аркуш

Антропометричні вимірювання



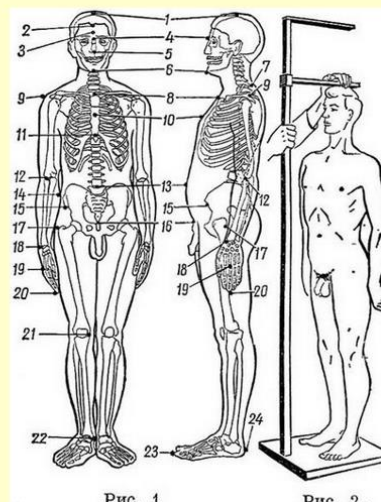
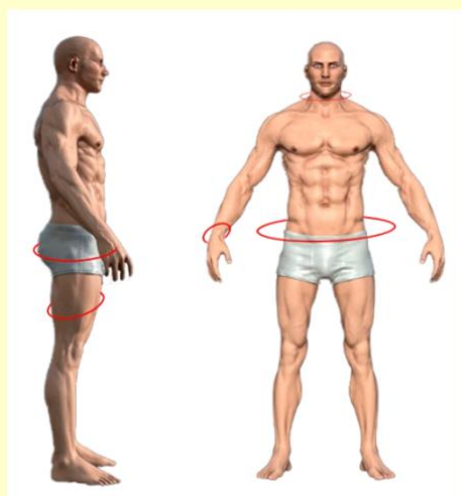
До основних антропометричних показників входять:

- | | |
|--|--------------------|
| – обхват біцепсу; | – зріст людини; |
| – обхват грудної клітки; | – обхват голови; |
| – довжина рук; | – обхват талії; |
| – обхват гомілки; | – обхват стегна; |
| – відстань від потилиці до тазового суглобу; | – обхват шиї; |
| – відстань від таза до колінного суглобу. | – обхват зап'ястя; |

2

Рисунок Г.2 – Антропометричні вимірювання

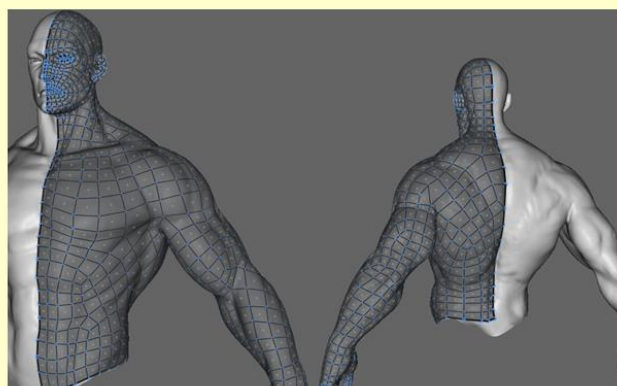
Галузі застосування антропометрії



3

Рисунок Г.3 – Галузі застосування антропометрії

Використання тривимірного моделювання в антропометрії



4

Рисунок Г.4 – Використання тривимірного моделювання в антропометрії

Роль ваги та площі людини в медичній практиці



Роль маси тіла

1. Дозування: вага є важливим фактором при визначенні дози багатьох ліків. Для дітей і дорослих із надмірною або недостатньою вагою дозування може бути скориговано для забезпечення ефективності та мінімізації побічних ефектів.
2. Оцінка стану здоров'я: вага в поєднанні зі зростом (індекс маси тіла - IMT) може вказувати на наявність або ризик ожиріння, недоїдання, анорексії та інших захворювань.
3. Моніторинг здоров'я: регулярний контроль ваги використовується для спостереження за станом здоров'я пацієнта, відстеження прогресу лікування або ефективності дієти.



Роль площі поверхні тіла

1. Дозування хіміотерапії: площа поверхні тіла використовується для розрахунку хіміотерапевтичних препаратів, оскільки це допомагає врахувати метаболічний вплив на організм пацієнта.
2. Фізіологічні та метаболічні оцінки: РРТ може бути корисним для оцінки основної швидкості метаболізму (BMR) та інших метаболічних процесів, дозволяючи точніше визначати потреби в калоріях і харчуванні.
3. Клінічні дослідження: РРТ використовується в клінічних дослідженнях для нормалізації біологічних параметрів, таких як кліренс препарату, до розміру тіла.

5

Рисунок Г.5 – Роль ваги та площі людини в медичній практиці

Розробка методів і програмних засобів антропометричних вимірювань з використанням тривимірного моделювання

- **Мета та завдання дослідження.** Метою дослідження є підвищення точності вимірювання антропометричних параметрів людини за рахунок використанням тривимірних моделей, що дозволяють реєструвати усі деталі геометрії тіла людини.
- **Об'єкт дослідження** – процес вимірювання антропометричних параметрів з використанням тривимірного моделювання.
- **Предмет дослідження** – методи та засоби вимірювання антропометричних параметрів з використанням тривимірного моделювання.
- **Методи дослідження.** У процесі дослідження застосовувалися: теорія антропометричних вимірювань; теорія алгоритмів для обробки та аналізу даних із тривимірних моделей та методи тривимірного моделювання для вимірювання антропометричних відстаней та визначення позицій точок антропометрії, комп'ютерне моделювання для аналізу та перевірки отриманих теоретичних положень.

6

Рисунок Г.6 – Мета дослідження



Рисунок Г.7 – Основні етапи підготовки тривимірної моделі

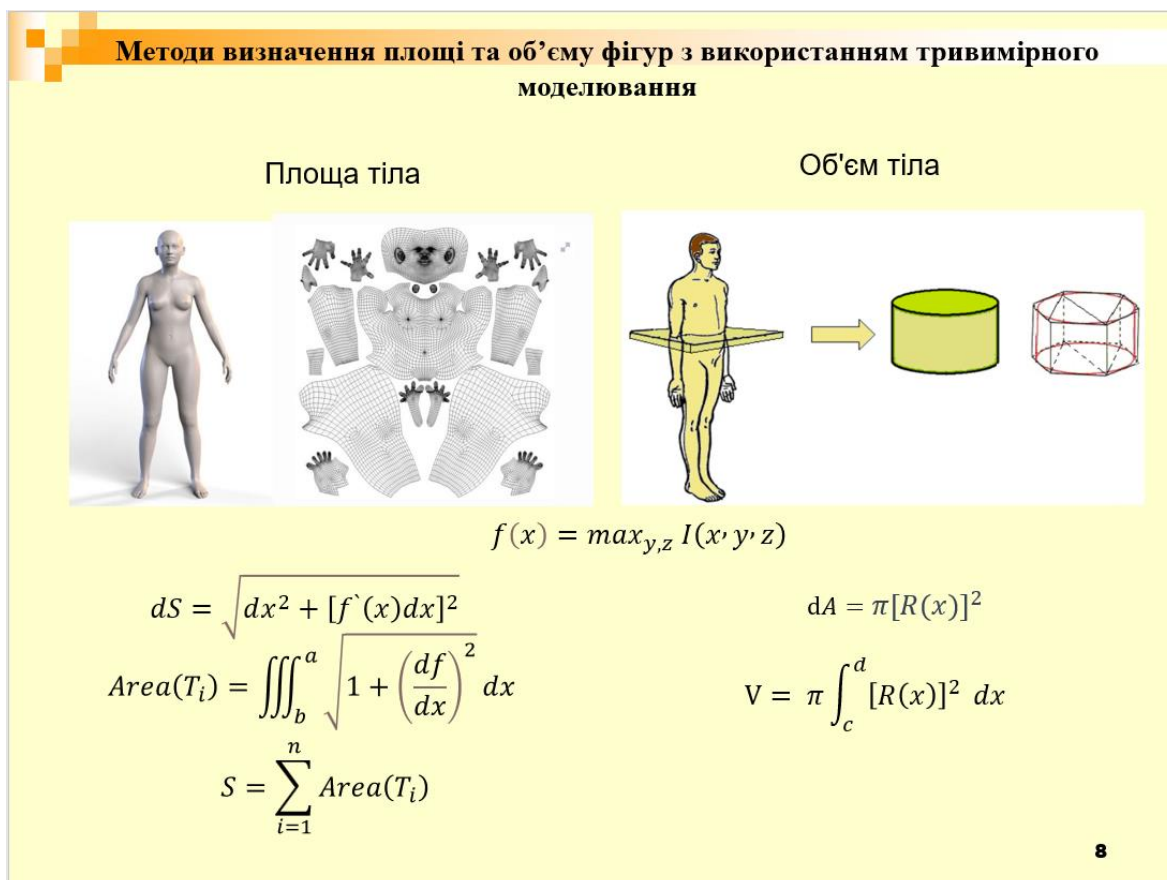


Рисунок Г.8 – Методи визначення площі та об'єму фігур з використанням тривимірного моделювання

Метод визначення точок антропометрії на основі даних тривимірної моделі

Для апроксимації точок антропометрії на поверхні тривимірної моделі використовується метод найближчих точок. Нехай P_i – антропометрична точка, яку необхідно визначити, Q_j – точка на поверхні тривимірної моделі. Тоді відстань між P_i та Q_j обчислюється за допомогою евклідової відстані:

$$d(P_i, Q_j) = \sqrt{(x_{P_i} - x_{Q_j})^2 + (y_{P_i} - y_{Q_j})^2 + (z_{P_i} - z_{Q_j})^2}$$

Наступним кроком, виразимо координати P_i – точки антропометрії, при умові, що Q_j – це відома точка суглобу на тривимірній моделі, а $d(P_i, Q_j)$ – відома відстань, отримана з міжнародних стандартів антропометрії. Виведемо формулу, що буде знаходити координати точки P_i . Для цього використаємо формули для визначення трьох координат $(x_{P_i}, y_{P_i}, z_{P_i})$.

$$\begin{aligned} x_{P_i} &= x_{Q_j} + \sqrt{d(P_i, Q_j)^2 - (y_{Q_j})^2 - (z_{Q_j})^2} \\ y_{P_i} &= x_{Q_j} + \sqrt{d(P_i, Q_j)^2 - (x_{Q_j})^2 - (z_{Q_j})^2} \\ z_{P_i} &= x_{Q_j} + \sqrt{d(P_i, Q_j)^2 - (x_{Q_j})^2 - (y_{Q_j})^2} \end{aligned}$$

9

Рисунок Г.9 – Метод визначення точок антропометрії на основі даних тривимірної моделі

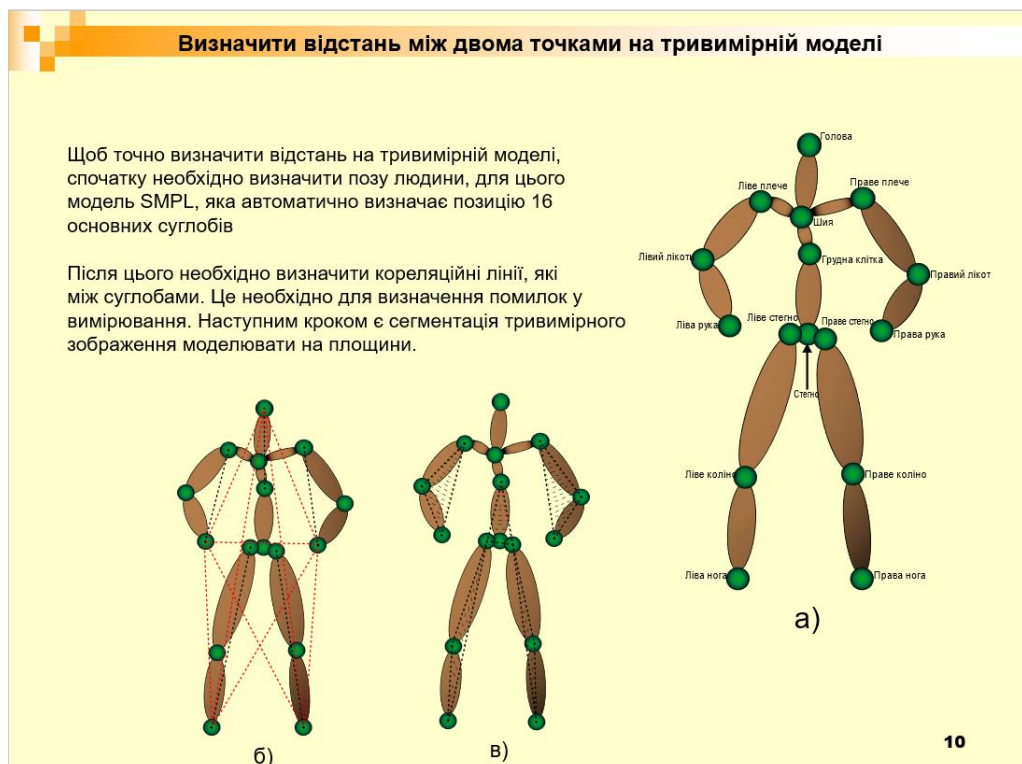
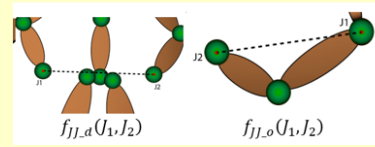


Рисунок Г.10 – Визначення відстані між двома точками на тривимірній моделі

Визначити відстань між двома точками на тривимірній моделі

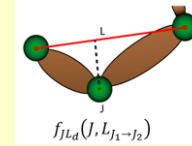
Для розрахунку відстані між стиками ми використовуємо формулу евклідової відстані

$$f_{JJ,d}(J_1, J_2) = \left| \overline{J_1 J_2} \right| \quad f_{JJ,o}(J_1, J_2) = \text{unit}(\overline{J_1 J_2})$$



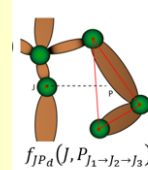
Розрахувати відстань між лінією стику $f_{JJ,d}(J_1, J_2)$ та окремого суглоба J використовується розширена формула Гелена:

$$f_{JL,d}(J, L_{J_1 \rightarrow J_2}) = 2 * \frac{S_{\Delta J J_1 J_2}}{f_{JJ,d}(J_1, J_2)},$$



Для розрахунку відстані між стиком J і площиною $P_{J_1 \rightarrow J_2 \rightarrow J_3}$ скористайтесь формулою перехресного добутку двох тривимірних векторів:

$$f_{JP,d}(J, P_{J_1 \rightarrow J_2 \rightarrow J_3}) = f_{JJ,o}(J_1, J_2) * \text{unit}(f_{JJ,o}(J_1, J_2) * f_{JJ,o}(J_1, J_3))$$



Відстань між двома точками на 3D-моделі:

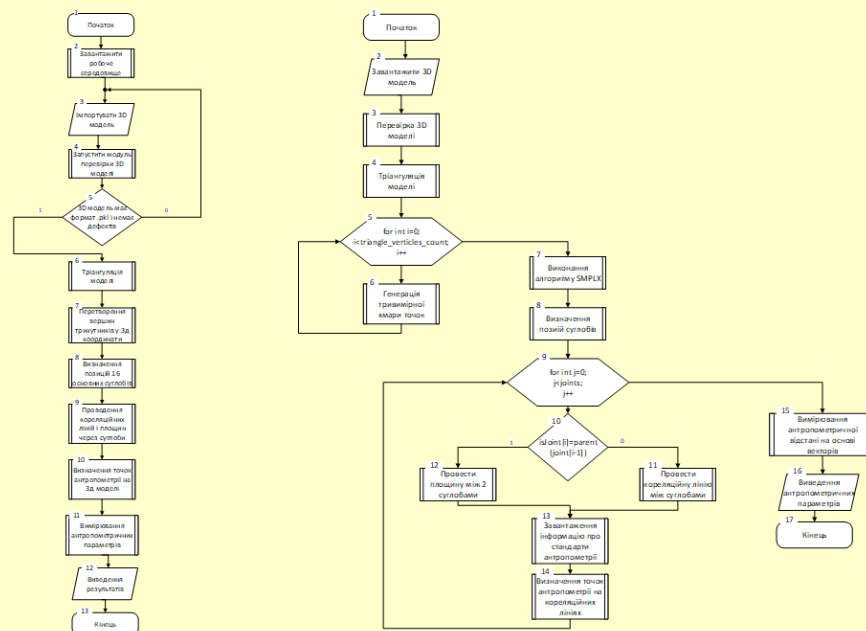
$$\sum_{i,j} |d_{ij} - \dot{d}_{ij}| = \sum_{i,j} (x_i - x_j)^T (x_i - x_j) - T * r * l$$

$$, ded_{ij} = \sum_{k=0}^m d_E^2(c_i^k, c_j^k) + \lambda \sum_{k=0}^m d_E^2(\dot{c}_i^k, \dot{c}_j^k)$$

11

Рисунок Г.11 – Розробка методу визначення відстані між двома точками на тривимірній моделі

Алгоритми роботи програмного забезпечення



12

Рисунок Г.12 – Алгоритми роботи програмного забезпечення



Рисунок Г.13 – Модуль визначення основних точок антропометрії на тривимірній моделі



Рисунок Г.14 – Модуль вимірювання антропометричних параметрів на тривимірній моделі

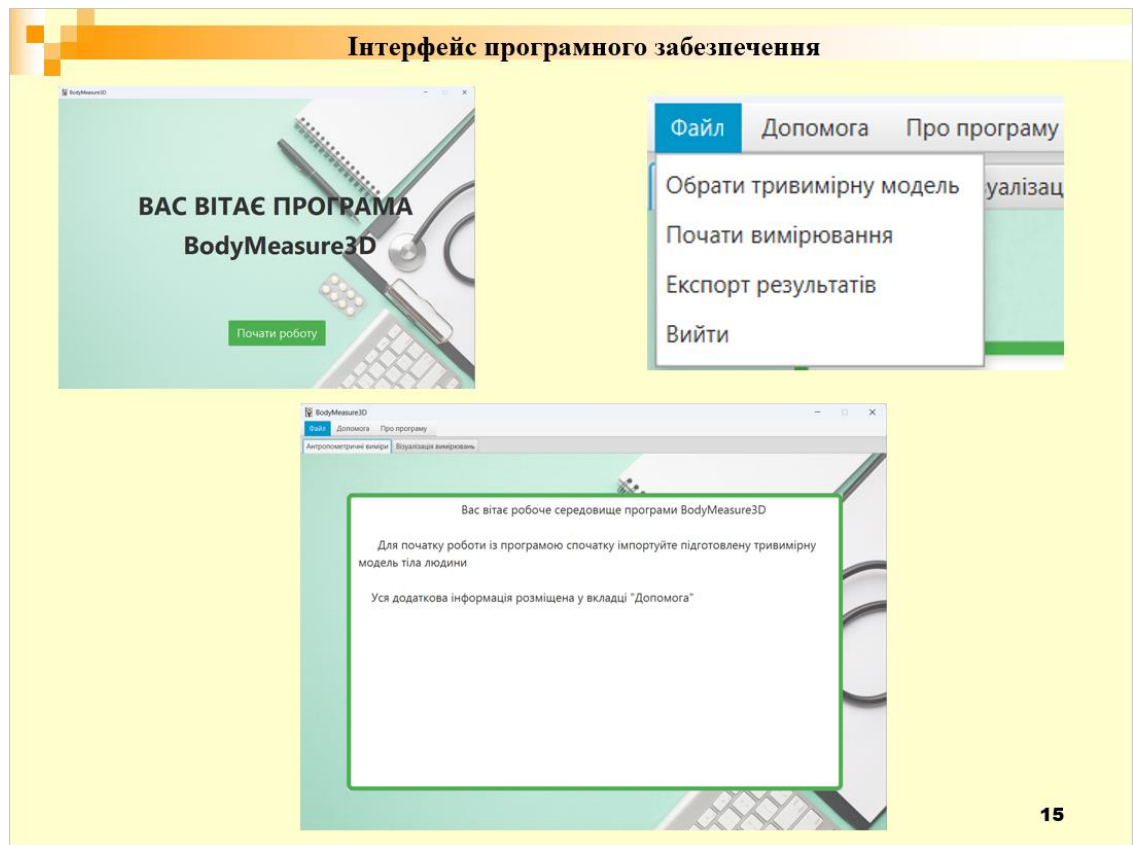


Рисунок Г.15 – Інтерфейс програмного забезпечення

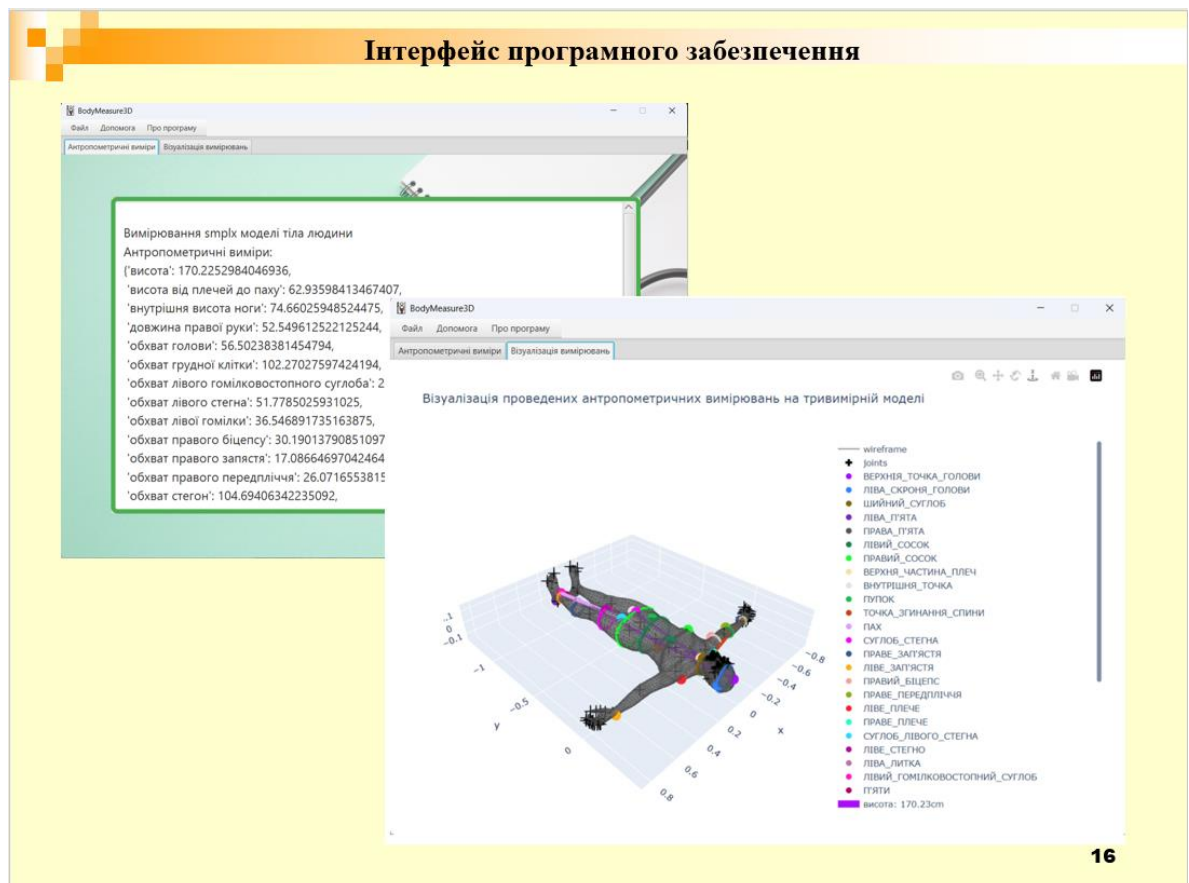
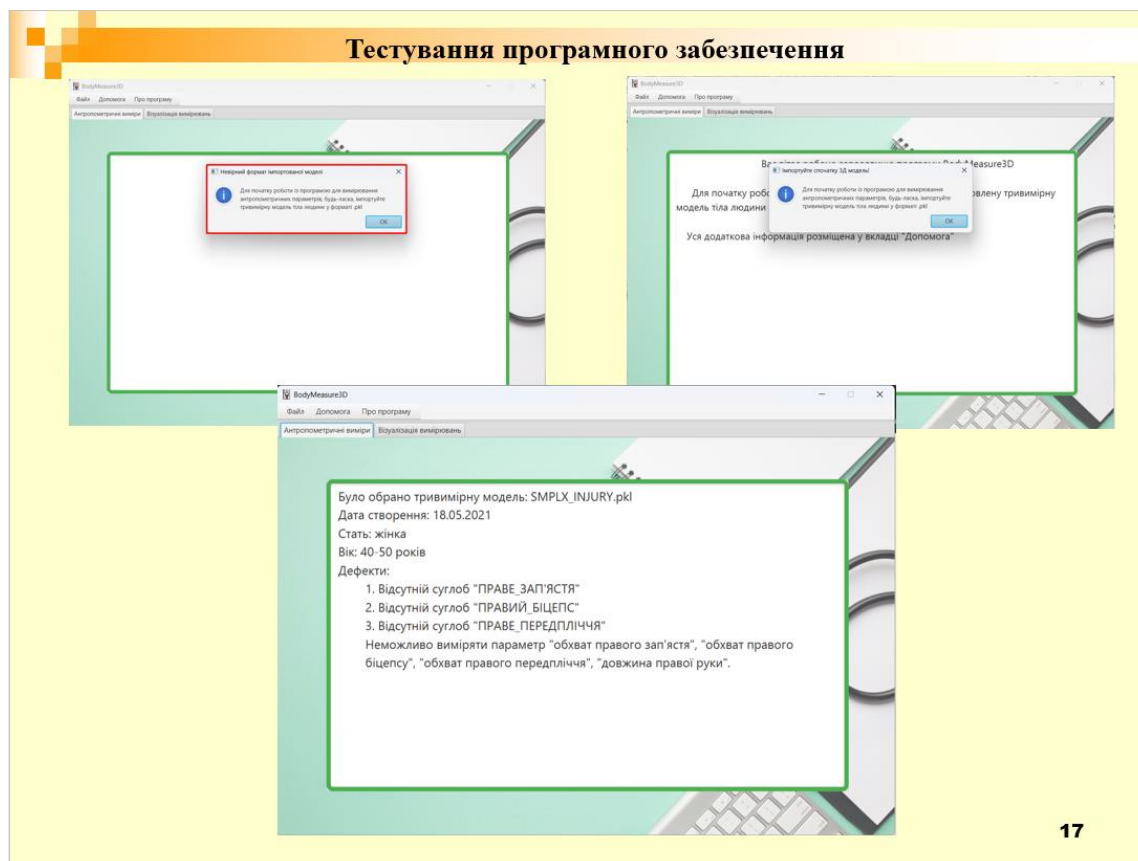


Рисунок Г.16 – Інтерфейс програмного забезпечення



17

Рисунок Г.17 – Тестування програмного забезпечення

Наукова новизна отриманих результатів

1. Вперше запропоновано метод визначення об'єму тіла людини з використанням тривимірного моделювання, особливість якого полягає у виконанні горизонтальних зрізів тривимірної моделі заданої товщини, виконання триангуляції та визначення сумарного об'єму отриманих складових призм, що дозволяє підвищити точність вимірювання.
 2. Вперше запропоновано метод визначення площі тіла людини з використанням тривимірного моделювання, особливість якого полягає у формуванні текстурної розгортки поверхні людини на горизонтальну площину з подальшим визначенням сумарної площі отриманих фігур, що дозволяє підвищити точність вимірювання.
 3. Вперше запропоновано метод визначення антропометричних відстаней на тривимірній моделі, на основі позицій 16 основних суглобів, особливість якого полягає у використанні нової формули визначення відстані між двома точками на тривимірній моделі тіла людини, у застосуванні горизонтальної кореляційної лінії через вершини трикутників та обрахування відстані використовуючи формули векторів. Це дає можливість автоматизувати процес вимірювання.
- **Практичне значення отриманих результатів** полягає в тому, що на основі теоретичних досліджень розроблено алгоритми та програми для антропометричних вимірювань та їх інтеграції в професійний графічний рушій idx3d.

18

Рисунок Г.18 – Наукова новизна

В бакалаврській дипломній роботі виконано такі задачі:

- аналіз предметної галузі;
- розробка алгоритму обробки та аналізу даних, отриманих із тривимірної моделі тіла людини.
- розробити алгоритм побудови тривимірних моделей тіла людини на основі антропометричних даних та їх тривимірної візуалізації;
- розробити алгоритми вимірювання, які забезпечать високу точність та надійність отриманих даних на основі тривимірних моделей;
- розробка алгоритму визначення позицій суглобів, який буде спроможний виявляти, відстежувати та відображати позиції 16 основних суглобів, необхідних для вимірювань;
- покращення процесу вимірювання антропометричних даних із забезпеченням точності даних.
- тестування розроблених програмних продуктів;

19

Рисунок Г.19 – Виконані задачі

Публікації

- За темою бакалаврської роботи надруковано 12 праць у матеріалах республіканських і міжнародних конференціях та 1 робота у фаховому виданні. Отримано авторське право на комп'ютерну програму.

20

Рисунок Г.20 – Публікації

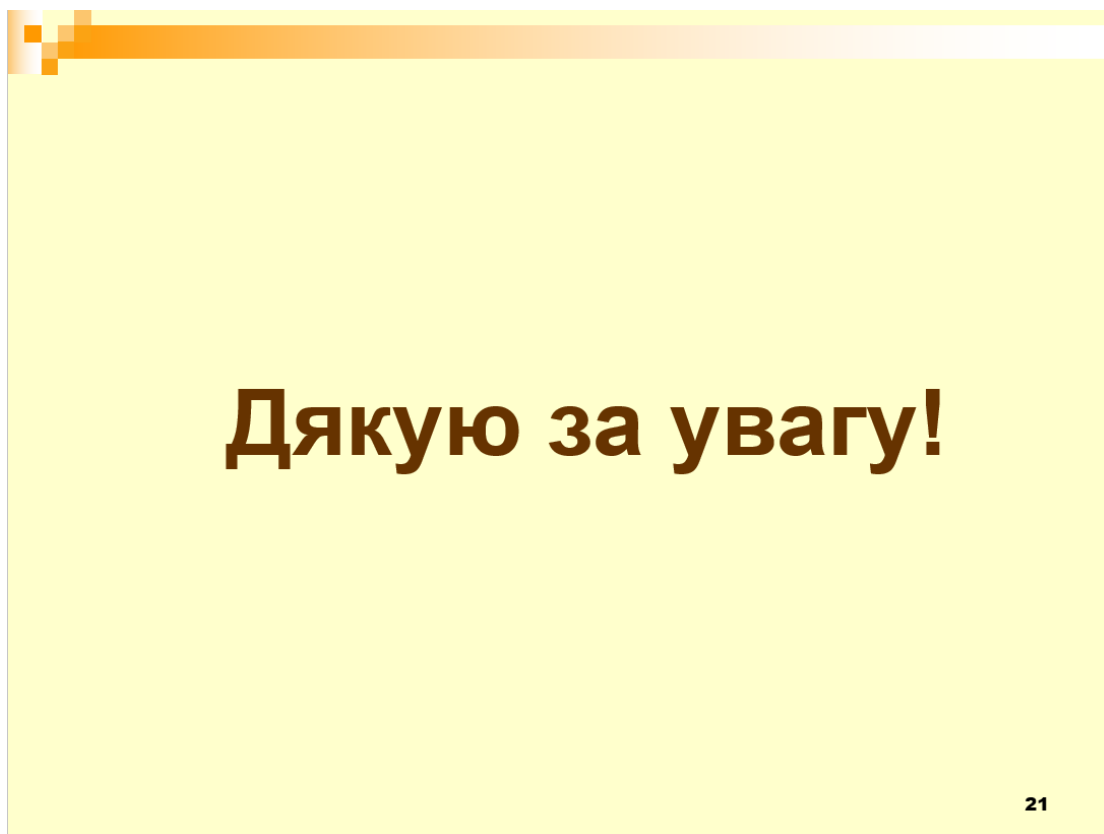


Рисунок Г.21 – Закінчення презентації