

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Інформаційних управляючих систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження моделей і методів виявлення зовнішніх пошкоджень ІТ-проекту
сканування багажу
(тема)

Виконав:
студент 5 курсу, групи УПГІТМ-20-1
Шипова В.С.
(прізвище, ініціали)

Спеціальність 122 - Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Управління проектами
в галузі інформаційних технологій
(повна назва освітньої програми)

Керівник д.т.н., проф. Левикін В.М
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ІУС

(підпис)

Петров К.Е.
(прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

(повна назва)

Кафедра Інформаційних управляючих систем

(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 - Комп'ютерні науки

(код і повна назва)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Управління проектами в галузі інформаційних технологій

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

« ____ » _____ 2021 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

Студенту Шипової Віолетти Сергіївни

(прізвище, ім'я, по батькові)

1. Тема роботи: Дослідження моделей і методів виявлення зовнішніх пошкоджень ІТ-проекту сканування багажу

затверджена наказом університету від 05.11.2021 р. №1646Ст

2. Термін подання студентом роботи до екзаменаційної комісії 13.12.2021 р.

3. Вихідні дані до роботи: Науково-технічні публікації, джерела інтернету, що стосуються теми атестаційної роботи.

4. Перелік питань, що потрібно опрацювати в роботі: Вступ; Аналіз предметної області та постановка задачі алгоритмування сканування багажу пасажирів; Вдосконалення моделі алгоритмів сканування багажу пасажирів; Побудова плану розробки проекту; Практичне використання вдосконаленої моделі сканування багажу клієнтів; Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій:

Побудова плану розробки проекту; діаграма Ганта – час початку і кінця робіт; діаграма Ганта проекту, із зазначенням ресурсів; логування в систему для веб-додатку; реєстрація в системі за допомогою QR-коду; мітка, котра містить в собі ідентифікаційний номер багажу; 3D-модель багажу з даними котрі були отримані після першого сканування.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз матеріалів з теми роботи	08.11.2021 – 10.11.2021	Виконано
2	Постановка задачі атестаційної роботи	09.11.2021 – 12.11.2021	Виконано
3	Обробка матеріалу	12.11.2021 – 15.11.2021	Виконано
4	Попереднє дослідження галузі завдання	15.11.2021 – 18.11.2021	Виконано
5	Аналіз існуючих методів і моделей сканування багажу	18.11.2021 – 25.11.2021	Виконано
6	Дослідження розробленої моделі алгоритмування	25.11.2021 – 28.11.2021	Виконано
7	Практичне використання вдосконаленого алгоритму	28.11.2021 – 01.12.2021	Виконано
8	Написання пояснювальної записки	01.12.2021 – 6.12.2021	Виконано
9	Підготовка презентації	06.12.2021 – 9.12.2021	Виконано
10	Перевірка на плагіат	09.12.2021	Виконано
11	Контроль норм	09.12.2021 – 13.12.2021	Виконано
12	Захист	14.12.2021	Виконано

Дата видачі завдання 08.11.2021 р.

Студент _____
(підпис)

Керівник роботи _____ д.т.н., проф. Левикін В.М.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи містить: 82 с., 21 рисуноків, 1 таблиця, 25 джерел.

АЕРОПОРТ, АЛГОРИТМИ, БАГАЖ, МЕТОДИ СКАНУВАННЯ,
МОДЕЛІ СКАНУВАННЯ.

Метою даної роботи є дослідження методів та моделей сканування багажу, з метою підвищення ефективності роботи сканерів, для підвищення оптимізації, прибутку та вдосконалення послуг сканування в логістичних пунктах та в будь-якій сфері в цілому.

Об'єктом дослідження в рамках магістерської атестаційної роботи, є дослідження та вивчення моделей сканування багажу та оптимізації сканування, в рамках аеропорту.

Предметом дослідження, є методи сканування багажу клієнтів IT-проекту, а саме оптимізація сканування багажу клієнтів в аеропорту.

Теоретичними результатами дослідження є описи удосконаленої моделі оптимізації та алгоритмізації сканування багажу.

Практичними результатами, є впровадження удосконаленої моделі оптимізації сканування багажу клієнтів в IT-проект, та приведені результати оптимізованого сканування.

Новизна дослідження полягає у вивченні існуючих моделей та методів в розробці, на підставі отриманої інформації, удосконаленої моделі оптимізації алгоритмів сканування багажу клієнтів в IT-проекті.

ABSTRACT

The explanatory note to the master's attestation work contains: 82 pages, 21 figures, 1 table, 25 sources.

AIRPORT, ALGORITHMS, LUGGAGE, SCAN METHODS, SCAN MODELS.

The purpose of this work is to study the methods and models of luggage scanning, in order to increase the efficiency of scanners, to increase optimization, profits and improve scanning services at logistics points and in any field in general.

The object of research in the framework of the master's certification work is the study and study of models of luggage scanning and scanning optimization, within the airport.

The subject of the study are the methods of scanning the luggage of customers of the IT project, namely the optimization of scanning the luggage of customers at the airport.

The theoretical results of the study are descriptions of an improved model of optimization and algorithmization of luggage scanning.

The practical results are the introduction of an improved model for optimizing the scanning of customers' luggage in the IT project, and the results of the optimized scan.

The novelty of the study is to study the existing models and methods in the development, based on the information obtained, an improved model for optimizing the algorithms for scanning customers' luggage in the IT project.

ЗМІСТ

Скорочення та умовні позначки і термінів.....	8
Вступ.....	9
1. Аналіз предметної області та постановка задачі і алгоритмування сканування багажу пасажирів.....	11
1.1 Опис предметно області застосування сканування 3D сканерів.....	11
1.2 Аналіз методів 3D сканування предметів.....	13
1.2.1 Контактний метод сканування.....	15
1.2.2 Безконтактні методи сканування.....	16
1.2.3 Лазерні сканери.....	16
1.2.4 Сканери, у яких використовується метод фокусного ряду.....	16
1.2.5 Сканери, в яких використовується стереоефект.....	17
1.2.6 Метод сканування, що ґрунтується на використанні структурованого світла	19
1.2.7 Тривимірне сканування, засноване на дифузному відображенні світла об'єктами.....	21
1.3. Постановка задачі дослідження.....	24
2. Вдосконалення моделі алгоритмів сканування багажу пасажирів.....	26
2.1 Дослідження моделей алгоритмування процесу сканування	27
2.1.1 Алгоритм плавучого горизонту.....	27
2.1.2 Алгоритм Робертса.....	31
2.1.2.1 Визначення нелицьових граней.....	31
2.1.2.2 Видалення невидимих ребер.....	33
2.1.3 Алгоритм Варнока.....	34
2.1.4 Алгоритм Вейлера-Азертонна.....	36
2.1.5 Алгоритм, який використовує z-буфер.....	38
2.1.6 Алгоритм RANSAC.....	41
2.1.7 Алгоритм ітерації найближчої точки (ICP).....	44
2.2 Порівняння двох груп алгоритмів для 3D сканування.....	46

2.3 Розробка моделі процесу сканування та відстеження багажу клієнтів.....	47
2.4 Удосконалення алгоритму Робертса та практичний приклад.....	48
2.4.1 Архітектура веб-додатку сканера.....	56
3. Побудова плану розробки проекту	59
3.1. Розробка плану проекту веб-додатку.....	59
4. Практичне використання вдосконаленої моделі сканування багажу клієнтів.....	66
4.1 План практичної реалізації проекту	66
4.2 Застосування вдосконаленої моделі сканування.....	66
Висновки.....	71
Перелік джерел посилань.....	72
Додаток А Графічний матеріал.....	73

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

АСУП – (Автоматизована система управління підприємством) – інтегрована автоматизована система;

BD – (Data Based) – база даних;

ICP – (Iterative Closest Point) - алгоритм, що використовується для мінімізації різниці між двома хмарами точок;

STL – (STereoLithography) - це форма технології 3D-друку, яка використовується для створення моделей, прототипів, візерунків та виробничих деталей пошарово за допомогою фотохімічних процесів;

RANSAC – (RANdom SAmple Consensus) - ітераційний метод для оцінки параметрів математичної моделі з набору спостережуваних даних, що містить викиди, коли викиди не повинні впливати на значення оцінок.

ВСТУП

На сьогоднішній день 3Д сканування розвивається зі швидкістю геометричної прогресії. Великий спектр професій та сфер діяльності задіяють цю функцію для поліпшення виробництва свого бізнесу чи виробництва. Багато компаній користуються сучасними 3д принтерами, відтворюючи з їх допомогою макети різних форм і розмірів. Існують і такі, які відтворюють предмети цілком – не тільки маленькі (наприклад, чохли для телефонів, сувеніри, кросівки), але й великі (будинки, запчастини для двигунів тощо). Ось тільки все це було б неможливим без 3д сканерів. Саме вони дозволяють точно копіювати практично все, що завгодно - від величезних будівель і споруд до людини, тварин, маленьких предметів та багато іншого [1].

У цій роботі будуть вивчені сфери діяльності, де використовуються, як використовуються і що для цього потрібно робити.

3D-моделювання – процес створення тривимірної моделі об'єкта. Завдання 3D-моделювання - розробити зоровий об'ємний образ бажаного об'єкта. При цьому модель може відповідати об'єктам з реального світу (автомобілі, будівлі, ураган, астероїд), так і бути повністю абстрактною (проекція чотиривимірного фракталу) [2].

Графічне зображення тривимірних об'єктів відрізняється тим, що включає побудову геометричної проекції тривимірної моделі сцени на площину (наприклад, комп'ютерний екран) за допомогою спеціалізованих програм. Однак із створенням і впровадженням 3D-дисплеїв і 3D-принтерів тривимірна графіка не обов'язково включає проектування на площину[3].

Мета 3D-сканера полягає в тому, щоб створити хмару точок геометричних зразків на поверхні об'єкта. Надалі ці точки можуть бути екстраповані для відтворення форми предмета (процес, що називається реконструкцією). Якщо були отримані дані і про колір, то колір реконструйованої поверхні також можна визначити.

Актуальність дослідження полягає в тому, що не всі веб-додатки встигають обробити безліч запитів. Від цього втрачається клієнтура і дохід, так як ніхто не хоче втрачати своїх потенційних клієнтів, потрібно організувати режим роботи медіаторів та самого сканера і час обробки запитів сервера. Також треба модернізувати модель реалізації запитів. Тоді спираючись на статистику необхідно проводити покращення алгоритмування сканерів, для мінімізації часу відповіді на запит клієнта. В рамках ІТ проекту покращення сканування багажу пасажирів і для цього потрібно модернізувати веб-додаток, за допомогою удосконалення функцій архітектури на якій буде побудований сайт.

Завдяки дослідженням моделей і методів, які використовуються зараз необхідно модернізувати модель яка буде враховувати недоліки існуючих моделей сканування, також модель дозволить мінімізувати час відповіді запиту клієнта та підвищити результати веб-додатку. В майбутньому ця модель може зазнати змін і стати основою для подальших моделей або методів, які зможуть більш якісно і точно обчислити мінімальний час для сканування більш габаритних грузів.

Модель повинна відкрити можливість для раннього прогнозування кількості сканованих предметів в певний період часу спираючись на час сканування та опрацювання даних, допоможе визначити методи для вдосконалення сайту веб-додатку спираючись на думку пасажирів. Проаналізувавши зібрані дані при використанні моделі на реальних проектах, можна буде зробити висновки по необхідності більш глибокого вивчення даного напрямку.

При успішній працездатності моделі управління запиту, будуть спрямовані зусилля на уніфікацію даної моделі і адаптацію під окремі випадки проектних рішень.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ АЛГОРИТМУВАННЯ СКАНУВАННЯ БАГАЖУ ПАСАЖИРІВ

1.1 Аналіз предметної області застосування сканування 3D сканерів.

Існує величезна кількість робіт і професій, яким 3D сканування облягає та покращує вироблений продукт. Ці професії та середовища використання сканерів класифікуються за типом використання та масштабом [4].

Існує кілька класифікацій:

- технічне проектування. Завдяки впровадженню сканерів у цю галузь вирішуються такі проблеми, як індустриальний дизайн, створення вручну тривимірних моделей та їх оцифрування з подальшим доопрацюванням методами машинної графіки, створення тривимірних моделей наявних штампів, прес-форм та інших виробів складної форми, наприклад, унікальних, у разі потреби, їх зміни, ремонту або повторного відтворення;

- архітектура. В архітектурі тривимірне сканування застосовується для реставрації шляхом створення банку електронних копій (віртуального архіву) та подальшого виготовлення або відновлення (промислового відтворення) оригіналів скульптур та створення віртуальних архітектурних музеїв;

- медицина. У медицині основне застосування такі системи знайшли у хірургічному плануванні, ортопедії, протезуванні, пластичній хірургії, косметології;

- археологія. Останнім часом системи тривимірного сканування широко використовують у археології на вирішення таких завдань, як віртуальне збереження творів мистецтва, створення документації із тривимірним зображенням об'єктів, віртуальна реставрація, а також копіювання для подальшої реплікації;

- промисловість. Лазерне сканування в промисловості використовується

як метод дослідження під реконструкцію або виконавчу зйомку. Швидкість методу дозволяє дуже швидко отримати точну та актуальну 3D-модель виробництва «як є» або «як побудовано». Наявність таких даних дозволяє іншому якісному рівні спроектувати реконструкцію чи проконтролювати хід будівництва. Реконструкції виробничих будівель та промислових майданчиків, проектування та будівництво нових об'єктів, тарування нафтоналивних резервуарів, створення 3D-моделі підприємства для АСУП з метою підвищення ефективності експлуатації, модернізації, проектуванні, спорудження суден та платформ;

– транспортна інфраструктура. На об'єктах транспорту ефективність лазерного сканування визначається можливістю зйомки лінійних об'єктів без зупинення руху. 3D сканування тунелів, мостів, доріг зазвичай виконується з метою створення топографічних планів, профілів, перерізів для проектування ремонту, реконструкції, для наповнення автоматизованого електронного банку даних об'єктивними просторовими даними про фактичний стан об'єктів транспорту. Проектування з використанням тривимірних даних геодезичних пошуків не тільки спрощує сам процес проектування, але головним чином підвищує якість проекту, що мінімізує подальші витрати на етапі будівництва; всі виміри проводяться вкрай швидким і точним методом, що виключає людський фактор, ступінь достовірності інформації підвищується в рази, зменшується ймовірність помилки, а також виміри проводяться невідбивним способом, дистанційно, що підвищує безпеку роботи; наприклад, немає необхідності перекидати автостраду для зйомки поперечних перерізів, зводити будівельні риштування для вимірювання фасаду;

– нафтогазова галузь. Система лазерного сканування формує 3D-модель діючої труби. Сканер збирає дані з точністю до 40 мікрон і відображає всі можливі параметри: результати зношування та вм'ятини;

– реверс-інжиніринг. У цій сфері застосовується сканування виробів складної форми. Процес характеризується створенням математичної моделі чи креслень за існуючим фізичним зразком;

– кіновиробництво. У кінотеатрах показують безліч сучасних фільмів, у сюжеті, яких можна побачити «ожилих» фантастичних персонажів, під час створення яких застосовувалася технологія захоплення рухів, що дозволило зробити максимально реалістичними і вражаючими;

– логістика. На сьогоднішній день сканери використовують в аеропортах та інших логістичних центрах. Застосування пов'язані з скануванням багажу чи предметів пасажирів, які здійснюють перевезення з точки А точку Б із пунктом огляду багажу [5].

Усі всі сфери можна поглянути на рисунку 1.

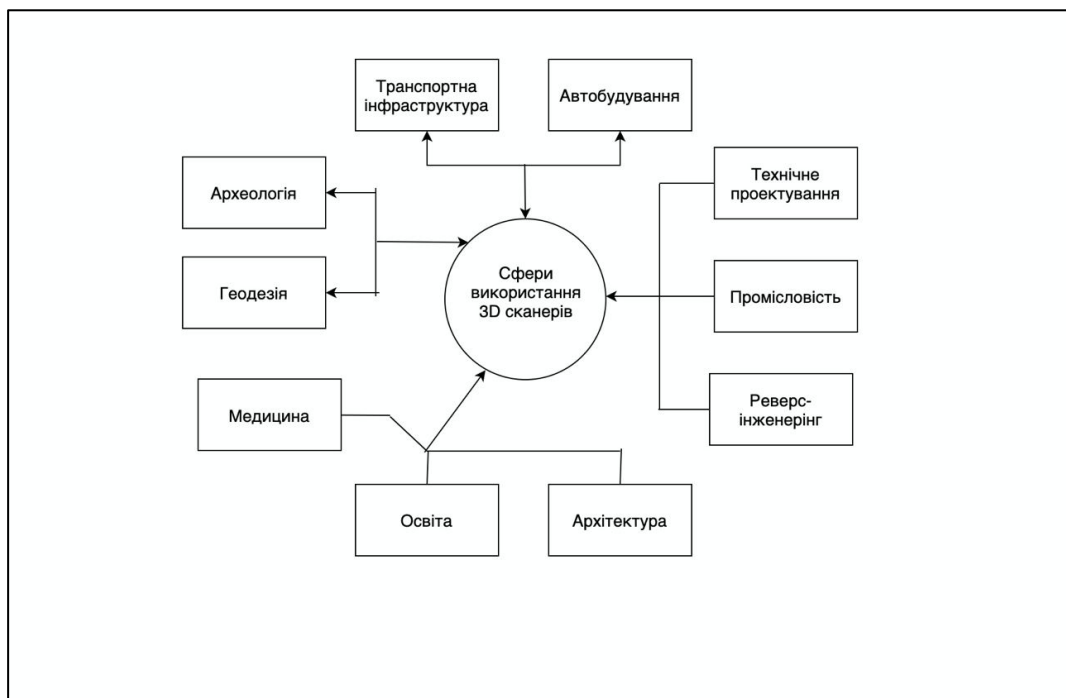


Рисунок 1 – Сфери використання 3D сканерів

1.2 Аналіз методів 3D сканування предметів.

Існує 2 метод відтворення 3D моделі. Перший метод – це метод малювання у спеціально створених програмах, таких як 3D Макс. Другий спосіб полягає в тому, що 3D-зображення об'єктів сцени отримують шляхом сканування реальних об'єктів, завдяки чому досягається реалістичність тривимірних зображень. Цей спосіб створення тривимірних сцен відрізняється великою трудомісткістю та високими грошовими витратами через високу вартість 3D-сканерів і великі витрати праці розробників.

1.2.1 Контактний метод сканування та сканери, що його реалізують.

В основі принципу дії контактного сканера лежить обведення поверхні об'єкта, що сканується оператором, яка виконується вручну. Сканери, засновані на цій технології, забезпечені спеціальним механічним пристосуванням, яке називається сенсором і є щупом. Перед скануванням на об'єкт, що сканується, наноситься сітка, розмір осередків який вибирається таким чином, щоб в областях високої кривизни поверхні сканованого об'єкта розмір осередків був би мінімальним, а в областях з малою кривизною поверхні — найбільшим. У точках, де лінії сітки перетинаються, за допомогою сенсора здійснюються вимірювання їх координат, які потім вводяться в комп'ютер [6].

Основною перевагою і одночасно недоліком цього типу сканерів є те, що тривимірна модель об'єкта створюється в пам'яті комп'ютера оператором вручну, що призводить до значних витрат часу. Другим недоліком безконтактних сканерів є неможливість читати текстуру. Крім того, сканери цього типу не можуть застосовуватися для сканування великих об'єктів, тому застосування контактних сканерів під час створення фільму або комп'ютерної

гри неможливе. Однак є області, де вони з успіхом знаходять своє застосування через їхню невисоку вартість, наприклад, для сканування скульптур, моделей корпусів всіляких невеликих деталей, де не потрібно зчитувати текстуру об'єкта, що сканується, і т.д.

1.2.2 Безконтактні методи сканування та сканери, що їх реалізують.

Безконтактні методи сканування поділяються на:

- лазерні методи сканування;
- оптичні методи сканування;
- ультразвукові методи сканування.

У свою чергу оптичні методи сканування поділяються на методи:

- метод фокусного ряду;
- стереоефект;
- структуроване світло;
- дифузне відображення світла об'єктами, що скануються.

Звертаючись до розгляду безконтактних методів сканування тривимірних об'єктів, слід зазначити наявність так званої проблеми зшивання частин, що отримується в результаті сканування тривимірного зображення. Істота проблеми полягає в тому, що при скануванні скануючий пристрій "бачить" лише частину об'єкта, що сканується, і для отримання 3D-зображення всього об'єкта необхідно просканувати його з усіх боків, а потім з'єднати (пошити) отримані частини 3D-зображень разом. Оскільки в більшості аналізованих далі методів координата глибини вимірюється лише для окремих точок об'єкта, що сканується, а для інших знаходиться шляхом інтерполяції, і, отже, на різних, зазвичай злегка перекриваються частинах зображень, отриманих шляхом сканування при різних взаємних положеннях об'єкта і скануючого устрою, її значення трохи розходяться, виникає

необхідність ручного узгодження. Це призводить до суттєвого збільшення трудомісткості [7].

1.2.3 Лазерні сканери

При використанні цього об'єкт сканується лазерним променем. У той час як горизонтальна і вертикальна координати точки об'єкта, що сканується, визначаються, виходячи з напрямку лазерного променя, третя координата, координата глибини x , знаходиться шляхом вимірювання часу затримки t сигналу, відображеного від поверхні сканованого об'єкта:

$$x = \frac{ct}{2}, \quad (1.1)$$

де c - швидкість поширення світла.

У сканерах цього типу використовуються лазерні датчики та цифрова фототехніка. Застосування цифрової фототехніки дозволяє створювати 3D-моделі об'єктів із текстурями. Достоїнством лазерних сканерів є можливість працювати "в автоматичному режимі", а також можливість сканувати великі об'єкти, що, як було зазначено, неможливо при використанні контактних сканерів. Слова "в автоматичному режимі" взяті в лапки, оскільки в повному розумінні автоматичних 3D-сканерів зараз не існує. Причиною є наявність проблеми зшивання зображень. Результати сканування вимагають зазвичай трудомісткого подальшого редагування, на яке йде багато часу [8].

1.2.4 Сканери, у яких використовується метод фокусного ряду

Як відомо, будь-яка оптична система забезпечує максимальну роздільну здатність тільки в одній площині. При віддаленні цієї площини на відстань d роздільна здатність зменшується пропорційно квадрату цієї відстані. Чим менша глибина дозволу, тим сильніше виражена ця залежність. У цьому методі ця властивість використана для визначення координати глибини сканованого об'єкта. Істота методу полягає в тому, що спочатку виходить ряд зображень при різних налаштуваннях оптичної системи, кожна з яких забезпечує максимальну роздільну здатність у своїй площині, видалення цієї площини від оптичної системи точно відоме, в результаті виходить так званий фокусний ряд. Потім для кожної точки зображення (піксела, стосовно цифрових зображень) методом порівняння значень сигналу у всіх зображеннях фокусного ряду знаходиться та площина, в якій зображення дрібних деталей об'єкта відтворюється з максимальною роздільною здатністю, що відповідає максимальному значенню сигналу від дрібної деталі об'єкта, розташованої в цьому місці. Знаючи видалення знайденої площини від оптичної системи, знаходять видалення (третю координату) цієї точки (піксела) 3D-зображення об'єкта. У тих областях зображень об'єкта, де дрібні деталі відсутні, третя координата знаходиться шляхом інтерполяції при використанні опорних точок тих пікселів, для яких вона визначена [9].

Недоліками розглянутого методу є:

- недостатня точність одержуваних результатів;
- проблема визначення третьої координати для дуже великих та дуже безструктурних областей об'єкта, що сканується, тобто для областей, в яких відсутні дрібні деталі.

1.2.5 Сканери, в яких використовується стереоефект

У цьому методі для вимірювання координат глибини 3D-зображення використовуються дві однакові камери, оптичні осі яких спрямовані паралельно один одному, а відстань (базис) між якими дорівнює b (рис. 1.2) [10].

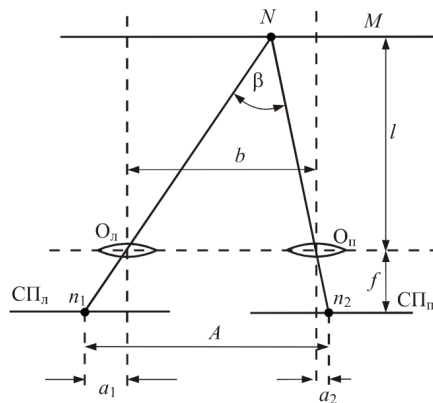


Рисунок 1.2 – Вимірювання відстані до об'єкта, що спостерігається

При відстані l між точкою об'єкта, до якої вимірюється відстань (глибина), і лінією, що з'єднує центри лівого та правого об'єктивів камер O_l і O_n , положення зображень точки об'єкта, створювані правою та лівою камерами на їх світлочутливих поверхнях $СП_l$ та $СП_n$, будуть зміщені на величини a_1 та a_2 щодо центральних осей. На цьому малюнку використані такі позначення: n_1 і n_2 - проєкції точкового об'єкта на зображеннях лівої та правої камер на їх світлочутливі поверхні, N - точковий об'єкт, M - площина, в якій розташований точковий об'єкт, b - відстань між об'єктивами камер, так званий базис, f - фокусна відстань об'єктивів [11].

Лінійний паралакс при цьому дорівнюватиме (1.2)

$$p = A - b = |a_1| + |a_2|. \quad (1.2)$$

З подоби трикутників n_1Nn_2 і $O_l NO_n$ випливає, що $b/A = l/(l + f)$, а, отже,

$$l = bf/(A - b) \quad (1.3)$$

Беручи до уваги вираз для паралаксу p , отримаємо остаточно формулу, яка приведена в 1.4.

$$l = bf/p \quad (1.4)$$

З отриманого виразу випливає, що величина паралаксу при відомому базисі і фокусній відстані камер дозволяє однозначно визначати відстань до будь-якої точки об'єкта, що спостерігається, що відрізняється за яскравістю та/або кольором від сусідніх точок.

Недоліки технології, заснованої на використанні стереозображень, полягають у тому, що при її застосуванні важко забезпечити точну відповідність для пікселів на зображеннях об'єкта, отриманих з двох камер у безструктурних областях, внаслідок чого неможливо забезпечити достатню точність 3D-зображень, що отримуються, а також наявність "проблеми" зшивання" [12].

1.2.6 Метод сканування структурованого світла

Проблема, що виникає при скануванні безструктурних областей об'єктів, з якою зустрічаються при використанні методу фокусного ряду та методу, заснованого на стереоскопічному ефекті, вирішується шляхом застосування так званого методу структурованого світла [13].

Відсутність дрібних деталей у цих галузях можна відшкодувати, створивши їх штучно шляхом проектування на них дрібних деталей за допомогою проектора. У цьому випадку проектор проектує на об'єкт зображення яскравої точки.

Далі зображення сканованого об'єкта фотографується за допомогою двох фотокамер. Оптична вісь першої фотокамер збігається з оптичною віссю проектора, а оптична вісь другої фотокамери по відношенню до оптичної осі першої фотокамери розташована під кутом 90° .

Маючи зображення точки на цих зображеннях, не важко визначити її положення в просторі. Таким чином, поміщаючи крапку в різні місця безструктурних областей об'єкта, можна легко визначити всі три її координати. Однак вирішення завдання можна суттєво спростити, використовуючи той факт, що у вигляді ліворуч одна з координат, а саме координата по вертикалі, повторює координату по вертикалі у вигляді спереду. Це дозволяє щодо третьої координати, саме координати глибини, проектувати на сканований об'єкт не точку, а пряму лінію.

На практиці при скануванні тривимірних об'єктів йдуть ще далі, проецирую на об'єкт, що сканується не одиночну лінію, а ґрати, що складається з ряду ліній. У цьому випадку при скануванні важливо з самого початку правильно задати відповідність ліній на обох фотографіях, що нескладно, розпочавши, наприклад, їх нумерацію на обох зображеннях. Таким чином, при використанні методу структурованого світла знаходяться

координати точок тієї частини поверхні, на яку спроектовані ґрати, незалежно від того, чи є на ній дрібні деталі, чи вони відсутні. Координати інших точок визначаються методом інтерполяції. Це є певним недоліком, оскільки потребує решіток із щільно розташованими тонкими лініями. Крім того, під час роботи з об'єктами, різні частини яких мають детальнізацію, що сильно відрізняється, необхідно вручну додавати точки в створювану модель, в областях високої детальності.

Незважаючи на простоту описаної технології і рекламовану виготовлювачами сканерів автоматичність режиму, процес отримання 3D-зображень виявляється також дуже трудомістким і вимагає великих витрат ручної праці, що є серйозним недоліком цього методу. Тут, як і раніше, при отриманні 3D-зображення об'єкта за декількома проекціями необхідно проводити зшивання цих зображень, а для цього потрібно попередньо провести калібрування камер, тобто визначити їхнє взаємне розташування [14].

1.2.7 Тривимірне сканування, засноване на дифузному відображенні світла об'єктами, що скануються.

Розглянутий далі метод 3D-сканування визначення координати глибини використовує розподіл світлотіней на двовимірних зображеннях об'єктів. Це виявляється можливим тому, що кількість світла, що потрапляє в об'єктив камери від кожної точки об'єкта, що сканується, визначається як відбивають властивостями поверхонь спостережуваних об'єктів, так і взаємним розташуванням цих поверхонь і джерел світла. Особливо просто цей метод реалізується при скануванні об'єктів, поверхня яких дифузно відображає світло, що подає на неї [15].

Розглянемо процес утворення світлотіней на зображенні об'єкта, при цьому вважатимемо, що об'єкт дифузно відображає світло, що падає на нього. Відповідно до закону косінусів Ламберта, сила відбитого світла I_1 дифузно відбиває поверхнею однаково у всіх напрямках і дорівнює:

$$I_1 = IK\cos\theta, \quad (1.5)$$

де I — сила падаючого світла, K — коефіцієнт відображення поверхні ($0 < K < 1$), θ — кут між нормаллю N до поверхні, що відбиває, і напрямком на джерело світла.

Сам факт залежності сили дифузно відбитого світла. Сам факт залежності сили дифузно відбитого світла від кутового положення поверхні, що відбиває по відношенню до напрямку, звідки відбувається її освітлення, дає принципову можливість визначати кутову орієнтацію відбиває поверхні в різних точках простору, а по них і всю поверхню. Ця обставина може бути використана для отримання тривимірних зображень при скануванні реальних об'єктів [16].

На цьому малюнку показана поверхня, що дифузно відображає світло, яка поміщена в початок координат, а також нормаль до цієї поверхні N , яка складає з напрямленням на джерело світла I кут θ (рис. 1.3).

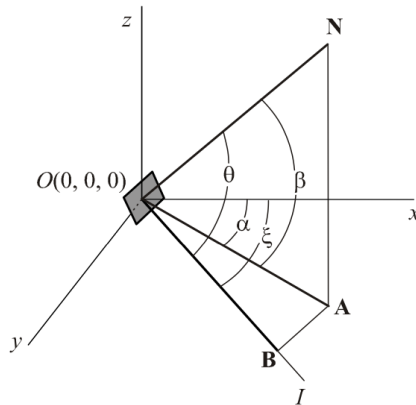


Рисунок 1.3 – Поверхня, що дифузно відображає світло, довільно орієнтована в просторі

Кут θ однозначно не визначає кутове положення в просторі нормалі до поверхні, що відбиває. Для того щоб однозначно задати кутове положення у просторі цієї нормалі, необхідно задати два кути, а саме: кут α – азимут та кут β – кут місця.

З вище написаного речення можна скласти співвідношення (1.6).

$$\cos\theta = \cos\beta \cos(\xi - \alpha). \quad (1.6)$$

Це співвідношення може бути отримане шляхом послідовного проектування нормалі N спочатку на координатну площину xu , а потім проектування проекції нормалі A на лінію I . При цьому матимемо співвідношення (1.7).

$$B = N \cos\beta \cos(\xi - \alpha) = N \cos\theta. \quad (1.7)$$

Для того, щоб зробити перетворення, потрібно з формули (1.6) зробити підстановку $\cos\theta$ в формулу (1.6) та отримаєм наступне перетворення (1.8).

$$I_1 = IK \cos(\xi - \alpha) \cos\beta, \quad (1.8)$$

де ξ - кут між віссю x і напрямком на джерело світла.

У цьому виразі відомими величинами вважаються сила світла I , яким висвітлюється поверхня, що дифузно відображає, сила світла I_1 , дифузно відображеного цією поверхнею, а також кут ξ , оскільки вони можуть бути виміряні. Невідомими ж величинами є кути α , β та коефіцієнт відображення поверхні K . Маючи всього одне рівняння та три невідомі, неможливо визначити ці невідомі. Однак, проблема може бути легко вирішена, якщо при вимірюваннях скористатися додатковими джерелами світла. Розташували додаткове джерело паралельних променів світла, наприклад, на осі x і

вимірявши силу світла I_0 , відображеного поверхнею, що розглядається від цього джерела, можна скласти додаткове рівняння (1.9).

$$I_0 = IK\cos\alpha\cos\beta. \quad (1.9)$$

Аналогічним чином можна отримати ще одне недостатнє рівняння, якщо додаткове джерело світла помістити так, щоб воно розташовувалося на лінії, що лежить у координатній площині xz (1.10).

$$I_2 = IK\cos(\xi - \beta)\cos\alpha, \quad (1.10)$$

де ξ - Кут між віссю x і напрямком на джерело світла.

Таким чином, для того щоб визначити третю координату (глибину) будь-якої точки об'єкта, що сканується на його двовимірному зображенні, необхідно в загальному випадку виміряти яскравість цієї точки при трьох різних розташуваннях джерела освітлення. Звернемо увагу на те, що при цих вимірюваннях освітлення точки об'єкта, для якого проводиться вимірювання, не повинно бути затінене. У разі сканування найпростіших об'єктів достатньо кожен їх точку по чергово висвітлити з трьох позицій. Більш складні об'єкти, наприклад особа людини, вимагають вже більшої кількості джерел світла, що використовуються по-змінно для освітлення різних її частин, тому що окремі її частини можуть бути затінені при деяких розташуваннях джерел освітлення. При цьому положення цих джерел можуть бути зручними для вимірювань.

Недоліком розглянутого методу, як і інших методів 3D-сканування, що використовують світло, є те, що він орієнтований на роботу з об'єктами, що дифузно відображають світло. Для сканування об'єктів, що мають блискучі, дзеркальні та прозорі поверхні, таких, як, наприклад, пофарбований кузов автомобіля, їх необхідно покривати антивідблиском (наприклад, як це

прийнято у використовуваних методах, шляхом розпилення з балончика), що створює білу матову поверхню [17].

1.3. Постановка задачі дослідження

На основі проведеного дослідження переваг і недоліків методів сканування, не всі методи можна застосувати до реалізації поставленого завдання, тому необхідно удосконалити метод і модель задоволеності клієнтів. Завдання досягнення мети є актуальність дослідження.

Об'єктом дослідження в рамках магістерської атестаційної роботи, є вивчення моделей оптимізації алгоритмів сканування та вивчення методів сканування, в рамках IT-проекту.

Предметом дослідження, є моделі оптимізації алгоритмів сканування IT-проекту, а саме оптимізація обробки сканів клієнтів в пунктах логістики.

Метою даної роботи є дослідження методів та моделей оптимізації алгоритмів сканування, з метою з метою збільшити пропускну спроможність у пунктах логістичних перевезень, для підвищення швидкості та точності, прибутку та лояльності клієнтів.

Для досягнення мети, досліджувати наступні питання:

- дослідити методи сканування в цілому;
- проаналізувати існуючі методи та моделі алгоритмів сканування;
- сформулювати або вдосконалити існуючу модель оптимізації сканування будь-якого багажу;
- проаналізувати архітектуру платформи додатку;
- удосконалити архітектуру додатку з використанням методів сканування;
- розробити план реалізації проекту;

- застосувати вдосконалену модель оптимізації сканування та алгоритмізування, разом з існуючими методами алгоритмізації, та оптимізації сканування.

2 ВДОСКОНАЛЕННЯ МОДЕЛІ АЛГОРИТМІВ СКАНУВАННЯ БАГАЖУ ПАСАЖИРІВ

Удосконалення – процес максимізації вигідних характеристик, співвідношень (наприклад, оптимізація виробничих процесів і виробництва), і мінімізації витрат.

Завдання удосконалення сформульована, якщо задані: критерій оптимальності (економічний, технологічні вимоги - вихід продукту, вміст домішок в ньому та інше); варіюють параметри (наприклад, температура, тиск, величини вхідних потоків в процесах переробки гірського і ін. сировини), зміна яких дозволяє впливати на ефективність процесу; математична модель процесу; обмеження, пов'язані з економічними і конструктивними умовами, можливостями.

Будь-яка компанія, яка має великі обсяги клієнтообігу та товарообігу, повинна орієнтуватися на оптимізацію всього процесу. Деякі дослідження показали, що оптимізація збільшує прибуток, клієнтообіг, скорочення часу та помилок при обробці документів, удосконалення контролю ваги та габаритних розмірів товару на стадії його отримання, використання штрих – кодів на аркушах підбору товару (комплектувальної накладної).

Організація може остаточно скористатися перевіреної систематичної моделлю сканування багажу клієнтів. Завдання організацій полягає в тому, щоб впровадити і забезпечити стандартизований процес сканування. Оптимізація процесу сканування, швидкість та мобільність можна розглядати як стратегічний інструмент розвитку бізнесу, і це позитивно впливає на прибутковість організації. Оптимізація процес становлять основу будь-якого успішного бізнесу, оскільки швидкість та мобільність веде до повторних використанню інструменту, лояльності до бренду та позитивним відгуками. Організаціям необхідно впровадити модель оптимізації, щоб поліпшити розвиток бізнесу і підвищити загальний рівень якості.

Метою всіх моделей оптимізації сканування, є отримання релевантних, надійних і достовірних результатів, які мають самий корінь фінансової здатності. Дослідження ступеня оптимізації будь-якого виробництва має проводитися з особливою ретельністю. Вимірювання та впровадження оптимізації алгоритмів має бути безперервним, послідовним, своєчасним, точним і надійним процесом. Саме тут, новий підхід оптимізації процесу, стає потужним стратегічним інструментом розвитку бізнесу для організації.

Теорія довела, що стійкі моделі та правила оптимізації повинні будуватися на чітко визначених прозорих процесах і на послідовному підході. Засоби, за допомогою яких досягається позитивний результат, можуть час від часу відрізнятися, і від групи / сегментації алгоритмів, незалежно від того, засноване це на географічній зоні, бізнес-підрозділі, країні, продукті або демографічній культурі, не має значення. Як наслідок вищесказаного, організації можуть отримати вигоду з чітко визначеною моделі оптимізації процесу.

2.1 Дослідження методів алгоритмування процесі сканування.

2.1.1 Алгоритм плаваючого горизонту

Алгоритм плаваючого горизонту можна зарахувати до класу алгоритмів, які у просторі зображення. Алгоритм плаваючого горизонту найчастіше використовується для видалення невидимих ліній тривимірного представлення функцій, що описують поверхню у вигляді (2.1) [18].

$$F = (x, y, z) = 0 \quad (2.1)$$

Головна ідея даного методу полягає у зведенні тривимірної задачі до двовимірної шляхом перетину вихідної поверхні послідовністю паралельних сіючих площин, що мають постійні значення координат x , y або z .

На рис. 2.1. наведено приклад, де вказані паралельні площини визначаються постійними значеннями z . Функція $F(x,y,z) = 0$ зводиться до послідовності кривих, що лежать у кожній з цих паралельних площин, наприклад, до послідовності $y=f(x,z)$ або $x=g(y,z)$, де z постійно на кожній із заданих паралельних площин.

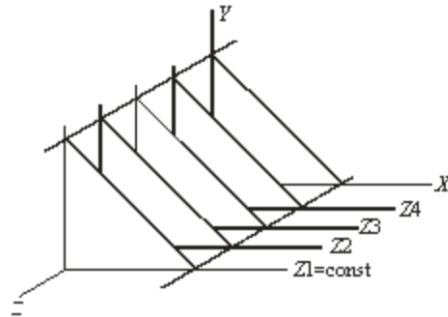


Рисунок 2.1 – Посічені площини з постійною координатою

Алгоритм спочатку упорядковує площини $z = \text{const}$ за зростанням відстані до них від точки спостереження. Потім для кожної площини, починаючи з найближчої до точки спостереження, будується крива, що лежить на ній, тобто для кожного значення координати в просторі зображення визначається відповідне значення y . Алгоритм видалення невидимої лінії полягає у наступному.

Якщо на поточній площині при деякому заданому значенні x відповідне значення y кривої більше значення y для всіх попередніх кривих при цьому значенні x , то поточна крива видима в цій точці; інакше вона невидима.

Невидимі ділянки показані пунктиром на рис. 2.2. Реалізація цього алгоритму досить проста. Для зберігання максимальних значень y при кожному значенні x використовується масив, довжина якого дорівнює числу точок (роздільна здатність) по осі x в просторі зображення. Значення, що зберігаються в цьому масиві, є поточними значеннями "горизонту". Тому в міру малювання кожної чергової кривої цей об'єкт "спливає".

Фактично цей алгоритм видалення невидимих ліній працює щоразу з однією лінією.



Рисунок 2.2 – Проекція кривих на площину $z = 0$

Подібні криві, природно, видимі і є нижньою стороною вихідної поверхні, проте алгоритм вважатиме їх невидимими. Нижня сторона поверхні робиться видимою, якщо модифікувати цей алгоритм, включивши до нього нижній горизонт, що опускається вниз у процесі алгоритму. Це реалізується за допомогою другого масиву, довжина якого дорівнює числу помітних точок по осі x в просторі зображення. Цей масив містить найменші значення у кожного значення x . Алгоритм тепер стає таким: якщо на поточній площині при певному заданому значенні x відповідне значення у на кривій більше максимуму або менше мінімуму за y для всіх попередніх кривих при цьому значенні x , то поточна крива видима. Інакше вона невидима.

У викладеному алгоритмі передбачається, що значення функції, тобто y відомо для кожного значення x в просторі зображення. Однак якщо для кожного значення x не можна вказати (обчислити) відповідне значення y , то неможливо підтримувати масиви верхнього і нижнього плаваючих горизонтів. У такому разі використовується лінійна інтерполяція значень y між відомими значеннями для того, щоб заповнити масиви верхнього та нижнього плаваючих горизонтів.

Але алгоритм має декілька дефектів, коли крива, що лежить в одній з більш віддалених від точки спостереження площин, з'являється ліворуч або праворуч з-під безлічі кривих, що лежать у площинах, які ближче до

зазначеної точки спостереження. Цей факт можна побачити на рис. 2.3. де вже оброблені площини $n-1$ і n розташовані ближче до точки спостереження. Після обробки кривих $n-1$ і n верхній горизонт для значень $x = 0$ і $x = 1$ дорівнює початковому значенню y ; для значень x від 2 до 17 він дорівнює ординатам кривої n ; а значень 18, 19, 20 - ординатам кривої $n-1$. Нижній горизонт для значень $x = 0$ та $x = 1$ дорівнює початковому значенню y ; для значень $x = 2, 3, 4$ - ординатам кривої n ; а значень x від 5 до 20 - ординатам кривої $n-1$. При обробці поточної кривої ($n+1$) алгоритм оголошує її видимою за $x = 4$. Аналогічний ефект виникає і праворуч при $x = 18$. Такий факт призводить до появи зазубрених бічних ребер. Проблема із зазубреністю бічних ребер вирішується включенням до масивів верхнього та нижнього горизонтів ординат, що відповідають штриховим лініям на рис. 2.3. Це можна виконати ефективно, створивши несправжні бічні ребра.

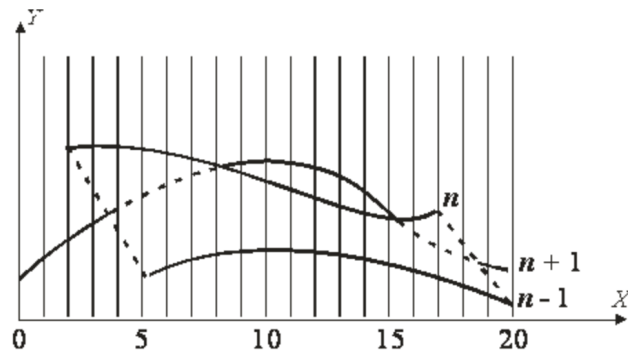


Рисунок 2.3 – Ефект зазубреного ребра

Якщо функція містить дуже гострі ділянки (піки), то наведений алгоритм може дати некоректні результати. Цей ефект викликаний обчисленням значень функції та оцінкою її видимості на ділянках, менших, ніж роздільна здатність екрану, тобто тим, що функція задана занадто малою кількістю точок. Якщо зустрічаються вузькі ділянки, то функцію слід обчислювати в більшій кількості точок.

2.1.2 Алгоритм Робертса

Алгоритм Робертса є першим відомим рішенням завдання про видалення невидимих ліній. Це математично елегантний метод, який працює в об'єктному просторі. Алгоритм насамперед видаляє з кожного тіла ті ребра чи грані, які екрануються самим тілом. Потім кожне з видимих ребер кожного тіла порівнюється з кожним з тіл, що залишилися для визначення того, яка його частина або частини, якщо такі є, екрануються цими тілами. Тому обчислювальна трудомісткість алгоритму Робертса зростає теоретично як квадрат числа об'єктів. Це у поєднанні із зростанням інтересу до растрових дисплеїв, що працюють у просторі зображення, призвело до зниження інтересу до алгоритму Робертса. Проте математичні методи, які у цьому алгоритмі, прості, потужні і точні. З іншого боку, цей алгоритм можна використовувати ілюстрації деяких важливих концепцій. Нарешті, пізніші реалізації алгоритму, що використовують попереднє пріоритетне сортування вздовж осі z та прості габаритні або мінімаксні тести, демонструють майже лінійну залежність від кількості об'єктів [19].

Алгоритм Робертса обробляє у 2 етапи:

- визначення нелицьових граней для кожного тіла окремо;
- визначення та видалення невидимих ребер.

2.1.2.1 Визначення нелицьових граней

Нехай F - деяка грань багатогранника. Площина, що несе цю грань, поділяє простір на два підпростори. Назвемо позитивним те з них, яке дивиться зовнішня нормаль до грані. Якщо точка спостереження – у позитивному підпросторі, то грань – лицьова, інакше – нелицьова. Якщо

багатогранник опуклий, видалення всіх нелицьових граней повністю вирішує завдання візуалізації з видаленням невидимих граней.

Для визначення, чи точка в позитивному підпросторі, використовують перевірку знака скалярного твору (l, n) , де l – вектор, спрямований до спостерігача, фактично визначає точку спостереження; n – вектор зовнішньої нормалі грані. Якщо $(l, n) > 0$, тобто кут між векторами гострий, грань є лицьовою. Якщо $(l, n) < 0$, тобто кут між векторами тупий, то грань є нелицьовою.

В алгоритмі Робертса потрібно, щоб усі ті чи об'єкти, що зображаються, були опуклими. Неопуклі тіла мають бути розбиті на опуклі частини. У цьому алгоритмі опукле багатогранне тіло з плоскими гранями має представитися набором площин, що перетинаються. Рівняння довільної площини у тривимірному просторі має вигляд:

$$ax + by + cz + d = 0 \quad (2.2)$$

У матричній формі цей результат виглядає так:

$$[xyzl][P]^T = 0, \quad (2.3)$$

де $[P]^T = [a \ b \ c \ d]$ є площиною.

Тому будь-яке опукле тверде тіло можна виразити матрицею тіла, що складається з коефіцієнтів рівнянь площин, тобто, де кожен стовпець містить коефіцієнти однієї площини.

Будь-яка точка простору уявна в однорідних координатах вектором $[S] = [x \ y \ z \ 1]$. Більше того, якщо точка $[S]$ лежить на площині, то $[S] \cdot [P]^T = 0$. Якщо ж $[S]$ не лежить на площині, то знак цього скалярного твору показує, по який бік від площини розташована точка. В алгоритмі Робертса передбачається, що точки, що лежать усередині тіла, дають негативний скалярний твір, тобто нормалі направлені назовні.

Цей метод є найпростішим алгоритмом видалення невидимих поверхонь для тіл, що є одиночними опуклими багатогранниками. Він також використовується для видалення нелицьових або задніх граней зі сцени перед застосуванням одного з алгоритмів видалення невидимих ліній, які обговорюються нижче. Цей спосіб часто називають відкиданням задніх площин. Для опуклих багатогранників число граней у своїй скорочується приблизно наполовину. Метод еквівалентний обчисленню нормалі до поверхні кожного окремого багатокутника. Даний метод визначення нелицьових граней у результаті формує аксонометрическую проекцію на якусь площину, розташовану нескінченно далеко від будь-якої точки тривимірного простору. Видові перетворення, включаючи перспективне, виробляються до визначення нелицевих площин. Коли видове перетворення включає у собі перспективу, потрібно використовувати повне перспективне перетворення одного тривимірного простору в інший, а чи не перспективне проектування на деяку двовимірну площину. Повне перспективне перетворення призводить до спотворення тривимірного тіла, яке потім проектується на якусь площину в нескінченності, коли вже невизначені площини. Цей результат еквівалентний перспективному проєціюванню з деякого центру на кінцеву площину проєкції.

2.1.2.2 Видалення невидимих ребер

Після першого етапу видалення нелицьових відрізків необхідно з'ясувати, чи існують такі відрізки, які екрануються іншими тілами у картинці чи сцені. Для цього кожен відрізок або ребро, що залишився, потрібно порівняти з іншими тілами сцени.

Можливі такі випадки виключень роботи алгоритму:

- грань ребра не закриває. Ребро залишається у списку ребер;

– грань повністю закриває ребро. Ребро видаляється зі списку ребер, що розглядаються;

– грань частково закриває ребро. І тут ребро розбивається кілька частин, видимими у тому числі є більше двох. Саме ребро видаляється зі списку ребер, що розглядаються, але в список ребер, що перевіряються додаються ті його частини, які даною гранню не закриваються.

Для удосконалення використовується пріоритетне сортування (z-сортировка), а також порівняння з прямокутними об'ємними оболонками тіл. Такий підхід дозволяє видалити цілі групи або кластери відрізків та тіл. Наприклад, якщо всі тіла в сцені впорядковані в певному пріоритетному списку, що використовує значення z найближчих вершин для подання відстані до спостерігача, то ніяке тіло з цього списку, у якого найближча вершина знаходиться далі від спостерігача, ніж найвіддаленіша з кінцевих точок ребра, не може закривати це ребро. Більше того, жодне з тіл, що залишилися, прямокутна оболонка якого розташована повністю праворуч, зліва, над або під ребром, не може екранувати це ребро. Використання цих прийомів значно скорочує число тіл, із якими потрібно порівнювати кожен відрізок чи ребро.

Рисунок 2.4 ілюструє завершену роботу алгоритму

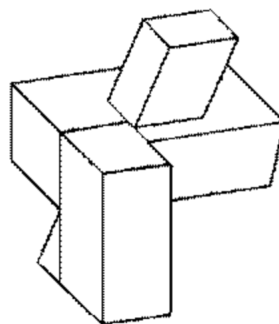


Рисунок 2.4 – Результат роботи алгоритму Робертса

2.1.3 Алгоритм Варнока

Алгоритм Варнока є одним із прикладів алгоритму, заснованого на розбиття картинної площини на частини, для кожної з яких вихідне завдання може бути вирішене досить просто.

Оскільки алгоритм Варнока націлений на обробку картинки, він працює у просторі зображення. У просторі зображення розглядається вікно і вирішується питання, чи порожньо воно, чи його вміст досить просто візуалізації. Якщо це не так, вікно розбивається на фрагменти до тих пір, поки вміст фрагмента не стане досить простим для візуалізації або його розмір не досягне необхідної межі дозволу [20].

Порівнюючи область з проекціями всіх граней, можна виділити випадки, коли зображення, що виходить в області, визначається відразу:

- проекція жодної грані не потрапляє до області;
- проекція лише однієї грані міститься в області або перетинає область. В цьому випадку проекції грані розбивають усю область на дві частини, одна з яких відповідає цій проекції;
- існує грань, проекція якої повністю накриває дану область, і ця грань розташована до картинної площини ближче, ніж інші грані, проекції яких перетинають цю область. У разі область відповідає цієї грані.

Якщо жоден із розглянутих трьох випадків немає місця, то знову розбиваємо область на чотири рівні частини і перевіряємо виконання цих умов кожної з частин. Ті частини, для яких таким чином не вдалося встановити видимість, знову розбиваємо і т.д.

Звісно, виникає питання критерії, виходячи з якого припиняти розбиття. Як очевидний критерій можна взяти розмір області. Як тільки розмір області стане не більше розміру одного пікселя, то проводити подальше розбиття не має сенсу і для цієї області найближча до неї межа визначається явно, як у методі трасування променів.

Іноді, для усунення сходового ефекту, процес розбиття проводиться до розмірів, менших за роздільну здатність екрана, на один піксель. При цьому усереднюються атрибути підпікселів, щоб визначити атрибути самих пікселів.

За допомогою викладеного алгоритму можна видалити або невидимі лінії або невидимі поверхні. Однак простота критерію розбиття, а також негнучкість способу розбиття призводять до того, що кількість подразбієнь виявляється великою. Можна підвищити ефективність цього алгоритму, ускладнивши спосіб та критерій розбиття.

На рисунку 2.5а показаний загальний спосіб розбиття і дано його порівняння з більш суворим способом розбиття, представленим на рисунку 2.5б.

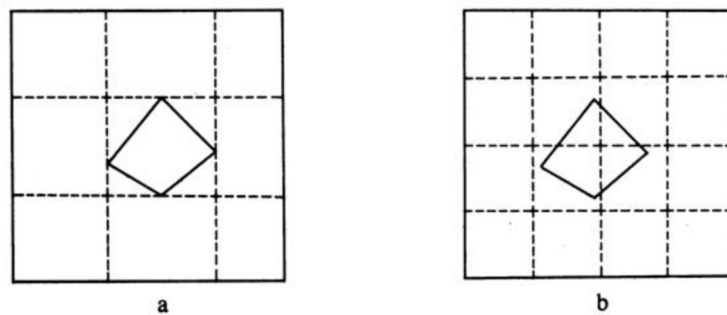


Рисунок 2.5 – Способи розбиття вікна

Розбиття, показане на рис. 2.5а, виходить з використанням прямокутної об'ємної оболонки багатокутника. Зауважимо, що оболонки у своїй можуть бути неквадратними. Цей спосіб можна рекурсивно застосувати до будь-якого багатокутника, повністю охопленого яким-небудь вікном або оболонкою. Якщо у вікні міститься лише один багатокутник і якщо він повністю охоплений цим вікном, його легко зобразити, не проводячи подальшого розбиття. Такий спосіб розбиття корисний, зокрема, за мінімізації числа розбиття для простих сцен. Однак зі зростанням складності сцени його перевага сходить нанівець.

2.1.4 Алгоритм Вейлера-Азертонна

Розбиття картинної площини можна проводити як прямими, паралельними координатним осям, а й у межах проєкцій граней. В результаті виходить точне розв'язання задачі. Пропонований метод працює з проєкціями граней на картинну площину. Як перший крок проводиться сортування всіх граней по глибині (front-to-back). Потім зі списку граней, що залишилися, береться найближча грань A і всі інші грані обрізаються по цій грані. Якщо проєкція грані B перетинає проєкцію грані A , то грань B розбивається на частини так, що кожна частина міститься в грані A , або не має з нею загальних внутрішніх точок [21].

Таким чином, виходять дві множини граней: $F(\text{in})$ - грані, проєкції яких містяться в проєкції грані A (сюди входить і сама грань A), і $F(\text{out})$ - грані, проєкції яких не мають спільних внутрішніх точок з проєкцією грані A . Безліч $F(\text{in})$ зазвичай називають безліччю граней, внутрішніх стосовно A .

Однак у безлічі $F(\text{in})$ можуть бути грані, що лежать до спостерігача ближче, ніж сама грань A (це можливо, наприклад, при циклічному накладення граней). У цьому випадку кожна така грань використовується для розбиття всіх граней із множини $F(\text{in})$ (включаючи вихідну грань A). Коли рекурсивне розбиття завершиться, то всі грані з першої множини виводяться з набору граней, що залишилися (їх вже ніщо не може закрити). Потім з набору граней $F(\text{out})$, що залишилися, береться чергова грань і процедура повторюється. На рисунку 2.6 можна побачити найпростіший випадок для двох граней A та B .

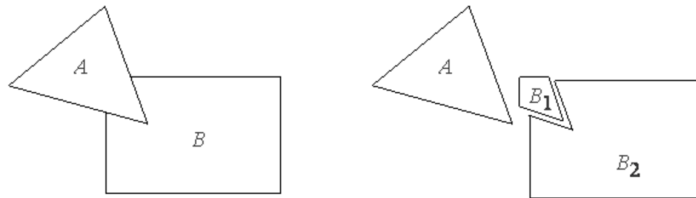


Рисунок 2.6 – Приклад розбиття граней

Грань А розташована ближче, ніж В. Тоді на першому кроці для розбиття використовується саме грань А. Грань В розбивається на дві частини. Частина B_1 потрапляє у перше безліч $F(in)$ і, оскільки лежить далі межі А, видаляється.

Після цього виводиться грань А і в списку граней залишається тільки грань B_2 . Оскільки крім неї інших граней не залишилося, то ця грань виводиться, і на цьому робота завершується.

2.1.5 Алгоритм, який використовує z-буфер

Алгоритм, що використовує z-буфер, це один із найпростіших алгоритмів видалення невидимих поверхонь. Вперше його було запропоновано Кетмулом. [22] Працює цей алгоритм у просторі зображення. Ідея z-буфера є простим узагальненням ідеї про буфер кадру. Буфер кадру використовується для запам'ятовування атрибутів (інтенсивності) кожного пікселя у просторі зображення, z-буфер - це окремий буфер глибини, який використовується для запам'ятовування координати z або глибини кожного видимого пікселя у просторі зображення. У процесі роботи глибина або значення z кожного нового пікселя, який потрібно занести до буфера кадру, порівнюється з глибиною того пікселя, який вже занесений до z-буфера. Якщо це порівняння показує, що новий піксель розташований попереду пікселя, що знаходиться в буфері кадру, новий піксель заноситься в цей буфер

i , крім того, проводиться коригування z -буфера новим значенням z . Якщо ж порівняння дає протилежний результат, то жодних дій немає. По суті алгоритм є пошуком по x і y найбільшого значення функції $z(x, y)$.

Головна перевага алгоритму – його простота. Крім того, цей алгоритм вирішує задачу про видалення невидимих поверхонь і робить очевидною візуалізацію перетинів складних поверхонь. Сцени можуть бути будь-якої складності. Оскільки габарити простору зображення фіксовані, оцінка обчислювальної трудомісткості алгоритму лише лінійна. Оскільки елементи сцени або картинки можна заносити в буфер кадру або z -буфер у довільному порядку, їх не потрібно попередньо сортувати за пріоритетом глибини. Тому економиться обчислювальний час, що витрачається на сортування за глибиною.

Основний недолік алгоритму - великий обсяг пам'яті. Якщо сцена піддається видовому перетворенню і відсікається до фіксованого діапазону значень координат z , можна використовувати z -буфер з фіксованою точністю. Інформацію про глибину потрібно обробляти з більшою точністю, ніж координатну інформацію на площині (x, y) ; зазвичай буває досить 20 біт. Буфер кадру розміром 512 512 24 біт в комбінації з z -буфером розміром 512 512 20 біт вимагає майже 1.5 мегабайт пам'яті. Однак зниження цін на пам'ять робить економічно виправданим створення спеціалізованих пристроїв для z -буфера і пов'язаної з ним апаратури.

Альтернативою створення спеціальної пам'яті для z -буфера є використання цієї мети оперативної пам'яті. Зменшення пам'яті досягається розбиттям простору зображення на 4, 16 або більше квадратів або смуг. У граничному варіанті можна використовувати z -буфер розміром один рядок розгортки. Для останнього випадку є цікавий алгоритм рядкового сканування. Оскільки кожен елемент сцени обробляється багато разів, сегментування z -буфера, взагалі кажучи, призводить до збільшення часу, необхідного для обробки сцени. Однак, сортування на площині, що дозволяє

не обробляти всі багатокутники в кожному з квадратів або смуг, може значно скоротити це зростання.

Інший недолік алгоритму z-буфера полягає у трудомісткості та високій вартості усунення сходового ефекту, а також реалізації ефектів прозорості та просвічування. Оскільки алгоритм заносить пікселі до буфера кадру в довільному порядку, то нелегко отримати інформацію, необхідну для методів усунення сходового ефекту, що ґрунтуються на попередній фільтрації. При реалізації ефектів прозорості та просвічування пікселі можуть заноситися до буфера кадру в некоректному порядку, що веде до локальних помилок.

Як попередній крок там, де це доцільно, застосовується видалення нелицьових граней.

Якщо відомо рівняння площини, що несе кожен багатокутник, то обчислення глибини кожного пікселя на рядку, що сканує, можна проробити покроковим способом. Грань у своїй малюється послідовно (рядок за рядком). Для знаходження необхідних значень використовується лінійна інтерполяція, рис 2.7.

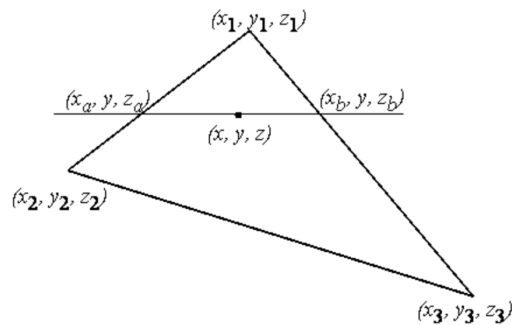


Рисунок 2.7 – Скануючий рядок за межею

Виходячи з рисунка 2.7 можна помітити, що y змінюється від y_1 до y_2 і далі до y_3 , причому для кожного рядка визначається x_a , z_a , x_b , z_b :

$$x_a = x_1 + (x_2 - x_1) * \frac{y - y_1}{y_2 - y_1}.$$

$$x_b = x_1 + (x_3 - x_1) * \frac{y - y_1}{y_3 - y_1}.$$

$$z_a = z_1 + (z_2 - z_1) * \frac{y - y_1}{y_2 - y_1}.$$

$$z_b = z_1 + (z_3 - z_1) * \frac{y - y_1}{y_3 - y_1}.$$

На скануючому рядку x змінюється від x_a до x_b і кожної точки рядка визначається глибина z :

$$z = z_a + (z_b - z_a) * \frac{x - x_a}{x_b - x_a} \quad (2.5)$$

Реалізація алгоритму вздовж скануючого рядка дозволяє поєднати алгоритм z -буфера з алгоритмами растрової розгортки ребер та алгоритмами забарвлення грані. Роботу алгоритму можна побачити на рисунку 2.8.

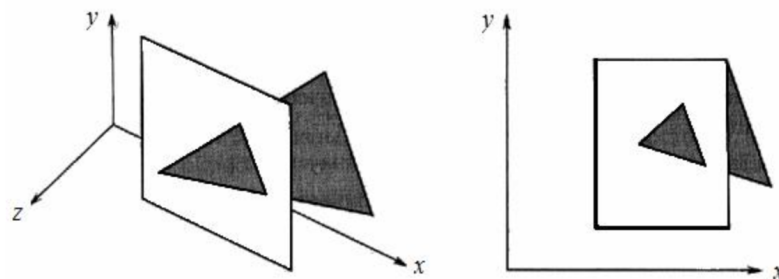


Рисунок 2.8 – Робота алгоритму z -буфера

На початку в буфері кадру та в z -буфері містяться нулі. Після растрової розгортки прямокутника вміст буфера кадру матиме вигляд двійкового коду в матриці.

2.1.6 RANSAC

Алгоритм RANdom Sample Consensus (RANSAC), запропонований Фішлером і Боллесом, є загальним підходом до оцінки параметрів, розробленим для того, щоб впоратися з великою часткою викидів у вхідних даних. На відміну від багатьох поширених методів надійної оцінки, таких як М-оцінки та найменші середні квадрати, які були прийняті спільнотою комп'ютерного зору зі статистичної літератури, RANSAC було розроблено в рамках спільноти комп'ютерного зору [23].

Це ітераційний метод для визначення параметрів математичної моделі за допомогою множини даних, що спостерігаються, які містять точки - викиди, що не задовольняють моделі (outliers). Тому RANSAC також можна назвати методом визначення вікидів. Це недетермінований алгоритм у тому сенсі, що він повертає прийнятну відповідь тільки з певною ймовірністю, і зі збільшенням цієї ймовірності при зростанні кількості ітерацій, що витрачаються. Базове припущення методу полягає в тому, що складаються з модельних прикладів і викидів, які описуються математичною моделлю. При цьому розподіл модельних прикладів може бути деяким безліччю параметрів.

Цей алгоритм має метод повторної вибірки, який генерує рішення-кандидати за допомогою мінімальної кількості спостережень (точок даних), необхідних для оцінки базових параметрів моделі. Як зазначають Фішлер і Боллес, на відміну від звичайних методів вибірки, які використовують якомога більше даних для отримання початкового рішення, а потім переходять до видалення викидів, RANSAC використовує найменший можливий набір і продовжує збільшувати цей набір за допомогою послідовних точок даних.

Даний алгоритм працює за наступним принципом:

- потрібно випадково вибрати мінімальну кількість точок, необхідну для визначення параметрів моделі;

- визначитися з параметрами моделі;
- визначитися, скільки точок із набору всіх точок відповідає попередньо визначеному допуску;
- якщо частка кількості інлієрів від загальної кількості точок у наборі перевищує попередньо визначений поріг t , повторно оцініть параметри моделі, використовуючи всі ідентифіковані інлієри, і заверште роботу.

Кількість ітерацій N , вибирається достатньо високою, щоб гарантувати, що ймовірність p (зазвичай дорівнює 0,99) того, що принаймні один із наборів випадкових вибірок не містить викидів. Нехай u представляє ймовірність того, що будь-яка вибрана точка даних є вбудованою та $v = 1 - u$ імовірність спостереження викиду.

Необхідні N ітерацій мінімальної кількості точок, позначених m будуть дорівнювати

$$1 - p = (1 - u^m)^N \quad (2.6)$$

Існує кілька модифікацій цього алгоритму:

- нерівномірний вибір вибірки на першому етапі кожної ітерації.

Одним із припущень завершення ітерацій у базовій версії є рівна ймовірність того, що всі точки вихідного хмари мають відповідну точку в цільовій хмарі. Це, очевидно, неправильно, оскільки через простих оклюзій сцени деякі точки вихідного хмари ніколи не будуть мати відповідної точки в цільовій хмарі. Існує оптимізація, коли кількість постійних гіпотез (~ 20) одно згенерована за допомогою вставок на попередньому кроці. Така модифікація призводить до найшвидшого знаходження термінальних критеріїв алгоритму;

- оптимізація перевірки гіпотез. Це один із методів, що дозволяють відкинути гіпотезу ще до оцінки матриці перетворення. Він порівнює полігони, створені з вихідним та цільовим хмарою, окремо. Якщо відношення відповідних сторін трикутників не перебуває у допустимому

інтервалі, гіпотеза відхиляється, і алгоритм повинен перейти до наступної ітерації. Цей підхід заснований на припущенні, що і вихідний, і цільовий об'єкти є твердотільні.

Інший спосіб оптимізації ґрунтується на "Td,d test". Перед перетворенням усієї вихідної хмари пропонують виконати перевірку випадково обраного підмножини d точок модельного хмари (де $d \mid S$). Співвідношення залишених точок можна було перевірити відразу після тесту Td, d - ймовірність відхилення вірної гіпотези мала, але отриманий виграш часу може бути помітний.

2.1.7 Алгоритм ітерації найближчої точки (ICP)

Метою алгоритму ICP є знаходження матриці геометричних перетворень, які вирівнюють хмари точок $X = \{x_1, x_2, \dots, x_n\}$ і $Y = \{y_1, y_2, \dots, y_n\}$ хмарою точок та найближчими точками є найближчими [24].

Вхідними даними алгоритму є два хмари точок, первинна оцінка трансформації та критерій для зупинення ітерацій. Вихідними даними є матриця геометричних перетворень.

ICP — алгоритм, який використовується для мінімізації різниці між двома хмарами точок. Алгоритм концептуально простий і часто використовується як реальна година. Він багаторазово застосовує перетворення (зміщення, обертання), необхідні для мінімізації відстані між точками із двох необроблених сканів. Входи: точки двох необроблених сканів, первинна оцінка трансформації, критерії для зупинки ітерації. Результат: досконале перетворення.

Недоліки алгоритму ICP: У стандартному алгоритмі ICP всі точки набору точок використовуються для обчислення відповідних точок. Кількість елементів у наборі точок, що використовується для реєстрації, зазвичай дуже

велика (близько кількох мільйонів точок). Розрахунок за цими наборами точок займає багато часу.

Відомо безліч варіацій алгоритму ICP. Основна мета цих методів - вибрати найменше кількість точок для подання всієї інформації про об'єкти вихідного набору точок. Година алгоритму зменшується за рахунок того, що зменшується кількість точок, що задіяні в алгоритмі.

У більшості випадків ICP дає хороший результат узгодження, але оскільки він заснований на тривимірних геометричних характеристиках поверхні об'єктів, складність обчислень висока.

Однією з основних проблем алгоритму ICP є обмеження області збіжності: алгоритм працює лише за умови, що хмари точок не сильно зрушені одна щодо одної. ICP також відомий своєю вразливістю до локальних мінімумів і добре працює тільки за хорошого початкового припущення про параметри перетворення.

2.2 Порівняння двох груп алгоритмів для 3D сканування

В даний час практично кожна людина намагається оточити себе корисними та красивими речами, однією з яких є комп'ютерна графіка (її можна зустріти у друкованій літературі, на екрані телевізора та рекламному плакаті, дизайнерському проекті). З розвитком комп'ютерних технологій комп'ютерна графіка дедалі більше проникає у життя людей, вона використовується в науці та освіті, бізнесі, кіноіндустрії, а також у різноманітних галузях промисловості. Але не слід забувати, що вивчення комп'ютерної графіки далеко не легкий процес. У цій статті ми розглянемо одне з найскладніших завдань комп'ютерної графіки – видалення невидимих ліній та поверхонь.

У процесі відображення тривимірної сцени на екрані може виникнути

ситуація, коли частина об'єктів сцени перекриває інші об'єкти. Невидимі спостерігачі частини об'єктів не повинні бути відображені на екрані, або повинні малюватись особливим чином (наприклад, пунктиром). Неправильне відображення видимих і невидимих ліній може призвести до спотворення зображення (до неправильного сприйняття). Вихід із подібних ситуацій можна знайти за допомогою алгоритмів видалення ліній та поверхонь. Невидимі лінії видаляються при відображенні сцени в каркасному вигляді (алгоритм виділяє частини ребер об'єкта, які заслонені та видаляє їх). Під час відображення об'єктів за допомогою зафарбованих поверхонь видаляються їх невидимі частини. Також визначаються видимі та невидимі для спостерігача частини обсягів, які можуть бути видалені аналогічним чином. В даний час існує безліч алгоритмів видалення невидимих ліній та поверхонь.

Таблиця 2.2 – Порівняльний аналіз алгоритмів видалення невидимих ліній

	Алгоритми, що працюють у просторі об'єкта	Алгоритми, що працюють у просторі зображення
Система координат	Алгоритми мають справу з фізичною системою координат, в якій описуються дані об'єкти	Алгоритми працюють із системою координат екрана, на який візуалізуються об'єкти
Точність	Висока точність	На точність обчислень впливає роздільна здатність екрану
Обсяг обчислень	зростає теоретично, як квадрат числа об'єктів n^2	зростає теоретично, як nN , де n - кількість об'єктів у сцені, N - число пікселів
Трудомісткість	менше	більше
Ефективність	менше	більше

2.3 Удосконалення алгоритму Робертса та практичний приклад

Для оптимізації та вдосконалення сканування багажу було обрано алгоритм Робертса. Алгоритм Робертса є одним з перших відомих рішень задачі про видалення невидимих ліній. Він працює в об'єктному просторі та вимагає, щоб грані були опуклими багатокутниками. Цей факт і став вибором застосування даного алгоритму для поточного ІТ-проекту. Для оптимізації використовується пріоритетне сортування (з-сортування), а також порівняння з прямокутними об'єктами оболонками тіл. Такий підхід дозволяє видалити цілі групи відрізків та тіл. Наприклад, якщо всі тіла в сцені впорядковані за значенням з найближчих до спостереження вершин, то жодне тіло з цього списку, у якого найближча вершина знаходиться далі від спостерігача, ніж найвіддаленіша з кінцевих точок ребра, не може закривати це ребро. Більш того, жодне з тіл, що залишилися, прямокутна оболонка якого розташована повністю праворуч, зліва, над або під ребром, не може екранувати це ребро. Використання цих прийомів значно скорочує кількість тіл, з якими потрібно порівнювати кожен відрізок чи ребро.

Алгоритм Робертса є одним з перших відомих рішень задачі про видалення невидимих ліній. Він працює в об'єктному просторі та вимагає, щоб грані були опуклими багатокутниками.

Даний алгоритм перебірного типу працює з ребрами, тобто більше підходить для каркасних зображень. І тут зображення формується видимими ребрами. Ребра, які стуляють дві невидимі грані, не розглядаються, т.к. вони свідомо невидимі.

Робота Алгоритм Робертса проходить у два етапи:

- визначення невидимих граней для кожного тіла у сцені окремо (тест видимості);
- визначення та видалення невидимих ребер.

Після першого етапу видалення нелицьових граней (рис. 2.9) визначається існування потенційно видимих ребер, які екрануються іншими тілами у сцені.

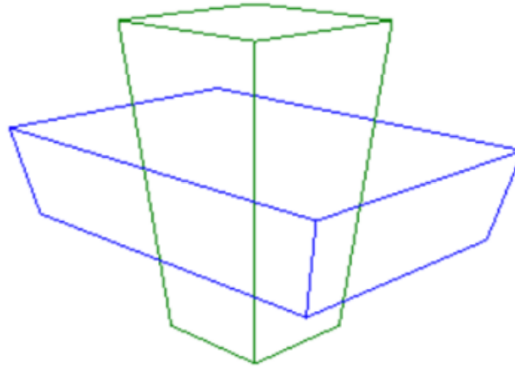


Рисунок 2.9 – Застосування тесту видимості для тестової сцени

Кожне з ребер, що залишилися, в сцені перевіряється на взаєморозташування з гранями кожного з інших об'єктів. Метою вважається визначення частин ребра, які екрануються цими об'єктами.

Можливі такі випадки розташування:

- грань ребра не закриває. Ребро залишається у списку ребер;
- грань повністю закриває ребро. Ребро видаляється зі списку розглянутих ребер;
- грань частково закриває ребро. І тут ребро розбивається кілька частин, видимими у тому числі є більше двох. Саме ребро видаляється зі списку ребер, що розглядаються, але в список ребер, що перевіряються додаються ті його частини, які даною гранню не закриваються.

Позначимо вихідні дані перед застосуванням алгоритму:

P_1P_2 – досліджуване ребро.

$P(t) = P_1 + t * (P_2 - P_1)$ або $\vec{v} = \vec{s} + t * \vec{d}$ – параметричне рівняння прямої, на якій лежить досліджуване ребро, де $\vec{s} = P_1, \vec{d} = (P_2 - P_1)$.

Необхідно знайти таке t , у якому змінюється видимість ребра.

Параметричне завдання відрізка від точки \vec{v} до \vec{g} – точки спостереження:

$$Q(a,t) = \vec{v} + a * \vec{g}, \text{ де } 0 \leq a.$$

Таким чином $Q(a,t) = \bar{s} + t\bar{d} + a\bar{g}$ – фактично є рівнянням площини.

Скалярний добуток будь-якої точки, розташованої всередині об'єкта та матриці об'єкта, позитивно (це твердження справедливе і для перетвореної матриці об'єкта). Крапка, що знаходиться всередині об'єкта, невидима. Отже, для перевірки на екранування вектор поточної точки відрізка по черзі множать на матрицю кожного об'єкта і визначають позитивне рішення, відповідне проходження відрізка всередині об'єкта.

Знайдемо такі t та a , для яких:

$$\bar{H} = (\bar{s} + t * \bar{d} + a * \bar{g}) * [V] > 0 \text{ або}$$

$$\bar{H} = \bar{s} * [V] + t * \bar{d} * [V] + a * \bar{g} * [V] > 0$$

Якщо всі компоненти \bar{H} для деяких t та a невід'ємні, то відрізок при цих t екранується тілом.

Позначимо:

$$\bar{p} = \bar{s} * [V]; \bar{q} = \bar{d} * [V]; \bar{w} = \bar{g} * [V];$$

Тоді:

$$h_i = p_i + t * q_i + a * w_i > 0, \text{ де } 0 \leq t \leq 1.0, 0 \leq a, i - \text{ номер грані.}$$

Далі потрібно вирішити задачу лінійного програмування при n невідомих та m обмежувачах.

Після прирівнювання h_i до 0 необхідно отримати всілякі пари систем 2-х рівнянь, при n гранях число розв'язків $\frac{n(n-1)}{2}$. Враховуючи обмеження $0 \leq a, t \leq 1, 0 \leq t$ буде $\frac{(n+3)(n-2)}{2}$ всіляких рішень.

Спочатку знаходяться t і a для однієї пари рівнянь і підставляються до інших.

Потім розраховуються t_{\min} – максимальне серед мінімальних значень t (що ближче до 0) та t_{\max} – мінімально серед максимальних значень t .

До того, як вирішувати систему, можна заздалегідь визначити факт повної видимості відрізка:

- $p_i < 0$ – перша вершина відрізка лежить по зовнішній бік від i -ої грані;
- $p_i + q_i < 0$ – друга вершина відрізка лежить по зовнішній бік від i -ої грані;
- $w_i \leq 0$ – i -а грань лицьова(видима);
- $w_i \leq 0$ та $p_i \leq 0$ – один кінець відрізка лежить або на видимій площині або між видимою площиною та точкою спостереження;
- при $w_i \leq 0$, $p_i \leq 0$ та $p_i + q_i \leq 0$ – гарантується, що відрізок повністю бачимо.

Застосування алгоритму можна поділити на такі етапи:

- визначення коефіцієнтів рівняння площини кожної грані, перевірка правильності знака рівняння та формування матриці об'єкта візуалізації;
- визначення нелицьових граней, видалення їх зі списку граней та відповідних ребер – зі списку ребер;
- визначення невидимих відрізків чи ділянок відрізків.
- формування списку можливих відрізків, що з'єднують точки протикання для пар об'єктів, пов'язаних відношенням протикання;
- перевірка видимості отриманих відрізків по відношенню до всіх об'єктів сцени відповідно до етапів 2 та 3;
- візуалізація зображення.

Як приклад застосування алгоритму розглядається одне з ребер у тестовій сцені (рис. 2.10).

На рисунку 2.11 представлена блок-схема алгоритму Робертса.

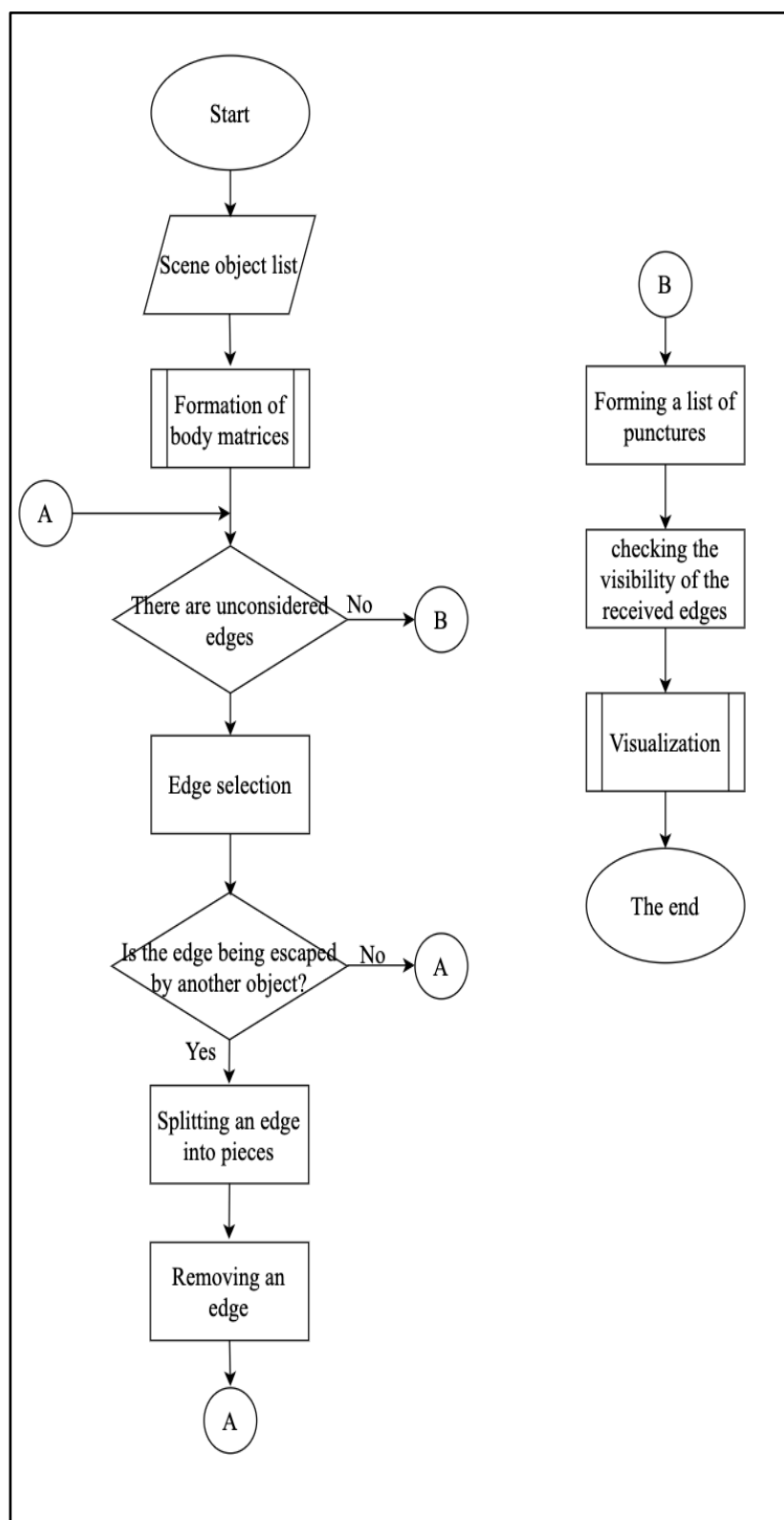


Рисунок 2.9 – Блок-схема алгоритму Робертса

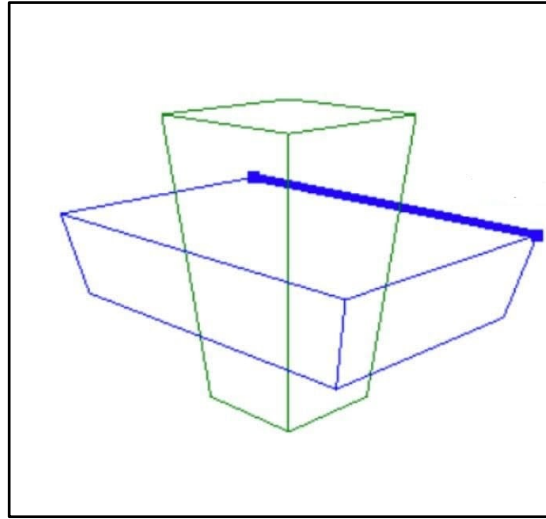


Рисунок 2.10 – Алгоритм Робертса для вибраного ребра

Початкові дані

$P_n = (-15,6; 8,4; -10,8)$, $P_k = (15,6; 8,4; -10,8)$ – тестоване ребро

$$[V] = \begin{bmatrix} 720 & -720 & 0 & 0 & 0 & 0 \\ 0 & 0 & 720 & -720 & 0 & 0 \\ 0 & 0 & 0 & 0 & 720 & -720 \\ 4320 & 4320 & 5011 & 3628 & 4320 & 4320 \end{bmatrix}$$

$[V]$ = матриця тіла зеленого об'єкта (нормалі до меж спрямовані всередину тіла);

$$\bar{d} = (31,2; 0; 0; 0); \quad \bar{s} = (-15,6; 8,4; -10,8); \quad \bar{g} = (40; 40; 40);$$

$$\bar{p} = \bar{s} * [V] = (-6912; 15552; 6220; 2419; -3456; 12096);$$

$$\bar{q} = \bar{d} * [V] = (22464; -22464; 0; 0; 0; 0);$$

$$\bar{w} = \bar{g} * [V] = (33120; -24480; 10771; -2131; 33120; -24480);$$

Система нерівностей $h_i = p_i + t * q_i + a * w_i > 0$ виглядає так:

$$\left\{ \begin{array}{l} -6912 + t * 22464 + a * 33120 > 0 \\ 15552 + t * -22464 + a * (-24480) > 0 \\ 6220,8 + t * 0 + a * 10771,2 > 0 \\ 2419,2 + t * 0 + a * (-2131,2) > 0 \\ -3456 + t * 0 + a * 33120 > 0 \\ 12096 + t * 0 + a * (-24480) > 0 \end{array} \right.$$

За умови, що $h_i = 0$ кожна з нерівностей фактично стає рівнянням площини. Пара рівнянь задають пряму, що належить обох площин. Необхідно отримати різні пари систем 2-х рівнянь.

Число рішень у такому разі дорівнює $\frac{n(n-1)}{2}$ або $\frac{6*5}{2} = 15$

Враховуючи обмеження $0 \leq a, t \leq 1, 0 \leq t$ було знайдено кілька пар:

1) $t = 0,154$ та $a = 0,494$;

2) $t = 0,579$ та $a = 0,104$;

За знайденими даними розраховуються вершини:

$P_1 = P(0.154) = (-15,6; 8,4; 10,8) + (31,2; 0; 0; 0) * 0.154 = (10,8; 8,4; 10,8)$;

$P_2 = P(0.579) = (-15,6; 8,4; -10,8) + (31,2; 0; 0; 0) * 0.579 = (2,45; 8,4; 10,8)$

Знайдені вершини утворюють нові відрізки з початковою та кінцевою вершинами тестового ребра. При цьому єдиним видимим відрізком є P_2P_k – нове видиме ребро.

На рисунку 2.11 товстою лінією виділено видиму частину ребра, а також грань, щодо якої змінюється видимість ребра.

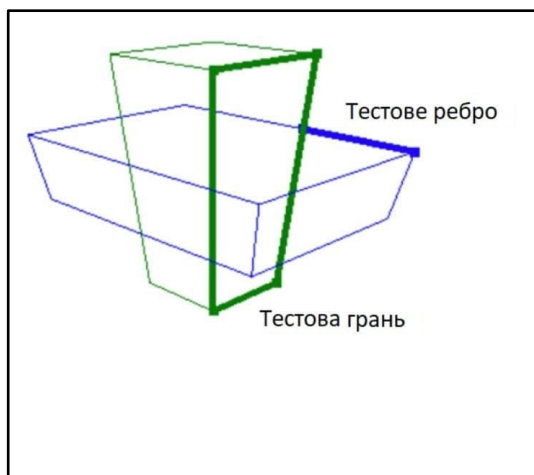


Рисунок 2.11 – Застосування алгоритму Робертса для тестового ребра

Після визначення частково видимих чи повністю невидимих відрізків (рисунок 2.12) визначають пари об'єктів, пов'язаних з ставленням протикання.

Для цього необхідно формувати рівняння $h_l=0$ і вирішити їх, об'єднуючи попарно і включивши в систему рівняння меж $t = 0$ та $t = 0$, а також включивши в систему рівняння кордону $a = 0$. Точки протикання запам'ятовуються.

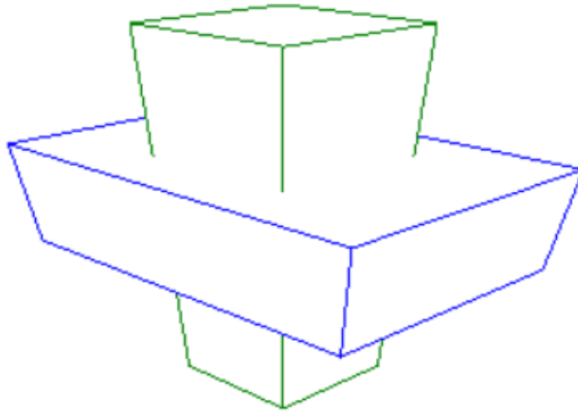


Рисунок 2.12 – Невидимі ребра видалені

Якщо точок протикання не виявлено, відбувається перехід до процедури візуалізації. В іншому випадку необхідно визначити видимі відрізки, що зв'язують точки протикання і потім сформувати всі можливі ребра, що з'єднують точки протикання, для пар тіл, пов'язаних ставленням протикання. Далі перевіряється екранування всіх ребер, що з'єднують, обома тілами, пов'язаними ставленням протикання. Зрештою виконується перехід до процедури візуалізації.

На рисунку 2.13 представлено ілюстрацію застосування алгоритму Робертса до всієї тестової сцени.

У ході тестування працездатності алгоритму оброблялася налагоджувальна інформація, що виводиться в консоль.

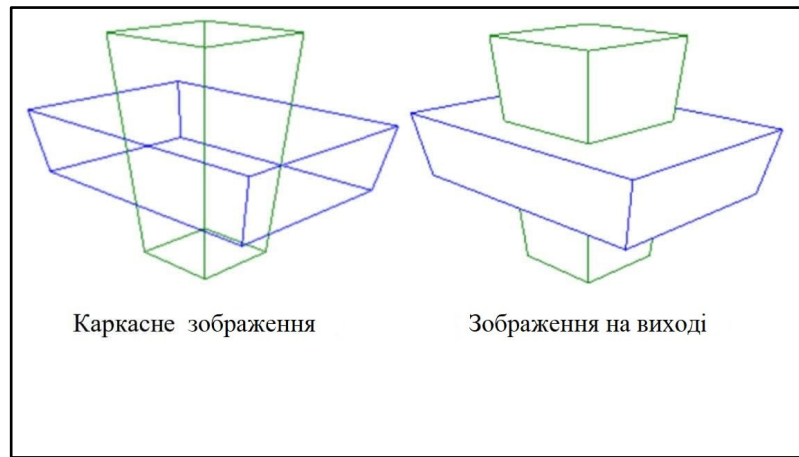


Рисунок 2.13 – Застосування алгоритму Робертса

Тимчасові витрати алгоритму пропорційні квадрату числа граней всіх об'єктів $O(n_2)$. Цей факт призвів до зниження інтересу до алгоритму Робертса. Однак математичні методи, що використовуються в цьому алгоритмі, прості, потужні та точні. З іншого боку, цей алгоритм можна використовувати ілюстрації деяких важливих концепцій.

2.4 Удосконалення ІТ-додатку, для використання даної функції

2.4.1 Архітектура сайту сканера

Діаграма розгортання (рис. 2.14) — діаграма в UML, на якій відображаються вузли, які обчислюються під час роботи програми, а також компоненти та об'єкти, що виконуються на цих вузлах.

Згідно діаграмі, яка приведена нижче наша система має п'ять реальних інстансів, на яких будуть запущені наступні складові системи [25].

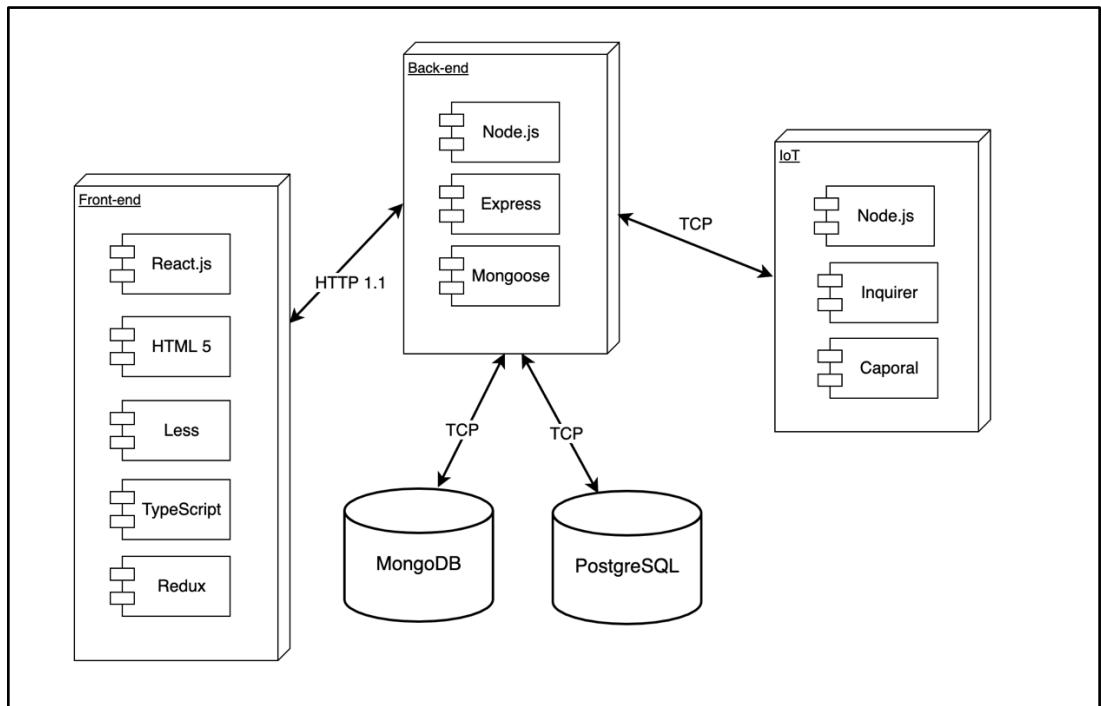


Рисунок 2.14 – Діаграма розгортання веб-додатку

Одна з складових нашої системи – це сервер. На стороні сервера обробляються головні функції програми. Саме на стороні back-end відбувається обробка всієї найважливішої бізнес-логіки проекту.

Друга частина програмної системи – це клієнтська сторона. На стороні клієнта розроблено візуальне відображення всієї системи. Спілкування серверної частини з клієнтської відбувається за допомогою HTTP 1.1. Цей протокол підходить для клієнт-серверної архітектури. Клієнтська програма формує запит і відправляє його на сервер, після чого серверне програмне забезпечення обробляє цей запит, формує відповідь і передає його назад клієнтові.

Також наш сервер використовуватиме технології Hangfire. Дана технологія дозволяє швидко, зручно та інтуїтивно зрозуміло працювати з чергами запитів. Між сервером та базою даних для цього сервера буде налагоджено зв'язок за допомогою протоколу TCP.

Найпопулярнішою сьогодні є дворівнева архітектура або клієнт-серверна архітектура, яка дає змогу масштабувати програмну систему вертикально – збільшувати обчислювальні можливості та обсяг пам'яті.

Зробивши аналіз сучасних архітектурних рішень, оцінивши складність рішення та час відведення на виконання роботи, були створені наступні архітектурні рішення.

Клієнтська частина буде являти собою веб-пристосування, що працює за клієнт-серверною архітектурою. Клієнтом буде виступати браузер, який за допомогою протоколу HTTP 1.1 відсилатиме запити, та отримуватиме відповідь у вигляді сторінок, що генеруються на стороні сервера сторінок. Серверна сторона містить у собі лише один головний сервер, який обробляє всі запити: які доходять до нього з клієнтської частини.

Архітектурний дизайн освітніх програм може відрізнитися один від одного. На сьогоднішній день можна вибрати два спільні підходи до розробки веб-додатків, а саме: Монолітна архітектура додатків, архітектура додатків Microservice. Для веб-додатку атестаційної роботи буде обрано монолітну архітектуру. У програмній інженерії монолітне застосування описує систему, де користувацький інтерфейс і код доступу до даних об'єднуються в єдину програму з однієї платформи.

Монолітне застосування використовує переваги простого багатокомпонентного зв'язку в рамках проекту. Всередині наступного користувацького інтерфейсу програми, бізнес-логіки, зберігання даних та виконання транзакцій виконуються всередині середовища, що в свою чергу знижує витрати на підтримку програми. Крім того, монолітні програми можуть бути горизонтально масштабовані з низькими витратами, що означає додавання нового примірника програми в задній частині балансувального пристрою. Крім цього, до моноліту легко додати нові функції, щоб побачити негайні результати через когерентність бази коду. Куля відображення використовує кулю бізнес логіки, кулю бізнес логіки використовує кулю доступу даних. Для виконання юніт-тестів існує окрема куля, яка має посилання на кулю бізнес логіки. Усі кулі можуть мати посилання на кулю сумісно використовуваної логіки, де є різні інтерфейси, сервіси та класи, що можуть бути корисними всюди.

3 ПОБУДОВА ПЛАНУ РОЗРОБКИ ПРОЕКТУ

Проект – це процес з чітко визначеними часовими рамками, мета якого створити унікальний продукт або отримати певні інноваційні результати. Відповідно, управління проектами – це конкретна діяльність, мета якої – реалізувати всі поставлені проектом завдання. Для цього прикладається максимальна кількість зусиль, знань, досвіду, методик та інструментарію.

Основу успішного впровадження будь-якої розробки становить задоволення потреб і очікувань зацікавлених сторін: користувачів послуг, замовників розробки, партнерів, інвесторів, працівників та ін. Управління проектами допомагає як найефективніше та за найбільш короткий час досягти виконання поставлених задач.

3.1. Розробка плану проекту інтернет магазину

Діаграма Ганта представляє собою відрізки, розташовані за горизонтальною шкалою часу, кожен з яких відповідає стадіям та етапам робіт. Початок, кінець і довжина відрізків на шкалі часу відповідає початку, кінцю і тривалості стадії або етапу. Етапи робіт і ресурси проекту у вигляді діаграми Ганта. Вертикальні лінії пов'язують етапи робіт і вказують на те, що наступна робота починається після завершення попередньої.

Якщо план проекту складається з великої кількості стадій та етапів робіт, неможливо детально проаналізувати весь план проекту. У таких ситуаціях потрібно проаналізувати ризики завдань які можуть стати критичними. Для цього буде використаний метод PERT, який за допомогою прогнозування песимістичних і оптимістичного часу виконання завдання, допоможе розрахувати очікувану тривалість операції. Метод PERT показаний у формулі (3.1).

$$PERT = (Pert_{\text{опт}} + Pert_{\text{пес}} + 4 * Pert_{\text{факт}}) / 6 \quad (3.1)$$

де, PERT – очікувана тривалість операцій; $Pert_{\text{опт}}$ – оптимістична тривалість операцій; $Pert_{\text{пес}}$ – песимістична тривалість операцій; $Pert_{\text{факт}}$ – фактична тривалість операцій.

Для даного проекту було спрогнозувати песимістичне і оптимістичне час виконання завдань проекту, обчислення наведені в формулі 3.2.

$$PERT = (30 + 35 + 33 * 4) / 6 = 32.83$$

Дані отримані з обчислення приблизно сходяться з часом на виконання проекту. Якщо протягом виконання проекту будуть виникати затримки часу на виконання роботи, тоді потрібно відразу додавати в програму Microsoft Project, і вносити зміни в план проекту.

Проектний менеджмент складається із ряду послідовних дій, а саме:

- визначення та формування вимог проекту;
- формування максимально чітких та зрозумілих цілей;
- встановлення та реалізація комунікації між задіяними у проекті сторонами;
- спілкування з командою, врахування їх потреб / побажань / очікувань и корекція існуючих планів відповідно до отриманих матеріалів.

Основні етапи побудова проекту складаються з:

- ініціація проекту - це стадія проекту, на якій виконується певний набір робіт для його успішного запуску. Зокрема, відбувається чітке визначення цілей и Завдання проекту, призначення керівника проекту, розробка статуту, ідентифікація учасників и зацікавлених осіб. У проект який був реалізований, в ініціативу проекту входять такі роботи, а саме, аналіз предметної області, постановка задачі, написання технічного завдання для всієї команди, і узгодження попереднього плану проекту. Ініціація проекту

становить 9 днів, що означає, поки не закінчиться ініціація проекту, не можна перейти на стадію планування робіт.

- планування (цей процес складається з трьох етапів: розробка загальних цілей, визначення конкретних цілей на даний період із послідуною їх деталізацією, визначення шляхів и способів досягнення цілей.);

- виконання и контроль (контроль за процесом досягнення поставленої цілей шляхом співставлення планових показників із фактичність та коригування цілей.) Цей етап слід чергувати з попереднім.

Ресурси проекту - це трудові, технічні та / або матеріальні одиниці, використовувані при виконанні завдань за проектом і забезпечують можливість реалізації всіх поставлених цілей проекту.

В даному проекті вказані трудові ресурси:

- менеджер проекту;
- seo-фахівець;
- верстальник;
- контент менеджер;
- програміст;
- тестувальник.

Трудові ресурси повинні виконати наступні роботи:

- аналіз методів задоволеності клієнтів. (Що дозволить визначитися з методами, які будуть використані під час реалізації проекту);
- аналіз моделей задоволеності клієнтів. (Що дозволить визначитися з моделями, які будуть проаналізовані під час реалізації проекту);
- розробка власної моделі задоволення клієнтів. (Після аналізу існуючих методів і моделей, буде розроблена нова модель або вдосконалена існуюча модель);
- впровадження створеної або вдосконаленої моделі задоволення клієнтів в сайт інтернет магазин;
- розробка структури сайту по створеної або удосконалення моделі

задоволеності клієнтів. (Як буде виглядає сайт для користувача, як має виглядати меню сайту, які категорії повинні бути на сайті, і тд. Дану роботу виконує Seo-спеціаліст);

- розробка макета сайту. (Розробка самого дизайну сайту, а саме верстку сайту. Дану роботу виконує верстальник);

- оптимізація швидкості шаблону сайту. (Після того, як шаблон інтернет магазину був створений, а саме візуальна частина, його потрібно оптимізувати, почистити від зайвого коду, для збільшення швидкості завантаження сайту, а саме візуальної частини. Дану роботу виконує верстальник);

- розробка шаблону сайту (Коли макет сайту створений, а саме його візуальна частина яка складається з HTML (HyperText Markup Language - «мова гіпертекстової розмітки»), CSS (Cascading Style Sheets), потрібно з'єднати шаблон з самої платформою Opencart на якій працює сайт , яка працює на мові програмування PHP (PHP: Hypertext Preprocessor), даною роботою займаються програміст і верстальник);

- збір ключових слів (В даному етапі проводиться аналіз ключових слів, запитів, які люди пишуть для пошуку того чи іншого товару або послуги яка клієнтів цікавить. Дану роботу проводить SEO- фахівець);

- тестування - етап тестування програмного продукту, сайту, наявності помилок, багів. Після виконання даного етапу при відсутності помилок або багів проект вважається завершеним і переходить на етап «Завершення проекту»;

- завершення проекту - передача замовнику опису проекту, протоколів тестування, звітів по проведеним перевіркам, заключний звіт по проекту і проектна документація, список відкритих питань та заключних робіт, дозвіл всіх спірних питань. Документування та аналіз досвіду виконання даного проекту.

	Название задачи	Длительность	Начало	Окончание	Названия ресурсов
0	Проект	33 дней?	Ср 02.09.20	Пт 16.10.20	
1	Ініціалізація	9 дней	Ср 02.09.20	Пт 14.09.20	Менеджер проекту
2	Аналіз предметної області	3 дней	Ср 02.09.20	Пт 04.09.20	Менеджер проекту
3	Постановка задачі	2 дней	Пн 07.09.20	Вт 08.09.20	Менеджер проекту
4	Написання ТЗ	3 дней	Ср 09.09.20	Пт 11.09.20	Менеджер проекту
5	Узгодження плану проекту	1 день	Пн 14.09.20	Пн 14.09.20	Менеджер проекту
6	Розробка	21 дней?	Вт 15.09.20	Вт 13.10.20	
7	Аналіз методів задоволення клієнтів	1 день	Вт 15.09.20	Вт 15.09.20	Менеджер проекту
8	Аналіз моделей задоволення клієнтів	2 дней	Ср 16.09.20	Чт 17.09.20	Менеджер проекту
9	Розробка власної моделі задоволення клієнтів	1 день?	Пт 18.09.20	Пт 18.09.20	Менеджер проекту
10	Впровадження моделі в сайт інтернет магазин	1 день?	Пн 21.09.20	Пн 21.09.20	Менеджер проекту
11	Розробка структури сайту	2 дней	Вт 22.09.20	Ср 23.09.20	Сео Спеціаліст
12	Розробка макету сайту	1 день?	Чт 24.09.20	Чт 24.09.20	Верстальник
13	Оптимізація шаблону на швидкість завантаження	1 день?	Пт 25.09.20	Пт 25.09.20	Верстальник
14	Розробка шаблону сайту	2 дней	Пн 28.09.20	Вт 29.09.20	Верстальник, Програміст
15	Збір ключових слів	2 дней	Ср 30.09.20	Чт 01.10.20	Сео Спеціаліст
16	Оптимізація сайту под пошукові системи	2 дней	Пт 02.10.20	Пн 05.10.20	Сео Спеціаліст, Контент менеджер
17	Заповнення сайту товарами	2 дней	Вт 08.10.20	Ср 07.10.20	Контент менеджер
18	Заповнення контенту сайту	3 дней	Чт 08.10.20	Пн 12.10.20	Контент менеджер
19	Оптимізація кошику оформлення заказу	1 день?	Вт 13.10.20	Вт 13.10.20	Програміст
20	Тестування	3 дней	Ср 14.10.20	Пт 16.10.20	Тестувальник
21	Тестування первинної реалізації	1 день	Ср 14.10.20	Ср 14.10.20	Тестувальник
22	Тестування кінцевої реалізації	2 дней	Чт 15.10.20	Пт 16.10.20	Тестувальник

Рисунок 3.1 - План проекту, розроблений у програмі MS Project

На Рисунку 3.1 наведено робочий простір програми MS Project у режимі відображення «Діаграма Ганта». Зображена діаграма Ганта яка відповідає всім роботам які повинні бути виконані за планом проекту.

На Діаграма Ганта, вказано час виконання проекту, а саме час початку виконання роботи, час закінчення виконуваної роботи приведення на рисунку 3.2, попередня завдання яка дотримується перед поточного завдання. З правій частині від кожної задачі яка виконується вказано відповідального за виконання даного завдання наведено на рисунку 3.3. Червоної стрілочкою, показаний критичний шлях виконання проекту.

Критичний шлях – це шлях моделі, тривалість якої дорівнює критичній.

Критична тривалість – це мінімальна тривалість, протягом якої може бути виконаний весь комплекс робіт проекту. Тобто виходячи з цього, критичний шлях є послідовністю робіт з нульовим резервом часу.

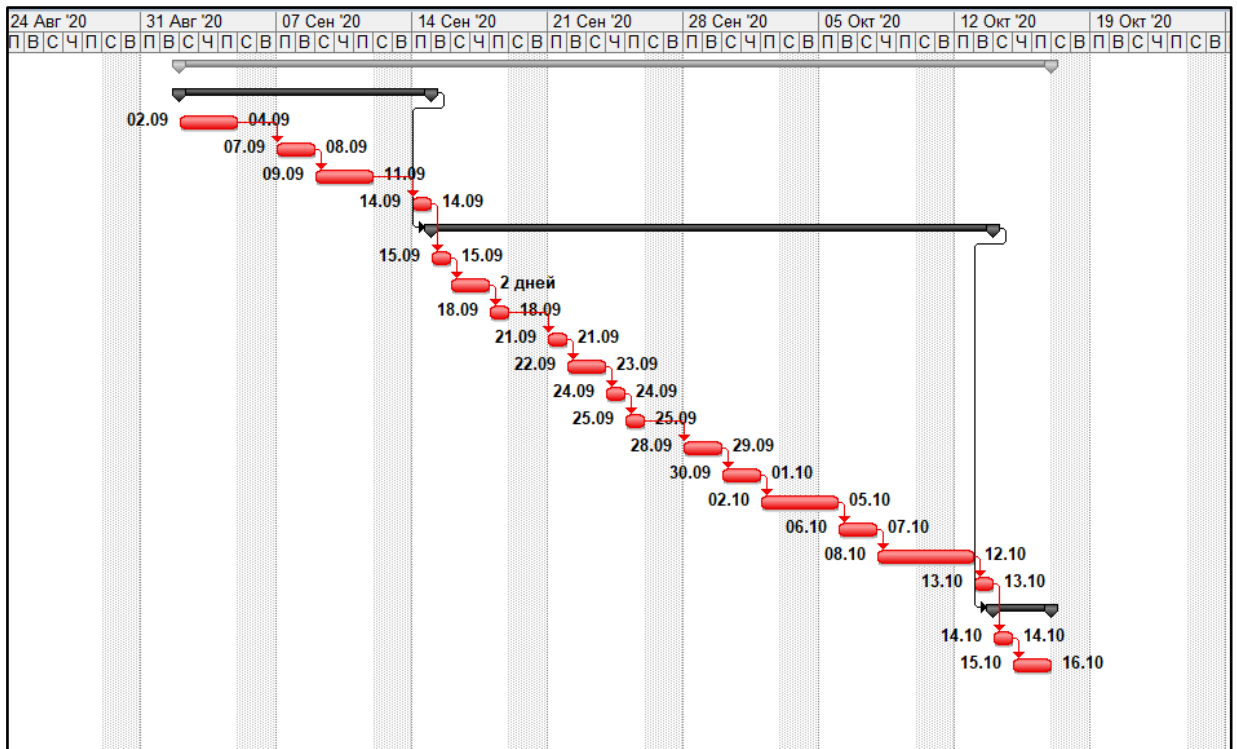


Рисунок 3.2 - Діаграма Ганта час початку і кінця робіт.

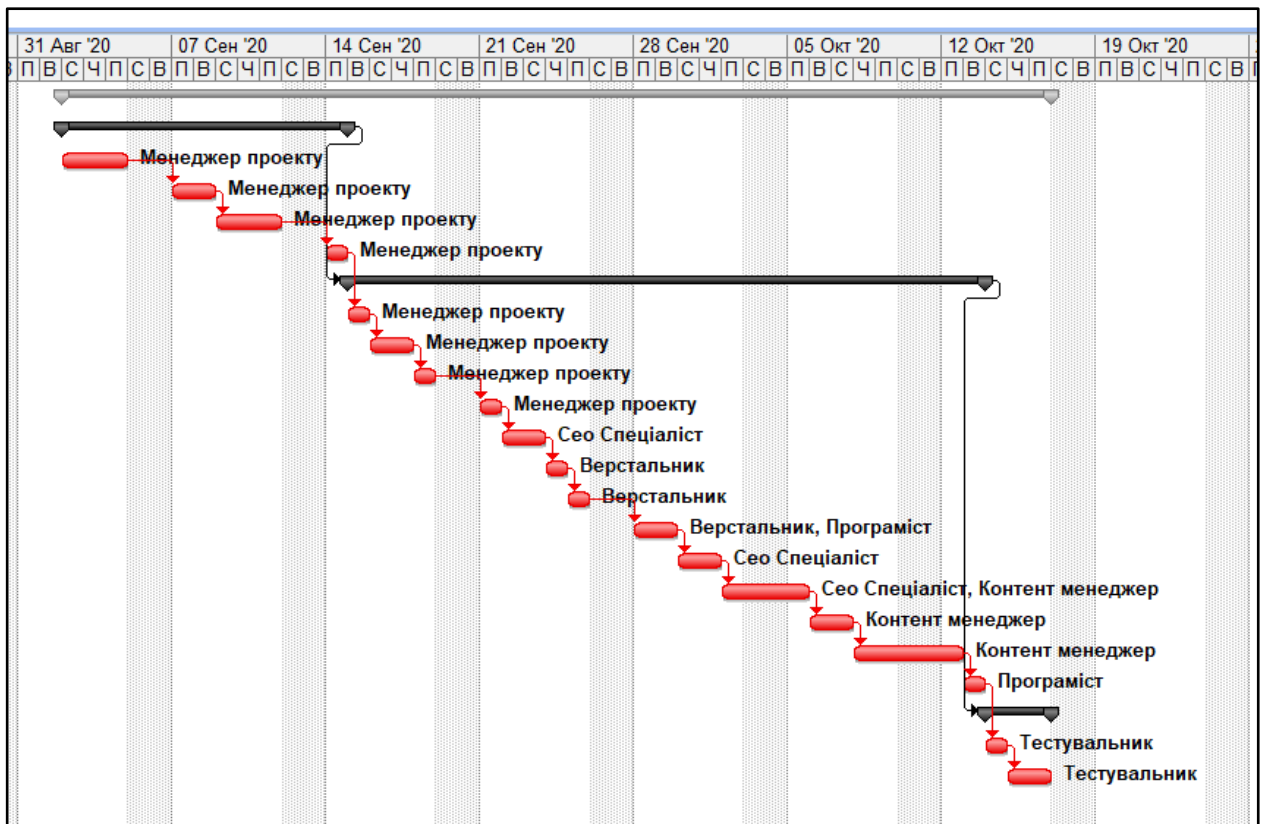


Рисунок 3.3 - Діаграма Ганта проекту, із зазначенням ресурсів

4 ПРАКТИЧНЕ ВИКОРИСТАННЯ ВДОСКОНАЛЕННОЇ СКАНУВАННЯ БАГАЖУ КЛІЄНТІВ

4.1 План практичної реалізації проекту

У даній роботі буде розроблений та вдосконалений метод сканування та алгоритмування, яка була впроваджена в веб-додатку:

- аналіз існуючих моделей та методів сканування;
- аналіз методів алгоритмування сканованих предметів;
- була розроблена і вдосконалена, на базі існуючих модель і методів, модель сканування по якій проводилися зміни в архітектурі веб-додатку;
- був розроблений план проекту;
- розроблено шаблон веб-додатку;
- була проведена настройка сайту під seo-оптимізацію;
- оптимізація швидкості завантаження веб-додатку.

4.2 Застосування вдосконаленої моделі сканування багажу.

В ході дослідження методів сканування була розроблена модель процесу сканування та відстеження багажу. Всі ці елементи пов'язані між собою. У даній моделі вказані такі параметри:

$M = \langle B, IK, SK, TC, TerC, M, HC_k, BB, VP, DO, CS, PX, AC \rangle$

- багаж користувача;
- історія користувача;
- статистика користувача;
- транспортна система;
- термінал сканування;
- модератор - людина, яка відповідає за цей сканер;
- № сканера;

- вага валізи;
- відсоток пошкодження після перевезення;
- детальний опис пошкодження після сканування ;
- час сканування;
- порівняльна характеристика до/після сканування;
- алгоритм сканування.

При розробці плану проекту, і розробці програмного продукту у вигляді веб-додатку були застосовані всі параметри удосконаленої моделі сканування багажу клієнтів.

Одними з перших факторів, які враховувалися, було зручність сайту, що мало на увазі, оптимізацію всього зовнішнього вигляду сайту під зручність користувачів. Також була проведена оптимізація швидкості завантаження сайту, що має позитивний ефект для користувачів зі слабкою швидкістю інтернету і для мобільних пристроїв. Також була проведена удосконалення архітектури сайту веб-додатку, а саме сторінка формування даних до/після сканування яка наведена на рисунку 4.1.

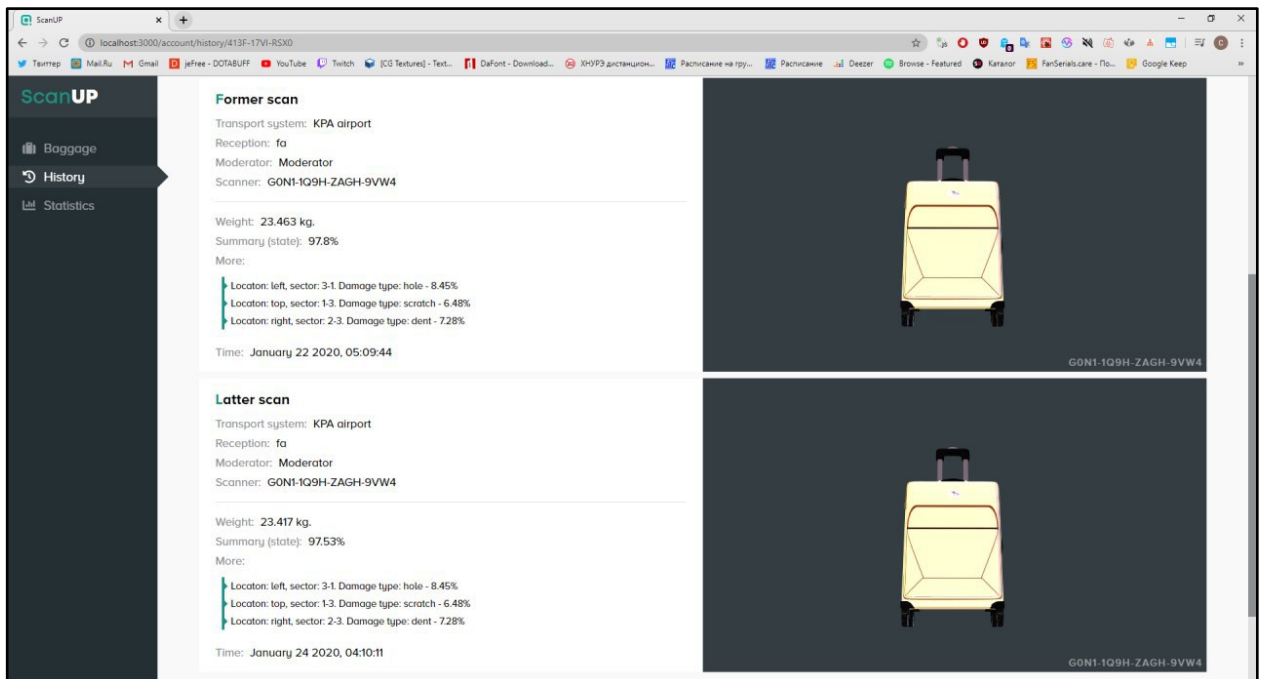
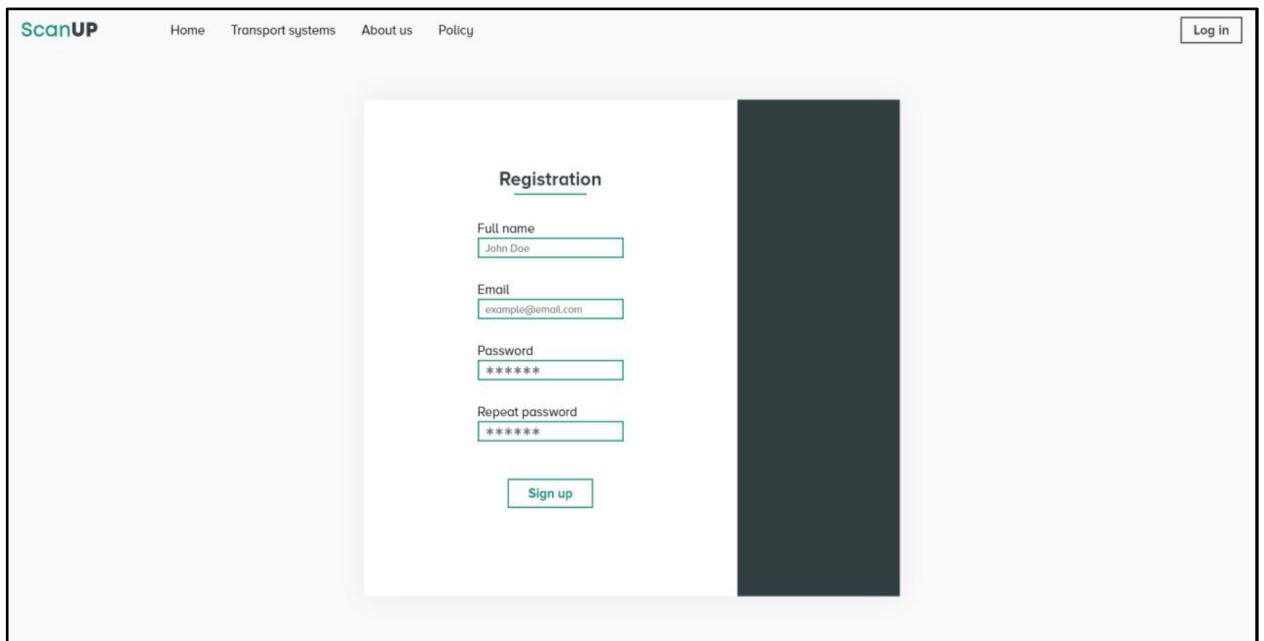


Рисунок 4.1 – Повне сканування багажу

Оскільки додаток проектувався з урахуванням досвіду типового користувача, система повинна мати приємний зовнішній вигляд та бути простою у використанні.

Для того, щоб зареєструватися до системи другим способом, необхідно зайти на сторінку реєстрації після першого сканування багажу. Вона повинна містити поле електронної адреси та паролю. Зображення сторінки реєстрації наведено на рисунку 4.2



The image shows a web browser window displaying the registration page of a system named 'ScanUP'. The page has a clean, modern design with a light gray background. At the top, there is a navigation bar with the 'ScanUP' logo on the left and links for 'Home', 'Transport systems', 'About us', and 'Policy' in the center. A 'Log in' button is located in the top right corner. The main content area is a white card with the title 'Registration' centered at the top. Below the title, there are four input fields: 'Full name' with the text 'John Doe', 'Email' with 'example@email.com', 'Password' with '*****', and 'Repeat password' with '*****'. A green 'Sign up' button is positioned below the 'Repeat password' field. A dark gray vertical bar is visible on the right side of the registration card.

Рисунок 4.2 – Сторінка реєстрації в системі

Головна сторінка повинна містити головну інформацію, що допоможе модераторам працювати швидше.

Головна сторінка в ролі модератора повинна містити головну інформацію, а саме:

- загальна кількість транспортних систем;
- загальна кількість багажів пасажирів;
- можливість друкувати мітки на багаж;
- можливість перегляду усього сканованого багажу;
- можливість відповідати на скарги від пасажирів.

Після першого сканування багажу пасажирові надається квиток та мітка, котра містить в собі ідентифікаційний номер багажу (рис 4.3).

Також, на клієнтській частині був використаний фреймворк JsGrid [5]. JsGrid – це легкий фреймворк для управління мережею даних на стороні клієнта на основі jQuery. Він підтримує основні операції з мережею, такі як вставка, фільтрація, редагування, видалення, підкачки та сортування. Даний фреймворк є гнучким і дозволяє настроїти його зовнішній вигляд і компоненти. Користувацький інтерфейс спроектовано на відображенні певних таблиць для додавання, перегляду, редагування та видалення даних.

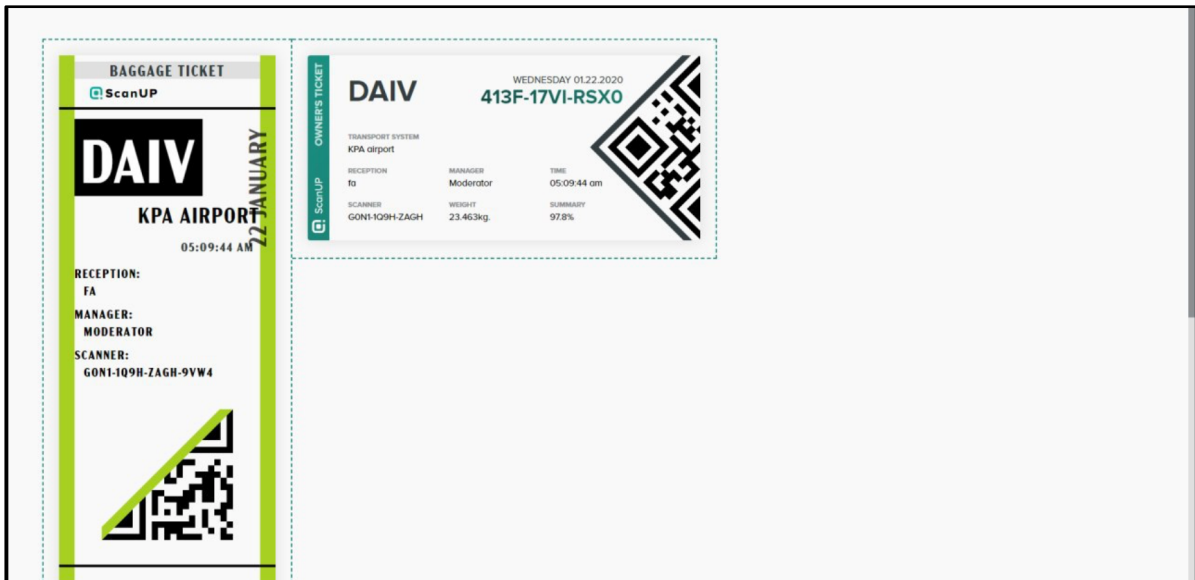


Рисунок 4.3 – Ідентифікаційний номер багажу

На рисунку 4.5 розглянемо сторінку процесу сканування багажу завдяки QR-коду. На цій сторінці ми зможемо побачити код завдяки якому можна відсканувати та побачити дані, які будуть доступні після всього сканування.

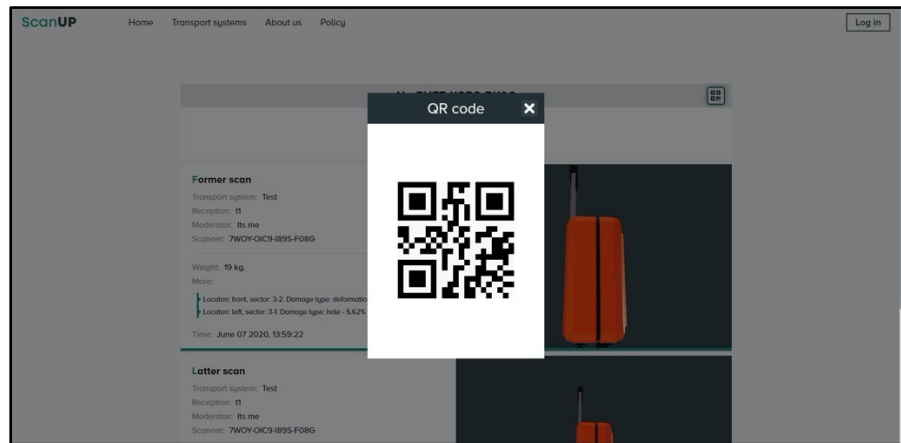


Рисунок 4.5 – QR-код для швидкого сканування багажу

Розробка зручного користувацького досвіду дуже важлива, оскільки адміністратор повинен зручно переглядати дані та без жодних зусиль редагувати їх. Також, користувацький інтерфейс не повинен бути перенасичений анімацією та різними кольорами.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було проведено дослідження моделей оптимізації сканування багажу, з цього дослідження було виявлено що потрібно вдосконалити алгоритм Робертса. В наслідок дослідження моделі сканування, була розроблена модель процесу сканування та відстеження багажу для чого були визначені та введені відповідні параметри спрямовані на підвищення оптимізації сканування багажу клієнтів. У рамках кваліфікаційної роботи була доведена актуальність дослідження, яка описує проблему веб-додатків, які вирішують питання щодо логістичних перевезень, а саме відсутність швидкої обробки сканувань, мала кількість сканерів, і не оптимізованої архітектури веб-додатку.

На основі цих досліджень була розроблена модель сканування та відстеження багажу клієнтів, на основі якої був розроблений план ІТ проекту, в якому вказані всі стадії для виконання та управління проектом, на базі даного плану був розроблений проект веб-додаток, який містить параметри сканування та сканування та обробки даних клієнта. Також була оптимізована робота сканерів для поліпшення якості і швидкості обробки даних пасажирів.

За результатами можна зробити висновок що, поставлена мета дослідження моделей оптимізації сканування ІТ-проекту була успішно виконана, і доведена наукова новизна.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки щодо розробки та оформлення магістерської атестаційної роботи за спеціальністю 122 – „Комп'ютерні науки” програма «Управління проектами в галузі інформаційних технологій» освітньокваліфікаційного рівня «магістр» / Упоряд.: Петров К.Е., Левикін В.М., Чалий С.Ф., Євланов М.В., Саєнко В.І., Міхнов Д.К., Міхнова А.В., Чала О.В. – Харків: ХНУРЕ, 2019. – 24 с. (дата звернення 04.11.2021)
2. Ярош, Е.С. Методы и средства представления графической информации: учебное пособие / Е.С. Ярош. – Челябинск: Издательский центр ЮУрГУ, 2017. – 230 с.
3. Алгоритм Робертса. - <http://compgraph.tpu.ru/roberts.htm>. Дата звернення: 01.11.2021.
4. Видалення невидимих ліній та поверхонь. - http://compgraph.tpu.ru/Del_hide_line.htm. Дата звернення: 15.02.2021.
5. Реалізація алгоритму Робертса. - <http://algotlist.ru/graphics/roberts.php>. Дата звернення: 01.11.2021.
6. Проблема пов'язаного surface removal. - http://learnwebgl.brown37.net/11_advanced_rendering/hidden_surface_removal.html. Дата звернення: 04.11.2021.
7. Operating System Market Share Worldwide. - <https://gs.statcounter.com/osmarket-share>. Дата звернення: 15.11.2021.
8. Roberts, L.G. Машина визнає три-dimensional solids. / L.G. Робертс. - Department of Electrical Engineering. Massachusetts Institute of Technology, 1963. - 82 с.
9. Seo-оптимізація. «Список загальних понять і термінів» [Електронний ресурс] // : [сайт]URL: <https://pixelplus.ru/samostoyatelno/stati/prodvizhenie-saytov/seo-terminy.html> (дата звернення 16.11.2021)

10. Релевантність сайту. «Визначення релевантності»[Електронний ресурс] // : [сайт]URL: <https://elama.ru/blog/chto-takoe-relevantnost-sayta/> (дата звернення 01.11.2021)
11. Appel, A. The Notion of Quantitative Invisibility and Machine rendering of Solids. / A. Appel. - Proceedings of ACM National Meeting, 1967. - 387 с.
12. Колдвелл Дж., 2001., Характеристики хорошого опитування задоволеності клієнтів. У JN Sheth, A. Parvatiyar & G. Shainesh, ред., «Управління відносинами з клієнтами», New Delhi, Tata McGraw-Hill, 2001., стор. 193-199. (дата звернення 05.11.2021)
13. Энджел Э. Интерактивная компьютерная графика. Вводный курс на базе OpenGL. Москва, 2006. – 248 с. (дата звернення 05.11.2021)
14. Robert C. Martin Clean Architecture: A Craftsman's Guide to Software Structure and Design. Prentice Hall, 2017. – 432 p. (дата звернення 06.11.2021)
15. Мартин Р. Чистый код: создание, анализ и рефакторинг. СПб: Питер, 2010. – 464 с. (дата звернення 08.10.2021)
16. White A. Major JavaScript Engines. JavaScript Programmer's Reference. 8-e 2nd Edition. Packt Publishing, 2017. – 246 с. (дата звернення 09.10.2021)
17. Сухов К. Node.js. Путеводитель по технологии. СПб: Питер, 2016. – 416 с. (дата звернення 11.11.2021)
18. Kyle Mew Learning Material Design. Packt Publishing, 2015. – 186 p. (дата звернення 11.11.2021)
19. Демінг, WE, 1981. Управління статистичними методами для забезпечення якості та продуктивності. Нью-Йоркський університет, «Вища школа бізнесу», Нью-Йорк. (дата звернення 12.11.2021)
20. Шведський барометр лояльності клієнтів (SCSB) [Електронний ресурс] //: [сайт] URL: <https://www.van-haaften.nl/customer-satisfaction/customer-satisfaction-models/64-the-swedish-customer-loyalty-barometer-scsb> (дата звернення 16.11.2021)

21. Ерланга В, С [Електронний ресурс] //: [сайт] URL: <https://www.callcentrehelper.com/erlang-c-formula-example-121281.htm> (дата звернення 19.11.2021)

22. Ерланга С [Електронний ресурс] //: [сайт] URL: <https://www.callcentrehelper.com/erlang-c-formula-example-121281.htm> (дата звернення 19.11.2021)

23. Ерланга Х [Електронний ресурс] //: [сайт] URL: <https://www.apexberg.ru/kb/articles/znakomstvo-s-erlang-x/> (дата звернення 29.11.2021)

24. Сучасний движок інтернет магазину [Електронний ресурс] // : [сайт] . URL: <http://opencart-russia.ru/> (дата звернення 23.11.2021)

25. Створюйте швидкі, чуйні сайти за допомогою Bootstrap [Електронний ресурс] // : [сайт] . URL: <https://getbootstrap.com/> (дата звернення 16.10.2021)