МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
УМАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет економіки і підприємництва

Кафедра інформаційних технологій

Допущено до захисту

"_____" _____ 2025 р.

Зав. кафедри, к.т.н., доцент

_____ **Ліщук Р.І.**

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Розробка додатку для створення об’ємних зображень

Виконав - студент 4 курсу, групи 41-кн
освітньої програми «Комп’ютерні науки»
спеціальності 122 Комп’ютерні науки
Нестеренко Ростислав Костянтинович
Науковий керівник - д.т.н., професор
Кулаков Павло Ігорович

Умань - 2025

Зміст

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ……..... 7
ВСТУП…… 8
РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ…… 10
1.1. Об’єкт та предмет дослідження…… 10
1.2 Порівняльний аналіз аналогів розробки…….. 15
1.3. Постановка задачі..... 21
РОЗДІЛ 2. ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ ДОДАТКУ ТА РОЗРОБКА АЛГОРИТМІВ……...…… 22
2.1. Розвиток та модифікація методу ray-tracing ……. 22
2.2. Аналіз і обґрунтування вибору засобів для реалізації програмного забезпечення ……... 26
2.3. Модель та алгоритм роботи додатку…… 30
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНИХ ДОДАТКУ ДЛЯ НАДАННЯ ОБ’ЄМНОСТІ ЗОБРАЖЕННЯМ…….. 34
3.1 Розробка модуля надання об’ємності зображенням…… 34
3.2 Розробка модуля завантаження обробленого зображення…….. 36
3.3 Розробка інтерфейсу програмного додатку…… 41
ВИСНОВОК…… 46
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ…… 47
ДОДАТОК А. Лістинг програмного коду……..……... 49

Вступ

Об'ємні зображення можуть створювати вражаючі візуальні ефекти, які привертають увагу глядачів та дозволяють зображенням виходити за межі звичайних плоских площин. Створення об'ємних ефектів на зображеннях є складним процесом, який вимагає високотехнологічних рішень та творчого підходу. Технології, які дозволяють надавати об'ємності зображенням, постійно розвиваються і удосконалюються, що відкриває нові можливості для використання їх у різних галузях. У сфері мистецтва об'ємні зображення можуть бути використані для створення унікальних творів, які вражають своєю красою та оригінальністю. Також важливою галуззю застосування об'ємних зображень є наука та освіта. Вони можуть бути використані для створення інтерактивних навчальних матеріалів, які допомагають студентам краще розуміти складні концепції та явища. Крім того, об'ємні зображення можуть бути використані у медицині для візуалізації медичних даних та діагностики хвороб.

У цьому контексті розробка мобільного застосунку для надання об'ємності 2D зображенням є важливим завданням, яке може принести користь у багатьох сферах життя. Такий застосунок може стати корисним інструментом для творців контенту, митців, дизайнерів, викладачів та багатьох інших людей, які цінують креативність та новаторство. З розвитком мобільних пристроїв та зростаючим попитом на інноваційні технології популярною тенденцією останніх років стала розробка мобільних систем з для редагування та надання об’ємності зображенням. Це зумовило потребу в програмних модулях, які можуть бути використані для реалізації таких систем. Ці програмні модулі можуть допомогти розробникам створювати мобільні системи, за допомогою яких можна в зручному доступі обробляти зображення.

Актуальність

У сучасному світі тривимірні зображення стали невід'ємною частиною щоденного життя. Технології, які дозволяють надавати об'ємності зображенням, є дуже популярними та використовуються в різних сферах, від розваг та

мистецтва до науки та бізнесу. Актуальність теми кваліфікаційної роботи бакалавра зумовлена тим, що розроблений додаток для створення об'ємних зображень може бути корисним інструментом для створювачів контенту, митців, дизайнерів, викладачів та багатьох інших людей. Аналогічні мобільні застосунки для обробки зображень не надають повний пакет інструментів для надання об'ємності зображенням.

Об'єкт дослідження - процес надання об'ємності зображенням у системах комп'ютерної графіки.

Предмет дослідження - методи та засоби надання об'ємності зображенням.

Мета дослідження

Метою та завданням кваліфікаційної роботи бакалавра є підвищення продуктивності формування тривимірних зображенням за рахунок розробки нових методів, використання сучасних бібліотек. У кваліфікаційній роботі бакалавра необхідно виконати такі завдання:

- провести аналіз існуючих методів і засобів надання об'ємності зображенням;

- запропонувати модифікований метод ray-tracing для рахування відблисків на

поверхні моделі;

- запропонувати метод розбиття поверхні моделі на трикутники для ідентифікації відблисків;

- розробити алгоритм надання об'ємності зображенням;

- розробити зручний та адаптивний графічний інтерфейс мобільної системи, який допомагатиме користувачу зручно обробляти зображення.

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Об’єкт та предмет дослідження

3D-візуалізація є комплексним процесом, спрямованим на створення віртуальних зображень об'єктів та сцен у тривимірному просторі [1, 2]. Цей метод графічного подання дозволяє користувачам сприймати об'єкти так, як вони виглядають у реальному світі. У сучасному контексті, 3D-візуалізація є невід'ємною частиною багатьох галузей, включаючи відеоігри, кіноіндустрію, дизайн продуктів, архітектуру та багато інших. Вона використовується для створення реалістичних та деталізованих зображень та анімації, що дозволяє створювати вражаючі візуальні ефекти та віртуальні світи. 3D-візуалізація здійснюється за допомогою комп'ютерної графіки та високотехнологічних програмних засобів. Вона включає в себе такі процеси, як моделювання, текстурування, освітлення, анімацію та рендеринг. Моделювання включає створення тривимірних об'єктів і сцен, текстурування – додавання текстур до цих об'єктів для надання їм реалістичного вигляду, освітлення - налаштування освітлення в сцені для створення правдоподібних тіней та відблисків, анімація - створення рухомих об'єктів та сцен, а рендеринг - перетворення тривимірної сцени у двовимірне зображення. У сфері 3D-візуалізації присутній комплекс специфічних термінів і понять, які визначають елементарні аспекти цього технічного дисциплінарного підходу. До них відносяться терміни, такі як модель, текстура, освітлення, анімація, рендеринг, а також поняття, такі як тріангуляція, затінення, камера, матеріал, сцена та багато інших. Загалом, 3D-візуалізація відкриває множину можливостей для творчості у різних галузях, дозволяючи створювати надзвичайно реалістичні та вражаючі візуальні ефекти.

Моделювання (Modeling): Це ключовий етап, який передбачає створення тривимірних об'єктів з урахуванням їх геометричної форми та взаємодії між ними [3]. Моделювання включає в себе використання математичних алгоритмів для визначення координат точок та їх взаємного розташування в просторі.

Текстурування (Texturing) [4, 5]: Термін, що визначає процес накладання текстур на поверхні моделей з метою надання їм візуального реалізму. Текстури надають об'єктам характер, враховуючи деталізацію та колірну інформацію.

Освітлення (Lighting) [6]: Це поняття об'єднує методи взаємодії світла та поверхонь об'єктів. Освітлення впливає на сприйняття форми об'єктів, забезпечуючи змогу реалістично відображати тіні, відбиття та інші оптичні ефекти.

Рендеринг (Rendering) [7]: Рендеринг є процесом генерації двовимірного зображення з тривимірної сцени за допомогою врахування всіх характеристик об'єктів та середовища, таких як кольори, тіні, освітлення та перспектива. Ці концепції становлять фундаментальний ланцюг в 3D-графіці, об'єднуючи технічні аспекти та творчий підхід для досягнення високого рівня візуальної реалістичності у віртуальних просторах. Інтегрування цих термінів у процес 3D-візуалізації визначає та розширює високотехнологічний характер цієї галузі, забезпечуючи можливість адекватного відтворення об'єктів та сцен у тривимірному просторі. У галузі 3D-візуалізації відбулися революційні досягнення, що дозволили значно покращити реалістичність та ефективність створення тривимірних зображень. Ці досягнення зазначено у період від 1980-х до 2000-х років і визначили нові стандарти в індустрії комп'ютерної графіки та 3D-візуалізації.

Рей-трейсинг. Введення рей-трейсингу в графічні обчислення представляє собою значний прорив. Рей-трейсинг дозволяє симулювати поведінку світла в найреалістичніший спосіб, що в результаті призвело до створення фото-реалістичних зображень. Він є важливим алгоритмом у галузі комп'ютерної графіки, який дозволяє створювати фото-реалістичні зображення, моделюючи поведінку світла та взаємодію з об'єктами в сцені. Основні концепції рей-рейсингу включають в себе відбиття світла, тіні, відблиски та розсіювання світла. Основні аспекти рей-трейсингу:

- Відбиття світла [8] - природнім чином об'єкти відбивають світло. Алгоритм рей-трейсингу включає в себе відправлення променів від камери до сцени, а коли промінь досягає поверхні об'єкта, відбувається відбиття, і розраховується колір відбитого світла (рисунок 1.1).

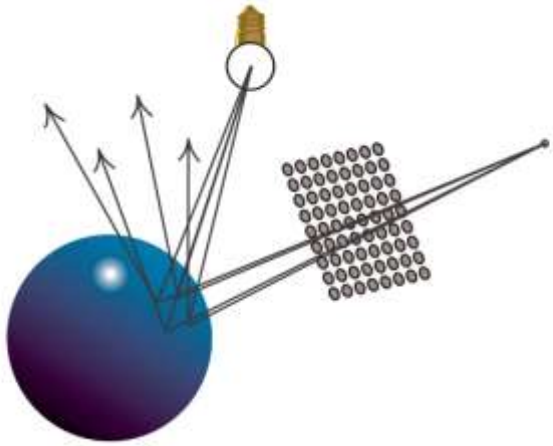


Рисунок 1.1 - Відбивання променів світла від поверхні кулі

- Променева тінь [9] - рей-трейсинг дозволяє визначити, чи освітлена певна точка сцени чи вона перебуває в тіні. Це досягається шляхом висилання додаткових променів від точок освітлення до точок на поверхні об'єктів (рис. 1.2).

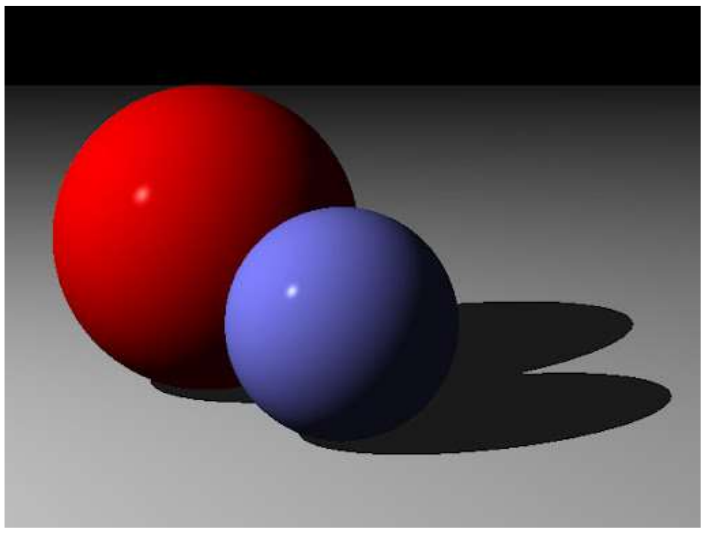


Рисунок 1.2 - Утворення тіней при потраплянні променя світла

- Відблиски [10] - рей-трейсинг враховує відблиски, які виникають при відбитті світла від гладких поверхонь. Це додає до зображення реалістичність, оскільки об'єкти можуть відображати оточуючі об'єкти (рисунок 1.3).



Рисунок 1.3 - Відблиски, що відображають оточуючі об'єкти

- Розсіювання світла [11] - цей аспект враховує розсіювання світла, коли світло розсіюється в різних напрямках при взаємодії з матеріалом. Це особливо важливо для матеріалів з матовою поверхнею (рисунок 1.4).

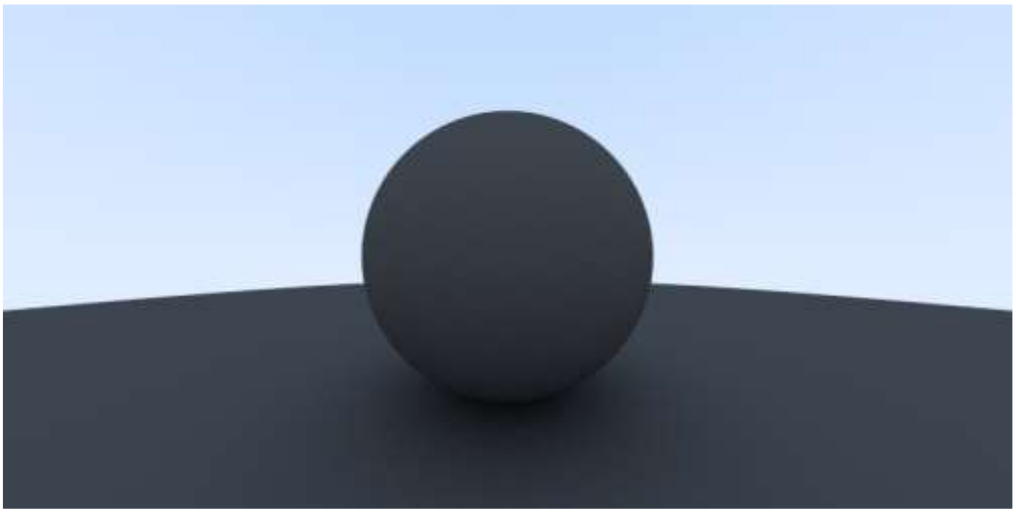


Рисунок 1.4 - Розсіювання світла на матовій поверхні

Рей-трейсинг є потужною технікою рендерингу, яка дозволяє створювати зображення з високим рівнем реалізму. Використання цієї техніки дає змогу моделювати складні ефекти освітлення, тіней та відображень, що робить її незамінною у візуалізації 3D сцен і створенні комп'ютерної графіки високої якості. Однак, через високі обчислювальні витрати, рей-трейсинг має високи вимоги до обладнання і часу, що обмежує його застосування у реальному часі для багатьох додатків. Мобільні застосунки для надання об'ємності зображенням є досить популярними і мають широкий спектр застосувань у різних галузях, включаючи рекламу, маркетинг, мистецтво, освіту та розваги. Деякі з цих застосунків надають користувачам можливість створювати об'ємні ефекти на зображеннях за допомогою доповненої реальності (AR), технологій віртуальної реальності (VR) та інших інноваційних методів [12]. Наприклад, деякі мобільні застосунки дозволяють користувачам створювати 3D об'єкти на основі зображень або накладати віртуальні об'єкти на реальний світ за допомогою AR. Це дозволяє створювати цікаві та захоплюючі візуальні ефекти, які можна використовувати у маркетингових кампаніях, рекламних матеріалах або для творчих проєктів. Деякі застосунки для обробки та надання об'ємності зображенням надають користувачам широкий спектр інструментів для редагування та покращення зображень. Використання цих інструментів дозволяє створювати вражаючі та професійні зображення без необхідності великих знань у галузі фотографії та дизайну. Однак, деякі мобільні застосунки можуть мати обмежені можливості або бути нестабільними через високі вимоги до обладнання та ресурсів пристрою. Також, вартість деяких застосунків та їх доступність для різних платформ може бути проблемою для деяких користувачів. У цілому, мобільні застосунки для надання об'ємності зображенням є важливим інструментом для творчих та професійних користувачів, які шукають шляхи для створення унікальних та вражаючих візуальних ефектів. Отже, розробка програмних модулів для впровадження

мобільного застосунку для надання об’ємності зображенням є критично важливим завданням у сучасному світі.

1.2 Порівняльний аналіз аналогів розробки

Використання ефектів для надання об'ємності зображенням є важливим етапом у редагуванні фотографій та створенні вражаючих візуальних ефектів. Ці ефекти дозволяють створювати ілюзію глибини та простору на плоскому зображенні, що додає йому реалістичності та емоційного зворушення. Використання ефектів надання об'ємності є особливо актуальним у сучасному світі, де візуальні зображення грають важливу роль у спілкуванні та враженні аудиторії. Такі ефекти дозволяють створювати унікальні та запам'ятовувані зображення, які виокремлюються серед інших та залишають глибоке враження на глядачів. Існує велика кількість програмних застосунків для надання об’ємності зображенням. Ось деякі з них:

- Photoshop Express;
- Snapseed;
- Pixlr.

Одним із прикладів обробки зображень в мобільній системі є «Photoshop Express» [13] (рис. 1.5).

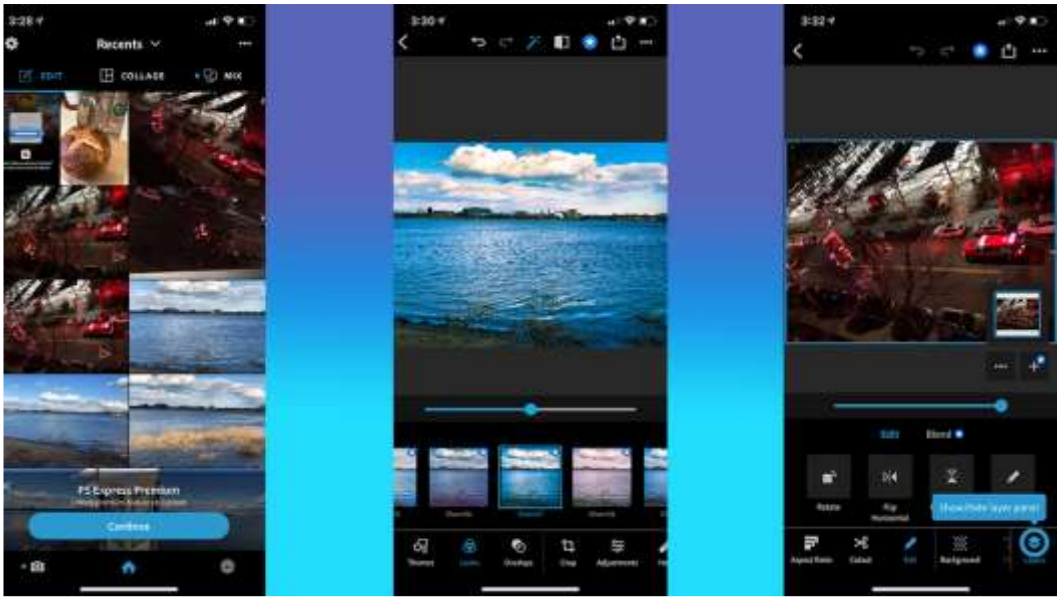


Рисунок 1.5 - Приклад роботи «Photoshop Express»

Це мобільний застосунок для редагування фотографій, розроблений компанією Adobe. Він надає користувачам широкий спектр інструментів для покращення та творчого редагування зображень простим і доступним способом. Завдяки своїй потужній функціональності та легкому використанню, «Photoshop Express» є популярним вибором серед користувачів, які шукають зручний спосіб редагувати свої фотографії на мобільних пристроях.

Інший приклад – додаток «Snapseed» [14] (рис. 1.6). Це мобільний застосунок для редагування фотографій, розроблений компанією Google. Застосунок надає широкий спектр інструментів для редагування зображень, включаючи різноманітні фільтри, ефекти, корекцію кольору та інші можливості. «Snapseed» відзначається простим та зручним інтерфейсом, що робить його популярним вибором серед користувачів, які шукають потужний інструмент для редагування фотографій на своїх мобільних пристроях.

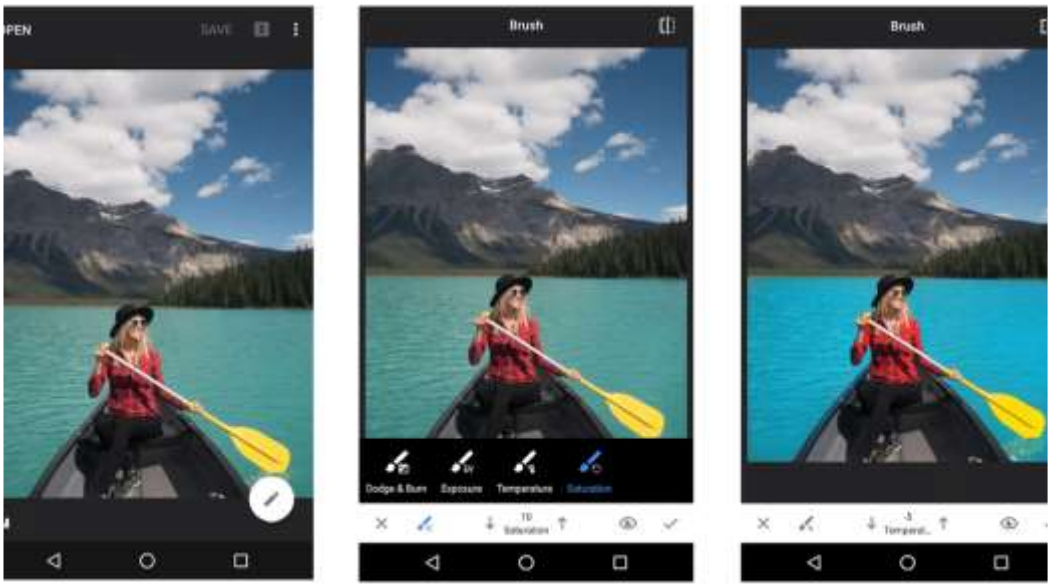


Рисунок 1.6 - Приклад роботи додатку «Snapseed»

Існує ще один приклад – додаток «Pixlr» [15] (рисунок 1.7). Це онлайн сервіс для редагування та обробки зображень, який також має мобільну версію для смартфонів і планшетів. Він надає користувачам можливість редагувати фотографії безпосередньо у веб-браузері або на мобільному пристрої, не

потребуючи встановлення спеціального програмного забезпечення. «Pixlr» має великий набір інструментів для редагування, включаючи фільтри, ефекти, інструменти для ретуші та корекції кольору, що робить його популярним серед користувачів, які шукають зручний спосіб обробки зображень.



Рисунок 1.7 - Приклад роботи додатку «Pixlr»

Розробка програмних модулів для впровадження мобільного застосунку для надання об’ємності зображенням вимагає досвіду обробці зображень, розробки програмного забезпечення та розробки користувацького інтерфейсу для мобільних додатків. Це передбачає створення власного функціоналу та інтегрування бібліотек для надання ефектів об’ємності. Результати порівняння аналогів зведено в таблиці 1.1. Відповідно до таблиці порівняльних характеристик розробка власного застосунку для надання об’ємності зображенням. Отриманий продукт зможе покрити недоліки існуючих рішень та забезпечити новий досвід та можливості застосування ефектів для надання об’ємності зображенням.

Таблиця 1.1 - Порівняльні характеристика онлайн платформ

Критерії	Photoshop Express	Snapseed	Pixlr	Модуль, що розробляється
AR-ефекти та фільтри	+	+	+	+
Зручний інтерфейс для обробки	+	-	+	+
Достатній спектр інструментів для обробки	-	+	-	+
Сумісність з декількома пристроями	-	-	+	+
Наявність мобільного додатку	+	+	+	+
Загальна оцінка	60%	60%	80%	100%

Отже, розробка програмних модулів для впровадження мобільного застосунку для надання об’ємності зображенням відкриває перед бізнесом широкі можливості для взаємодії зі своїми клієнтами, створення унікального контенту, а також підвищення рівня залученості та утримання користувачів.

Розробка програмних модулів для реалізації мобільного застосунку для надання об’ємності зображенням вимагає ретельного розгляду найкращих підходів до вирішення проблеми. Аналіз методів розв'язання поставленої задачі, що полягає у наданні об'ємності зображенням за допомогою мобільного застосунку, включає огляд існуючих підходів та технологій, які можуть бути використані для досягнення цієї мети. Основні методи включають використання

технологій доповненої та віртуальної реальності, обробку зображень, застосування спеціальних ефектів та фільтрів.

Перший метод, що варто розглянути – це використання технології доповненої реальності (AR). AR дозволяє накладати віртуальні об'єкти на реальний світ через камеру мобільного пристрою. За допомогою AR можна створювати об'ємні ефекти на 2D зображеннях, що додає їм реалістичності та привабливості. Наприклад, користувач може розгорнути 2D зображення книги у 3D модель, яка буде виглядати як справжня книга на столі. AR також може бути використаний для створення інтерактивних маркерів на зображеннях, що дозволить користувачам взаємодіяти з ними за допомогою своїх мобільних пристроїв.

Другий метод - використання технології віртуальної реальності (VR). VR дозволяє користувачам зануритися в інші виміри та взаємодіяти з об'ємними об'єктами у віртуальному середовищі. Хоча цей підхід більш складний у реалізації та може вимагати спеціального обладнання, він може забезпечити більш глибокий та іммерсивний досвід для користувачів. Наприклад, можна створити додаток, який дозволяє користувачам переглядати та взаємодіяти з об'ємними моделями архітектурних споруд у віртуальному просторі. VR також може бути використаний для створення тренувальних симуляторів для різних професій, де реалістична візуалізація грає важливу роль.

Третій метод - обробка зображень та застосування спеціальних ефектів та фільтрів. Цей підхід включає в себе використання алгоритмів обробки зображень для створення ефектів глибини, тіней та освітлення, що додають об'ємності зображенням. Наприклад, можна реалізувати функцію додавання ефекту "паралаксу" до зображень, щоб вони виглядали більш об'ємними та динамічними. Також можна використовувати різноманітні фільтри та ефекти для створення вражаючих візуальних елементів.

Кожен з цих методів має свої переваги та обмеження. Вибір конкретного методу може залежати від конкретних потреб та можливостей програмного

застосунку. Розглядаючи ці методи, можна знайти оптимальний шлях для реалізації задачі та створення мобільного застосунку, що надає об'ємність зображенням. Додавши об'ємність зображенням за допомогою технологій доповненої та віртуальної реальності (AR та VR), можна створити унікальний досвід для користувачів. Ці технології дозволяють створювати об'ємні ефекти, які додають реалістичності та привабливості зображенням. AR може бути особливо ефективним для створення об'ємних ефектів на 2D зображеннях. Наприклад, можна створити додаток, який дозволяє користувачам переглядати інтерактивні об'єкти на зображеннях у реальному часі через камеру свого смартфона. Це може бути корисно для віртуального огляду продуктів або мистецьких творів перед покупкою. VR, з іншого боку, може забезпечити більш глибокий та цікавий досвід, де користувачі можуть взаємодіяти з об'ємними об'єктами у віртуальному середовищі. Це може бути особливо ефективно для тренувальних симуляторів або віртуального туризму, де люди можуть досліджувати нові місця та оточення. Обробка зображень та застосування спеціальних ефектів та фільтрів також можуть додати об'ємності зображенням. Використання цих методів може бути ефективним для створення цікавих та креативних ефектів, які роблять зображення більш привабливими та цікавими для глядачів. Розглянувши переваги та недоліки кожного методу, було вирішено використати метод розробки власного мобільного застосунку для надання об’ємності зображенням та подальшій обробці з використанням спеціальних ефектів та фільтрів для покращення вигляду зображення. Такий підхід дозволить отримати повноцінний програмний продукт без необхідності використання потужного апаратного забезпечення.

Photoshop Express від Adobe є потужним інструментом з багатьма функціями редагування, але може бути складним у використанні для початківців. Snapseed від Google має простий інтерфейс та багато інструментів для редагування, а Pixlr надає широкий спектр фільтрів та ефектів. Однак жоден з цих додатків не надає повного спектру можливостей для надання об'ємності

зображенням, що підтверджує актуальність розробки власного програмного рішення. У результаті порівняння було виявлено, що досліджувані платформи набули значної популярності завдяки своїй зручності та доступності. Проаналізувавши переваги та недоліки відомих платформ, було виявлено, що вони не надають достатній спектр інструментів для надання об’ємності зображенням. Таким чином було доведено доцільність розробки власного програмного рішення. На основі отриманої інформації було сформовано перелік задач, які необхідно виконати для розробки власного мобільного застосунку для надання об’ємності зображенням.

1.3 Постановка задачі

Метою кваліфікаційної роботи бакалавра є розробка додатку для підвищення продуктивності формування тривимірних зображенням за рахунок розробки нових методів, використання сучасних бібліотек. У кваліфікаційній роботі бакалавра необхідно провести аналіз існуючих методів і засобів надання об’ємності зображенням, запропонувати модифікований метод ray-tracing для рахування відблисків на поверхні моделі, запропонувати метод розбиття поверхні моделі на трикутники для ідентифікації відблисків, розробити алгоритм надання об’ємності зображенням, розробити зручний та адаптивний графічний інтерфейс мобільної системи, який допомагатиме користувачу зручно обробляти зображення.

Завданням розробки є аналіз предметної області та проведення порівняльного аналізу аналогічних додатків та систем створення об’ємних зображень, обґрунтування вибору технологій для розробки додатку, розробка загальної структури додатку та розробка програмного забзпечення.

РОЗДІЛ 2. ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ ДОДАТКУ ТА РОЗРОБКА АЛГОРИТМІВ

2.1 Розвиток та модифікація методу ray-tracing

Рендеринг фото-реалістичних зображень є критично важливим завданням у галузі комп'ютерної графіки. Один з найефективніших методів досягнення високої якості зображень - це метод відстеження променів, відомий як ray-tracing. Цей метод базується на симуляції реального світла, включаючи його відбиття та розсіяння на поверхнях об'єктів.

Ray-tracing використовується для моделювання того, як світло взаємодіє з об'єктами в сцені. При його використанні для кожного пікселя на зображенні обчислюється шлях променя світла від джерела світла до камери через сцену.

Цей процес включає обчислення відбиття світла від поверхонь об'єктів, урахування тіней, розсіяння та інших фізичних ефектів, що спричиняють реалістичний вигляд зображення [16].

Однією з важливих переваг відстеження променів є його здатність до точного відтворення складних ефектів світла, таких як відбиття, переломлення та розсіяння. Це дозволяє отримати зображення з високою ступенем реалізму, що є особливо важливим у візуалізації архітектурних проєктів, створенні спеціальних ефектів у кіно та відеоіграх.

Наприклад, при моделюванні скляних об'єктів, ray-tracing дозволяє відтворити їхні відбиття та переломлення, що додає зображенню реалізму та деталізації. Такий підхід використовується у великій кількості візуальних ефектів, що ми бачимо у фільмах та відеоіграх, і дозволяє створити вражаючі, фото-реалістичні сцени.

У традиційному методі ray-tracing для кожного променя, що випускається з камери, розраховується його шлях у просторі, відбиття та кольори, які він зустрічає. Однак, цей метод вимагає великої обчислювальної потужності, особливо при рендерингу складних сцен з великою кількістю об'єктів та джерел

світла.

Рей-трейсинг (англ. Ray-tracing) — техніка комп'ютерної графіки, яка полягає у симуляції простору за допомогою променів світла. У традиційному методі ray-tracing для кожного променя, що випускається з камери, розраховується його шлях у просторі, відбиття та кольори, які він зустрічає. Однак, цей метод вимагає великої обчислювальної потужності, особливо при рендерингу складних сцен з великою кількістю об'єктів та джерел світла. Променевий трасування може бути дуже витратним з точки зору обчислювальних ресурсів через потребу у великій кількості обчислень для кожного променя. При рендерингу складних сцен з великою кількістю об'єктів та джерел світла ця проблема стає ще більш відчутною. У таких випадках для зменшення обчислювального навантаження можуть використовуватися різні техніки оптимізації, наприклад, кешування результатів обчислень, розпаралелювання обчислень або застосування методів підтримки обчислювання на відеокарті (GPU).

Модифікований метод ray-tracing, який досліджується і модифікується, спрямований на зменшення обчислювальної складності за рахунок аналізу матеріалу поверхні об'єкта. Ідея полягає в тому, що якщо поверхня об'єкта є матовою, тобто не відбиває світло, то відбиття на цій поверхні не обчислюється. Це дозволяє заощадити обчислювальні ресурси та прискорити процес створення реалістичних зображень. Цей підхід може мати значний вплив на розвиток комп'ютерної графіки, оскільки дозволяє ефективніше використовувати обчислювальні ресурси та покращує продуктивність алгоритму ray-tracing. Аналіз матеріалу поверхні об'єкта є важливим аспектом візуалізації сцен, оскільки дозволяє забезпечити більш реалістичні та деталізовані зображення без зайвого навантаження на систему.

Вибір матеріалу поверхні об'єкта має велике значення для відтворення правдоподібних зображень. Аналіз і класифікація матеріалів дозволить

оптимізувати алгоритм ray-tracing для кожного типу поверхні, що покращить якість зображення і зменшить час обчислень.

Для аналізу матеріалу поверхні фігури використовується спеціальний алгоритм, який визначає тип поверхні (матова або блискуча) на основі її властивостей. Для матової поверхні відбиття не обчислюється, що дозволяє значно зменшити обчислювальну складність алгоритму.

Згідно методу виконується аналіз параметрів матеріалу, такі як коефіцієнт розсіювання світла, шорсткість поверхні, та інші. Наприклад, для матової поверхні характерно, що світло, що падає на неї, розсіюється у різних напрямках, внаслідок чого воно не відбивається прямо назад, як на блискучій поверхні. Це можна виявити шляхом аналізу кута відбиття світла від поверхні. Для блискучої поверхні характерно, що вона відбиває світло у визначений кут, відповідно до закону відбиття світла. Така поведінка світла допомагає визначити, що поверхня є блискучою. Метод може враховувати інші фізичні властивості матеріалу, такі як індекс рефракції, що може впливати на те, як світло взаємодіє з поверхнею. Враховуючи ці параметри, алгоритм може точно визначити тип поверхні і відповідно до цього обчислити, чи потрібно обчислювати відбиття світла для даної точки на поверхні фігури.

Основні етапи методу:

- Аналіз кольору - перед обчисленням відбиття проводиться аналіз кольору пікселя. Якщо він наближений до чорного або сірого, імітуємо матову поверхню та не обчислюємо відбиття.

- Матеріали - для кожного матеріалу об'єкта визначаємо коефіцієнти дифузності k_d та відбивності k_s. Ці коефіцієнти використовуються для обчислення інтенсивності відбитого світла.

- Формула Ламберта - якщо поверхня матова, використовується формула Ламберта для обчислення інтенсивності відбитого світла.

- Відбиття - якщо поверхня блискуча, використовуємо формулу для відбиття світла, де враховується кут падіння та кут відбиття та використовуємо модифікований метод ray-tracing.

- Рекурсивне відбиття - для підвищення реалістичності можна використовувати рекурсивне відбиття, коли промінь відбивається від поверхні декілька разів, зменшуючи інтенсивність відбитого світла кожного разу.

Список матеріалів наведений в таблиці 2.1.

Таблиця 2.1 - Порівняння різних типів матеріалів

Матеріал	***kd***	***ks***	**Тип поверхні**
Метал	0.6	0.4	Блискуча
Дерево	0.8	0.2	Матова
Скло	0.2	0.8	Блискуча
Пластик	0.7	0.3	Матова
Бетон	0.9	0.1	Матова
Поліуретан	0.3	0.7	Блискуча
Текстиль	0.85	0.15	Матова
Папір	0.95	0.05	Матова
Мармур	0.4	0.6	Блискуча
Глина	0.75	0.25	Матова
Шкіра	0.65	0.35	Матова
Вода	0.1	0.9	Блискуча
Гума	0.5	0.5	Матова
Пісок	0.8	0.2	Матова
Шовк	0.7	0.3	Матова

Модифікований метод ray-tracing, який враховує тип поверхні для обчислення відбиття світла, може значно покращити швидкодію та ефективність

рендерингу складних сцен. Аналіз кольору та типу поверхні дозволяє ефективно використовувати ресурси для обчислень, забезпечуючи високу якість зображень.

2.2. Аналіз і обґрунтування вибору засобів для реалізації програмного забезпечення

Для реалізації програмного забезпечення, яке додає об'ємність до 2D зображень, необхідно вибрати оптимальні засоби розробки. У даному розділі проведено варіантний аналіз доступних засобів та обґрунтовано вибір конкретних технологій. Для розробки програмного забезпечення, яке додає об'ємність до 2D зображень, необхідно вибрати оптимальні засоби розробки, які дозволять ефективно втілити функціональні вимоги роботи. У цьому розділі проведено варіантний аналіз доступних засобів та обґрунтовано вибір конкретних технологій.

'Одним з ключових виборів є мова програмування, на якій буде реалізовано програмне забезпечення. Для розробки додатків для мобільних пристроїв Android, Kotlin є однією з найбільш популярних і сучасних мов програмування, яка надає широкі можливості для реалізації складних функцій. Використання Kotlin дозволить забезпечити високу продуктивність та швидкість роботи програмного забезпечення, що є важливим для забезпечення задоволення користувачів.

Для розробки на мові Kotlin для Android-пристроїв оптимальним вибором є використання Android Studio, яке є офіційним інтегрованим середовищем розробки для платформи Android. Android Studio надає широкий набір інструментів для розробки, тестування та налагодження Android-додатків, що робить його ідеальним вибором для нашого програмного забезпечення. Вибір мови програмування Kotlin обумовлений її сучасністю та широкими можливостями для реалізації функцій програмного забезпечення. Використання Android Studio є логічним вибором для розробки на Android, оскільки це офіційне

середовище розробки, яке надає всі необхідні інструменти для створення Android-додатків. Використання OpenGL ES дозволить забезпечити високу якість обробки зображень та додавання об'ємності до 2D зображень. Аналіз доступних засобів:

- OpenCV [17]: Open Source Computer Vision Library є однією з найпопулярніших бібліотек для обробки зображень та комп'ютерного зору. OpenCV надає широкий набір функцій для роботи з зображеннями, включаючи фільтри, виявлення контурів, згладжування та підтримку роботи з 3D зображеннями.

- Android SDK [18]: Android Software Development Kit містить інструменти та бібліотеки для розробки мобільних додатків для платформи Android. Він включає в себе функції для роботи з графікою та зображеннями, такі як масштабування, обрізка та робота з різними форматами зображень.

- JavaFX [19]: JavaFX є бібліотекою для розробки графічних інтерфейсів користувача в Java. Вона має підтримку для відображення зображень та можливості для обробки графічних ефектів.

Для реалізації програмного забезпечення, яке додає об'ємність до 2D зображень, було вирішено використати OpenCV та Android SDK. Обґрунтування цього вибору полягає в наступному:

- Ефективність: OpenCV надає широкий набір оптимізованих алгоритмів для обробки зображень, що дозволяє швидко та ефективно додавати об'ємність до 2D зображень.

- Інтеграція з платформою: Android SDK дозволяє легко інтегрувати розроблене програмне забезпечення з платформою Android, що дозволить використовувати його на широкому колі пристроїв.

Отже, в результаті аналізу було вирішено використати OpenCV та Android SDK для реалізації програмного забезпечення, яке додає об'ємність до 2D зображень. Ці засоби мають всі необхідні функціональність та можливості для успішної реалізації цієї задачі.

Для реалізації мобільної системи для надання об’мності зображенням будуть використані Kotlin [20] та такі бібліотеки як OpenCV, OpenGL ES та Android CameraX. Kotlin - це статично типізована мова програмування, що працює поверх JVM і розробляється компанією JetBrains. Також компілюється в JavaScript. Мову названо на честь острова Котлін у Фінській затоці, на якому розміщена частина Кронштадту.

Головні переваги OpenCV полягають у його потужності та ефективності. Бібліотека написана на C/C++ і має велику кількість оптимізованих алгоритмів, що дозволяє швидко обробляти великі обсяги зображень. Крім того, OpenCV підтримується на різних платформах, включаючи Windows, Linux, macOS, Android та iOS, що робить його універсальним інструментом для розробки комп'ютерних зорових додатків на різних пристроях.

Бібліотека включає в себе багато готових модулів і функцій для роботи з зображеннями, таких як фільтри для обробки зображень, алгоритми виявлення облич, знаходження ключових точок на зображенні та багато іншого. Крім того, OpenCV постійно оновлюється і розширюється новими функціями та можливостями, що робить його потужним інструментом для реалізації різних завдань у сфері комп'ютерного зору.

Android CameraX є потужною бібліотекою, яка спрощує роботу з камерою на пристроях з операційною системою Android. Вона надає розробникам простий та зручний інтерфейс для доступу до функціональності камери, що дозволяє швидко та ефективно реалізувати різноманітні функції зйомки та обробки зображень. Однією з ключових особливостей CameraX є низький рівень абстракції, що дозволяє розробникам працювати з камерою на більш низькому рівні, отримуючи при цьому велику гнучкість та контроль. Це дозволяє ефективно використовувати ресурси пристрою та забезпечує високу продуктивність додатків, які використовують камеру. В цілому, Android CameraX є потужним інструментом для розробників Android, який спрощує роботу з камерою на пристроях з операційною системою Android та відкриває

нові можливості для створення високоякісних фотографій та відео. Розробка програмних модулів для реалізації мобільної системи для додавання 3D ефектів об'ємності 2D зображенням передбачає кілька етапів.

Спочатку потрібно створити модуль, який може захоплювати зображення з пристрою та обробляти його додаючи ефекти для об'ємності. Цей модуль буде відповідати за оброблення зображень за допомогою додавання різноманітних ефектів, в результаті чого зображення будуть мати більш об'ємний та привабливий вигляд. Далі буде розроблений модуль для перетворення 2D у 3D. Цей модуль буде перетворювати зображення у 3D модель за допомогою алгоритмів обробки зображень та комп'ютерного зору. Після захоплення та обробки зображень буде розроблений модуль для рендерингу та відображення 3D-зображень. Цей модуль буде обробляти зображення за допомогою розроблених модулів обробки та відображати його користувачу в реальному часі. Наступним буде модуль для взаємодії користувача з мобільним додатком. Цей модуль дозволить користувачам взаємодіяти з 3D-зображеннями, зберігати та змінювати їх, тощо. Розробка цих програмних модулів вимагає графічного програмування та програмування на Kotlin. Для реалізації цих модулів будуть використані різноманітні інструменти та бібліотеки програмування, зокрема OpenCV, OpenGL ES та Android CameraX. Програмні модулі будуть протестовані та доопрацьовані, щоб забезпечити безперебійну роботу з доповненою реальністю для користувачів.

Отже, розробка програмних модулів для реалізації мобільної системи для надання об'ємності зображенням на Kotlin та OpenCV, OpenGL ES та Android CameraX є комплексним завданням, яке включає кілька етапів. Модулі повинні бути ретельно розроблені, реалізовані та протестовані, щоб забезпечити високу якість роботи з доповненою реальністю. Незважаючи на це, володіючи необхідними знаннями та інструментами, можна створити мобільну систему, яка допомагатиме надавати об'ємність зображенням.

2.3. Модель та алгоритм роботи додатку

Для кращого розуміння роботи програмних модулів мобільної системи для надання об’ємності зображенням було вирішено розробити модель роботи системи на прикладі UML діаграми діяльності (рис. 2.1).

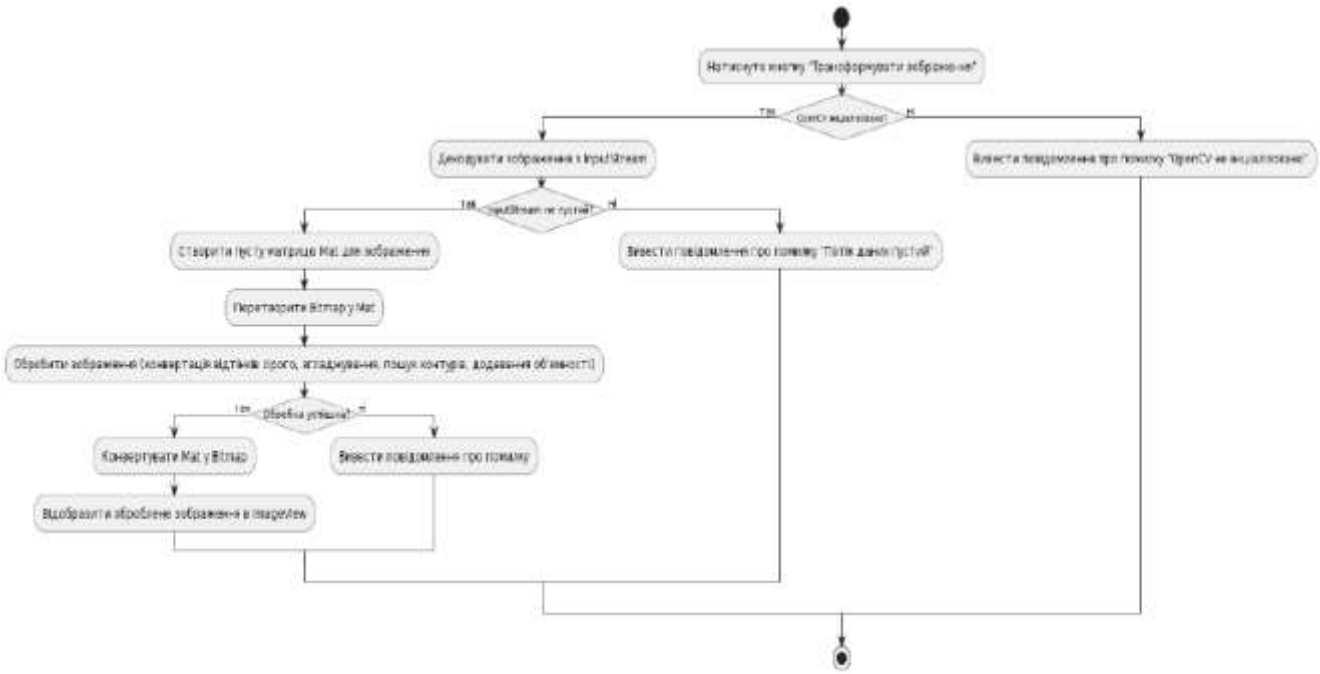


Рисунок 2.1 - Діаграма діяльності мобільної системи

Для роботи мобільного додатка, який надає об'ємність зображенням, використовується алгоритм, що дозволяє додавати глибину та об'єм до зображень, які спочатку були двовимірними. Після завантаження зображення, користувач може вибрати різні ефекти та параметри для створення об'ємного ефекту. Процес перетворення 2D зображення у 3D модель включає в себе кілька кроків. Спочатку, зображення аналізується для виявлення контурів та областей, які потребують обробки. Потім, за допомогою цієї інформації, створюється висотна карта, яка визначає глибину кожного пікселя на зображенні. Нарешті, використовуючи цю висотну карту, створюється об'ємний ефект, який додає зображенню відчуття тривимірності.

Діаграма UML діяльності описує процес обробки зображення в додатку за допомогою OpenCV. Процес починається з натискання користувачем кнопки "Трансформувати зображення". Перш ніж продовжити, перевіряється, чи було успішно ініціалізовано OpenCV. Далі зображення декодується з `InputStream`, отриманого з URI зображення. Перевіряється, чи не є `InputStream` порожнім. Якщо дані у `InputStream` є, то створюється порожня матриця `Mat` для подальшої обробки зображення. Отримане зображення в форматі `Bitmap` конвертується у формат `Mat` для обробки OpenCV.

Зображення обробляється за допомогою ряду операцій, таких як конвертація у відтінки сірого, згладжування, пошук контурів та додавання об'ємності. Перевіряється, чи була успішно завершена обробка зображення. Оброблене зображення у форматі `Mat` конвертується у формат `Bitmap` для відображення у `ImageView`. Оброблене зображення відображається у `ImageView` на екрані додатку. Процес завершується.

Для розробки мобільного додатку, який надає об'ємність зображенням, був зроблений алгоритм, що дозволяє користувачам додавати глибину та об'ємність до їхніх 2D зображень (рисунок 2.2). Додаток пропонує інтуїтивний інтерфейс, який дозволяє швидко та ефективно виконувати цю задачу. Після завантаження зображення користувач може вибрати різні ефекти та налаштування для додавання об'ємності.

Один із ключових етапів алгоритму - перетворення 2D зображення у 3D модель. Цей процес включає в себе виявлення контурів на зображенні, створення маски для цих контурів та використання цієї маски для створення висотної мапи. Після цього, за допомогою цієї висотної мапи, створюється 3D ефект, який додає об'ємність та глибину до зображення. Додаток також має функціонал для додавання інших обробок та ефектів до зображення, що дозволяє користувачам налаштовувати вигляд та стиль своїх зображень. Результат може бути збережений у високій якості та легко поділений через різноманітні канали комунікації.



Рисунок 2.2 - Алгоритм роботи програмного застосунку

Для створення мобільного додатка, який надає об'ємність зображенням, використовується алгоритм, що дозволяє додавати глибину та об'єм до зображень, які спочатку були двовимірними. Після завантаження зображення, користувач може вибрати різні ефекти та параметри для створення об'ємного ефекту.

Процес перетворення 2D зображення у 3D модель включає кілька кроків. Спочатку, зображення аналізується для виявлення контурів та областей, які потребують обробки. Потім, за допомогою цієї інформації, створюється висотна карта, яка визначає глибину кожного пікселя на зображенні. Нарешті, використовуючи цю висотну карту, створюється об'ємний ефект, який додає зображенню відчуття тривимірності.

Крім додавання об'ємності до зображень, додаток також може включати інші цікаві функції, такі як автоматичне видалення фону зображення, зміна освітлення та колірний корекції, а також додавання різноманітних фільтрів та текстур для створення унікальних ефектів. Користувачі можуть експериментувати з різними комбінаціями ефектів, щоб отримати бажаний результат. Такий підхід до обробки зображень може стати відмінним інструментом для творчих проєктів, рекламних кампаній або особистого використання, дозволяючи створювати вражаючі та оригінальні зображення без складних програм або навичок у графічному дизайні.

Додаток також має можливість додавати інші ефекти та обробки до зображень, щоб користувачі могли налаштовувати їх вигляд. Результат обробки може бути збережений у високій якості та легко росповсюджений через різні платформи соціальних мереж.

Цей підхід дозволяє створювати вражаючі об'ємні ефекти для зображень, що робить додаток привабливим для широкого кола користувачів, які цікавляться створенням унікальних та креативних зображень.

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНИХ ДОДАТКУ ДЛЯ НАДАННЯ ОБ'ЄМНОСТІ ЗОБРАЖЕННЯМ

3.1 Розробка модуля надання об'ємності зображенням

Для реалізації можливості додавання об'ємності до 2D зображень у мобільному додатку був розроблений спеціальний модуль. Першим кроком було визначення вимог до модуля. Основні вимоги включали в себе: можливість вибору різних ефектів об'ємності для зображення; простий та зрозумілий інтерфейс для користувачів; ефективність роботи модуля для швидкої обробки зображень.

Модуль складався з наступних компонентів: інтерфейс користувача для вибору та налаштування об'ємних ефектів; механізм обробки зображень, використовувався для обробки та додавання ефектів до зображень; модуль відображення результату, відображав оброблені зображення з доданими 3D ефектами. Для коректного відображення та завантаження елементів застосунку вирішено розробити екран завантаження, діаграма класів якого наведена на рисунку 3.1.

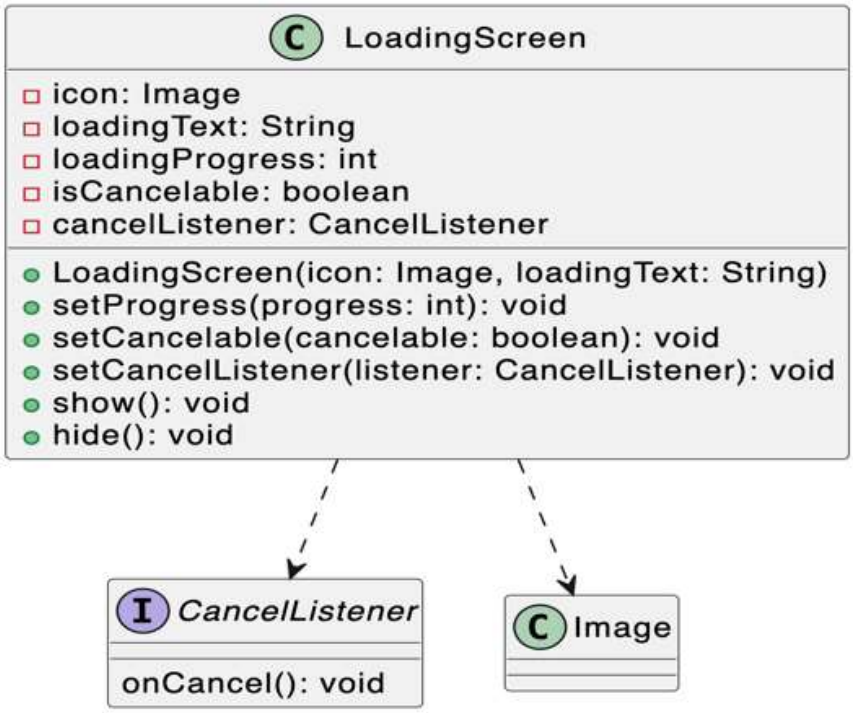


Рисунок 3.1 - Діаграма класу екрану завантаження

Було вирішено розробити окремий клас для маніпуляцій фото, який включає в себе весь можливий функціонал для надання об’ємності зображенням (рис. 3.2).

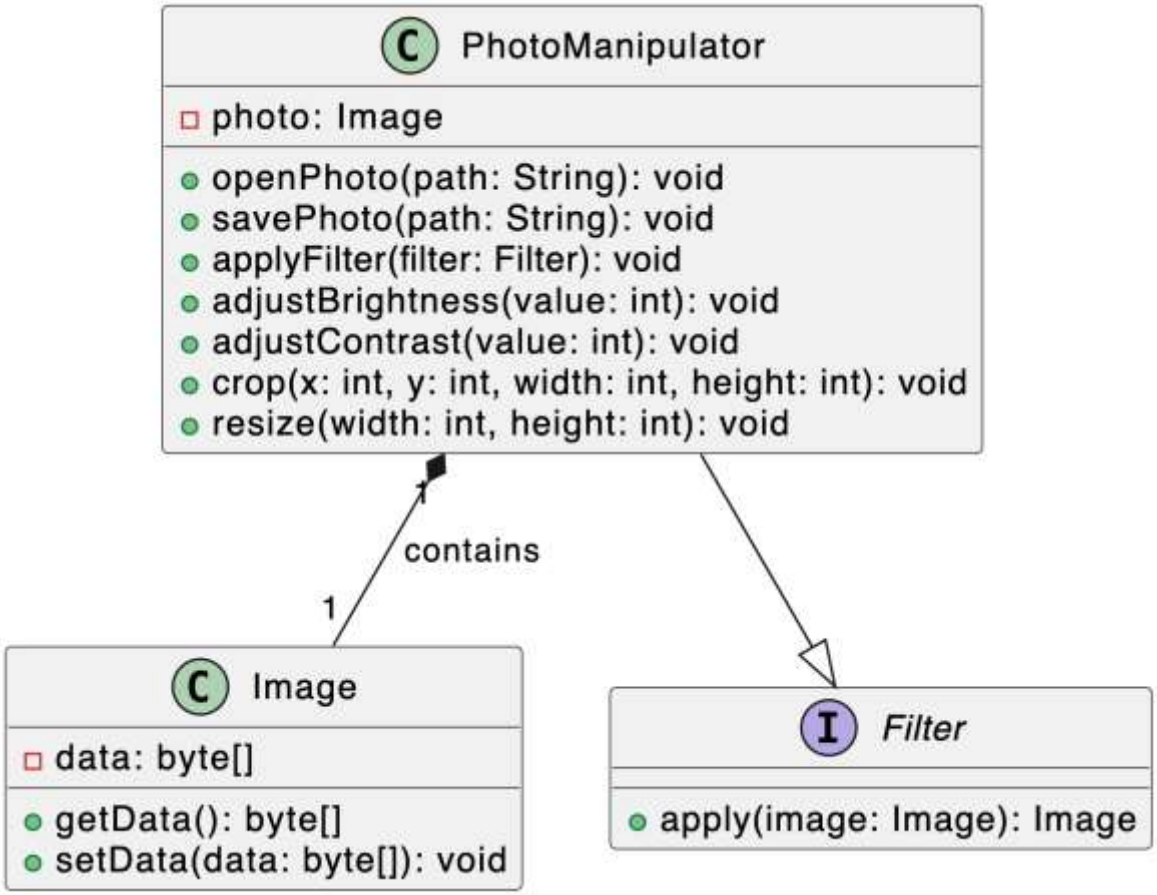


Рисунок 3.2 - Діаграма класу для маніпуляцій з зображеннями

Для додавання зображення до екрану потрібно мати код імпорту зображень з телефону в мобільний застосунок (рисунок 3.3). В даному фрагменті коду реалізовано додавання зображення до екрану Activity додатку та в подальшому мати змогу маніпулювати зображення. Після редагування зображення, нове зображення з доданими ефектами додається в код замість початкового завантаженого.

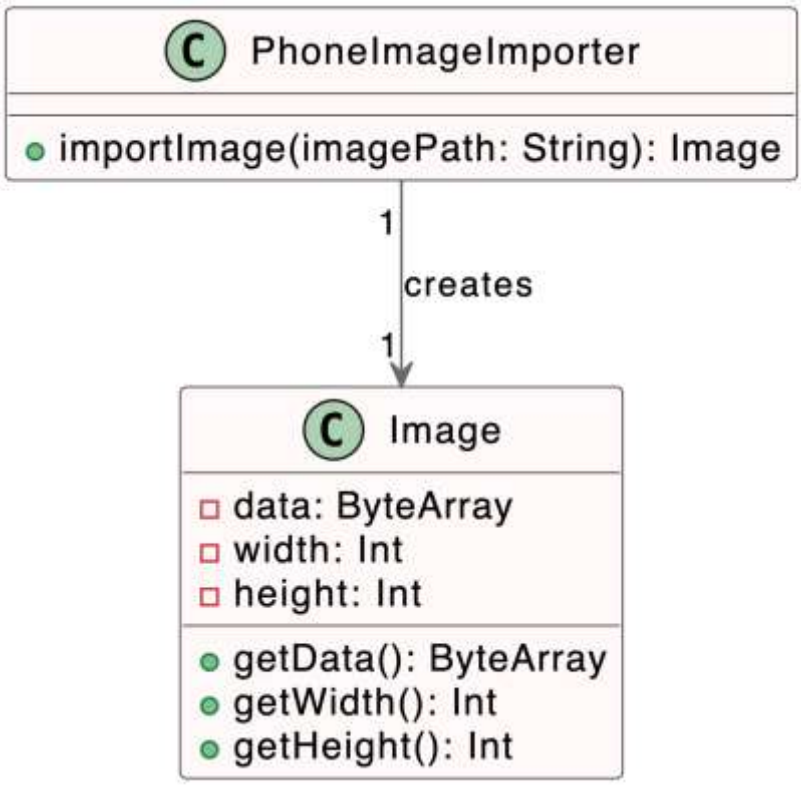


Рисунок 3.3 - Діаграма класу імпорту зображень

3.2. Розробка модуля завантаження обробленого зображення

Модуль завантаження обробленого зображення є важливою частиною програмного забезпечення, оскільки від нього залежить правильність та ефективність відображення оброблених зображень для користувачів. У цьому підрозділі описано розробку модуля, що відповідає за завантаження та відображення обробленого зображення в програмі. Перш за все, необхідно визначити основні вимоги до модуля завантаження обробленого зображення. Основні вимоги включають: можливість завантаження зображення з різних джерел, таких як локальне сховище, мережа або зовнішні джерела; ефективність та швидкість завантаження зображення; підтримка різних форматів зображень, таких як JPEG, PNG, BMP тощо; можливість оброблення та відображення обробленого зображення з урахуванням особливостей обробки, які можуть включати ефекти об'ємності та інші графічні ефекти.

Для реалізації модуля завантаження обробленого зображення було обрано

архітектурний підхід, що базується на розділенні відповідальностей. Модуль складається з наступних компонентів:

Клас ImageLoader: Відповідає за завантаження зображення з вказаного джерела та передачу його для подальшої обробки (рисунок 3.4).

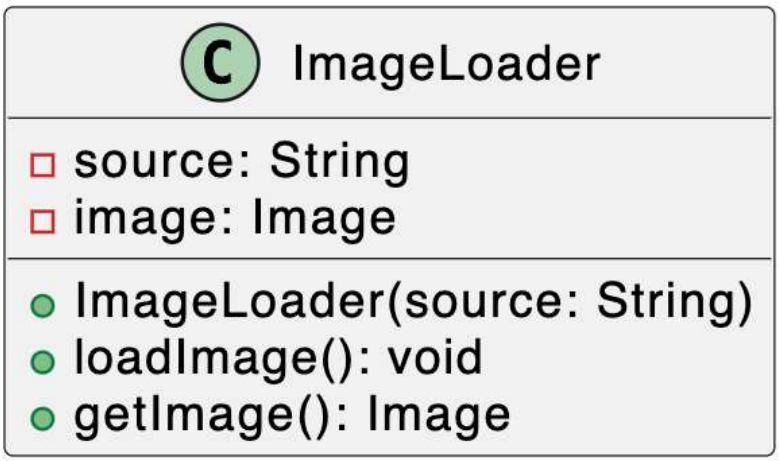


Рисунок 3.4 - Діаграма класу ImageLoader

Клас ImageProcessor: Відповідає за обробку завантаженого зображення, включаючи застосування ефектів об'ємності та інших графічних ефектів (рис. 3.5).

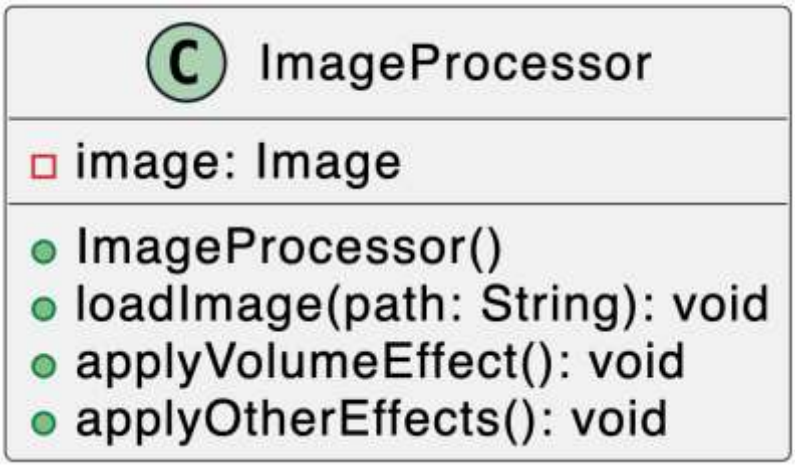


Рисунок 3.5 - Діаграма класу ImageProcessor

Клас ImageRenderer: Відповідає за відображення обробленого зображення на екрані для користувача (рисунок 3.6).

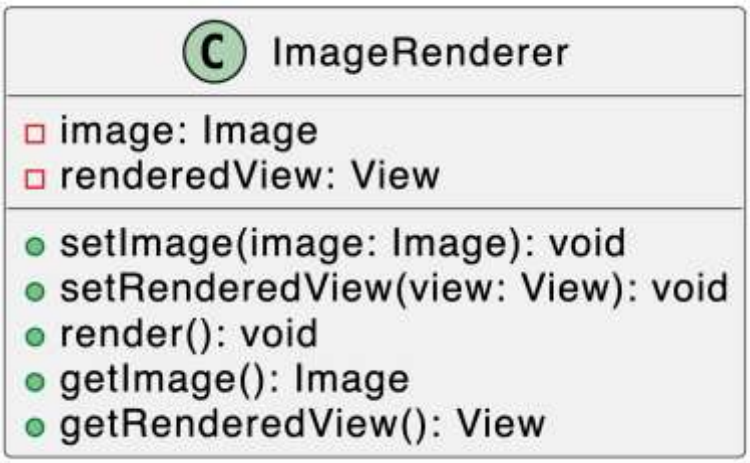


Рисунок 3.6 - Діаграма класу ImageRenderer

Також для коректної роботи бібліотеки OpenCV, потрібно трансформувати вхідні типи зображень в bitmap, а потім в формат mat, з яким працює дана бібліотека (рисунок 3.7).

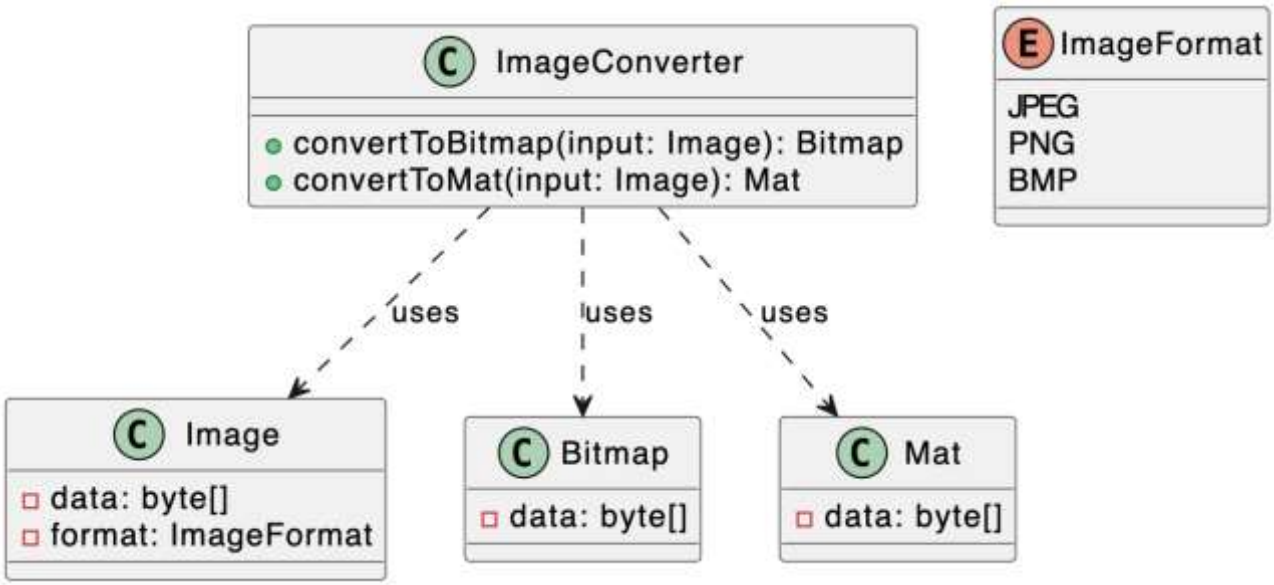


Рисунок 3.7 - Діаграма класу трансформації зображення

Також для створення зрозумілого та зручного інтерфейсу, було написано відповідні до екранів файли xml, котрі описують будову екрану застосунку, стилізації тощо. Фрагмент xml файлу головного екрану зображено на рис. 3.8.

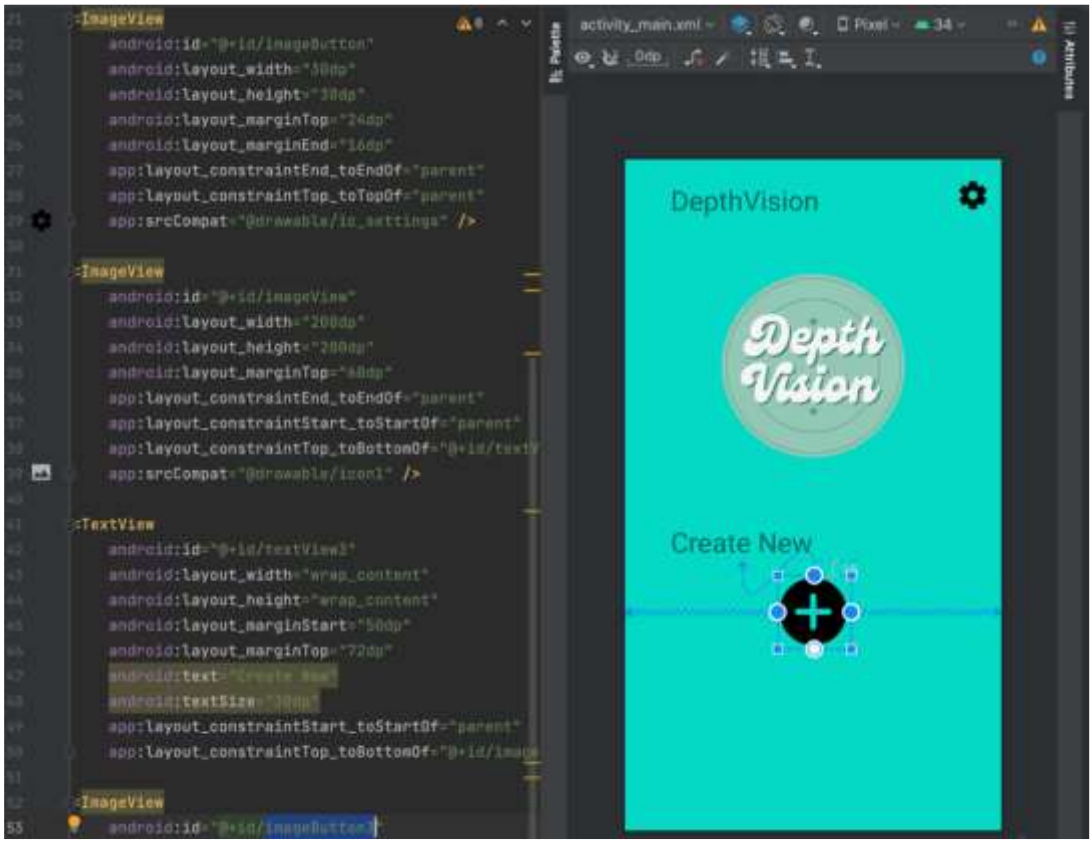


Рисунок 3.8 - Фрагмент xml файлу головного екрану

Нижче наведений фрагмент програмного коду розробленого програмного забезпечення.

```
private final CameraDevice.StateCallback mStateCallback = new
CameraDevice.StateCallback() {
@Override
public void onOpened(CameraDevice cameraDevice) {
mCameraDevice = cameraDevice;
mCameraOpenCloseLock.release();
createCameraPreviewSession();
```

```
}
@Override
public void onDisconnected(CameraDevice cameraDevice) {
cameraDevice.close();
mCameraDevice = null;
mCameraOpenCloseLock.release();
}

@Override
public void onError(CameraDevice cameraDevice, int error) {
cameraDevice.close();
mCameraDevice = null;
mCameraOpenCloseLock.release();
}
};

private void createCameraPreviewSession() {
int w=mPreviewSize.getWidth(), h=mPreviewSize.getHeight();
        Log.i(LOGTAG, "createCameraPreviewSession("+w+"x"+h+")");

if(w<0 || h<0)
return;
try {
mCameraOpenCloseLock.acquire();
if (null == mCameraDevice) {
mCameraOpenCloseLock.release();
Log.e(LOGTAG, "createCameraPreviewSession: camera isn't opened");
return;
}
```

```
if (null != mCaptureSession) {
mCameraOpenCloseLock.release();
Log.e(LOGTAG, "createCameraPreviewSession: mCaptureSession is already
started");
return;
}

if(null == mSTexture) {
mCameraOpenCloseLock.release();
Log.e(LOGTAG, "createCameraPreviewSession: preview SurfaceTexture is null");
return;
        }

mSTexture.setDefaultBufferSize(w, h);
Surface surface = new Surface(mSTexture);
mPreviewRequestBuilder = mCameraDevice
.createCaptureRequest(CameraDevice.TEMPLATE_PREVIEW);
        mPreviewRequestBuilder.addTarget(surface);
```

3.3. Розробка інтерфейсу програмного додатку

Необхідністю є розробити інтуїтивно зрозумілий інтерфейс, за допомогою якого користувач зручно та швидко може надавати об’ємність будь яким зображенням. Основним кольором додатку був обраний м’ятний колір, оскільки він безтурботний та легкий, тим самим користувач може повністю поринути в обробку зображень, не відволікаючись на інші фактори. Допоміжними кольорами було обрано чорний та білий, оскільки вони дуже добре доповнюють основний колір додатку та є універсальними у дизайні. При відкритті додатку, користувача зустрічає екран завантаження із логотипом та назвою системи (рисунок 3.9).



Рисунок 3.9 - Екран завантаження

Після завантаження додатку відкривається головний екран де користувач може перейти в налаштування застосунку або вже розпочинати роботу по обробці зображень (рисунок 3.10).



Рисунок 3.10 - Головний екран

Після натискання на кнопку "створення нового зображення" користувачу відкривається екран додавання зображення, де він може вибрати зображення з галереї свого пристрою або зробити нове фото за допомогою камери. Після вибору зображення відображається попередній перегляд на екрані, щоб користувач міг побачити, як воно виглядатиме з ефектами об'ємності. На цьому екрані користувач може також вибрати різні параметри для обробки зображення, такі як розмір зображення, колірна палітра, чи використовувати автоматичні налаштування для покращення зображення. Крім того, на екрані можуть бути доступні інші функції, наприклад, можливість додавати текст, малюнки або інші графічні елементи до зображення перед його обробкою. Після завершення обробки користувач може зберегти зображення або поділитися ним через різні соціальні мережі або месенджери. (рисунок 3.11).

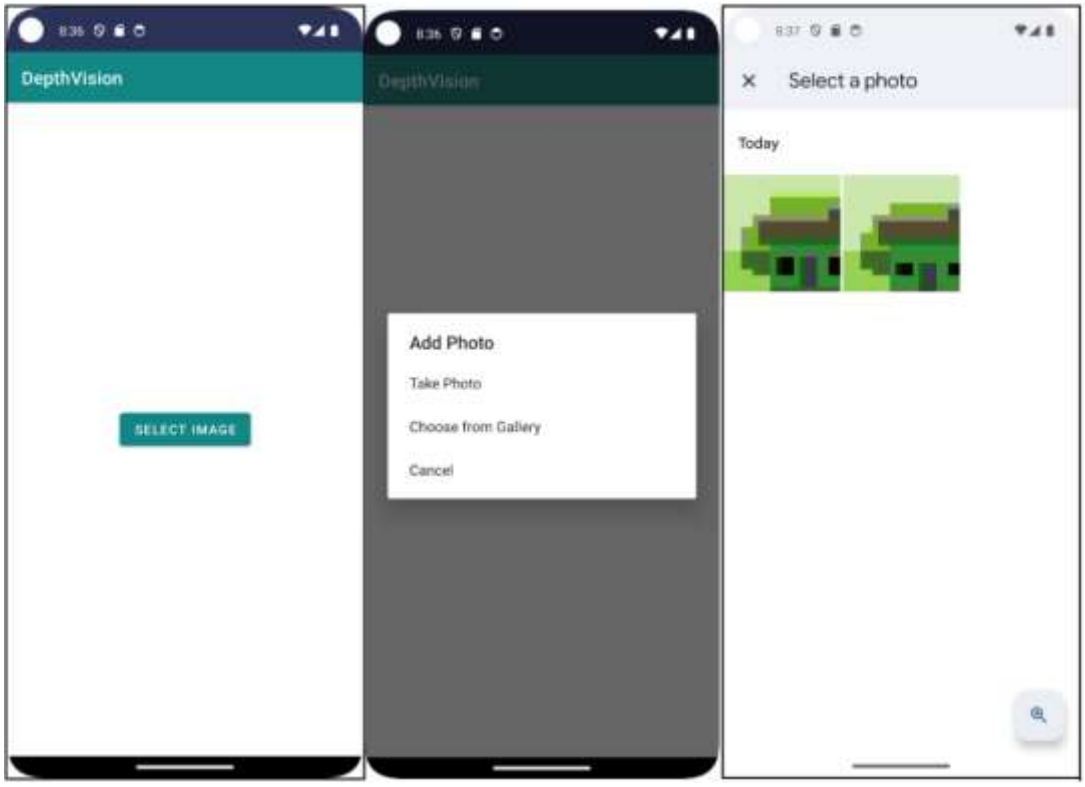


Рисунок 3.11 - Екран додавання зображення

Після вибору фото для надання об’ємності з’являється екран з відображенням самого обраного зображення та кнопка за допомогою якої можна змінити зображення (покращити) (рисунок 3.12).



Рисунок 3.12 - Екран редагування зображення

Після натискання користувач отримує оброблене зображення яке виглядає об’ємним та більш привабливим для користувача (рисунок 3.13).



Рисунок 3.13 - Екран з відображенням обробленого зображення

Розробка графічного інтерфейсу онлайн платформи була проведена у середовищі розробки Android Studio з використанням технології OpenCV та мови програмування Kotlin. Тестування та валідація результатів є важливою частиною будь-якого дослідження або розробки.

Для тестування результатів методу ідентифікації відблиску було використано низку стандартних сценаріїв з реальними об'єктами із відомими характеристиками поверхонь. Кожен сценарій включав в себе об'єкти з різними типами матеріалів і ступенем відблиску. Результати тестування показали, що запропонований метод ідентифікації відблиску виявляє високу точність у визначенні спекулярної складової кольору на ребрах трикутника. Він ефективно розпізнає відблиски на різних типах поверхонь і дозволяє відтворювати їх з великою точністю. Для валідації результатів було порівняно відтворені відблиски з реальними об'єктами, використовуючи стандартні методи оцінки візуальної якості. Оцінка показала високу ступінь відповідності між симуляцією та реальними відблисками.

Застосування методу ідентифікації відблиску показало високу ефективність та точність у відтворенні спекулярних складових кольору на поверхнях об'єктів. Результати тестування та валідації підтвердили його придатність для застосування у графічних застосунках для реалістичного моделювання відблисків.

ВИСНОВОК

У кваліфікаційній роботі бакалавра було розроблено програмні модулі реалізації мобільного додатку для надання об’ємності зображенням. Проведено детальний аналіз предметної галузі. Запропоновано модифікований метод ray-tracing, який відрізняється від відомих аналізом типів матеріалів для вилучення надлишкових обчислень, що дає можливість підвищити продуктивність формування тривимірних графічних сцен за рахунок комбінування методів зафарбовування. Подальшого розвитку отримав метод зафарбовування тривимірних об’єктів, який відрізняється від відомих попереднім аналізом наявності відблисків у межах трикутників і використання спрощених обчислень за їх відсутністю. Це дає можливість підвищити продуктивність рендерінгу зображень. Практична цінність одержаних результатів полягає в тому, що на основі отриманих в кваліфікаційній роботі бакалавра теоретичних положень запропоновано алгоритми та розроблено програмні засоби для надання об’ємності зображенням.

Для розробки було використано середовище програмної розробки Android Studio. Під час виконання кваліфікаційної роботи було проаналізовано сучасний стан проблеми. Було розглянуто основні аналоги програмного продукту, визначено їхні недоліки та порівняно з власним програмним продуктом. Базуючись на порівнянні було встановлено основні завдання.

Під час аналізу технологій розробки було обґрунтовано вибір мови програмування Kotlin та бібліотеки OpenCV. Було розроблено схеми загального алгоритму роботи мобільної системи. Також було розроблено модель системи з використанням UML діаграм. Тестування програми довело повну працездатність даного програмного продукту.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Гавриляк М.С. Теорія кольору і кольороутворення / автор.: М.С. Гавриляк Чернівці: Чернівец. нац. ун-тет, 2022, с. 263.

2. 3D-візуалізація [Електронний ресурс] – Режим доступу до ресурсу: https://oxvisual.com/ua/blog/architectural-3d-renderings-types

3. Modeling [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/3D-%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8E%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F

4. Tissainayagam P. Object Tracking in Image Sequences Using Point Features / P. Tissainayagam, D. Suter // Pattern Recognition .— Vol. 38. — 2005. — p.105—113.

5. Texturing [Електронний ресурс] – Режим доступу до ресурсу: https://www.adobe.com/ua/products/substance3d/discover/3dtexturing.html

6. Lighting [Електронний ресурс] – Режим доступу до ресурсу: https://www.adobe.com/ua/products/substance3d/discover/3d-lighting.html

7. Rendering [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BD%D0%B4%D0%B5%D1%80%D0%B8%D0%BD%D0%B3

8. Reflected Ray : навч. посіб. / Foley, J. D., van Dam, A., Feiner, S. K., & Hughes, J. F. "Computer Graphics: Principles and Practice" (3rd ed.). Addison-Wesley, 2022. – 127с.

9. Radiant shadow : навч. посіб. / Akenine-Moller, T., Haines, E., & Hoffman, N. "Real-Time Rendering" (4th ed.). AK Peters/CRC Press, 2018. – 274с.

10. Glare : навч. посіб. / Foley, J. D., van Dam, A., Feiner, S. K., & Hughes, J. F. "Computer Graphics: Principles and Practice" (3rd ed.). Addison-Wesley, 2022. – 142с.

11. Scattering of light : навч. посіб. / Pharr, M., Jakob, W., & Humphreys, G. "Physically Based Rendering: From Theory to Implementation" (4th ed.). Morgan Kaufmann, 2019. – 119с.

12. Augmented Reality – The Past, The Present and The Future [Електронний ресурс] – Режим доступу до ресурсу: https://www.interactiondesign. org/literature/article/augmented-reality-the-past-the-present-and-thefuture

13. Photoshop Express [Електронний ресурс] – Режим доступу до ресурсу: https://www.adobe.com/ua/products/photoshop-express.html

14. Snapseed [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Snapseed

15. Pixlr [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Pixlr

16. Ray-tracing : навч. посіб. / Pharr, M., Jakob, W., & Humphreys, G. "Physically Based Rendering: From Theory to Implementation" (4th ed.). Morgan Kaufmann, 2019. – 119с.

17. OpenCV: Open Source Computer Vision Library [Електронний ресурс] – Режим доступу до ресурсу: https://opencv.org/

18. Android SDK: Android Software Development Kit [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Android_SDK

19. JavaFX: JavaFX [Електронний ресурс] – Режим доступу до ресурсу: https://openjfx.io/

20. Kotlin [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Kotlin