

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики

(інститут)

Факультет інформаційних технологій

(факультет)

Кафедра Програмного забезпечення комп'ютерних систем

(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня

бакалавра

(назва освітньо-кваліфікаційного рівня)

студента

Біденка Володимира Володимировича

(ПІБ)

академічної групи

122-19-4

(шифр)

спеціальності

122 Комп'ютерні науки

(код і назва спеціальності)

освітньої програми

Комп'ютерні науки

(назва освітньої програми)

на тему:

Розробка програмного забезпечення для застосування

комп'ютерної графіки у сфері кінематографу за допомоги хромакею

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи				
розділів:				
спеціальний	<i>проф. Бердник М.Г.</i>			
економічний	<i>проф. Вагонова О.Г.</i>			
Рецензент	<i>доц. Шедловский І.А</i>			
Нормоконтролер	<i>доц. Гуліна І.Г.</i>			

Дніпро
2023

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»

ЗАТВЕРДЖЕНО:

завідувач кафедри
програмного забезпечення комп'ютерних систем

(повна назва)

М.О. Алексєєв

(підпис)

(прізвище, ініціали)

« » 2023 року

ЗАВДАННЯ

на кваліфікаційну роботу
бакалавра

(назва освітньо-кваліфікаційного рівня)

студента 122-19-4 Біденка В.В.

(група)

(прізвище та ініціали)

тема кваліфікаційної роботи Розробка програмного забезпечення для застосування комп'ютерної графіки у сфері кінематографу за допомоги хромакею

затверджена наказом ректора НТУ «ДП» від 16.05.2023 № 350-с

Розділ	Зміст виконання	Термін виконання
Спеціальний	<i>На основі матеріалів проєктно-технологічної практики та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми</i>	<i>13.05.2023 р.</i>
Економічний	<i>Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПЗ й тривалості його розробки</i>	<i>27.05.2023 р.</i>

Завдання видав

проф. Бердник М.Г

(підпис)

(посада, прізвище, ініціали)

Завдання прийняв до виконання

Біденко В.В.

(підпис)

(прізвище, ініціали)

Дата видачі завдання: 14.01.2023 р.

Термін подання кваліфікаційної роботи до ЕК: 12.06.2023 р.

РЕФЕРАТ

Пояснювальна записка: 87 с., 12 рис., 2 табл., 3 дод., 25 джерел.

Об'єкт розробки: програмне забезпечення для застосування комп'ютерної графіки у сфері кінематографу за допомоги хромакею.

Мета кваліфікаційної роботи: розробка програмного забезпечення для застосування комп'ютерної графіки у сфері кінематографу за допомоги хромакею з використанням мови Python.

У вступі розглядається аналіз та сучасний стан проблеми, конкретизується мета кваліфікаційної роботи та галузь її застосування, наведено обґрунтування актуальності теми та уточнюється постановка завдання.

У першому розділі проаналізовано предметну галузь, визначено актуальність завдання та призначення розробки, сформульовано постановку завдання, зазначено вимоги до програмної реалізації, технологій та програмних засобів.

У другому розділі проаналізовані наявні рішення, обрано платформи для розробки, виконано проектування і розробка програми, описана робота програми, алгоритм і структура її функціонування, а також виклик та завантаження програми, визначено вхідні і вихідні дані, охарактеризовано склад параметрів технічних засобів.

В економічному розділі визначено трудомісткість розробленої інформаційної системи, проведений підрахунок вартості роботи по створенню програми та розраховано час на його створення.

Практичне значення полягає у створенні додатка, що надає можливість працювати із хромакеєм, відділяти фон, застосовувати корекцію кольору та мати можливість додавати спецефекти.

Актуальність інформаційної системи визначається великим попитом кіноіндустрійного ринку на подібні розробки, що дають змогу розробляти вигадані світи та спрощувати роботу над їх створенням, а також залучати менше ресурсів, таким чином пришвидшуючи процес роботи при менших фінансових вкладеннях, а також підвищуючи їх якість та відповідність потреб сучасного глядача.

Список ключових слів: ХРОМАКЕЙ, ФОН, ОБРОБКА, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ГРАФІКА.

ABSTRACT

Explanatory note: 87 pp., 12 fig., 2 table, 3 appendix, 25 sources.
Development object: software for the application of computer graphics in the field of cinematography with the help of chromakey.

The goal of the qualification work: development of software for the application of computer graphics in the field of cinema with the help of chromakey using the Python language.

In the introduction, the analysis and current state of the problem is considered, the purpose of the qualification work and the field of its application are specified, the justification of the relevance of the topic is given, and the statement of the task is clarified.

In the first section, the subject area is analyzed, the relevance of the task and the purpose of the development is determined, the task statement is formulated, and the requirements for software implementation, technologies and software tools are specified.

In the second section, available solutions are analyzed, platforms for development are chosen, program design and development is performed, program operation, algorithm and structure of its functioning are described, as well as program calling and loading, input and output data are determined, the composition of technical means parameters is characterized.

In the economic section, the labor intensity of the developed information system is determined, the cost of work on creating the program is calculated, and the time for its creation is calculated.

The practical value is to create an application that provides the ability to work with chroma key, separate the background, apply color correction and be able to add special effects.

The relevance of the information system is determined by the great demand of the film industry market for similar developments, which make it possible to develop fictional worlds and simplify the work on their creation, as well as to attract fewer resources, thus speeding up the work process with smaller financial investments, as well as indicating their quality and compliance with the needs of modern the viewer.

List of keywords: CHROMAKEY, BACKGROUND, PROCESSING, SOFTWARE, GRAPHICS.

ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT.....	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ.....	10
1.1. Загальні відомості з предметної галузі.....	10
1.1.1. Історія виникнення та розвитку хромакею.....	10
1.1.2. Історія використання хромакею у кіноіндустрії.....	14
1.1.3. Поняття візуальних ефектів.....	17
1.1.4. Основні принципи ключування та корекції хромакею.....	20
1.1.5. Використання різних форматів відео та зображень у кіноіндустрії.....	30
1.2. Призначення розробки та галузь застосування.....	33
1.3. Підстава для розробки.....	34
1.4. Постановка завдання.....	34
1.5. Вимоги до програми або програмного виробу.....	35
1.5.1. Вимоги до функціональних характеристик.....	35
1.5.2. Вимоги до складу та параметрів технічних засобів.....	36
1.5.3. Вимоги до якості та контролю якості візуальних ефектів.....	39
1.5.4. Вимоги до хромакею, який взаємодіє з програмним забезпеченням...	42
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	48
2.1. Функціональне призначення програмного забезпечення.....	48
2.2. Опис застосованих математичних методів.....	51
2.3. Опис використаних технологій та мов програмування	52
2.4. Опис структури програми та алгоритмів її функціонування.....	58
2.5. Обґрунтування та організація вхідних та вихідних даних програми...	59
2.6. Опис розробленої системи	59
2.6.1. Використані технічні засоби.....	59

2.6.2. Використані програмні засоби.....	60
2.6.3. Виклик та завантаження програми.....	60
2.6.4. Опис інтерфейсу користувача.....	60
РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ.....	70
3.1. Розрахунок трудомісткості та вартості розробки програмного продукту.....	70
3.2. Рахунок витрат на створення програми.....	74
ВИСНОВКИ.....	76
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	77
Додаток А. Код програми.....	79
Додаток Б. Відгук керівника економічного розділу.....	86
Додаток В. Перелік файлів на диску.....	87

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

VFX – visual effects;

AK – альфа канал;

AR – augmented reality;

VR – virtual reality;

UV mapping – розгортка об'єкта на координатній сітці;

GS – green screen;

BS – blue screen;

ПТ – проекційне текстурування;

ФТ – фотографічне текстурування;

Проц.Т – процедурне текстурування;

РТ – ручне текстурування;

ЦК – цифровка камери;

СМΥК – Cyan, Magenta, Yellow, key;

RGB – Red, Green, Blue;

ПЗ – програмне забезпечення;

GPU – Graphics Processing Unit.

ВСТУП

Розробка програмного забезпечення для використання комп'ютерної графіки в кінематографії за допомогою хромакею є вкрай актуальною темою в даний час. Щороку кількість фільмів, що використовують комп'ютерну графіку, зростає, і це призводить до підвищення вимог до якості та реалізації візуальних ефектів. Хромакей – це одна з технологій, яка дозволяє створювати ефекти, відокремити об'єкти від фону на відео та додавати до них візуальні ефекти. У цьому контексті розробка програмного забезпечення для використання комп'ютерної графіки в кінематографії за допомогою хромакею є актуальною темою дослідження з кількох причин.

По-перше, за допомогою хромакею можна значно покращити якість та реалізм візуальних ефектів у фільмах. Ця технологія дозволяє створювати різні ефекти, наприклад, зробити людину невидимою або додати кадри, яких не було під час зйомок. Для досягнення максимально реалістичного ефекту візуальних ефектів потрібна висока точність та якість обробки відео. Розробка програмного забезпечення, що забезпечує цю точність та якість, може значно підвищити привабливість фільмів для глядачів.

По-друге, створення програмного забезпечення для використання комп'ютерної графіки в кінематографії за допомогою хромакею може значно скоротити час, необхідний створення візуальних ефектів. Коли ефекти створюються вручну, це забирає багато часу і вимагає великої команди фахівців. Розробка програмного забезпечення, яке дозволяє створювати візуальні ефекти автоматично, може значно скоротити час, необхідний для їхнього створення. Це дозволить студіям кінематографу заощадити гроші та тимчасові ресурси, що є важливим фактором у конкурентній індустрії.

По-третє, використання хромакею в кінематографії може призвести до появи нових можливостей для створення візуальних ефектів. Це може включати створення ефектів, які раніше були недоступні, а також більш реалістична взаємодія об'єктів на екрані.

Щодо хромакею та його використання у кінематографі, своєю думкою ділилися відомі режисери. Джеймс Кемерон, який став популярним візнявши фільм “Титанік” , казав що, хромакей - це ключ до створення будь-яких візуальних ефектів, які ви можете уявити, він відкриває двері уяві, щоб дозволити вам робити все, що ви хочете, на екрані, а Джордж Лукас, котрий відомий людям “Зоряними війнами” стверджує, що хромакей - це незамінний інструмент який дозволяє створювати цілі світи та істот, які раніше були неможливі для реалізації на екрані.

Тому метою кваліфікаційної роботи є розробка програмного забезпечення для засосування компю'терної графіки у сфері кінематографу за допомоги хромакея з використанням мови Python.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1. Загальні відомості з предметної галузі

1.1.1. Історія виникнення та розвитку хромакею

Хромакей, також відомий як "зелений екран", є технікою відео та фотографії, яка використовується для заміни фону на певний колір або зображення. Ця техніка дозволяє створювати візуальні ефекти, розширювати сцени та покращувати візуальну презентацію. Історія хромакею налічує безліч віх та розвитку, і починається вона досить давно.

Перші кроки у створенні хромакею було зроблено 1930-х роках. Компанія RKO Radio Pictures розробила процес, який використовував камеру із чорно-білою плівкою та зелене тло. Ідея полягала в тому, щоб замінити зелене тло на бажане тло, щоб створити враження, ніби об'єкти знімалися на цьому тлі. Проте техніка мала свої обмеження, оскільки вимагала ретельної ручної обробки фотографій та відеоматеріалів.

У 1940-х роках художник і фотограф Лоуренс Баттен зробив значний внесок у розвиток хромакею. Він розробив спеціальну систему, яка використовувала двошарову оптичну плівку та зелений екран. Перший шар плівки експонувався на зеленому екрані, а другий шар використовувався для зйомки об'єктів перед зеленим фоном. При подальшій обробці фотографій можна було видалити зелене тло і замінити його на бажане тло. Це значно спростило процес обробки та дозволило отримувати якісніші результати.

У 1960-х роках компанія Ultimatte розробила перший комп'ютерний хромакей, який використовувався у телевізійній індустрії. Це був прорив у технології, оскільки комп'ютерний хромакей дозволяв у реальному часі видаляти зелене тло і замінювати його на інший. Ultimatte розробила апаратне та програмне забезпечення, яке визначало зелений колір та дозволяло видалити його, а потім додати новий фон. Це дало можливість створювати ефекти на льоту, без потреби у тривалій постобробці. Спочатку

ця технологія була доступна на перших телевізійних студіях, де використовувалась для створення різних ефектів та задніх фонів. Це значно скоротило час і витрати на виробництво, тому що не потрібно було знімати кожну сцену в реальному середовищі, а можна було додати фони та ефекти на етапі постпродакшну.

У 1970-х роках з появою кольорової телевізії та відеотехніки хромакей став більш поширеним та використовуваним. Процес став більш покращеним та автоматизованим, завдяки розробці спеціалізованого обладнання та програмного забезпечення. Використання хромакею стало звичайною практикою у телевізійній індустрії, особливо у передачах з віртуальними наборами та спеціальними ефектами.

З розвитком комп'ютерної графіки та відеотехнологій у 1990-х роках хромакей став ще більш популярним та доступним для широкої аудиторії. Компанії, такі як Adobe, розробили спеціалізовані програми обробки хромакею, що зробило його доступним широкому колу користувачів. Завдяки цьому, хромакей став застосовуватися у професійних студіях, а й у домашніх умовах, де люди могли створювати свої власні відеоматеріали з використанням ефектів хромакею.

В даний час, хромакей техніка продовжує розвиватися та удосконалюватися. З появою більш потужних комп'ютерів, програмного забезпечення та нових матеріалів, таких як синтетичні тканини, які мають більш рівномірне зелене забарвлення, процес хромакею став ще більш точним і ефективним. Технології розпізнавання та обробки зображень продовжують розвиватися, що дозволяє більш точно виділяти об'єкти та реалістично замінювати фони. Таким чином, історія хромакею свідчить про його поступовий розвиток від перших експериментів із плівкою та зеленим екраном до сучасних комп'ютерних технологій та програмного забезпечення.

Сьогодні хромакей є невід'ємною частиною індустрії розваг, кіно та телебачення. Він застосовується у створенні спеціальних ефектів, віртуальних наборів, реклами, музичних кліпів, відеоігор, стрімінгу та багато

іншого. З розвитком соціальних медіа та відеохостингів, хромакей став широко використовуватися в контент-створенні для платформ, таких як YouTube, TikTok, Instagram та інші.

Блогери, стримери та творчі люди використовують хромакей для створення професійного вигляду своїх відеоматеріалів, застосовуючи різні ефекти, задні фони та віртуальні мережі. Це дозволяє створювати унікальний і якісний контент, залучаючи широку аудиторію. Сучасні технології хромакею також включають розширену реальність (AR) і віртуальну реальність (VR). За допомогою хромакею та спеціальних гарнітур AR/VR користувачі можуть поринути у віртуальні світи та взаємодіяти з візуальними елементами, створюючи приголомшливі візуальні ефекти та досвід, що занурює.

Майбутнє хромакею обіцяє ще більше інновацій та можливостей. Розвиток штучного інтелекту, машинного навчання та комп'ютерного зору дозволяє автоматизувати та покращувати процес обробки хромакею, роблячи його більш точним, швидким та доступним для широкої аудиторії. Технології, такі як розпізнавання об'єктів та трекінг рухів, дозволяють більш точно виділяти об'єкти та взаємодіяти з ними на фоні.

На закінчення, історія хромакею є показником постійного розвитку та застосування передових технологій для створення візуальних ефектів та покращення процесу створення відео та фотографій. Від перших експериментів до сучасних інновацій, хромакей став невід'ємною частиною індустрії розваг та контент-створення. Завдяки хромакею, творці контенту можуть втілювати свої творчі ідеї, розширювати межі візуального сприйняття та досягати нових висот як виробництво.

Одним із нових напрямків у розвитку хромакею є застосування його у сфері віртуальної та доповненої реальності. З використанням хромакею в AR/VR-технологіях, користувачі можуть створювати інтерактивні та захоплюючі візуальні світи, в яких вони можуть взаємодіяти та насолоджуватися вражаючими ефектами.

Варто відзначити, що хромакей знаходить застосування не тільки у розважальній промисловості, а й у інших областях. Наприклад, в освіті він використовується для створення інтерактивних уроків та презентацій, де викладачі можуть показувати демонстрації на віртуальних фонах або створювати візуалізації для покращення розуміння навчального матеріалу.

У медицині хромакей застосовується розробки тренажерів, симуляторів і візуальних моделей на навчання і планування операцій. Майбутнє хромакею обіцяє ще більше інновацій та можливостей. З розвитком технологій та удосконаленням алгоритмів обробки зображень очікується покращення якості виділення об'єктів на передньому плані та більш точна заміна фону. Можливо, з'являться нові методи та матеріали, які дозволять ще гнучкіше та ефективніше застосовувати хромакей.

У результаті хромакей відіграє важливу роль у світі візуальної творчості та комунікації. Він відкриває двері до нескінченних можливостей створення та втілення ідей, дозволяючи людям втілити свою креативність та реалізувати свої творчі бачення. Разом з розвитком технологій та покращенням процесу хромакею, ми можемо очікувати ще більш захоплюючу сторінку втілення віртуальних світів та вражаючих візуальних ефектів.

Безперечно, що хромакей продовжить еволюціонувати та вносити нові інновації в галузь візуального мистецтва, телебачення, кіно, стримінгу, освіти та інших галузей. З його допомогою творчі можливості розширюватимуться, а візуальні та віртуальні світи стануть ще більш захоплюючими та реалістичними. Таким чином, історія хромакею від його ранніх експериментів та розробок до сучасних технологій та застосувань є історією прагнення людства до створення вражаючих та візуально збагачених світів.

З кожним новим кроком у його розвитку, хромакей продовжує відкривати перед нами необмежені можливості вираження та втілення наших творчих ідей. Майбутнє хромакея обіцяє нам ще більш захоплюючі

можливості свого функціоналу, які надихатимуть нас і перенеситимуть нас у світи уяви та фантазії.

1.1.2. Історія використання хромакею у кіноіндустрії

Історія використання хромакею в кіноіндустрії налічує багато років і починається зі шляхетного екрану, який використовувався в кіно на початку 20-го століття. Шляхетний екран був зеленого або синього кольору і використовувався для заміни фону або додавання спеціальних ефектів до зйомок. Але в ті часи процес хромакею був складним і вимагав великого вміння і терпіння з боку операторів. У 1930-х роках хромакей екран був удосконалений і зарекомендував себе як важливий інструмент в кіноіндустрії.

Великі студії, такі як Universal і Warner Bros., почали активно використовувати хромакей для створення спеціальних ефектів і фантастичних світів. Використання хромакею дозволяло режисерам зняти сцени, які були неможливі або дуже дорогі для відтворення у реальному житті. Одним з відомих прикладів використання хромакею у кіно є "Зелена година" (1938) режисера Майкла Кертиса. У цьому фільмі вперше був застосований зелений хромакей екран для створення сцени, в якій головний герой з'являється на задньому плані різних відеозаписів з допомогою спеціального екрану. Протягом наступних десятиліть хромакей продовжував використовуватися в кіноіндустрії, постійно удосконалюючись.

Розвиток цифрової технології та комп'ютерної графіки у 1990-х роках значно змінив підхід до використання хромакею. З'явилися спеціалізовані комп'ютерні програми та апаратне забезпечення для обробки хромакею, що спростило процес і зробило його більш доступним для кінематографічних студій. З появою цифрової епохи хромакей став невід'ємною частиною великих голлівудських продукцій.

Завдяки хромакею кінорежисери можуть перетворити непривабливі або небезпечні місця в екзотичні локації або вигадані світи. Також хромакей використовується для створення спеціальних ефектів, які додають магію і фантазію до фільмів. Він дозволяє реалізувати неможливі сцени, такі як літання героїв, зміна образу персонажів або зміна фону в реальному часі. Однак, використання хромакею вимагає великого технічного досвіду та витрат на обладнання. Для створення якісного ефекту необхідне правильне освітлення сцени, точне використання камери і комп'ютерної обробки зображення. Крім того, важливо забезпечити взаємодію акторів з віртуальними об'єктами або фонами, щоб забезпечити природність та реалістичність сцен.

У сучасних фільмах хромакей використовується не тільки для створення спеціальних ефектів, але і для полегшення зйомок. Наприклад, використання хромакею дозволяє замінити фонові масиви або виконавців, які створюють масовки, на віртуальні об'єкти або людей за допомогою комп'ютерної графіки. Це забезпечує ефективність і зручність зйомок, а також дає можливість режисерам зосередитися на виконанні акторів та сюжетних лініях, не витрачаючи багато часу і коштів на реалізацію складних сцен.

Крім кіно, хромакей також широко використовується в телевізійному виробництві, зокрема в новинах, ток-шоу, спортивних трансляціях та рекламних роликах. Завдяки хромакею можна створювати віртуальні студії, в яких ведучі або коментатори знаходяться перед відображеним на екрані віртуальним фоном, що робить трансляції більш ефектними та цікавими для глядачів. Важливо зазначити, що хромакей залишається еволюційною технологією.

Сучасні кіностудії вдосконалюють та використовують нові методи його застосування, такі як використання LED-екранів замість фізичних шпалетних екранів або використання високоточних сенсорних систем для взаємодії

акторів з віртуальними об'єктами. У майбутньому можна очікувати ще більшого розширення використання хромакею у кіноіндустрії.

З розвитком віртуальної та доповненої реальності, можливостей штучного інтелекту та комп'ютерної графіки, хромакей стане ще більш потужним і ефективним інструментом для створення новаторських та захоплюючих кінематографічних досвідів. Одним з можливих напрямків розвитку хромакею є використання вирізних панорамних екранів, які забезпечують ще більшу іммерсію та реалізм.

Зараз хромакей також активно використовується в інших галузях, наприклад, в розвагах та ігровій індустрії. Він знаходить своє застосування в створенні відеоігор, віртуальної реальності, а також в стрімінгових платформах та контент-креаторському середовищі. Хромакей дозволяє стрімерам та відеоблогерам створювати захоплюючий контент, додавати спеціальні ефекти, використовувати віртуальні фони та взаємодіяти з глядачами у реальному часі.

У майбутньому хромакей може знайти застосування в нових сферах, таких як розширена реальність у вигляді віртуального заміщення, медицина, дизайн та багато інших. З розвитком технологій та зростанням інтересу до віртуального досвіду, хромакей відіграє ключову роль у створенні нових можливостей та відкриває шлях для безмежного творчого вираження.

Отже, історія використання хромакею у кіноіндустрії свідчить про його важливе місце у творенні захоплюючих та реалістичних кінематографічних ефектів. Він дозволяє режисерам творити неймовірні світи та сцени, які раніше були неможливими.

З розвитком технологій та творчого підходу хромакей продовжує інновувати та забезпечувати безмежні можливості для кіноіндустрії та інших галузей, де використовується візуальне мистецтво. Завдяки постійному розвитку технологій та креативному підходу, майстри хромакею продовжують вражати глядачів новими захоплюючими світами та ефектами, що робить кіномистецтво ще більш захопливим та непередбачуваним.

1.1.3. Поняття візуальних ефектів

Візуальні ефекти (VFX) – це спеціальні елементи, що використовуються в кіно, телебаченні та комп'ютерних іграх, щоб зробити сцени більш реалістичними, захоплюючими чи драматичними. Візуальні ефекти часто використовуються у фантастичних фільмах, науково-фантастичних картинах та пригодницьких фільмах, але також можуть застосовуватись у документальних фільмах, передачах новин та рекламних відео.

Один із основних інструментів для створення візуальних ефектів є композитинг, який є процесом об'єднання різних елементів, таких як справжні зйомки, комп'ютерна графіка та інші ефекти в одному кадрі. Процес композитингу може включати використання альфа-каналу, який вказує прозорість пікселів в зображенні. Альфа-канал може бути використаний визначення того, які елементи зображення повинні бути видимими, а які ні.

Крім того, при створенні візуальних ефектів також використовується хромакеєва техніка, при якій фон замінюється на інший елемент, такий як комп'ютерна графіка. Для цього використовується зелений або синій екран, на якому знімається актор або інший об'єкт. Також часто використовується клавішний чек, який є технікою відокремлення об'єкта від фону за допомогою альфа-каналу.

Для створення візуальних ефектів також використовується рендеринг, який є процесом створення зображення із комп'ютерної моделі, включаючи моделювання світла, матеріалів та тіней. Також процес створення візуальних ефектів може включати корекцію кольору, яка є процесом зміни кольору або насиченості зображення для досягнення певного ефекту.

Візуальні ефекти можуть включати різні елементи, такі як тіні, світло, воду, вогонь, дим, електричні розряди та інші. Всі ці елементи використовуються для того, щоб зробити сцени більш реалістичними, цікавими та захоплюючими. Візуальні ефекти також можуть включати і

створення різних істот або об'єктів, яких немає в реальному світі. Для цього використовується моделювання, текстурування та анімація.

Моделювання називають - процес створення тривимірних об'єктів за допомогою полігонів, які складаються з вершин гранів та ребер. Полігони можуть набувати різної форми, але найчастіше використовують саме трикутну форму завдяки універсальності та більш простій обробки, завдяки чому робота просувається швидше, а потреби об'ємів потужності техніки стають меншими. Однією з найголовніших основ у 3D - є контроль кількості полігонів, адже ці малі фігури у великому обсязі можуть добряче навантажити техніку і щоб позбутися цієї проблеми, розробники активно моніторять кількість полігонів і нерідко виставляють ліміти трикутників, кількість яких вони не можуть перевищити. На цьому етапі починається клопітливий процес, тому як усі полігони, які людина не зможе побачити – видаляються. Наприклад скеля на сцені буде полою, адже користувач не побачить, що знаходиться усередині за задумом розробника.

Текстурування - це процес додавання зображення на поверхню 3D-моделі, щоб створити ефект деталізації, реалізму та візуального інтересу. Текстури можуть включати, зображення, малюнки або інші елементи, які можуть бути застосовані до поверхні об'єкта, але цей список не завершується лише на штучно створених у ручну зображень, бо текстури можуть включати у себе ще і фотографії чи подібні готові ресурси. Текстури активно використовуватися для створення реалістичних поверхонь і без них неможливо уявити накладання на об'єкт шкіри, металу, дерева, тканин різних типів, чи тим паче більш абстрактних або ж фантастичних ефектів. Застосування текстур може також дозволити суттєво зменшити кількість полігонів, необхідних для створення поверхні, що зменшує навантаження на обчислення та підвищує продуктивність програми, покращуючи її оптимізацію.

Процедурне текстурування - створення текстур з урахуванням математичних алгоритмів. Воно дозволяє створювати складні та реалістичні

текстури, які можуть бути анімовані та змінені у реальному часі. Процедурне текстурування може використовуватися для створення текстур для поверхонь, таких як каміння, дерево, вода та багато інших.

Фотографічне текстурування – це використання фотографій або зображень для створення текстур. Це дозволяє створювати текстури, які мають високий рівень деталізації та реалістичності. Фотографічне текстурування може бути використане для створення текстур для поверхонь, таких як цегла, камінь, дерево та трава.

UV-мапінг - це метод текстурування, який використовує двовимірні зображення, які називаються UV-картами, для створення текстур на поверхні тривимірного об'єкта. UV-мапінг дозволяє більш точно контролювати розміщення текстур на поверхні об'єкта і може використовуватися для створення деталізованих текстур для об'єктів, таких як меблі, транспортні засоби та архітектурні елементи.

Проекційне текстурування - це метод, який використовує проекційне обладнання для нанесення зображення на поверхню об'єкта в реальному часі. Цей метод часто використовується у кіноіндустрії для створення ефектів, таких як хромакей та спеціальні ефекти.

Ручне текстурування – це метод створення текстур з використанням ручних інструментів, таких як пензлі та фарби. Цей метод часто використовується в анімації та в комп'ютерних іграх для створення унікальних та оригінальних текстур.

Анімацією називають - процес створення ілюзії руху шляхом послідовного показу зображень, що називають кадрами, які швидко змінюють один одного. Ці кадри можуть включати зміни форми, розміру, положення чи зовнішнього вигляду об'єктів, які справляють враження руху. Анімація може бути створена як у 2D, так і в 3D просторі. У 3D-анімації об'єкти створюються в тривимірному просторі і можуть бути анімовані за допомогою кадрової анімації або різних технік морфінгу. Морфінг використовується для зміни форми об'єктів, а кадрова анімація

використовується для створення руху об'єктів у просторі. Анімація широко використовується не тільки у кіно, а і в різних областях, таких як мультфільми, ігри, рекламні ролики, відео та телевізійні шоу. У сучасній комп'ютерній графіці анімація часто поєднується з іншими технологіями, такими як тривимірне моделювання, композитинг, цифрова камера і т.д., що дозволяє створювати реалістичні візуальні ефекти. Одним із ключових елементів анімації є крива руху, адже вона являє собою графічне уявлення зміни позиції об'єкта у часі. Вона використовується для визначення траєкторії руху об'єкта та його швидкості. Крива руху може бути змінена, щоб змінити швидкість, додати прискорення, уповільнити чи поміняти напрямку руху об'єкта.

1.1.4. Основні принципи ключування та корекції хромакею

Ключування хромакею у кінематографі - це процес виділення об'єктів на передньому плані від хромакейного фону, щоб їх можна було замінити на інше тло або інтегрувати в інші візуальні елементи. Цей процес вимагає використання спеціальних методів та інструментів, щоб досягти точного виділення об'єктів та забезпечити їх плавний перехід на новий фон. Спочатку вибирається хромакейний фон, який часто має рівномірний зелений або синій колір. Ці кольори обираються, оскільки вони мають мінімальну спільну складову з кольором шкіри або іншими об'єктами, які необхідно виділити. Після вибору фону проводиться зйомка, де об'єкти переднього плану фіксуються на хромакейному фоні. Важливо мати належне освітлення для уникнення тіней та нерівномірностей на фоні, що може вплинути на якість ключування.

Процес ключування починається одразу після зйомки, у ньому використовується програмне забезпечення або комп'ютерні алгоритми. Ці інструменти аналізують кольори пікселів у відео та порівнюють їх з певним діапазоном кольорів, які вважаються фоновими. Пікселі, які відповідають фоновому кольору, видаляються або замінюються, тоді як пікселі, що

відносяться до об'єктів переднього плану, залишаються. Різні алгоритми ключування можуть використовувати різні підходи, такі як ключування на основі кольору, форми або текстури.

Коли об'єкти виділені починається корекція кольору об'єктів і це також є невід'ємним етапом, який дозволяє досягнути бажаного візуального ефекту і створити потрібну атмосферу в сцені. Цей процес включає редагування, виправлення та покращення кольору об'єктів, знятих з різних камер або в різних умовах освітлення. Перший крок у корекції кольору об'єктів - це правильне встановлення балансу білого. Це допомагає виправити відтінки кольорів і забезпечити, що білий колір виглядає насправді білим.

Встановлення правильного балансу білого враховується як важливий етап, оскільки він визначає загальну кольорову температуру сцени. Після встановлення балансу білого проводиться корекція тональності. Вона включає регулювання яскравості, контрастності, насиченості та тону зображення для створення бажаної настрою і візуального ефекту. Цей крок допомагає досягнути потрібної якості зображення, забезпечити рівновагу між світлими і темними областями та підкреслити деталі.

Наспутним кроком проводиться корекція кольорового балансу, яка включає редагування окремих каналів кольору, таких як червоний, зелений та синій. Це дозволяє змінювати загальний колорит сцени або підкреслювати певні кольорові акценти. Наприклад, можна збільшити насиченість червоних тонів для підкреслення важливих об'єктів або зменшити зелений тон для створення більш холодного візуального ефекту.

Корекція кольору об'єктів також дозволяє впливати на кольорові характеристики окремих об'єктів у кадрі. Цей процес дозволяє змінити вигляд об'єктів, виправити кольорові недоліки або створити ефекти, які підкреслюють або змінюють сприйняття об'єктів. Корекція кольору об'єктів виконується за допомогою спеціалізованих програм і інструментів, які надають широкий набір функцій і можливостей для точного редагування кольору. Програми, такі як Adobe Premiere Pro, DaVinci Resolve, Final Cut Pro

та інші, дозволяють змінювати кольори, налаштовувати градієнти, криві, використовувати маски та інші інструменти для редагування кольору.

У кінематографі корекція кольору об'єктів також є творчим процесом, який дозволяє створювати певний стиль, настрій та естетику. Вона може відображати відчуття та інтерпретацію оператора або режисера і впливати на сприйняття глядача. Відтворення кольору може бути реалістичним, насиченим, приглушеним або навіть фантастичним, залежно від вимог сценарію та візії фільму.

Далі, використовують контурну корекцію, яка є важливим елементом пост-продакшну в кінематографі. Вона використовується для підвищення чіткості та визначеності об'єктів на екрані шляхом зміни контурів і ліній об'єктів. Цей процес включає в себе редагування тонів та контроль над границями об'єктів для підсилення їх чіткості і роздільної здатності. Контурна корекція допомагає виокремити об'єкти на зображенні, зробити їх більш видимими та детальними, а також покращити їх розмежування від фону. Для контурної корекції в кінематографі використовуються різні інструменти та техніки. Один з найпоширеніших методів - використання фільтрів та плагінів у спеціалізованих програмах для обробки відео. Ці інструменти дозволяють визначати границі об'єктів, регулювати контрастність, яскравість та різні параметри, що впливають на видимість контурів. Одним з основних методів контурної корекції є використання масок.

Маски дозволяють вибирати конкретні області на зображенні і застосовувати до них окремі налаштування контурної корекції. Наприклад, можна створити маску для об'єкта, який потребує більшої чіткості, і змінити контрастність, різкість або яскравість лише в цій області, не впливаючи на решту зображення. Крім того, існують спеціалізовані алгоритми обробки зображень, такі як алгоритм "унішення контурів" (unsharp mask), який застосовується для підвищення контрастності на контурних деталях об'єктів. Цей алгоритм включає розмиття зображення, вилучення деталей високої

частоти та подальше їх підсилення, що призводить до збільшення різкості контурів.

Контурна корекція також може включати використання додаткових ефектів, таких як обробка з м'яким ефектом, що забезпечує більш гладкі та менш різкі контури. Це може бути корисно для створення більш м'якого і естетичного вигляду об'єктів. Окрім технічних аспектів, контурна корекція є також творчим процесом, що дозволяє досягти певного стилю, настрою та естетики візуальної композиції. Вона може відтворювати реалістичний вигляд об'єктів, забезпечувати їх більш видиму та роздільну здатність або, навпаки, створювати спеціальні ефекти, такі як контурна підсвітка або зміна форми об'єктів. Таким чином, контурна корекція в кінематографі направлена на покращення вигляду об'єктів на екрані.

Подальша робота стосується додавання тіней та відблисків і використовуються для створення реалістичного освітлення, поглиблення простору та надання об'єктам глибини і реалізму. Тіні додаються для імітації впливу джерел світла на об'єкти в сцені. Вони можуть бути м'якими, розпливчастими або виразними, залежно від характеру джерела світла та поверхні об'єкта. Тіні можуть підкреслювати форму об'єктів, створювати відчуття глибини та простору, а також надавати сцені більш реалістичного вигляду.

Для додавання тіней використовуються різні техніки, залежно від потреб сцени та вимог режисера. Один з популярних методів - використання програм та плагінів для комп'ютерної графіки, які дозволяють створювати реалістичні тіні шляхом налаштування параметрів, таких як напрям, інтенсивність та розмір тіні.

Відблиски використовуються для створення відображення світла на поверхнях об'єктів. Вони можуть бути яскравими, розпливчастими або контрастними, в залежності від типу поверхні та характеру світла. Відблиски додають реалізм до сцени, відображаючи вплив джерел світла на оточуючі об'єкти.

Для додавання відблисків також використовуються спеціалізовані програми та плагіни, які дозволяють налаштовувати параметри відблисків, такі як розмір, інтенсивність, фокусування та розташування відблиску на поверхні. Це дозволяє створювати реалістичні відблиски, які відображають форму та характеристики джерела світла.

При додаванні тіней та відблисків важливо враховувати особливості сцени, освітлення та характеристики об'єктів. Це вимагає детального аналізу світлових джерел, їх розташування, інтенсивності та кольорової температури. Також потрібно враховувати матеріали та текстури поверхонь об'єктів, які впливають на сприйняття тіней та відблисків. Крім того, можуть застосовуватися додаткові техніки для покращення реалізму тіней та відблисків. Наприклад, використання масок та альфа-каналів дозволяє точно визначити області, на які впливають тіні та відблиски, і забезпечити більш точне позионування та вигляд ефектів. Також можуть використовуватися техніки блендингу та розмивання для забезпечення плавного переходу між освітленою та затіненою областями.

Додавання тіней та відблисків в кінематографі вимагає уважного планування, візуального аналізу та технічних навичок. Ці ефекти допомагають створити більш реалістичну та привабливу візуальну презентацію, надають об'єктам глибину та реалізм, а також впливають на настрій та естетику кіношедеврів. Відповідно підготовлені тіні та відблиски допомагають створювати вражаючі зображення, які переносять глядача у фантастичний світ кіно.

Додавання тіней та відблисків робить зображення більш реалістичними, динамічними та виразними. При роботі з тінями та відблисками важливо враховувати контекст сцени та задум режисера. Тіні можуть використовуватися для створення певної атмосфери, підкреслення настрою або виділення певних об'єктів у кадрі. Вони можуть мати різний ступінь прозорості, інтенсивності та форми, що залежить від бажаного ефекту.

Додавання відблисків допомагає створити візуальну глибину та реалізм. Відблиски можуть бути яскравими, блискучими або розсіяними, залежно від поверхні об'єкта та характеристик джерела світла. Вони можуть додати блиск і витонченість до сцени, підкреслити деталі або створити ефект засліплення.

Для додавання тіней та відблисків використовуються різноманітні техніки і інструменти. Це може бути ручна робота з використанням програмного забезпечення для редагування зображень, а також використання спеціальних ефектів і методів комп'ютерної графіки.

Команда пост-продакшну працює над кожним кадром, щоб забезпечити точне позиціонування тіней та відблисків, реалістичний перехід між освітленою та затіненою областями, а також підкреслити форму та текстуру об'єктів. Відблиски, зі свого боку, додають блиск і сяйво на поверхні об'єктів, що відображають світло. Вони можуть бути яскравими та блискучими, або розсіяними та ніжними, залежно від характеру джерела світла та властивостей поверхні.

Відблиски можуть створювати реалістичний ефект вологості, скла, металу чи інших матеріалів, а також підкреслювати текстурні деталі та форму об'єктів. Для додавання тіней та відблисків використовуються різні методи. Це може включати ручну ретуш або використання спеціального програмного забезпечення для редагування зображень, комп'ютерну графіку або використання фізичних ефектів під час зйомки.

Деякі програми мають спеціалізовані інструменти для створення тіней та відблисків, які дозволяють змінювати їхню форму, інтенсивність, колір та інші параметри. Процес додавання тіней та відблисків вимагає вміння сприймати та розуміти освітлення сцени, розташування джерел світла та їхній вплив на об'єкти. Важливо враховувати візуальну консистентність тіней та відблисків в усій сцені, щоб забезпечити їхню реалістичність і натуральність. Також потрібно враховувати перспективу і геометрію сцени,

щоб правильно позиціонувати тіні та відблиски відносно об'єктів і джерел світла.

Пост-продакшн спеціалісти, такі як композитори, художники по світлу та колористи, мають велике значення у створенні ефектів тіней та відблисків. Вони використовують різні інструменти, програми та техніки, щоб досягти бажаного ефекту. Наприклад, вони можуть використовувати маски та альфа-канали для точного контролю над областями, на які впливають тіні та відблиски, або використовувати методи розмивання та блендингу, щоб забезпечити плавний перехід між світлими і затіненими ділянками.

Композиція та інтеграція є важливими елементами в процесі створення візуальних ефектів в кинематографі. Ці поняття описують способи організації образу та його взаємодії з оточуючим середовищем для створення злитого, гармонійного зображення. Композиція визначає розташування та взаєморозміщення об'єктів, персонажів та інших елементів в кадрі. Вона визначає геометрію кадру, розташування об'єктів у просторі, рівень деталізації та ієрархію елементів.

Головна мета композиції - створити збалансоване та естетично привабливе зображення, що сприймається глядачем зручно та приємно. Елементи композиції включають кадрування, лінії, форми, кольори, текстури та пропорції. Кадрування визначає межі кадру і вибір пропорцій, таких як плани (широкий, середній, близький), а також кут зйомки (високий, низький, рівний). Лінії та форми впливають на спрямованість погляду глядача, створюють глибину та рух у кадрі. Кольори та текстури допомагають створити атмосферу та відтворити емоційний настрій.

Інтеграція відноситься до процесу злиття візуальних ефектів зі зйомкою, таким чином, щоб вони створювали враження, що вони є частинами одного цілого. Це вимагає уваги до деталей, відповідності освітлення та кольорової палітри, а також аналізу та врахування властивостей реального середовища.

Робота з тінями завершується і наступним кроком стає процес інтеграції, який поділений на декілька етапів, що передбачають взаємодію візуальних ефектів із зйомкою:

1. Аналіз зйомки: спочатку важливо ретельно дослідити вихідний матеріал, зокрема камерні кути, освітлення, кольорову гаму та загальну атмосферу сцени. Це дозволить зрозуміти контекст і настрої, які треба врахувати при інтеграції візуальних ефектів.
2. Оптична відповідність: щоб досягти реалістичного об'єднання зйомки та візуальних ефектів, необхідно забезпечити відповідність оптики, освітлення та колірної палітри. Це може включати корекцію тонів, контрасту та кольору, щоб зробити ефекти більш природними і відповідними зйомці.
3. Розміщення об'єктів: важливо правильно розмістити візуальні ефекти у просторі зйомки, враховуючи перспективу, освітлення та глибину поля. Це може включати зміну розмірів, позицій та орієнтації елементів, щоб вони гармонійно вписувалися в зйомку.
4. Контактна точність: якщо візуальний ефект взаємодіє з реальними об'єктами або персонажами, важливо забезпечити точний контакт і взаємодію між ними. Це може означати врахування тіней, відблисків, взаємодії світла та фізичних властивостей.
5. Композитинг: після розміщення та інтеграції візуальних ефектів в кадрі виконується процес композитингу, під час якого різні елементи зображення (зйомка та візуальні ефекти) поєднуються в одне ціле. Композитинг включає в себе роботу з альфа-каналами, масками, режимами злиття та іншими інструментами, що дозволяють змішувати, налаштовувати і контролювати різні елементи зображення. Під час композитингу ретельно працюють над гармонізацією кольору, контрасту та освітлення між різними елементами, щоб досягти єдиного візуального стилю. Застосовуються різні корекції, наприклад, колірна корекція,

зменшення чи збільшення рівнів яскравості, налаштування тону та насиченості, щоб забезпечити злиття та гармонію між реальними об'єктами та візуальними ефектами.

В процесі композитингу також можуть використовуватися техніки ретуші, заміни фону, видалення нежаданих елементів та інші корекції для поліпшення загального зображення та створення бажаного ефекту.

Композиція та інтеграція в кінематографі вимагають уваги до деталей, технічної майстерності та творчого бачення. Основною метою є створення злитого, реалістичного та естетичного зображення, де візуальні ефекти і реальні об'єкти взаємодіють так, що глядач навіть не помічає розділення між ними.

Під час композиції важливо враховувати пропорції, перспективу, лінії та форми об'єктів. Відповідне розташування та взаємодія елементів у кадрі допомагає створити глибину, рух та візуальну цікавість. Композиція також включає в себе вибір кадру, кута зйомки та плану, що допомагає передати емоції, створити настрій та виразно викласти задум режисера.

Інтеграція вимагає уваги до деталей, освітлення, колірної гами та оптичної відповідності. Візуальні ефекти повинні гармонійно вписуватися у зйомку, мати відповідність оптики та освітлення, а також бути візуально злитими з реальними об'єктами. Важливо досягти єдності в кольорі, контрасті та освітленні між різними елементами, щоб створити враження, що вони існують в одному просторі та часі.

Для досягнення успішної композиції та інтеграції візуальних ефектів важливо мати добре сплановану продакшн-студію та команду фахівців з візуальних ефектів, композитингу, кольорокорекції та інших візуальних технологій. Кожен член команди вносить свою унікальну експертизу та сприяє створенню візуальної цілісності.

Важливо також враховувати творчі задуми режисера та зберігати єдність стилю, настрою та художнього виразу в усьому процесі створення фільму. Композиція та інтеграція вимагають багато уваги до деталей,

технічної точності та творчого підходу для досягнення максимального ефекту та задоволення глядача.

Завдяки композиції та інтеграції в кінематографі створюються захоплюючі, вражаючі та магічні світи, де візуальні ефекти і реальність переплітаються в одне незабутнє візуальне досвід. Це дозволяє режисерам та творчим командам розкрити свою фантазію, створити неповторні образи та перенести глядачів в нові виміри кіношедеврів.

Зібравши та поєднавши усі матеріали лишається тільки довести все до ідеалу і цей етап виконується у фінальній обробці. Це є заключним етапом і він включає ряд технічних та творчих процесів, які мають на меті досягнути бажаного вигляду та якості фінального продукту.

Одним з ключових елементів фінальної обробки є генеральна колірна корекція. Цей процес включає в себе регулювання колірної палітри та налаштування кольорових тонів фільму. Він може бути використаний для створення певної атмосфери, передачі настрою або наголошення на певних деталях. Колірна корекція також допомагає забезпечити консистентність кольорів у всьому фільмі.

Іншим важливим аспектом є монтаж звуку. Під час фінальної обробки звукові ефекти, діалоги та музика синхронізуються та змішуються, щоб створити балансований та іммерсивний звуковий доріжку. В цьому процесі також можуть застосовуватися різні ефекти звуку, наприклад, ехо, звуки оточуючого середовища або спеціальні звукові ефекти.

Під час фінальної обробки можуть бути застосовані спеціальні ефекти та візуальні покращення. Це можуть бути комп'ютерні графічні ефекти, ретуші образу, видалення нежаданих об'єктів або заміна зеленого екрану. Всі ці процеси допомагають покращити якість зображення та створити більш реалістичний або фантастичний вигляд фільму.

Крім технічних аспектів, фінальна обробка також включає остаточне редагування фільму, збалансування його структури та ритму. Режисер та монтажер працюють разом, щоб вирішити питання темпу, хронології подій,

показувати сюжетні лінії та розвивати персонажів. Вони можуть вирішувати, які сцени залишити, а які видалити, які дії показати з різних ракурсів та які моменти підкреслити для досягнення максимального емоційного ефекту.

Після завершення редагування, фінальний фільм може пройти кілька раундів тестового перегляду та звукової корекції. Це дозволяє збирати фідбек від глядачів або тестової аудиторії та внести останні зміни або виправлення, які можуть бути необхідні.

Коли фінальна версія фільму готова, вона може бути експортована у різні формати відповідно до вимог кінотеатрального показу, телевізійного трансляції або веб-платформ. У цьому процесі важливо забезпечити високу якість зображення та звуку, а також правильну кодировку для відтворення на різних пристроях. Завдяки фінальній обробці в кінематографі який поєднує у собі творчий та технічний процеси, стає можливим створення кінематографічний шедевр, де кожен крок має велике значення для створення наочного та емоційного досвіду для глядачів.

1.1.5. Використання різних форматів відео та зображень у кіноіндустрії

Формати збереження відео та зображень використовуються для того, щоб зберігати дані відео та зображень в файловій формі, яку можна перенести на інші пристрої та зберегти для подальшого використання. У кіноіндустрії використовуються різні формати збереження зображень і відео в залежності від їх призначення та вимог до якості. Цих форматів велика кількість і у кожного з них є свої недоліки та переваги, свої обмеження та унікальні функції, тому деякі з найпоширеніших форматів для зберігання і обробки відео та зображень в кіноіндустрії буде наведено нижче, щоб на їх прикладі можна було роздивитися їх функціонал та зрозуміти на якій етапі і при яких обставинах та вимогах використовуються ті чи інші формати:

1. RAW цей формат має досить символічну назву, тому як відноситься до формату запису цифрового відео без стиснення. Це означає, що

матеріал, що знімається, зберігається в необробленому вигляді з мінімальною обробкою і стисненням, щоб зберегти максимальну кількість інформації про сцену. Файли у форматі RAW записуються в некомпресованому вигляді і містять повний динамічний діапазон, колірний простір та роздільну здатність. Це робить формат raw кращим вибором для професійної обробки та редагування відеоматеріалу у кіноіндустрії. Незважаючи на те, що формат RAW є стандартом для збереження високоякісного відео в кіноіндустрії, він не є універсальним форматом і може відрізнитися залежно від виробника обладнання та моделі камери.

2. TIFF (Tagged Image File Format) – це формат файлів для зберігання растрових зображень. Формат TIFF був розроблений компанією Aldus (нині Adobe) у 1986 році і є одним із найпоширеніших форматів для зберігання зображень. У кіноіндустрії формат TIFF використовується для зберігання та передачі високоякісних цифрових зображень. TIFF використовується як формат між різними етапами постпродакшну і для створення фінальних версій зображень, таких як постери, рекламні банери та інші. Основною перевагою формату TIFF в кіноіндустрії є можливість збереження зображень без стиснення і без втрат якості. Це дозволяє зберегти максимальну кількість інформації про колір та деталізацію зображення, що важливо для професійної обробки та редагування. Файли у форматі TIFF у кіноіндустрії зазвичай містять високу роздільну здатність та широкий динамічний діапазон, що дозволяє створювати якісні кадри для цифрових кінопроекторів. TIFF також підтримує метадані, що спрощує ідентифікацію, організацію та керування великими обсягами зображень, що використовуються у кіноіндустрії. Однак TIFF також має деякі обмеження, пов'язані з великим розміром файлів. Файли у форматі TIFF можуть бути дуже великими та займати багато місця на жорстких дисках. Це може

стати проблемою при передачі та зберіганні великих обсягів даних у кіноіндустрії. Тому, як альтернатива, виробники можуть використовувати стислі формати, такі як JPEG2000, для зберігання та передачі зображень у кіноіндустрії.

3. Apple ProRes - це формат відеофайлів, розроблений Apple для використання у кіноіндустрії. Він був вперше представлений у 2007 році та призначений для збереження та передачі високоякісних відеофайлів у кіноіндустрії. Apple ProRes дозволяє зберігати відео без стиснення та втрати, що дозволяє зберегти високу якість відео і при цьому зменшити розмір файлу. Формат ProRes підтримує різні дозволи, включаючи стандартні, HD, 2K, 4K і навіть вище. Він також підтримує різні формати кольору, включаючи Res. 709, Res. 2020 та інші. Однією з головних переваг формату Apple ProRes є його висока швидкість роботи. Він дозволяє професійним відеомонтажерам швидко та легко обробляти та редагувати відео, що важливо для кіноіндустрії, де час – це гроші. Також формат ProRes підтримується багатьма програмами для редагування відео, включаючи Final Cut Pro, Adobe Premiere та інші. Крім того, Apple ProRes також має менший розмір файлів, ніж формати без стиснення, що дає змогу зменшити обсяг зберігання даних. Це зручно для професійних кінематографістів, які можуть знімати багато матеріалу та мати обмежений об'єм сховища. У той же час формат ProRes зберігає високу якість зображення, що є важливою вимогою для кіноіндустрії.
4. DPX (Digital Picture Exchange) - це формат файлів для зберігання та обміну цифровими зображеннями у професійній кіно- та відеоіндустрії. DPX був розроблений в 1984 для уніфікації обміну цифровими кіноплівками між різними комп'ютерними системами, що використовуються на кіновиробництві. Формат DPX підтримує різні роздільні здатності зображень (від SD до 4K і більше), глибину

кольору (від 8 до 16 біт на канал) та різні колірні моделі (RGB, YUV, СМУК). Він призначений для збереження цифрових зображень як стиснені послідовності кадрів. Основними перевагами формату DPX є збереження максимально можливої якості зображення, можливість роботи із зображеннями на різних платформах та в різних програмах і можливість обробки та редагування зображень із збереженням вихідної якості.

Формат DPX широко використовується в кіно- та відеоіндустрії для зберігання, передачі та обробки цифрових зображень, але цей формат завдяки своїй універсальності також використовується в наукових та медичних дослідженнях, а також у графічному дизайні та анімації.

1.2. Призначення розробки та галузь застосування

Призначення розробки програмного забезпечення для роботи з хромакеєм полягає в створенні інструментів, які допомагають виконувати ефективну та точну обробку зображень і відео з використанням хромакею. Головною метою такого програмного забезпечення є створення візуальних ефектів, заміна фону і створення сцен, які були б неможливі без використання цієї техніки. Розробка програмного забезпечення для роботи з хромакеєм має широку галузь застосування.

Кіноіндустрія активно використовує хромакей для створення сцен, які були б неможливі або дорогими в реальних умовах. Завдяки розробці програмного забезпечення для хромакею, створення візуальних ефектів, фантастичних місць та реалістичних фонів стало легшим і доступнішим.

Багато телевізійних передач, новинних програм, музичних відео та рекламних роликів використовують хромакей для створення ефектів заміни фону та створення віртуальних сцен.

Розробка програмного забезпечення для хромакею також знаходить своє застосування в онлайн-контенті. Багато відеоблогерів, стримерів та контент-мейкерів використовують хромакей для створення професійного вигляду

своїх відео та стрімів, дозволяючи їм замінити фон на щось більш цікаве або відповідне їхній тематиці.

Хромакей технологія є важливим компонентом у створенні віртуальної реальності та розширеної реальності. Розробка програмного забезпечення для хромакею допомагає створювати інтерактивні віртуальні середовища, де користувачі можуть взаємодіяти з об'єктами та сценами у реальному часі.

Хромакей широко використовується в професійній фотографії та студійній роботі для створення художніх та креативних зображень. Розробка програмного забезпечення для хромакею допомагає фотографам використовувати цю техніку швидше й ефективніше. Розробка програмного забезпечення для роботи з хромакеєм відкриває безліч можливостей для творчих і професійних застосувань. Ця технологія постійно розвивається, що дозволяє розширювати межі творчості та створювати вражаючі візуальні ефекти.

1.3. Підстава для розробки

Підставами для розробки та виконання кваліфікаційної роботи є:

- освітня програма 122 Комп'ютерні науки;
- навчальний план та графік навчального процесу;
- наказ ректора Національного технічного університету «Дніпровська політехніка» № 350-с від 16.05.2023 р;
- завдання на кваліфікаційну роботу на тему «Розробка програмного забезпечення для засосування комп'ютерної графіки у сфері кінематографу за допомоги хромакею».

1.4. Постанова завдання

В данній кваліфікаційній роботі було необхідно створити програмне забезпечення, яке матиме можливість взаємодіяти із технологією хромакею.

У якості вимог до програми можуть виступати наступні чинники:

- інтуїтивно зрозумілий інтерфейс;
- можливість завантажувати підготовлені зображення та видаляти їх;
- підтримка роботи хромакею;
- підтримка формату TIFF та можливість збереження результату;
- вбудоване вікно захоплення камери.

1.5. Вимоги до програми або програмного виробу

1.5.1. Вимоги до функціональних характеристик

До вимог щодо функціональних характеристик програмного забезпечення входять:

1. Запуск програми: користувач може запустити програму "ChromaKey App" на своєму пристрої.
2. Налаштування хромакею: користувач може ввести значення нижньої та верхньої меж кольору для хромакею. Введені значення меж кольору відображаються у відповідних полях вводу.
3. Налаштування корекції кольору: користувач може ввести значення яскравості, контрастності та насиченості для корекції кольору. Введені значення параметрів корекції кольору відображаються у відповідних полях вводу.
4. Завантаження фонового зображення: користувач може вибрати фонове зображення з файлової системи свого пристрою. Обране фонове зображення відображається у програмі.
5. Видалення фону: користувач може натиснути кнопку "Remove Background", щоб видалити фон з поточного кадру. Фон у поточному кадрі замінюється на прозорий фон.
6. Переключення хромакею: користувач може натиснути кнопку "Toggle ChromaKey", щоб увімкнути або вимкнути ефект хромакею. При включеному режимі хромакею, фон у поточному кадрі

замінюється на обране фонове зображення з використанням заданих меж кольору. При вимкненому режимі хромакею, застосовується лише корекція кольору до поточного кадру.

7. Застосування параметрів корекції кольору: користувач може натиснути кнопку "Apply", щоб застосувати введені параметри корекції кольору до поточного кадру. Застосовані параметри корекції кольору відображаються у відповідних полях вводу.
8. Збереження результату: користувач може натиснути кнопку "Save Result", щоб зберегти поточний кадр з врахуванням застосованих ефектів. Відкривається діалогове вікно для вибору місця збереження та введення назви файлу. Результат зберігається у форматі TIFF.
9. Закриття програми: користувач може закрити програму "ChromaKey App" шляхом закриття головного вікна програми.

Ці функціональні характеристики дозволять користувачу взаємодіяти з програмою, налаштовувати ефекти хромакею та корекції кольору, вибирати фонове зображення та зберігати результати роботи програми.

1.5.2. Вимоги до складу та параметрів технічних засобів

Вимоги до параметрів техніки, яка використовується при роботі з хромакеєм у кінематографі, включають вимогу до високої якості зображення. Це означає, що технічне обладнання повинно мати високу роздільну здатність, здатність зберігати деталізацію зображення для точного відокремлення зеленого або синього фону від об'єктів. Крім того, воно повинно мати велику чутливість до світла, щоб збирати достатню кількість світлової інформації з фону та об'єктів.

Широкий динамічний діапазон також є важливим, оскільки обладнання повинно відтворювати деталі як у темних, так і у яскравих частинах зображення. Мінімальний рівень шуму теж вимагається для збереження чіткості та чистоти зображення.

Ці вимоги допомагають забезпечити високу якість зображення та точне відокремлення фону та об'єктів, що є необхідним при роботі з хромакеєм у кінематографі. Але окрім цього залізо повинно мати високий рівень потужності, яка використовується при роботі з хромакеєм у кінематографі. Це означає, що технічне обладнання повинно мати достатню кількість обчислювальних ресурсів для ефективної обробки великих обсягів даних та відеоматеріалів.

Робота з хромакеєм вимагає обробки великої кількості інформації, такої як відеозаписи з камер, відслідковування руху об'єктів, відокремлення фону та інтеграція з новим фоном. Для цього необхідні потужні обчислювальні ресурси, такі як процесори, графічні прискорювачі та оперативна пам'ять. Достатня оброблювальна потужність дозволяє забезпечити швидку та ефективну обробку великих обсягів даних, зменшує час обробки та покращує продуктивність процесу роботи з хромакеєм. Важливо мати належно налаштовану та оптимізовану систему, яка здатна обробляти дані в реальному часі і забезпечувати швидкість та точність виконання необхідних алгоритмів. Таким чином, вимога до достатньої оброблювальної потужності техніки є важливою при роботі з хромакеєм у кінематографі, оскільки це допомагає забезпечити ефективну та швидку обробку даних, що є ключовим елементом для успішного використання цієї технології.

Окрім внутрішньої потужності, технічне обладнання повинно мати міцну та стійку конструкцію, яка зможе витримувати вібрації під час руху. Це може бути досягнуто за допомогою стабільних кріплень, амортизуючих матеріалів або інших механізмів, які зменшують вплив зовнішніх факторів на обладнання. Додатково, технічне обладнання може мати вбудовані функції або алгоритми, які допомагають компенсувати рух і вібрації. Наприклад, системи можуть використовувати активну стабілізацію зображення або електронну обробку зображення для зниження впливу руху на якість

зображення. Це може включати алгоритми стабілізації зображення, корекцію зміщення та компенсацію руху об'єктів на зображенні.

Стійкість до вібрацій та руху є важливою вимогою, оскільки допомагає забезпечити якість зображення та точність відокремлення фону та об'єктів, навіть у рухомих умовах зйомки. Це дає змогу створювати чисті, реалістичні та професійні візуальні ефекти з використанням хромакея в кінематографі. Але сама конструкція має бути міцною і при цьому не громіздкою, щоб не виникало труднощів при її транспортуванні, таким чином забезпечуючи її гнучкість і мобільність.

Таким чином обладнання повинно бути здатним швидко адаптуватися до різних зйомочних ситуацій і забезпечувати мобільність для зручного використання на знімальному майданчику. Гнучкість вимагає, щоб технічне обладнання мало різноманітні налаштування та опції для адаптації до різних умов зйомки. Наприклад, це можуть бути регульовані камерні налаштування, різноманітні об'єктиви з різними фокусними відстанями, а також можливість вибору різних хромакейних засобів та освітлення.

Гнучкість дозволяє операторам вибирати оптимальні параметри для отримання бажаного результату. Мобільність означає, що технічне обладнання повинно бути портативним і зручним для переміщення. Це важливо для зйомок на різних майданчиках або в локаціях з обмеженим простором. Таке обладнання може бути компактним, легким, зручно збирається і має можливість швидкого монтажу і демонтажу. Важливо, щоб мобільність не впливала на якість роботи обладнання і зберігала його функціональні можливості.

Гнучкість і мобільність є важливими вимогами для потужності техніки, яка взаємодіє з хромакеєм у кінематографі. Ці характеристики дозволяють операторам ефективно працювати в різних зйомочних умовах, швидко налаштовувати обладнання та забезпечувати гнучкість в реалізації творчих ідей і проектів. Щоб будь – які потреби могли бути реалізовані, ретельна увага уділяється надійності.

Надійність є однією з ключових вимог до потужності техніки, оскільки кіноіндустрія вимагає високої якості та надійності обладнання, це важливо забезпечити, щоб уникнути можливих помилок та втрати даних під час зйомок. Надійна техніка має бути стійкою до відмов та несправностей, що дозволяє уникнути перерв у роботі та забезпечити безперебійну зйомку. Для досягнення надійності, потужність техніки повинна бути підтримувана відповідними процедурами обслуговування, регулярним оновленням програмного забезпечення та контролем якості.

Також важливо, щоб обладнання мало достатню міцність та довговічність для роботи в інтенсивному режимі. Висока якість компонентів, стійкі матеріали та витривалі конструкції допомагають запобігти поломкам та зберегти працездатність протягом тривалого часу. Надійна техніка також повинна мати вбудовані заходи безпеки, такі як захист від перенапруги, перегріву або короткого замикання. Це допомагає запобігти можливим пошкодженням обладнання та забезпечити безпеку для користувачів.

Враховуючи вимогу до надійності, технічне обладнання, яке використовується при роботі з хромакеєм у кінематографі, повинно проходити необхідні перевірки, тестування та сертифікацію, щоб гарантувати його надійну роботу на протязі тривалого часу.

1.5.3. Вимоги до якості та контролю якості візуальних ефектів

Якість візуальних ефектів (VFX) у кінематографі має величезне значення, оскільки їхня недостатня якість може суттєво знизити загальне враження від фільму. Тому контроль якості візуальних ефектів є важливим етапом у процесі створення фільму.

Контроль якості починається із планування візуальних ефектів. Режисер, VFX-супервайзер та інші фахівці обговорюють, які ефекти потрібні, як вони будуть використовуватись у сюжеті та як вони мають виглядати. Потім створюється дизайн-борд, який містить приклади того, як мають виглядати візуальні ефекти. Коли починається виробництво візуальних

ефектів, кожен кадр уважно перевіряється на відповідність до дизайну. Кожен спеціаліст у команді перевіряє свою роботу на різних етапах виробництва, щоб переконатися, що вона відповідає стандартам якості.

Початок контролю якості починається із процесу попереднього перегляду (previs) – це важливий етап у створенні візуальних ефектів (VFX) у кінематографі. Previs - це процес створення простих комп'ютерних моделей та анімації, які використовуються для планування та візуалізації сцен у фільмі перед початком зйомок. Previs допомагає команді творців фільму зрозуміти, як виглядатимуть візуальні ефекти в кінцевому вигляді, і як вони поєднуються зі зйомками на майданчику. Це дозволяє режисеру та іншим членам команди точніше планувати кадри та вибирати відповідні ракурси, а також визначити, які ефекти будуть реалізовані в реальному часі на майданчику, а які будуть створені на комп'ютері у постпродакшні, таке планування дозволяє команді спеціалістів чітко зрозуміти у якому напрямі рухатися завдяки чьому зменшити кількість задіяних ресурсів і підняти якість виробництва.

Далі процес контролю якості відбувається у постпродакшні, що є невід'ємною частиною кіноіндустрії адже цей етап дозволяє переконатися в тому, що остаточний продукт відповідає заданим стандартам і вимогам.

Контроль якості в постпродакшні включає ряд кроків і процедур, які дозволяють оцінити і поліпшити якість фільму. Один із важливих аспектів контролю якості – це перевірка на помилки. Це може включати перевірку корекції кольору, звукових доріжок, візуальних ефектів і монтажу. Щоразу, коли виявляється помилка, вона має бути виправлена, щоб забезпечити відповідність якості. Ще один важливий аспект контролю якості – це перевірка на відповідність стандартам кінематографічної індустрії. Під час контролю якості повинні бути переконливими, що фільм відповідає певним стандартам, таким як формат експорту, стандарти корекції кольору, звукова сумісність та багато інших.

Важливим аспектом контролю якості є зворотний зв'язок із режисером та продюсерами. Це дозволяє їм оцінити якість кінцевого продукту та запропонувати необхідні зміни. Крім того, дизайнери звукової доріжки та візуальних ефектів повинні працювати у тісній співпраці з режисером та продюсерами, щоб задовольнити їхні вимоги та отримати максимально точне уявлення про те, як має виглядати та звучати фільм.

Ще одним необхідним кроком щодо контролю якості є тестування фільму на аудиторії. Це дозволяє отримати зворотний зв'язок від глядачів та оцінити реакцію на фільм. Цей процес може допомогти визначити, які елементи фільму працюють, а які потребують покращення. Етап остаточної перевірки якості у кінематографі – це заключний етап виробництва фільму перед його виходом у прокат. Він включає перевірку якості візуальних ефектів, звуку, монтажу та інших аспектів фільму, щоб переконатися в тому, що остаточний продукт відповідає вимогам і стандартам кінематографічної індустрії, але цей етап найбільш скрупульозний і включає у себе декілька пунктів:

1. Перевірка візуальних ефектів: на цьому етапі здійснюється перевірка якості візуальних ефектів, таких як комп'ютерна графіка, анімація та спецефекти. Це включає в себе перевірку корекції кольору, відображення руху і реалістичності ефектів. Усі вони повинні відповідати заданим стандартам та вимогам.
2. Перевірка звуку: на цьому етапі здійснюється перевірка звукових доріжок, звукових ефектів, музичного оформлення та дублювання. Звук повинен відповідати заданим стандартам якості та повинен бути правильно змонтований, щоб синхронізуватися із зображеннями.
3. Перевірка монтажу: на цьому етапі здійснюється перевірка якості монтажу, щоб переконатися, що сцени коректно переміщуються одна в одну і що темп і ритм фільму відповідають заданим стандартам та вимогам.

4. Загальна перевірка якості: на цьому етапі проводиться загальна перевірка якості, щоб переконатися, що всі аспекти фільму відповідають заданим стандартам та вимогам, а також що немає помилок та дефектів, які можуть негативно вплинути на оцінку якості фільму.
5. Перевірка на відповідність стандартам індустрії: на цьому етапі здійснюється перевірка на відповідність стандартам кінематографічної індустрії, таким як формат експорту, корекція кольорів, звукова сумісність та інші стандарти.
6. Перевірка на аудиторії: на цьому етапі проводиться тестування фільму на аудиторії, щоб отримати зворотний зв'язок і оцінити їх реакцію, щоб робити висновки на майбутні продукти.

1.5.4. Вимоги до хромакею, який взаємодіє з програмним забезпеченням

Вибір кольору хромакею в кіноіндустрії є важливим складником досягнення якісних візуальних ефектів. Правильний вибір кольору хромакею залежить від ряду факторів, включаючи умови освітлення, кольори об'єктів та акторів, а також вимоги постобробки. У цьому тексті ми докладно розглянемо вибір кольору хромакею та наведемо приклади його використання у кіноіндустрії.

Зелений хромакей є найпоширенішим вибором у кіноіндустрії. Це пов'язано з кількома перевагами, які він надає. По-перше, зелений колір знаходиться далі від шкірного тону та багатьох кольорів об'єктів, що полегшує виділення переднього плану при постобробці. Зелений хромакей дозволяє більш точно виділити акторів та об'єкти та замінити фон на будь-яке зображення чи відео, створюючи реалістичні візуальні ефекти. Цей колір широко використовується в багатьох жанрах кіно, включаючи фантастичні блокбастери, телевізійні програми та рекламу.

Приклади використання зеленого хромакею у кіноіндустрії різноманітні. У сценах, де персонажі перебувають у лісі чи віртуальному середовищі, де зелений є природним тлом, зелений хромакей є ідеальним вибором. Розглянемо, наприклад, фантастичний фільм, де головні герої борються із інопланетною загрозою у зеленому лісі. Зйомка з використанням зеленого хромакею дозволяє акторам вільно взаємодіяти з віртуальним оточенням, а постпродакшн команді легко замінити фон на приголомшливі цифрові пейзажі.

Синій хромакей – ще один варіант кольору хромакею, що використовується у кіноіндустрії. Синій хромакей застосовується у випадках, коли зелений колір на фоні або об'єктах створює проблеми при ключуванні. Це може бути пов'язане з наявністю зеленого одягу у акторів або зеленого фону всередині сцени. Синій колір хромакею дозволяє ефективно виділяти передній план та відокремлювати його від фону у таких ситуаціях. Наприклад, у супергеройських фільмах, де персонажі носять зелені костюми або мають зелені елементи у своєму образі, використання синього хромакею може бути кращим вибором. Це дозволяє створити візуально дивовижні ефекти, такі як літаючі герої, спеціальні сили та захоплюючі сутички.

Крім зеленого та синього, у кіноіндустрії також можуть використовуватися інші кольори хромакею залежно від конкретних вимог зйомки та постобробки. Наприклад, червоний і жовтий кольори можуть бути застосовані в деяких сценах, де вони краще поєднуються з палітрою кольорів об'єктів на передньому плані або контрастують з ними для більш ефектного виділення переднього плану. Це дозволяє створювати унікальні і незабутні візуальні образи. Наприклад, у фільмах жахів чи фентезі використання червоного хромакею, адже він може допомогти створити атмосферу незвичайності та містики.

Важливо, що вибір кольору хромакею має бути заснований на аналізі та розумінні контексту зйомки, освітлення та кольорової гами об'єктів та акторів на передньому плані. Також необхідно врахувати вимоги

постобробки та можливості, які надають інструменти для ключування та заміни фону. Консультація з досвідченими фахівцями в галузі візуальних ефектів та постпродакшну допоможе прийняти правильне рішення при виборі кольору хромакею для конкретного проекту.

Розглянемо приклади використання різних кольорів хромакею у кіноіндустрії. У фільмі "Сутінки. Сага" (2008-2012) використовувався синій хромакей для створення візуальних ефектів, пов'язаних з героями перевертнями. Оскільки перевертні мали зелений відтінок, використання зеленого хромакею могло призвести до змішування кольорів та проблем при виділенні переднього плану. Натомість, синій хромакей дозволив легко відокремити перевертнів від фону та створити реалістичні ефекти та трансформації.

У фільмі "Матриця" (1999) режисери брати Вачовські використовували зелену та синю хромакей у різних сценах. Зелений хромакей використовувався для зйомки сцен усередині матриці, де головні герої перебували у віртуальній реальності. Синій хромакей застосовувався для зйомки сцен у реальному світі, де актори взаємодіяли із фізичними об'єктами. Це дозволило створити контраст і чудово виділити різні світи та ситуації у фільмі.

Але не рідко зустрічаються випадки використання геть інших кольорів. Одним із прикладів фільму, де використовувався нестандартний колір хромакею, є "Синя оксамитова" (1986) режисера Девіда Лінча. У цьому фільмі був використаний червоний хромакей для створення містичної та сюрреалістичної атмосфери. Червоний хромакей допомагав відокремити персонажів від фону та створити візуальні ефекти, які відповідали тематиці та стилістиці фільму.

Ще одним прикладом послужить фільм "Скубі-Ду!" (2002) режисера Раггі Блю. У цій комедії з елементами фентезі використали жовтий хромакей для зйомок деяких сцен. Жовтий колір хромакею створював контраст і

виділяв головних героїв – команду Скубі-Ду, надаючи їм яскравості та виразності.

Цей список поповнює ще і сучасний фільм "Месники: Війна нескінченності" (2018) режисерів Ентоні та Джо Руссо. У деяких сценах цього супергеройського блокбастера було використано фіолетовий хромакей. Фіолетовий колір хромакею допомагав створити ефектну візуалізацію для героїв, пов'язаних із Камнем Душі та з деякими космічними елементами фільму. Це лише кілька прикладів фільмів, де було використано нестандартні кольори хромакею.

У кіноіндустрії існує безліч творчих рішень, і вибір кольору хромакею може бути визначений виходячи з вимог сценарію, бажаних візуальних ефектів і атмосфери, що створюється.

Однак окрім кольору хромакею не менш важливим є сам матеріал, а також його якість. Вибір відповідного матеріалу для хромаку є важливим аспектом при зйомці в кіноіндустрії. Якість матеріалу має прямий вплив на результати зйомок та подальший процес постпродакшну. Хромакей - це техніка, що використовує однорідно забарвлений фон, який потім замінюється на інший задній план під час постпродакшну.

При виборі матеріалу для хромаку кілька факторів слід врахувати. По-перше, колір є важливим аспектом. Зелений колір найбільш поширений та універсальний для хромакею. Він найменше зустрічається в природі і добре відокремлюється від шкіри та волосся. Синій колір також може бути використаний, особливо якщо в передньому плані є зелений одяг або зелені об'єкти. Важливо вибрати насичений колір, який буде однорідним по всій поверхні фону.

Однорідність матеріалу також відіграє важливу роль. Хромакей-матеріал повинен мати однорідне забарвлення без видимих плям чи текстур. Будь-які нерівномірності або плями можуть викликати проблеми під час ключування та заміни фону. Матеріал повинен бути рівномірно забарвлений і не мати помітних відмінностей у відтінках кольору. Крім того, слід звернути увагу на

віддзеркалення світла. Бажано вибрати матеріал із мінімальним відображенням світла. Це допоможе уникнути небажаних відображень та відблисків на об'єктах переднього плану, що полегшить подальший процес ключування. Чим менший матеріал, тим краще.

Матовість матеріалу теж є необхідністю. Хромакей-матеріал має бути матовим, щоб мінімізувати відблиски та відображення. Матова поверхня поглинає світло замість відображення його назад на передній план. Це допомагає створити більш реалістичний ефект та забезпечити чіткий поділ між переднім планом та фоном. Матовий матеріал також сприяє кращій інтеграції об'єктів переднього плану з фоном, що замінюється в постпродакшн процесі.

Інший важливий аспект при виборі матеріалу для хромакею - це його фізичні властивості та якість. Матеріал повинен бути достатньо міцним та стійким до пошкоджень, щоб витримувати інтенсивне використання під час зйомок. Він повинен легко монтуватися на стійки чи рами та зберігати свою форму без прогинів чи складок, які можуть створювати небажані тіні на передньому плані. Додатково, слід враховувати простоту використання та зручність матеріалу.

Ідеальний матеріал має бути легким та портативним, щоб його можна було легко транспортувати та встановлювати на знімальному майданчику. Він також повинен бути легко миючим, щоб видалити сліди пилу або бруду, які можуть з'явитися під час зйомок. Важливо пам'ятати, що вибір матеріалу для хромакею має відповідати вимогам конкретного проекту та умовам зйомки.

У деяких випадках може знадобитися використання спеціальних матеріалів, наприклад, хромакей-тканин або плівок, які мають певні властивості для досягнення оптимального результату. Насамкінець, вибір матеріалу для хромакею при зйомках у кіноіндустрії є важливим кроком для досягнення якісного результату. Необхідно враховувати колір, однорідність,

відображення, матовість, фізичні властивості, зручність використання та відповідність вимогам проекту.

Ретельний підбір матеріалу допоможе створити ефективний хромакей та полегшить подальший постпродакшн процес. Насамкінець, слід зазначити, що вибір матеріалу для хромакею при зйомках у кіноіндустрії є важливим кроком для досягнення якісного результату. Необхідно враховувати колір, однорідність, відображення, матовість, фізичні властивості, зручність використання та відповідність вимогам проекту. Ретельний підбір матеріалу допоможе створити ефективний хромакей та полегшить подальший постпродакшн процес.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Функціональне призначення інформаційної системи

Програмне забезпечення "ChromaKey App" призначене для роботи з ефектом хромакею. Основна функціональність програми включає у себе відтворення відеопотоку з веб-камери або іншого джерела в реальному часі, налаштування параметрів хромакею, таких як нижня та верхня межі кольору, налаштування параметрів корекції кольору, таких як яскравість, контраст та насиченість, вибір фонового зображення для заміни відео фону, перемикання режиму хромакею та збереження результату.

Про кожен цей пункт варто поговорити більш докладно. У кодї, реалізовано відтворення відеопотоку з веб-камери в реальному часі, завдяки чому програма використовує відеострім, що надається вебкамерою, як основним джерелом вхідних даних. Кожен кадр з відеостріму обробляється незалежно для застосування хромакею, корекції кольору та заміни фону. Оброблені кадри відображаються на головному вікні програми. Для цього використовується бібліотека OpenCV (cv2). Клас cv2.VideoCapture відповідає за створення об'єкту, який може отримувати відео з джерела, такого як веб-камера, відеофайл або інші джерела відеозапису.

У конструкторі класу ChromaKeyApp, self.cap ініціалізується як об'єкт cv2.VideoCapture(0), де аргумент 0 вказує на веб-камеру за замовчуванням (перший доступний відео пристрій). За допомогою цього об'єкту cap, програма може отримувати кадри з веб-камери. У методі update_frame викликається self.cap.read(), що зчитує наступний кадр з веб-камери. Потім отриманий кадр обробляється, застосовуючи хромакей, корекцію кольору та інші ефекти відповідно до вибраних налаштувань. Оновлений кадр конвертується з формату BGR (використовуваного OpenCV) в RGB (для відображення зображення у вікні Tkinter). Далі, зображення кадру image встановлюється як зображення в Tkinter-етикетці self.frame_label, що

дозволяє відображати кадри в реальному часі у вікні програми. Таким чином, код забезпечує відтворення відеопотоку з веб-камери або іншого джерела в реальному часі та відображення оброблених кадрів у вікні програми.

Деякі налаштування параметрів хромакею, яке пропонує програмне забезпечення відповідає за визначення діапазону кольорів, які будуть замінені на фонове зображення. Це важливо для створення ефекту хромакею, де зелений або обраний колір вважається прозорим, і може бути замінений на інше зображення або фон. У класі `ChromaKeyApp`, параметри хромакею задаються за допомогою двох масивів `self.lower_color` та `self.upper_color`. `self.lower_color` відповідає за нижню межу кольору, а `self.upper_color` - за верхню межу кольору. Ці параметри встановлюються при створенні інтерфейсу програми у відповідних текстових полях.

Користувач може ввести значення для нижньої та верхньої межі кольору, які відповідають діапазону кольорів, які він бажає використовувати для хромакею. Наприклад, в коді задано початкові значення `[40, 50, 50]` для `self.lower_color` і `[80, 255, 255]` для `self.upper_color`. Після того, як користувач внесе зміни до цих параметрів у вікні програми та натисне кнопку «Apply», значення параметрів оновлюються відповідно до введених користувачем значень. Ці оновлені значення будуть використовуватись при обробці кадрів в методі `replace_background`, де виконується хромакей, тобто заміна кольорів у визначеному діапазоні на фонове зображення.

У межах коду, користувач має доступ до налаштування параметрів корекції кольору дозволяє змінювати яскравість, контраст та насиченість кольорів об'єктів на відео. Це дозволяє вам впливати на візуальні характеристики зображення та підкреслювати або змінювати кольори, що використовуються під час хромакею.

У класі `ChromaKeyApp`, параметри корекції кольору встановлюються за допомогою трьох змінних: `self.brightness`, `self.contrast` та `self.saturation`. Ці параметри також встановлюються при створенні інтерфейсу програми у відповідних текстових полях. Користувач може ввести значення для кожного

з цих параметрів, щоб визначити бажану яскравість, контраст та насиченість кольорів у відео.

Після того, як користувач внесе зміни до цих параметрів у вікні програми та натисне кнопку «Apply», значення параметрів оновлюються відповідно до введених користувачем значень. Ці оновлені значення будуть використовуватись при обробці кадрів в методі `replace_background`, де виконується корекція кольору, що включає зміну яскравості, контрасту та насиченості кольорів зображення.

У рамках коду, передбачена функція «перемикання режиму хромакею», яка дає можливість користувачеві увімкнути або вимкнути ефект хромакею, що дозволяє замінити фон на відео. У класі `ChromaKeyApp` існує змінна `self.chroma_key_enabled`, яка слугує прапорцем для визначення стану хромакею. Початково цей прапорець встановлений у значення `False`, що означає, що ефект хромакею вимкнений.

Користувач може перемкнути стан хромакею, натиснувши на кнопку «Toggle ChromaKey» у вікні програми. Кожне натискання цієї кнопки змінює значення прапорця `self.chroma_key_enabled`. Якщо прапорець має значення `True`, то хромакей увімкнено; якщо значення `False`, то хромакей вимкнено. Коли хромакей увімкнено, програма використовує параметри нижньої та верхньої межі кольору, встановлені користувачем, для визначення діапазону кольорів, які будуть замінені. Далі застосовується ефект хромакею до кадру в методі `replace_background`. Якщо хромакей вимкнено, то фон не замінюється, але можуть все ще використовуватись параметри корекції кольору (яскравість, контраст, насиченість), які також можуть бути встановлені користувачем. Це дозволяє користувачеві змінювати візуальні характеристики зображення без заміни фону.

Загалом, перемикання режиму хромакею дозволяє користувачеві контролювати відображення відео з ефектом заміни фону або без нього, залежно від їхніх потреб та вимог. У контексті коду розробленого програмного забезпечення, «збереження обробленого відео з заміненим

фоном у форматі TIFF» означає, що програма може зберегти оброблене відео у форматі TIFF замість відтворення його на екрані.

Для збереження використовується бібліотека OpenCV, яка має функціональність роботи з відеофайлами. Для цього створюється об'єкт `cv2.VideoWriter`, який дозволяє записувати відеофайли з використанням різних кодеків та налаштувань. При ініціалізації об'єкта `cv2.VideoWriter` передаються наступні аргументи:

- ім'я файлу, в який буде збережено відео. Наприклад, 'output.tiff';
- чотирьохсимвольний кодек, що визначає формат збереження відео. У даному випадку 'tiff' вказує на формат TIFF;
- частота кадрів (FPS), яка визначає, скільки кадрів в секунду буде записано у відео. Наприклад, 30 означає 30 кадрів на секунду;
- розмір кадра у форматі (ширина, висота). Наприклад, (640, 480) вказує на ширину 640 пікселів та висоту 480 пікселів.

Після ініціалізації об'єкта `cv2.VideoWriter`, у циклі `while` після обробки кадру, якщо ефект хромакею увімкнено та фон був замінений, викликається метод `out.write(frame)`, де `frame` є обробленим кадром. Цей метод додає кадр до вихідного відеофайлу. Після завершення обробки всіх кадрів відео та виходу з циклу `while`, об'єкт `cv2.VideoWriter` закривається за допомогою методу `out.release()`. Це виконується для завершення запису вихідного відеофайлу.

Як результат, після виконання коду, оброблене відео з заміненим фоном буде збережено у форматі TIFF з вказаним ім'ям файлу та налаштуваннями кодеку, частоти кадрів та розміру кадра.

2.2. Опис застосованих математичних методів

У цьому коді використовуються деякі математичні методи для досягнення різних ефектів хромакею і корекції кольору.

Кольоровий простір HSV призначений для використання хромакею, зображення перетворюється з колірного простору BGR в HSV. Компоненти HSV (відтінок, насиченість, значення) надають можливість визначати діапазон кольорів для видалення або заміни.

Порігова обробка (Thresholding) використовується для створення маски, яка виділяє області зображення, що відповідають встановленому діапазону кольорів хромакею. Функція `cv2.inRange()` використовується для порогової обробки зображення в просторі HSV та отримання бінарної маски.

Побітова операція (Bitwise Operation) є маскою, яка отримана з порігової обробки, застосовується до вихідного зображення за допомогою побітової операції `cv2.bitwise_and()`. Це дозволяє видалити або замінити пікселі, які відповідають хромакею, залишаючи лише об'єкти переднього плану.

Корекція кольору потрібна для покращення якості зображення. У цьому коді використовується простий підхід, використовуючи просторовий колірний простір LAB. Компоненти LAB (яскравість, зелений-червоний, синій-жовтий) змінюються за допомогою коефіцієнтів корекції яскравості, контрастності та насиченості, і потім перетворюються назад в простір BGR.

Масштабування фонового зображення відбувається автоматично, щоб відповідати розмірам поточного кадру, щоб плавно інтегрувати його з вихідним зображенням.

Ці методи дозволяють виконувати операції хромакею та корекції кольору для отримання бажаного ефекту обробки зображень.

2.3. Опис використаних технологій та мов програмування

Даний код реалізує програму для хромакею - видалення фону з відеопотоку та заміна його на інше зображення. У його розробці був обрана та використана мова програмування Python, а отже код програми був написаний саме цією мовою програмування. Python - це високорівнева, інтерпретована мова програмування з простим і елегантним синтаксисом.

Вона володіє широким спектром бібліотек, які допомагають у розробці різноманітних додатків.

У даному випадку, Python використовується для розробки програми для обробки зображень та роботи з графічним інтерфейсом користувача. Python одна з найпопулярніших мов, яка забезпечує простоту розробки до списку бібліотек для обробки зображень входять такі бібліотеки як: OpenCV, Pillow, Tkinter, NumPy, Tiff File та багато інших. OpenCV (Open Source Computer Vision Library) є бібліотекою з відкритим вихідним кодом для обробки зображень та комп'ютерного зору.

У цьому коді OpenCV використовується для захоплення відеопотоку з камери, обробки кадрів, використання хромакею та заміни фону. NumPy є бібліотекою для обробки числових даних в Python. В коді вона використовується для роботи з масивами та математичними операціями, зокрема для роботи з колоровими просторами та масками.

Tkinter є стандартною бібліотекою Python для створення графічного інтерфейсу користувача (GUI). Вона надає набір інструментів для створення вікон, кнопок, полів введення та інших елементів інтерфейсу. У даному коді Tkinter використовується для створення вікна програми та розміщення на ньому елементів управління.

Pillow - це форк бібліотеки PIL (Python Imaging Library) і є інструментом для роботи з зображеннями, такими як завантаження, збереження, зміна розміру, перетворення кольорів тощо. У даному коді використовується для перетворення кадрів зображень з формату OpenCV у формат, який може використовувати Tkinter для відображення.

Tiff File є бібліотекою для читання та запису TIFF-зображень. В коді вона використовується для збереження результату обробки у форматі TIFF, так як це є один з кращих форматів для подібного використання, адже формат TIFF налаштований на збереження файлів великого обсягу, не змінюючи якість зображення. Ці технології та мова програмування дозволяють реалізувати функціонал хромакею, забезпечують можливість

захоплення відеопотоку, обробки зображень, взаємодії з користувачем та збереження результату відео чи зображення у різних форматах.

Конкретні методи та функції, які використовуються у програмному забезпеченні будуть наведені нижче для більш детального ознайомлення (табл. 2.1) та (табл. 2.2).

Таблиця 2.1

Функції програми

<code>cv2.VideoCapture()</code>	Ця функція з бібліотеки OpenCV відкриває з'єднання з камерою або відеофайлом і повертає об'єкт, який може використовуватися для отримання кадрів з відеопотоку.
<code>cv2.cvtColor()</code>	Ця функція з бібліотеки OpenCV використовується для зміни кольорового простору зображення. У коді вона використовується для перетворення кадрів зображень з простору BGR (Blue-Green-Red) у HSV (Hue-Saturation-Value).
<code>cv2.inRange()</code>	Ця функція з бібліотеки OpenCV використовується для створення маски, яка виділяє область зображення, що задовольняє задані межі кольору. У коді вона застосовується до кадрів зображень у просторі HSV для створення маски хромакею.
<code>cv2.bitwise_and()</code>	Ця функція з бібліотеки OpenCV виконує побітову кон'юнкцію (AND) між двома зображеннями з використанням маски. У коді вона використовується для застосування маски хромакею до кадрів зображень і отримання відфільтрованого зображення.

Cv2.add()	Ця функція з бібліотеки OpenCV виконує додавання елементів двох зображень. У кодї вона використовується для додавання коригованих кадрів зображень до фонового зображення.
cv2.resize()	Ця функція з бібліотеки OpenCV використовується для зміни розміру зображення. У кодї вона використовується для масштабування фонового зображення до розміру поточного кадру.
cv2.imwrite()	Ця функція з бібліотеки OpenCV використовується для збереження зображення на диск. У кодї вона використовується для збереження результату обробки відеопотоку.
Image.fromarray()	Ця функція з бібліотеки PIL (Python Imaging Library) використовується для створення об'єкту зображення з масиву NumPy. У кодї вона використовується для створення зображення з обробленого кадру зображення перед його відображенням у вікні Tkinter.
ImageTk.PhotoImage()	Цей клас з бібліотеки PIL використовується для створення об'єкту зображення, який можна використовувати у вікні Tkinter для відображення зображення.
tk.Tk(), tk.Label(), tk.Entry(), tk.Button(), tk.Frame(), tk.mainloop()	Ці класи та функції з модуля Tkinter використовуються для створення графічного інтерфейсу користувача (GUI) програми, включаючи вікна, елементи керування (наприклад, мітки, кнопки, поля введення) та основний цикл подій для відображення та обробки подій.

Методи програми

Метод create_interface()	Створює інтерфейс програми за допомогою бібліотеки Tkinter. Створює елементи керування, такі як поля введення для налаштування параметрів хромакею та корекції кольору, додавання/видалення фону, перемикачів ефекту хромакею та збереження результатів.
Методи create_color_entry() і create_value_entry()	Створюють елементи введення для завдання кольору та числових значень відповідно.
Метод apply_settings()	Обновляє параметри хромакея і корекції цвета на основі значень, введених користувачем.
Метод parse_color()	Перетворює рядок з кольором на масив чисел.
Метод load_background_image()	Відкриває діалогове вікно для вибору фонового зображення та завантажує вибране зображення.

Метод toggle_chromakey()	Перемикає стан ефекту хромакею (ввімкнений/вимкнений) і додає фонове зображення, якщо воно не було вибрано.
Метод replace_background()	Замінює фон поточного кадру на фонове зображення. Застосовує хромакей та корекцію кольору за потреби.
Методи apply_chromakey() і apply_color_correction ()	Застосовують ефект хромакею та корекцію кольору відповідно до заданого кадру.
Метод remove_background()	Видаляє додане фонове зображення.
Метод save_result()	Зберігає поточний кадр відеопотоку із застосованим тлом у файл. Пропонує користувачеві вибрати місце збереження та формат файлу.
Метод update_frame()	Оновлює поточний кадр відеопотоку, застосовує ефекти хромакею та корекції кольору, замінює фонове зображення за потреби, відображає кадр у вікні програми. Викликається рекурсивно через певний проміжок часу для оновлення зображення в реальному часі.

2.4. Опис структури програмного забезпечення та алгоритмів її функціоналу

Система Chromakey App є інтерфейсною віконною програмою, яка дозволяє користувачеві додавати фонові зображення, застосовувати ефект хромакею, видаляти доданий фон і зберігати результати. Структури системи застосовують клас ChromakeyApp: Головний клас програми, що відповідає за створення інтерфейсу, керування відеопотоком та обробку зображень. Root є головним вікном програми. Cap це об'єкт захоплення відеопотоку з веб-камери. lower_color та upper_color відповідають за нижній та верхній пороги кольору для хромакею, а brightness, contrast і saturation за корекції кольору. background_image відповідає за завантажене фонове зображення. chromakey_enabled виступає у якості прапора, що вказує, чи увімкнено ефект хромакею. frame_label є своєобразною міткою для відображення поточного кадру у вікні. Створення екземпляра класу ChromakeyApp.

До алгоритму функціонування відноситься створення інтерфейсу за допомогою методу create_interface(). Запуск оновлення поточного кадру відеопотоку за допомогою update_frame(). У процесі роботи програми користувач може взаємодіяти з елементами інтерфейсу, такими як поля введення, кнопками вибору фонове зображення, кнопками додавання/видалення фону та іншими. При увімкненому ефекті хромакею та вибраному фоновому зображенні поточний кадр обробляється методом replace_background(), який замінює фон на задане зображення із застосуванням хромакею та корекції кольору. Оновлений кадр відображається у вікні програми. Користувач може зберегти поточний результат за допомогою кнопки "Save Result", при цьому поточний кадр із фоном зберігається у файл. Після завершення роботи програма вікно закривається. Це загальна структура та алгоритм роботи системи Chromakey App. При необхідності ви можете доопрацювати та розширити функціональність програми відповідно до ваших потреб.

2.5. Обґрунтування та організація вхідних та вихідних даних програми

Для обґрунтування та організації вхідних та вихідних даних програми, використовується бібліотека OpenCV для роботи з відеострімом та обробки зображень. Вхідні дані програми включають відеострім з веб-камери та фонове зображення. Вихідні дані програми - відображення обробленого кадру з використанням хромакею та корекції кольору. Основна логіка програми зосереджена в класі ChromakeyApp. При створенні об'єкта цього класу, він ініціалізує вікно програми та відеострім з веб-камери (self.cap). Також відбувається ініціалізація параметрів хромакею, корекції кольору та фонового зображення. Метод create_interface() створює графічний інтерфейс користувача з допомогою бібліотеки Tkinter. У цьому інтерфейсі є налаштування для хромакею (нижня та верхня межі кольору), налаштування для корекції кольору (яскравість, контраст, насиченість) та кнопки для виконання різних операцій, таких як вибір фонового зображення, виконання хромакею, збереження результату тощо. Методи load_background_image(), toggle_chromakey(), replace_background(), apply_chromakey(), apply_color_correction(), remove_background(), save_result() використовуються для виконання відповідних операцій з фоновим зображенням, хромакею та корекцією кольору. Метод update_frame() оновлює кадр відеостріму та відображає оброблений кадр у вікні програми.

2.6. Опис розробленої системи

2.6.1. Використані технічні засоби

Розробляючи дане програмне забезпечення використовувався персональний комп'ютер на основі системи Windows 10 який відповідає наступним характеристикам:

- чотирьохядерний процесор Intel Core i5 с тактовою частотою 2.5 ГГц;

- 8 ГБ оперативної пам'яті DDR4;
- Windows 10 Pro;
- дискретна графічна карта AMD Radeon RX 560 на 4Гб пам'яті.

2.6.2. Використані програмні засоби

Для створення програми було задіяно відкритий вихідний редактор коду Visual Studio Code розроблений компанією Microsoft та встановлене розширення Python для редактору разом з необхідними бібліотеками для функціонування коду.

2.6.3. Виклик та завантаження програми

Для завантаження та використання програми потрібно мати файл зі скопійованим кодом.

2.6.4. Опис інтерфейсу користувача

При запуску програми користувач бачить перед собою програму, яка має функціонал взаємодії з хромакеєм. Перед користувачем розташоване вікно інтерфейсу програми (рис. 2.1).

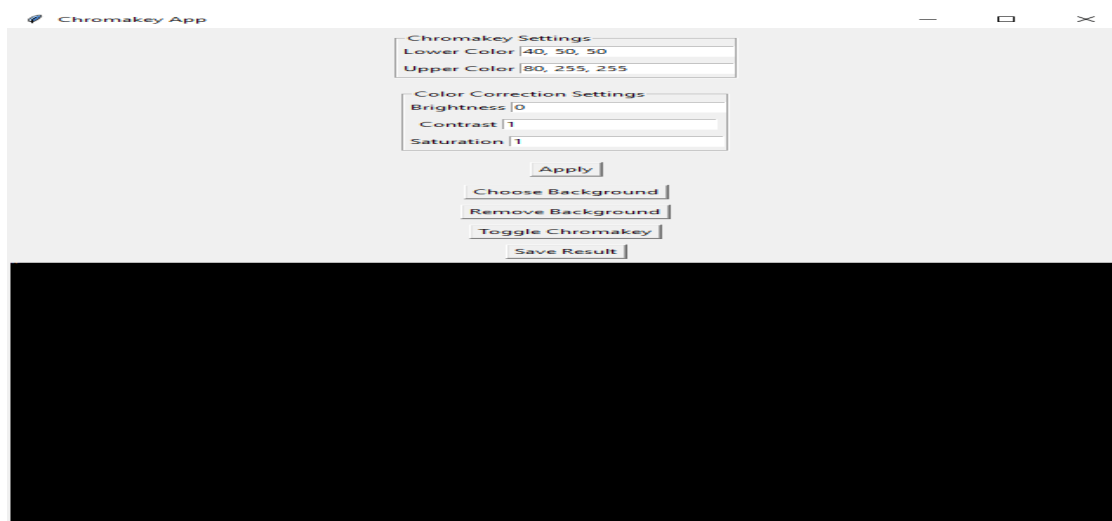


Рис. 2.1. Зображення повного інтерфейсу програмного забезпечення

Вікно програми: вікно має назву «Chromakey App»

Налаштування хромакею:

- елемент «Chromakey Settings» - це рамка, яка містить налаштування хромакею;
- елемент «Lower Color» - це поле введення, яке дозволяє користувачу ввести значення для нижньої межі кольору (зеленого);
- елемент «Upper Color»: це поле введення, яке дозволяє користувачу ввести значення для верхньої межі кольору (зеленого) (рис. 2.2).

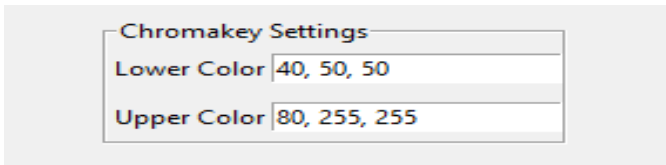


Рис. 2.2. Налаштування хромакею

Налаштування корекції кольору:

- елемент «Color Correction Settings» - це рамка, яка містить налаштування корекції кольору;
- елемент «Brightness» - це поле введення, яке дозволяє користувачу ввести значення для яскравості кадру;
- елемент «Contrast» - це поле введення, яке дозволяє користувачу ввести значення для контрастності кадру;
- елемент «Saturation» - це поле введення, яке дозволяє користувачу ввести значення для насиченості кадру (рис. 2.3).

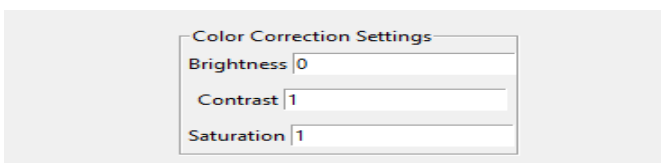


Рис. 2.3. Налаштування яскравості

Кнопки:

- кнопка «Apply» при натисканні її застосовуються всі введені налаштування хромакею та корекції кольору;
- кнопка «Choose Background» при натисканні її відкривається вікно із файлами завдяки якому користувач може вибрати фонове зображення для заміни фону (2.4);

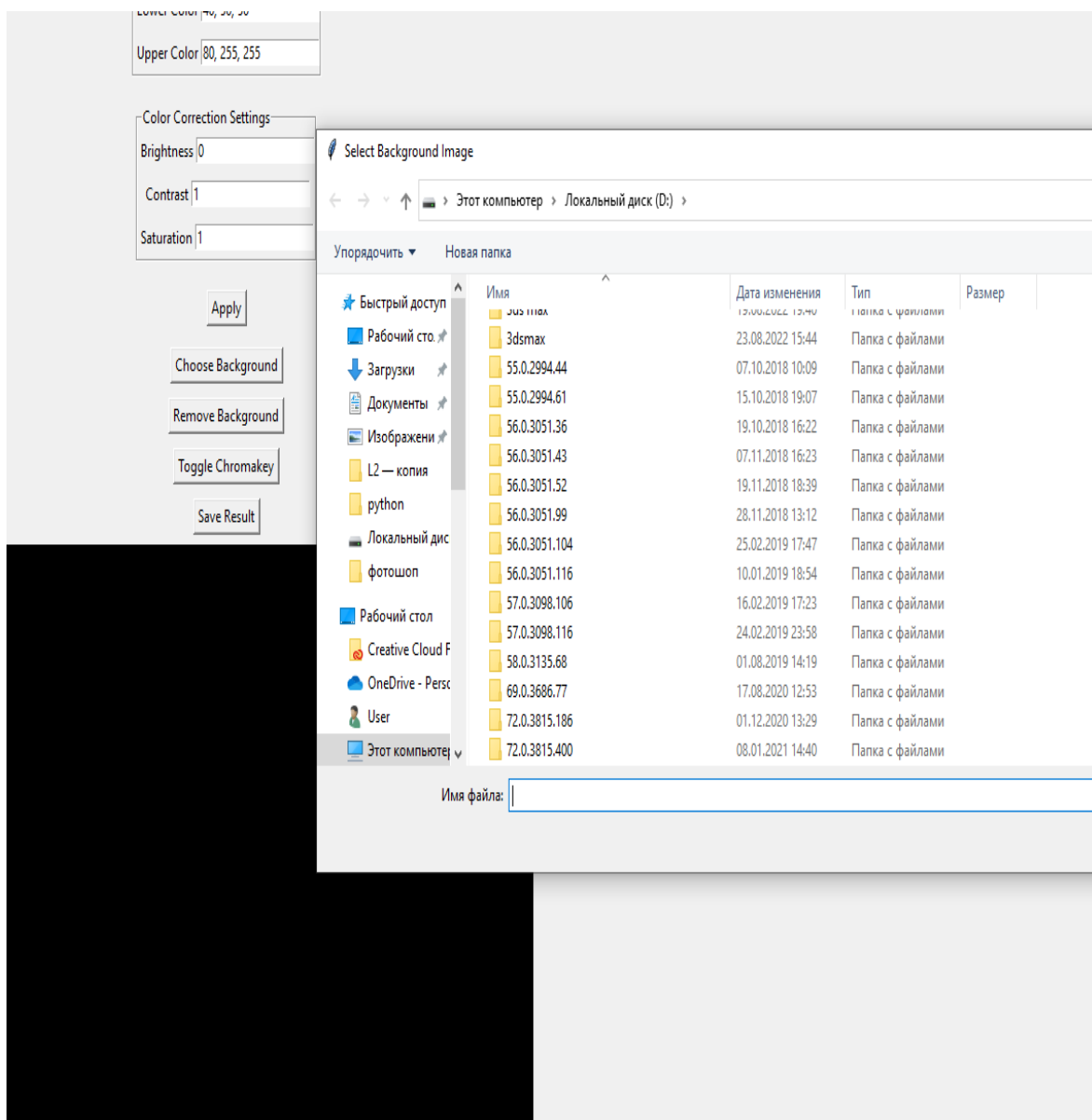


Рис. 2.4. Додавання підготовленого фону

- кнопка «Remove Background» при натисканні її фонове зображення, яке було додано користувачем видаляється, і відеострім відображається без заміни фону;

- кнопка «Save Result» при натисканні її користувач може зберегти результат заміни фону як TIFF-файл (рис 2.5).

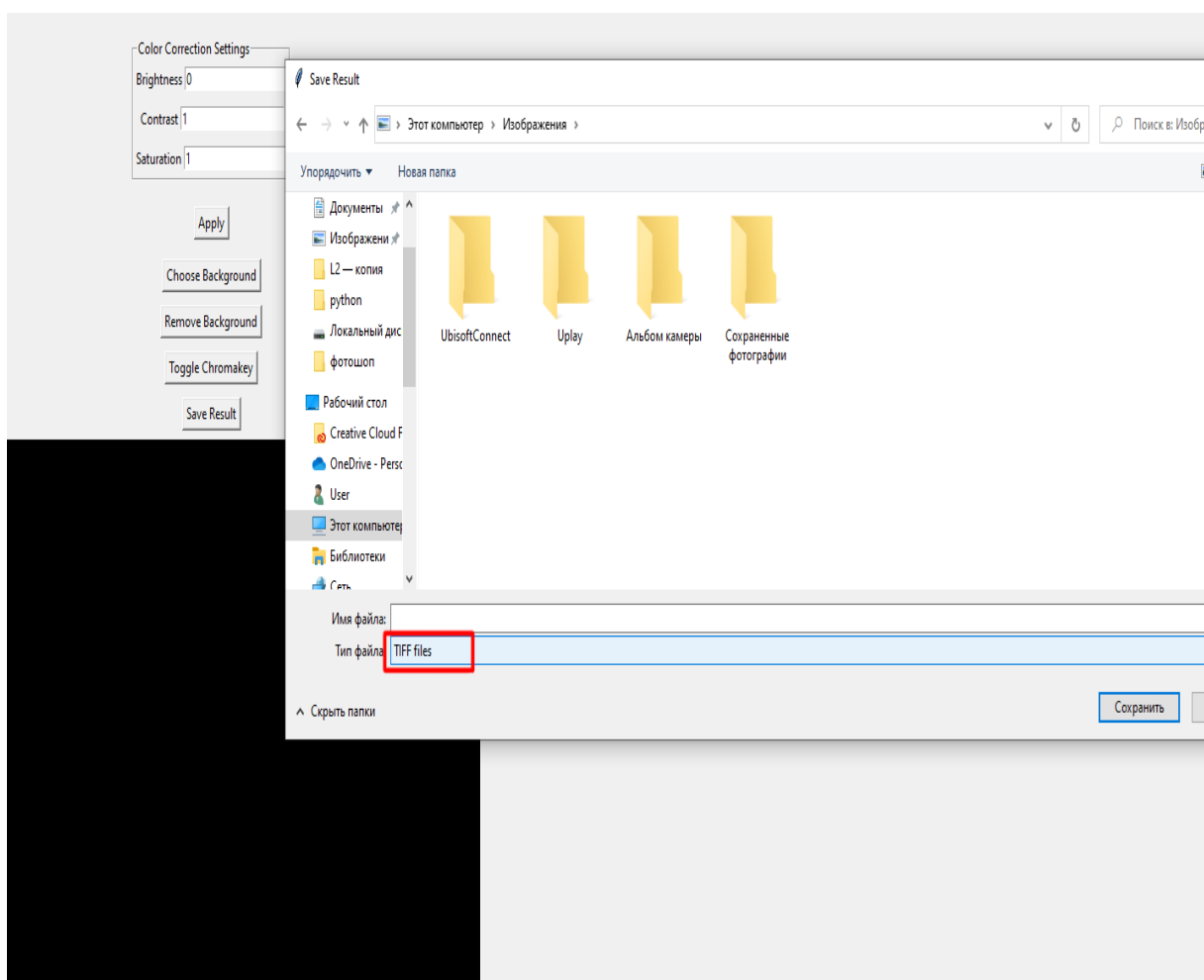


Рис. 2.5. Збереження результату у форматі TIFF

Панель керування, завдяки якій здійснюються операції, підтримку яких передбачає програмне забезпечення. Кожна функція залежить від потреб користувача і здійснюється за допомогою відповідних кнопок (рис 2.6).

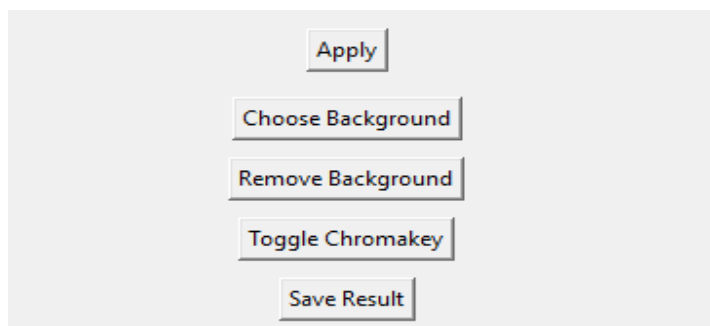


Рис. 2.6. Кнопки

Під кнопками знаходиться вікно захоплення камери, яке виступає у ролі сцени впроваджені до програмного забезпечення, воно транслює зображення через веб-камеру, яку необхідно під'єднати до комп'ютера перед запуском програми (рис. 2.7).

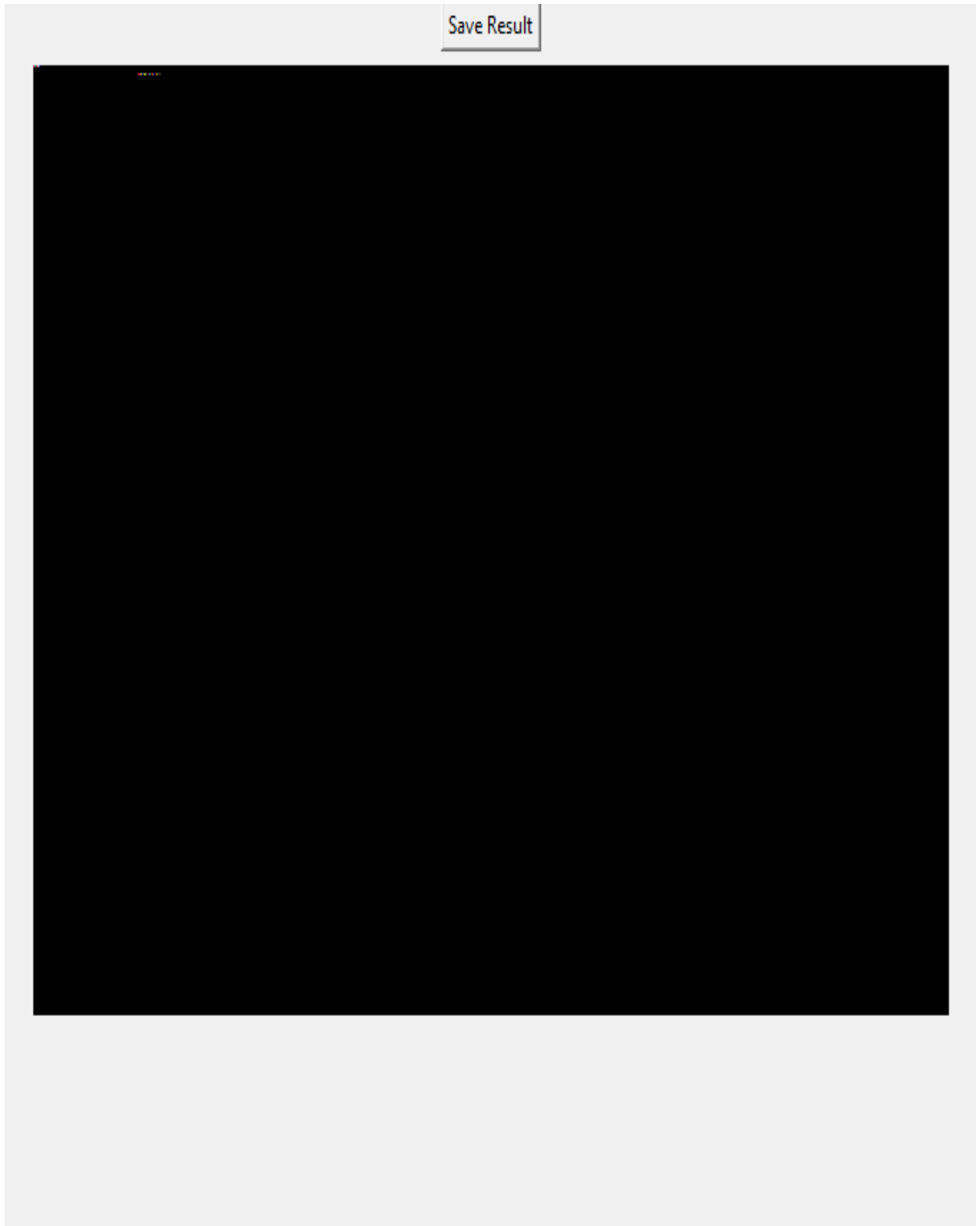


Рис. 2.7. Вікно перегляду кадра

Для запуску вікна програми необхідно під'єднати веб-камеру (рис.2.8).

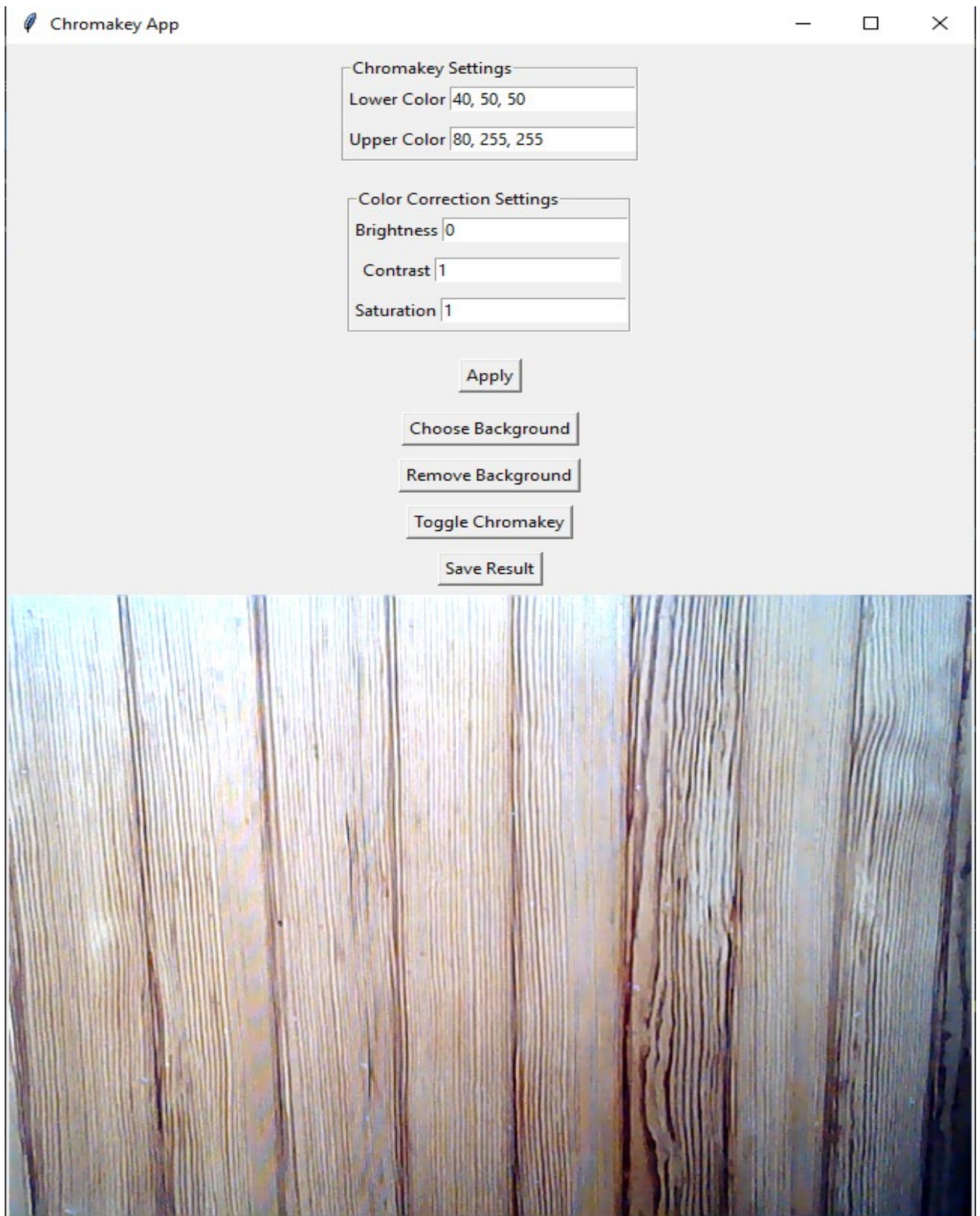


Рис. 2.8. Робота вікна захоплення камери

Щоб замінити фон у камері захоплення, необхідно відкрити спеціальний підготовлений файл у форматі PNG, таким чином він замінить собою старий фон у рамках потреби користувача. (рис. 2.9).

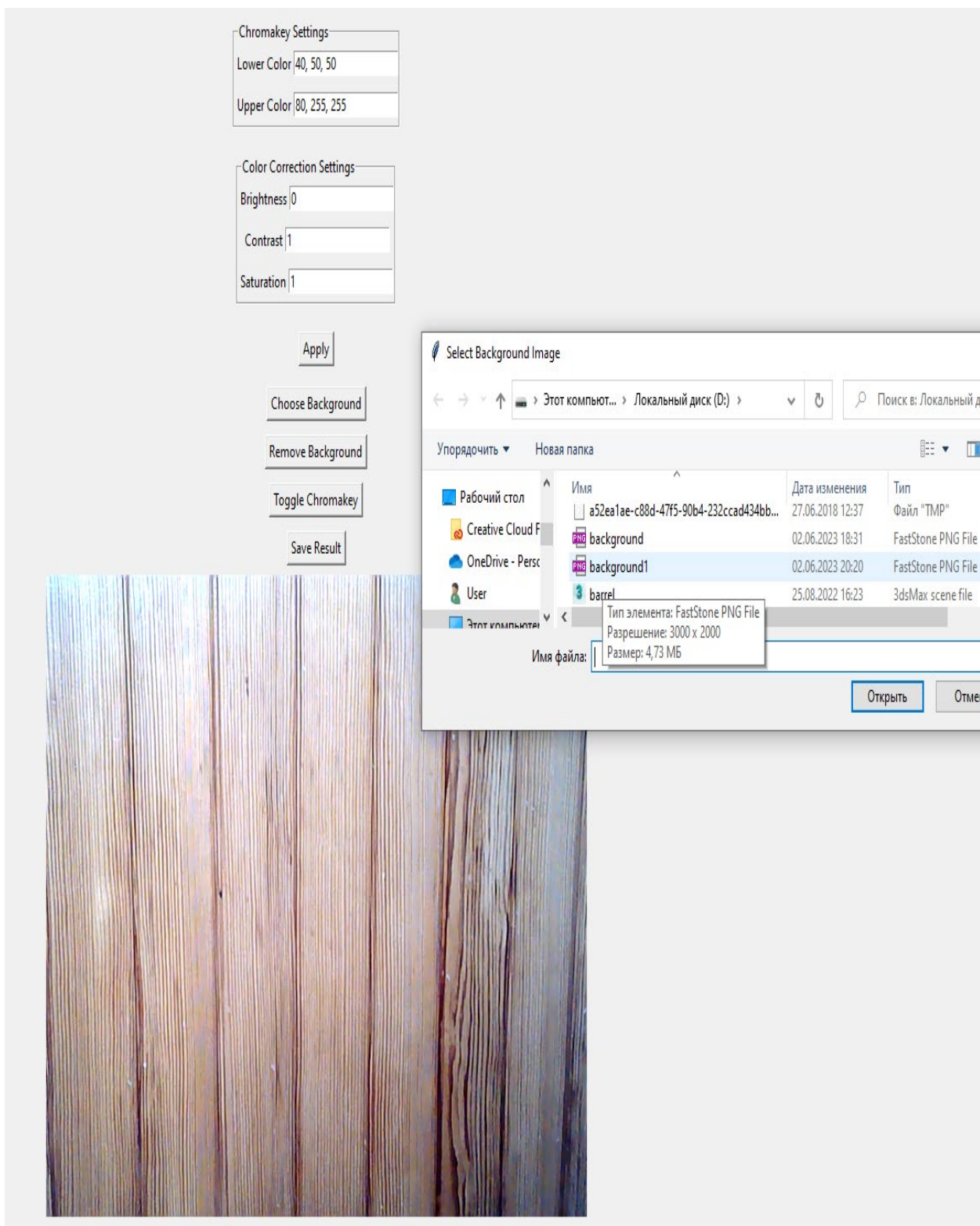


Рис. 2.9. Вибір необхідного фону

Щоб фон наклався і був готовий до увімкнення хромакею, необхідно спочатку змінити значення «Brightness» підвищивши його показник до 1 (рис. 2.10).

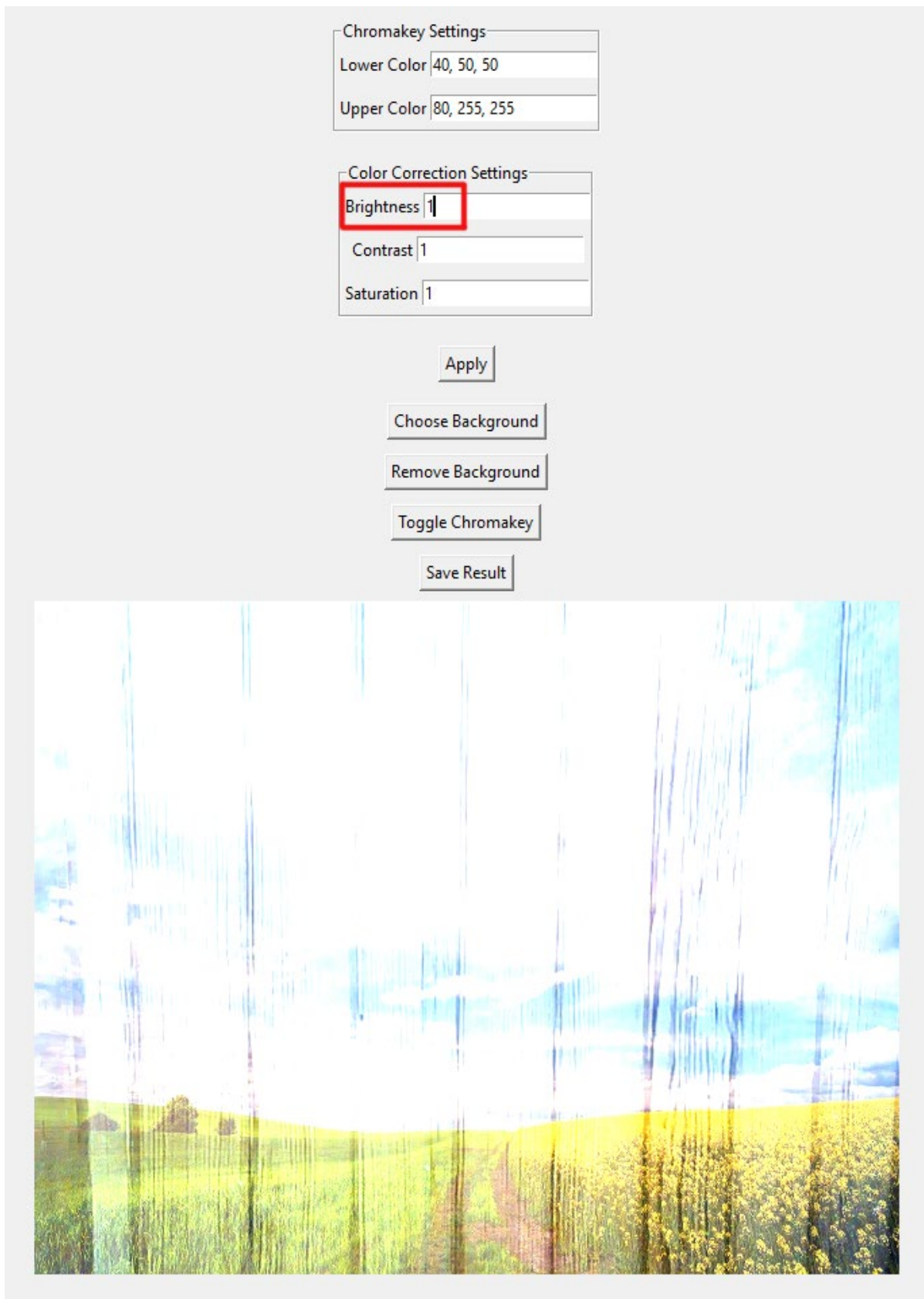


Рис. 2.10. Додавання фону та зміна значень налаштувань хромакею призводить до накладання фонів один на одного

Для того щоб фон виглядав нормально залишилося увімкнути саму функцію хромакею (рис 2.11).

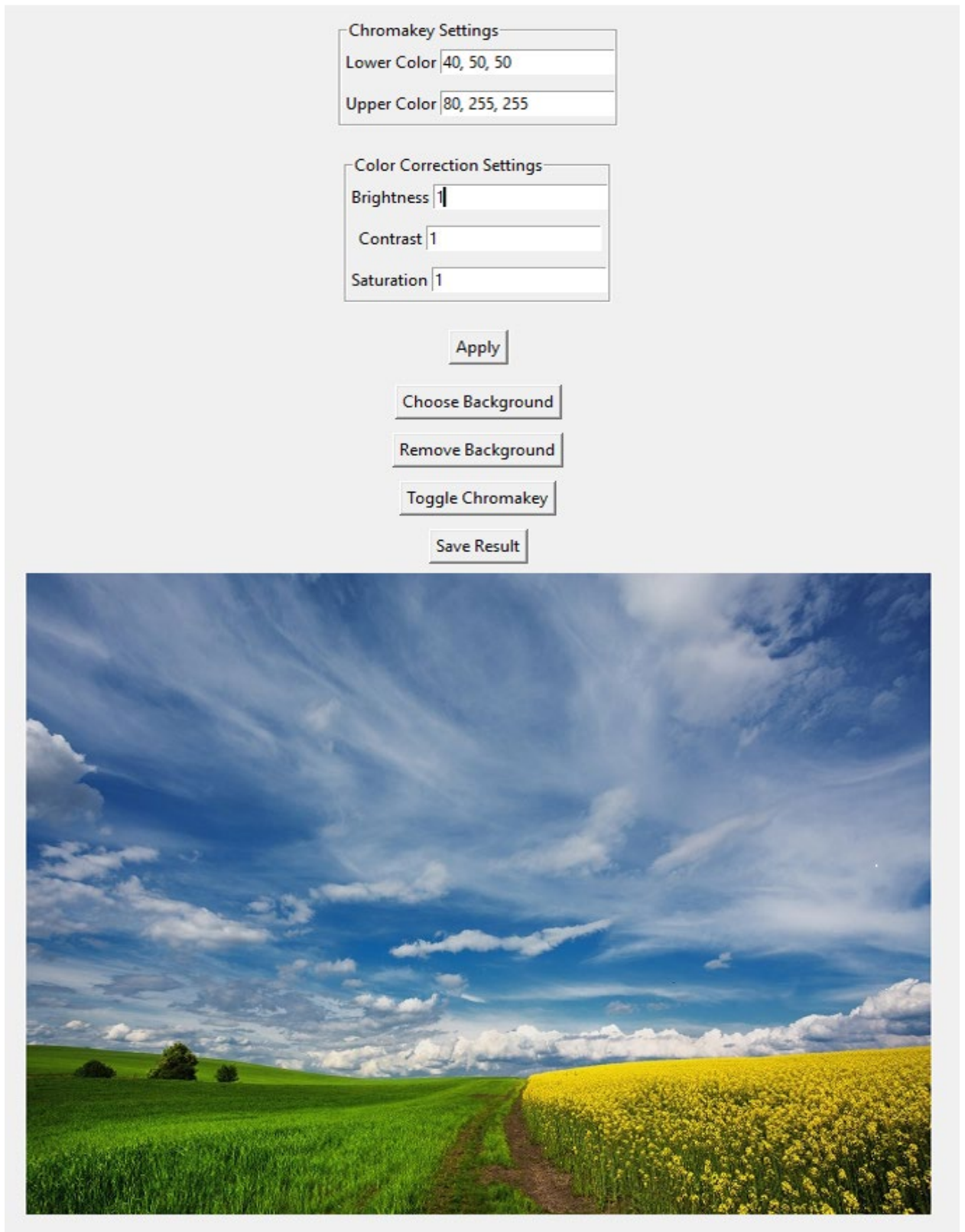


Рис 2.11. Зображення, яке накладено замість попереднього фону із увімкненою функцією хромакею

Додавання зелених об'єктів відбувається у режимі реального часу, завдяки чьому користувач може додати будь-ку кількість кадрів до вікна (рис.2.12).

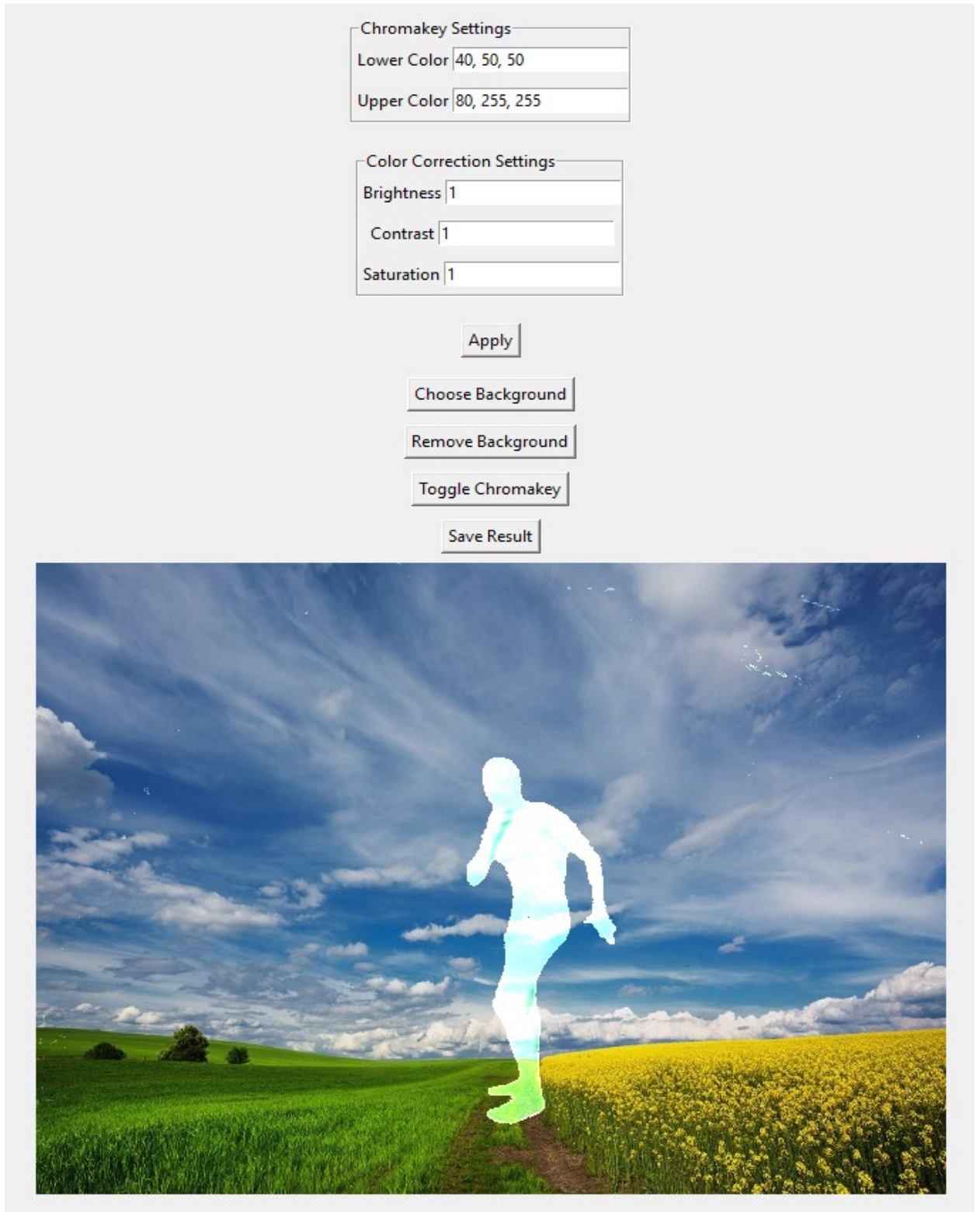


Рис. 2.12. Додавання зелених об'єктів до вікна захоплення камери

РОЗДІЛ 3 ЕКОНОМІЧНИЙ РОЗДІЛ

3.1. Визначення трудомісткості розробки програмного забезпечення

Початкові дані:

1. передбачуване число операторів програми - 200;
2. коефіцієнт складності програми – 1,5;
3. коефіцієнт корекції програми в ході її розробки – 0,08;
4. годинна заробітна плата Python розробника – 157 грн/год;
5. коефіцієнт збільшення витрати праці внаслідок недостатнього опису задачі – 1,1;
6. коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності – 0,8;
7. вартість машино-години ЕОМ – 14 грн/год.

Годинну заробітну плату Python розробника було розраховано за допомогою даних, узятих з сайту «Української спільноти програмістів (DOU.ua)». Середня заробітна плата Python розробника з досвідом роботи до року, по Україні, складає приблизно 750 американських доларів на місяць. Станом на зараз, початок червня 2023 року, один американський долар дорівнює 36,75 грн, виходячи з цього, середня заробітна плата у гривнях складає 27560 грн. При звичайному восьмигодинному робочому графіку (приблизно 176 годин на місяць) середня погодинна заробітна плата буде складати 156,6 грн/год.

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{отл} + t_d, \text{ людино-годин} \quad (3.1)$$

де t_o – витрати праці на підготовку й опис поставленої задачі (приймається 50 людино-годин);

t_u – витрати праці на дослідження алгоритму рішення задачі;

t_o – витрати праці на розробку блок-схеми алгоритму;

t_n – витрати праці на програмування по готовій блок-схемі;

$t_{отл}$ – витрати праці на налагодження програми на ЕОМ;

t_d – витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у програмному забезпеченні, яке розробляється.

Умовне число операторів:

$$Q = q \cdot C \cdot (1 + p) \quad (3.2)$$

де q - передбачуване число операторів (200),

C - коефіцієнт складності програми (1,5),

p - коефіцієнт корекції програми в ході її розробки (0,08).

$$Q = 200 \cdot 1,5 \cdot (1 + 0,08) = 324$$

Витрати праці на вивчення опису задачі t_u визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75 \cdot 85) \cdot k}, \text{ людино-годин} \quad (3.3)$$

де B – коефіцієнт збільшення витрат праці внаслідок недостатнього

опису задачі,

k – коефіцієнт кваліфікації програміста, обумовлений стажем роботи з даної спеціальності.

$$t_u = \frac{324 \cdot 1,1}{75 \cdot 0,8} = 5,94 \text{ людино-годин.}$$

Витрати праці на розробку алгоритму рішення задачі:

$$t_a = \frac{Q}{(20 \cdot 25) \cdot k}, \text{ людино-годин} \quad (3.4)$$

де Q – умовне число операторів програми,

k – коефіцієнт кваліфікації програміста.

Підставивши відповідні значення в формулу (3.4), людино-годин:

$$t_a = \frac{200}{20 \cdot 0,8} = 12,5 \text{ людино-годин}$$

Витрати на складання програми по готовій блок-схемі:

$$t_a = \frac{Q}{(20 \cdot 25) \cdot k}, \text{ людино-годин} \quad (3.5)$$

$$t_a = \frac{200}{25 \cdot 0,8} = 10 \text{ людино-годин}$$

Витрати праці на налагодження програми на ЕОМ:

– за умови автономного налагодження одного завдання:

$$t_{отл} = \frac{Q}{(4 \cdot 5) \cdot k}, \text{ людино-годин} \quad (3.6)$$

$$t_{\text{отл}} = \frac{200}{4 \cdot 0,6} = 83,3 \text{ людино-годин}$$

– за умови комплексного налагодження завдання:

$$t_{\text{отл}}^{\text{к}} = 1,5 \cdot t_{\text{отл}}, \text{ людино-годин} \quad (3.7)$$

$$t_{\text{отл}}^{\text{к}} = 1,5 \cdot 83,3 = 125 \text{ людино-годин}$$

Витрати праці на підготовку документації:

$$t_{\text{д}} = t_{\text{др}} + t_{\text{до}}, \text{ людино-годин} \quad (3.8)$$

де $t_{\text{др}}$ - трудомісткість підготовки матеріалів і рукопису.

$$t_{\text{др}} = \frac{Q}{(15 \cdot 20) \cdot k}, \text{ людино-годин} \quad (3.9)$$

$$t_{\text{др}} = \frac{200}{15 \cdot 0,8} = 16,67 \text{ людино-годин}$$

$t_{\text{до}}$ - трудомісткість редагування, печатки й оформлення документації

$$t_{\text{до}} = 0,75 \cdot t_{\text{др}}, \text{ людино-годин} \quad (3.10)$$

$$t_{\text{до}} = 0,75 \cdot 16,67 = 12,5 \text{ людино-годин}$$

$$t_{\text{д}} = 16,67 + 12,5 = 29,17 \text{ людино-годин}$$

Отримаємо трудомісткість розробки програмного забезпечення:

$$t = 5,94 + 12,5 + 10 + 83,3 + 125 + 29,7 = 266,44 \text{ людино-годин}$$

У результаті ми розрахували, що в загальній складності необхідно 266,44 людино-годин для розробки даного веб-додатку.

3.2. Витрати на створення програмного забезпечення

Витрати на створення ПЗ $K_{\text{ПО}}$ включають витрати на заробітну плату виконавця програми $Z_{\text{ЗП}}$ і витрат машинного часу, необхідного на налагодження програми на ЕОМ.

$$K_{\text{ПО}} = Z_{\text{ЗП}} + Z_{\text{МВ}}, \text{ грн} \quad (3.11)$$

де $Z_{\text{ЗП}}$ заробітна плата виконавців, яка визначається за формулою:

$$Z_{\text{ЗП}} = t \cdot C_{\text{пр}}, \text{ грн} \quad (3.12)$$

де t – загальна трудомісткість людино-годин,

$C_{\text{пр}}$ – середня годинна заробітна плата програміста, грн/год.

$$Z_{\text{ЗП}} = 266,44 \cdot 157 = 41831 \text{ грн}$$

$Z_{\text{МВ}}$ – вартість машинного часу, необхідного для налагодження програми на ЕОМ.

$$Z_{\text{МВ}} = t_{\text{отл}} \cdot C_{\text{мч}}, \text{ грн} \quad (3.13)$$

де $t_{\text{отл}}$ – трудомісткість налагодження програми на ЕОМ, год,

$C_{\text{мч}}$ – вартість машино-години ЕОМ, грн/год.

$$З_{\text{МВ}} = 266,44 \cdot 14 = 3730,16 \text{ грн}$$

$$К_{\text{ПО}} = 41831 + 3730,16 = 45561,16 \text{ грн}$$

Визначені в такий спосіб витрати на створення програмного забезпечення є частиною одноразових капітальних витрат на створення АСУП.

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p}, \text{ міс} \quad (3.14)$$

де B_k – число виконавців,

F_p – місячний фонд робочого часу (при 40 годинному робочому тижні $F_p = 176$ годин).

$$T = \frac{266,44}{1 \cdot 176} = 1,51 \text{ міс}$$

Висновок: На розробку даного веб-додатку піде 266,44 людино-годин. Тобто, ймовірна очікувана тривалість розробки складатиме 1,51 місяців при стандартному 40-годинному робочому тижні і 176-годинному робочому місяці. Очікувані витрати на створення веб-додатку складатимуть 45561,16 грн.

ВИСНОВКИ

Метою кваліфікаційної роботи було створення програми, яка дає можливість користувачеві взаємодіяти з технологією хромакею, яке буде зрозуміле для використання за рахунок інтуїтивно зрозумілого та простого інтерфейсу.

В результаті виконання кваліфікаційної роботи було створено програмне забезпечення яке здатне взаємодіяти з технологією хромакею.

Створена програма надає наступні функціональні можливості:

1. Завантажувати та видаляти підготовлений фон до вікна захоплення камери.
2. Змінювати кольорову корекцію.
3. Додавати зелені об'єкти у режимі реального часу.
4. Вмикати режим хромакей.
5. Зберігати результат у форматі Tiff.

Програма розроблена з використанням мови Python.

В ході кваліфікаційної роботи було вираховано трудомісткість та порахована приблизна вартість розробленого додатку. Загальна вартість програмного продукту становить 45561,16 грн, а час створення - 176 годин або 1,51 місяці.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. «Цифровий композитинг для кіно та відео» - Стів Райт (переклав Валерій Кульчицький)
2. «Мистецтво та наука цифрового композитингу» - Рон Брінкман (переклав Володимир Чернишов)
3. «Володарювання цифровим 2D та 3D мистецтвом» - Лес Пардью (переклав Олег Яковенко)
4. «Хрома: Книга про колір» - Дерек Джармен (переклав Сергій Зубов)
5. «Техніка монтажу фільму та відео: історія, теорія та практика» - Кен Данцигер (переклав Андрій Кушнір)
6. «Посібник зеленого екрану: техніки реального виробництва» - Джефф Фостер (переклав Олександр Павловський)
7. «Посібник з корекції кольору: професійні техніки для відео та кіно» - Алексіс Ван Хюркман (переклав Дмитро Кузнецов)
8. «Цифрове монтажування для кіно і відео» - Джон Декер (переклав Іван Сорокін)
9. «Ключування та композитинг у After Effects» - Марк Кристіансен (переклав Олександр Коваленко)
10. «Фільм без кінця: композитинг і спецефекти» - Лео Гільберт (переклав Владислав Замшинський)
11. «Посібник по спецефектам: ключування, композитинг та створення візуальних ефектів» - Рік Граветт (переклав Андрій Шевченко)
12. «Кіно та відео монтажування: Основи теорії та практики» - Стіві Дженкінс (переклав Євген Бондаренко)
13. «Цифрова пост-продакшн: Композитинг та спецефекти з використанням програмного забезпечення Adobe» - Ліз Гіллер (переклав Денис Москаленко)
14. «Збирання реальності: візуальні ефекти з використанням графіки» - Стів Райт (переклав Валерій Кульчицький)
15. «Цифрове зображення і композитинг: техніки та застосування» - Том Гаскінс (переклав Олександр Хлебников)
16. «Майстерня візуальних ефектів: практичні поради для кінематографістів» - Джон Гейтс (переклав Андрій Перцов)
17. «Теорія та практика композитингу в кіно та відео» - Пол Лінді (переклав Микола Таран)
18. «Композитинг: Візуальні ефекти з використанням Adobe After Effects» - Лі Фрідлендер (переклав Віктор Степаненко)

19. «Техніки графічного дизайну для кіно та відео» - Лінда Урбан (переклав Іван Сорокін)
20. «Зелений екран зроблений легко: техніки ключування та композитингу для незалежних кінематографістів» - Джеремі Ханке та Мішель Ямазакі (переклав Олександр Гладун)
21. «Посібник VES з візуальних ефектів: індустрійні стандарти та процедури візуальних ефектів» - редагували Сьюзан Зверман та Джефрі Окун (переклав Микола Таран)
22. «Посібник кінематографіста: комплексний підручник для цифрової ери» - Стівен Ашер та Едвард Пінкус (переклав Михайло Шкарупа)
23. «Майстерність цифрового композитингу: візуальні ефекти та пост-продакшн» - Білл Меддокс та Петер Плат (переклав Віктор Степаненко)
24. Хромакей / URL <https://uk.wikipedia.org/wiki>
25. dou.ua зарплати / URL: <https://jobs.dou.ua/salaries>

КОД ПРОГРАМИ

```
import cv2

import numpy as np

import tkinter as tk

from PIL import Image, ImageTk

from tkinter import filedialog

import tiffio

class ChromakeyApp:

    def __init__(self, root):

        self.root = root

        self.root.title("Chromakey App")

        # Відеострім

        self.cap = cv2.VideoCapture(0)

        # Параметри хромакею

        self.lower_color = np.array([40, 50, 50]) # Нижня межа кольору (зелений)

        self.upper_color = np.array([80, 255, 255]) # Верхня межа кольору (зелений)

        # Параметри корекції кольору

        self.brightness = 0

        self.contrast = 1

        self.saturation = 1

        # Фонове зображення

        self.background_image = None
```

```

# Переключение хромакея

self.chromakey_enabled = False

# Создание интерфейса

self.create_interface()

# Обновление кадра

self.update_frame()

def create_interface(self):

    # Настройки хромакею

    chromakey_frame = tk.LabelFrame(self.root, text="Chromakey Settings")

    chromakey_frame.pack(padx=10, pady=10)

    self.lower_color_entry = self.create_color_entry(chromakey_frame, "Lower Color",
self.lower_color)

    self.upper_color_entry = self.create_color_entry(chromakey_frame, "Upper Color",
self.upper_color)

    # Настройки коррекции коьору

    color_correction_frame = tk.LabelFrame(self.root, text="Color Correction Settings")

    color_correction_frame.pack(padx=10, pady=10)

    self.brightness_entry = self.create_value_entry(color_correction_frame, "Brightness",
self.brightness)

    self.contrast_entry = self.create_value_entry(color_correction_frame, "Contrast", self.contrast)

    self.saturation_entry = self.create_value_entry(color_correction_frame, "Saturation", self.saturation)

    # Кнопка "Применить"

    apply_button = tk.Button(self.root, text="Apply", command=self.apply_settings)

    apply_button.pack(pady=10)

```



```

# Кнопка "Выбрать фон"

select_background_button = tk.Button(self.root, text="Choose Background",
command=self.load_background_image)

select_background_button.pack(pady=5)

# Кнопка "Удалить фон"

remove_background_button = tk.Button(self.root, text="Remove Background",
command=self.remove_background)

remove_background_button.pack(pady=5)

# Кнопка "Переключить хромакей"

toggle_chroma_key_button = tk.Button(self.root, text="Toggle Chroma Key",
command=self.toggle_chroma_key)

toggle_chroma_key_button.pack(pady=5)

# Кнопка "Сохранить результат"

save_result_button = tk.Button(self.root, text="Save Result", command=self.save_result)

save_result_button.pack(pady=5)

# Окно для показа кадра

self.frame_label = tk.Label(self.root)

self.frame_label.pack()

def create_color_entry(self, parent, label_text, color):

    entry_frame = tk.Frame(parent)

    entry_frame.pack(pady=5)

    label = tk.Label(entry_frame, text=label_text)

    label.pack(side=tk.LEFT)

```

```
color_entry = tk.Entry(entry_frame)
color_entry.pack(side=tk.LEFT)
color_entry.insert(tk.END, f" {color[0]}, {color[1]}, {color[2]}")
```

```
return color_entry
```

```
def create_value_entry(self, parent, label_text, value):
```

```
    entry_frame = tk.Frame(parent)
```

```
    entry_frame.pack(pady=5)
```

```
    label = tk.Label(entry_frame, text=label_text)
```

```
    label.pack(side=tk.LEFT)
```

```
    value_entry = tk.Entry(entry_frame)
```

```
    value_entry.pack(side=tk.LEFT)
```

```
    value_entry.insert(tk.END, str(value))
```

```
    return value_entry
```

```
def apply_settings(self):
```

```
    # Обновление параметров хромакея
```

```
    lower_color_str = self.lower_color_entry.get()
```

```
    upper_color_str = self.upper_color_entry.get()
```

```
    self.lower_color = self.parse_color(lower_color_str)
```

```
    self.upper_color = self.parse_color(upper_color_str)
```

```
    # Обновление параметров коррекции колюору
```

```
    self.brightness = float(self.brightness_entry.get())
```

```
    self.contrast = float(self.contrast_entry.get())
```

```
    self.saturation = float(self.saturation_entry.get())
```

```

def parse_color(self, color_str):
    color_values = color_str.split(',')
    color = [int(value.strip()) for value in color_values]
    return np.array(color)

def load_background_image(self):
    file_path = filedialog.askopenfilename(title="Select Background Image")
    if file_path:
        self.background_image = cv2.imread(file_path)

def toggle_chromakey(self):
    self.chromakey_enabled = not self.chromakey_enabled
    if self.background_image is None:
        self.add_background()

def replace_background(self, frame, background_image):
    if self.chromakey_enabled:
        chromakey_frame = self.apply_chromakey(frame, self.lower_color, self.upper_color)
        corrected_frame = self.apply_color_correction(chromakey_frame, self.brightness, self.contrast,
self.saturation)
    else:
        corrected_frame = self.apply_color_correction(frame, self.brightness, self.contrast, self.saturation)

# Масштабируем фоновое изображение до размеров текущего кадра
resized_background = cv2.resize(background_image, (frame.shape[1], frame.shape[0]))

# Создаем маску, где фон не черный
foreground_mask = cv2.bitwise_not(cv2.cvtColor(corrected_frame, cv2.COLOR_BGR2GRAY))

```

```

# Объединяем фоновое изображение и текущий кадр с помощью маски
result = cv2.bitwise_and(resized_background, resized_background, mask=foreground_mask)

# Добавляем откорректированный кадр к результату
result = cv2.add(result, corrected_frame)

return result

def apply_chromakey(self, frame, lower_color, upper_color):
    hsv_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    mask = cv2.inRange(hsv_frame, lower_color, upper_color)
    masked_frame = cv2.bitwise_and(frame, frame, mask=mask)
    return masked_frame

def apply_color_correction(self, frame, brightness, contrast, saturation):
    corrected_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2LAB)
    corrected_frame[:, :, 0] = np.clip(corrected_frame[:, :, 0] * brightness, 0, 255)
    corrected_frame[:, :, 1] = np.clip(corrected_frame[:, :, 1] * contrast, 0, 255)
    corrected_frame[:, :, 2] = np.clip(corrected_frame[:, :, 2] * saturation, 0, 255)
    corrected_frame = cv2.cvtColor(corrected_frame, cv2.COLOR_LAB2BGR)
    return corrected_frame

def remove_background(self):
    self.background_image = None

def save_result(self):
    file_path = filedialog.asksaveasfilename(title="Save Result", defaultextension=".tiff",
filetypes=[("TIFF files", "*.tiff")])
    if file_path:
        frame = self.cap.read()[1]

```

```

    if self.background_image is not None:
        frame = self.replace_background(frame, self.background_image)
        cv2.imwrite(file_path, frame)

def update_frame(self):
    _, frame = self.cap.read()

    if self.background_image is not None:
        frame = self.replace_background(frame, self.background_image)

    frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    image = Image.fromarray(frame)
    image = ImageTk.PhotoImage(image)

    self.frame_label.configure(image=image)
    self.frame_label.image = image

    self.root.after(15, self.update_frame)

root = tk.Tk()
app = ChromakeyApp(root)
root.mainloop()

```

ДОДАТОК Б

ВІДГУК КЕРВІНИКА ЕКОНОМІЧНОГО РОЗДІЛУ

ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

Ім'я файлу	Опис
Пояснювальні документи	
Кваліфікаційна робота Біденко.doc	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
Кваліфікаційна робота Біденко.pdf	Пояснювальна записка до кваліфікаційної роботи в форматі PDF
Програма	
Program.rar	Архів, який містить код програми і откомпільовану програму
Презентація	
Презентація_Біденко.ppt	Презентація кваліфікаційної роботи