

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО”**

Навчально-науковий інститут атомної та теплової енергетики

Кафедра цифрових технологій в енергетиці

"На правах рукопису"

УДК 004.4

«До захисту допущено»

Завідувач кафедри

Наталія АУШЕВА

“ ” 2023 р.

Магістерська дисертація
на здобуття ступеня магістра
за освітньо-професійною програмою
“Цифрові технології в енергетиці”
зі спеціальності 122 “Комп’ютерні науки”

на тему: Калібрація об’єктиву для сферично-панорамної зйомки на основі
катадіоптричної камери

Виконав: студент 2 курсу, групи ТР-21 мп

ГОНЧАРУК Костянтин Григорійович

(прізвище, ім’я, по батькові)

(підпис)

Науковий керівник: доцент, к.т.н., Демчишин А.А.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент: доцент, к.т.н., Гагарін О.О.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

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

Студент _____
(підпис)

Київ – 2023

**Національний технічний університет України
“Київський політехнічний інститут ім. Ігоря Сікорського”**

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ ЕНЕРГЕТИКИ

Кафедра **ЦИФРОВИХ ТЕХНОЛОГІЙ В ЕНЕРГЕТИЦІ**

Рівень вищої освіти **другий (магістерський)**

За освітньою програмою **“Цифрові технології в енергетиці”**

Спеціальності 122 Комп’ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ **Наталія АУШЕВА**

«__» _____ 2023 р.

**З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ**

ГОНЧАРУКУ Костянтину Григорійовичу

(прізвище, ім’я, по батькові)

1. Тема дисертації **Калібрація об’єктиву для сферично-панорамної зйомки на основі катадіоптричної камери**

науковий керівник дисертації

Демчишин Анатолій Анатолійович, к.т.н., доцент

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від **«06» листопада 2023р. № 5152-с**

2. Строк подання студентом дисертації **18 грудня 2023 року**

3. Об’єкт дослідження – програмні системи обробки зображення отриманого з катадіоптричної камери.

4. Вихідні дані: мова програмування JavaScript, GLSL, бібліотека WebGL, камера Defender c-090, сферичне дзеркало.

5. Перелік завдань, які потрібно виконати: провести огляд існуючих камер з великим кутом зору та зробити вибір засобів розробки, розробити математичну модель та провести її реалізацію, провести виготовлення прототипу катадіоптричної камери, провести тестування розробленого програмно-апаратного комплексу.

6. Орієнтовний перелік ілюстративного матеріалу: презентація, камери з великим кутом зору, режими зйомки, методи проєціювання, блок-схеми, прототип катадіоптричної камери, інтерфейс програмного продукту, таблиці для стартап-проекту.

7. Орієнтовний перелік публікацій: міжнародна інтернет – конференція «SCIENCE 2023: RESEARCH AND INNOVATION» 1-2 грудня 2023 року, публікація на тему «CALIBRATION OF LENS FOR SPHERICAL PANORAMIC IMAGING BASED ON CATADIOPTRIC CAMERA».

8. Дата видачі завдання 31 травня 2023 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Затвердження теми роботи	01.09.23	
2	Вивчення та аналіз задачі	04.09.23-09.09.23	
3	Розробка математичної моделі	04.09.23-18.09.23	
4	Розробка архітектури та загальної структури системи	18.09.23-14.10.23	
5	Розробка структур окремих підсистеми	04.09.23-09.09.23	
6	Програмна реалізація системи	01.10.23-14.10.23	
7	Виготовлення макету системи	07.10.23-14.10.23	
8	Оформлення пояснювальної записки	14.10.23-30.11.23	
9	Захист програмного продукту	27.10.23	
10	Передзахист	04.12.23	
11	Захист		

Студент

_____ **Костянтин ГОНЧАРУК**
(підпис) (ім'я ПРІЗВИЩЕ)

Науковий керівник

_____ **Анатолій ДЕМЧИШИН**
(підпис) (ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Актуальність теми. Актуальність теми полягає в попиті на камери з кутом огляду більше 180 градусів на ринку безпілотних систем різного роду, систем відеоспостереження та ринку стаціонарних камер для онлайн-конференцій.

Метою роботи є створення програмно-апаратного комплексу, що складається з катадіоптричної системи та програмного продукту, який обробляє зображення отримане катадіоптричною камерою та дозволяє переглядати оброблене зображення на плоскому моніторі.

Задачі роботи:

1. Провести огляд існуючих камер з великим кутом зору та зробити вибір засобів розробки.
2. Розробити математичну модель та провести її реалізацію.
3. Провести виготовлення прототипу катадіоптричної камери.
4. Провести тестування розробленого програмно-апаратного комплексу.

Об'єкт дослідження – програмні системи обробки зображення отриманого з катадіоптричної камери.

Предмет дослідження – програмна система роботи з катадіоптричною камерою з кутом огляду 360 градусів, отримання та обробки зображення.

Практична цінність роботи полягає у виготовленні прототипу катадіоптричної системи з програмним забезпеченням, що дозволяє отримувати оброблене зображення у прямокутному форматі з частотою кадрів 30 на секунду. Прототип в комплексі з програмним продуктом можна застосовувати для онлайн-конференцій, систем відеоспостереження та у безпілотних системах, і зокрема БПЛА, які використовують у війні проти росії.

Апробація результатів дисертації. Міжнародна інтернет – конференція «SCIENCE 2023: RESEARCH AND INNOVATION». США, м.Філадельфія, 1-2 грудня 2023.

Дисертація складається зі вступу, п'яти розділів. Повний обсяг дисертації складає 97 сторінок, в тому числі 78 сторінок основного тексту, 17 таблиць, 32

рисунка, 4 сторінки списку використаних джерел у кількості 50 найменувань та 1 додатку.

Ключові слова: *катадіоптрична камера, обробка зображення, сферично-панорамна зйомка, WebGL*

ABSTRACT

Relevance of the topic. The relevance of this project is in the market demand for a camera with a viewing angle of more than 180 degrees in unmanned autonomous vehicles, video surveillance systems and the market of stationary cameras for online conferences.

The goal of the project is creation of a software-hardware complex consisting of a catadioptric optical system and a software product that processes the image received by the catadioptric camera and allows you to view the processed image.

Tasks:

1. Conduct a survey of existing cameras with a wide viewing angle and make a choice of development tools.
2. Develop a mathematical model and implement it in software.
3. Carry out the manufacturing of a prototype of a catadioptric camera.
4. Test the developed hardware and software complex.

The object of research is software systems for image processing obtained from a catadioptric camera.

The subject of the research is a software system for working with a catadioptric camera with an angular view of 360 degrees, received and processed images.

The practical value of the research is that a prototype catadioptric system with software was made, which allows you to receive a processed image with a frequency of 30 frames per second. The prototype in a complex with a software product can be used for online conferencing, video surveillance and in unmanned systems.

Approbation of the results of the dissertation. International Internet – conference "SCIENCE 2023: RESEARCH AND INNOVATION". December 1-2, 2023.

The dissertation consists of an introduction, five chapters and conclusions. The full volume of the dissertation is 91 pages, including 79 pages of the main text, 17 tables, 32 figures, 4 pages of the list of used sources in the amount of 50 names.

Keywords: catadioptric camera, image processing, spherical panoramic photography

ЗМІСТ

ВСТУП.....	8
1 КАМЕРИ З КУТОМ ЗОРУ БІЛЬШЕ 180 ГРАДУСІВ.....	10
1.1 Аналіз пропозицій на ринку камер з великим кутом зору.....	10
1.1.1 Огляд існуючих камер з великим кутом зору	10
1.1.1.1. Insta360 One X3	11
1.1.1.2. GoPro MAX 360.....	11
1.1.1.3. Kodak PIXPRO ORBIT360 4K 360° VR Camera.....	12
1.1.2 Огляд режимів зйомки.....	14
1.1.2.1. «360° сферичний» або «режим віртуальної реальності».	14
1.1.2.2. «197° ультраширокий фронтальний режим».	14
1.1.2.3. Режим «Купола»	15
1.1.3 Огляд технології, що закладена у камерах з великим кутом зору	15
1.2 Принцип роботи катадіоптричної камери	19
2 ЗАСОБИ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ	22
2.1 Огляд засобів розробки програмного продукту для обробки зображення .	22
2.1.1 Огляд мов програмування з бібліотеками, що дозволяють виконувати паралельні обчислення	22
2.1.2 Огляд середовищ розробки	32
2.2 Порівняння мов програмування використаних при розробці програмного продукту	36
2.3 Методи проєціювання поверхні сфери на площину.....	38
2.3.1 Циліндричний метод.....	39
2.3.2 Конічний метод	40
2.3.3 Гномонічний метод.....	41

3 ОПИС ПРОГРАМНО-АПАРATНОЇ РЕАЛІЗАЦІЇ КАТАДІОПТРИЧНОЇ СИСТЕМИ	45
3.1 Розробка математичної моделі обробки зображення	45
3.2 Шлях розробки програмного продукту для обробки зображення	48
3.3 Програмна реалізація продукту по обробці зображення	52
3.3.1 Клас Model	54
3.3.2 Клас ShaderProgram.....	54
3.4 Виготовлення прототипу катадіоптричної системи	55
3.5 Тестування катадіоптричної системи.....	57
4 ОГЛЯД ІНТЕРФЕЙСУ КОРИСТУВАЧА. СИСТЕМНІ ВИМОГИ	62
4.1 Огляд інтерфейсу користувача застосунку	62
4.2 Мінімальні системні вимоги програми.....	64
5 РОЗРОБКА СТАРТАП-ПРОЄКТУ	67
5.1 Опис ідеї проєкту	67
5.2 Технологічний аудит стартапу.....	69
5.3 Аналіз ринкових можливостей запуску стартап-проєкту.....	69
5.4 Розроблення ринкової стратегії проєкту	82
ВИСНОВКИ.....	85
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	86
ДОДАТОК А.....	91

ВСТУП

Використання катадіоптричних камер набуло значного поширення у галузі комп'ютерного зору та панорамних зображень завдяки їхній унікальній здатності отримувати ширококутні зображення з мінімальними спотвореннями. Катадіоптричні системи, що поєднують лінзи та дзеркала, надають потрібний функціонал для галузей, де необхідно отримувати зображення з камери з кутом огляду 360 градусів, включаючи відеоспостереження, безпілотних системах, що фіксують візуальну інформацію навколишнього середовища та віртуальну реальність для керування пілотованим пристроєм. Однак обробка зображень, отриманих з катадіоптричних систем, створює проблеми, які пов'язані зі значними спотвореннями зображень, а також тим, що необроблене зображення складно сприймається людиною.

Магістерську дисертацію присвячено розробці програмного продукту, призначеного для обробки сферичних панорам, отриманих з катадіоптричних систем. Виходячи з того, що катадіоптричної системи фіксують зображення в півкулі сферичного дзеркала, отримані дані потребують алгоритмів для виправлення спотворень перспективи. Запропонований програмний продукт спрямовано на виправлення спотворення та виведення зображення у звичному для людині форматі.

У роботі досліджено аспекти катадіоптричних систем та проблем, пов'язаних з обробкою сферичної панорами, і представлено програмний продукт, розроблений для виправлення спотворень перспективи та виведення зображення у прямокутному форматі. Робота направлена на обробку зображень, зосереджуючись на точності та ефективності перетворення спотворених катадіоптричних зображень, що полягає у мінімізації спотворень перспективи зображення та підвищенню кількості кадрів на секунду відео, за рахунок використання паралельних обчислень, які пропонує графічний процесор.

Катадіоптрична система поєднує в одній оптичній системі ефекти заломлення і відбиття світлових променів, як правило за допомогою лінз

(діоптрика) і сферичних дзеркал (катоптрика). Катадіоптрична система складається з камери та сферичного дзеркала.

Розроблений програмний продукт націлено на збільшення обсягу знань та алгоритмів у галузі панорамних зображень і наданні ресурсу для дослідників, інженерів і практиків, які прагнуть використовувати потенціал катадіоптричних систем у своїх застосунках.

Актуальність цієї полягає в попиті на ринку камер з кутом огляду більше 180 градусів на ринку безпілотних систем різного роду, систем відеоспостереження та ринку стаціонарних камер для онлайн-конференцій.

Метою роботи є створення програмно-апаратного комплексу, що складається з катадіоптричної системи та програмного продукту, який обробляє зображення отримане катадіоптричною камерою та дозволяє переглядати оброблене зображення на плоскому моніторі.

Було поставлено наступні задачі:

1. Провести огляд існуючих камер з великим кутом зору та зробити вибір засобів розробки.
2. Розробити математичну модель та провести її реалізацію.
3. Провести виготовлення прототипу катадіоптричної камери.
4. Провести тестування розробленого програмно-апаратного комплексу.

Практична цінність полягає в тому, що було виготовлено прототип катадіоптричної системи з програмним забезпеченням, що дозволяє отримувати оброблене зображення у прямокутному форматі з частотою кадрів 30 на секунду. Прототип в комплексі з програмним продуктом можна застосовувати для онлайн-конференцій, систем відеоспостереження та у безпілотних системах, і зокрема БПЛА, які використовують у війні проти росії.

1 КАМЕРИ З КУТОМ ЗОРУ БІЛЬШЕ 180 ГРАДУСІВ

В цьому розділі проведено аналіз пропозицій на ринку камер з кутом зору більше 180 градусів, для того щоб зрозуміти які камери є продуктом масового вжитку та які технології використовують розробники камер. Також розглянуто безпосередньо технологію, що складає магістерську дисертацію, а саме катадіоптричну систему: за яким принципом формується зображення в такій системі та фізичних процесів, що робить формування зображення з надзвичайно великим кутом зору можливим.

1.1 Аналіз пропозицій на ринку камер з великим кутом зору

Сьогодні на ринку представлено широкий вибір камер з ширококутним об'єктивом. Камер з надзвичайно великим кутом(більше 180 градусів) огляду на ринку представлено не так багато, більшість з них використовується для зйомки фото та відео у вільному форматі. Стаціонарних камер для конференцій та систем відеоспостереження були знайдені в мережі лише у вигляді прототипів – таких, що не досягли масового користувача. Було зроблено огляд камер з надзвичайно великим кутом огляду. Всі вони досягають такого результату за рахунок об'єднання двох ширококутних об'єктивів.

1.1.1 Огляд існуючих камер з великим кутом зору

В огляд увійшли камери, що доступні пересічному користувачу. І хоча ринок пропонує більше різноманітних пропозицій, вони нелегко доступні користувачу, адже здебільшого виробники таких камер працюють за системою бізнес-бізнесу. Саме з цієї причини до огляду було включено Insta 360 One X3, GoPro MAX 360 та Kodak PIXPRO ORBIT360 4K 360° VR Camera.

1.1.1.1. Insta360 One X3

Панорамна екшн-камера з роздільною здатністю 5.7K з безліччю вбудованих інтелектуальних функцій та режимів. Девайс має ряд покращень в апаратній частині, сенсорі, але найпомітнішим виступає великий екран 2.29".

Зовні камера має плоский вертикально витягнутий форм-фактор. Винятком є дещо змінені габарити, підлаштовані під апаратне оснащення. X3 представлена виключно у темному забарвленні [2], що можна побачити на рисунку 1.1.



Рисунок 1.1 – Insta360 One X3

1.1.1.2. GoPro MAX 360

Використовуючи об'єктиви з протилежного боку, MAX може знімати все, що відбувається під кутом огляду 360°, у вигляді відео з роздільною здатністю до 4992 x 2496 і фотографій 5760 x 2880. Фотографії та відео автоматично зшиваються в камері для гнучкості редагування та легкого обміну. Він також може використовувати лише один об'єктив для зйомки традиційних ширококутних кадрів, коли це необхідно [3]. Дизайн камери можна побачити на рисунку 1.2.



Рисунок 1.2 – GoPro MAX 360

1.1.1.3. Kodak PIXPRO ORBIT360 4K 360° VR Camera

Цю портативну камеру легко використовувати, щоб знімати чудові 360° фотографії та відео, автоматично зшивати їх та ділитися ними в дорозі. Orbit360 4K поміщається у долоні та захищений від пилу, морозу, бризок і ударів.

Використовуючи безкоштовну мобільну програму Pixpro для iOS і Android, інтелектуальні пристрої тепер можуть без зусиль завантажувати 360° відео на YouTube, а 360° фотографії та відео на Facebook [4]. Зовнішній вигляд камери можна побачити на рисунку 1.3.



Рисунок 1.3 – Kodak PIXPRO ORBIT360 4K 360° VR Camera

Розглянемо та порівняємо загальні технічні характеристики камер, що пропонують надзвичайно широкий кут огляду в таблиці. Також в таблиці 1.1 будуть розглянуті більш детальні технічні характеристики для основного об'єктива, адже кожна з представлених камер мають по 2 об'єктива.

Таблиця 1.1 – Порівняльна таблиця камер з широким кутом огляду

Характеристика	Камера		
	Insta360 One X3	GoPro MAX 360	Kodak PIXPRO ORBIT360
Кількість об'єктивів	2	2	2
Фокусна відстань, мм	6,7	8,9	1,633

Продовження таблиці 1.1

Кут поля зору, градуси	170	149	197
Максимальна роздільна здатність фото	11968 x 5984	5760 x 2880	7360 x 3680
Максимальна роздільна здатність відео / максимальна кількість кадрів на секунду при вказаній роздільній здатності	5760x2880 / 30	4992 x 2496 / 30	3840 x 1920 / 24
Ємність акумулятора, мАг	1800	1600	1250
Розміри, мм	46 x 114 x 33,1	64 x 69 x 40	55 x 55 x 67,6
Вага, г	180	163	156
Ціна, грн	25 999	22 999	18 000

Отже, в ході огляду існуючих рішень і одночасного аналізу предметної області, що стосується теми дипломної було виявлено брак пропозицій стаціонарних камер з надзвичайно широким кутом огляду, які б підходили для використання у офісі для онлайн-конференцій та/або відеоспостереження “з коробки”. Попри чудові технічні характеристики оглянутих камер очевидною стала відсутність камер з надзвичайно широким кутом огляду в бюджетному сегменті, адже ціни на розглянуті камери стартують від 18000 гривень. Важко не погодитися, що компактні камери з запропонованим функціоналом це звичайно зручно для пересічного користувача, та при використанні їх для, наприклад, відеоспостереження або онлайн-конференцій, де важлива безперервна робота, ми б спостерігали брак надійності, через їх обмежений час роботи з одним зарядом акумулятора.

1.1.2 Огляд режимів зйомки

Камери зазначені вище пропонують багато різноманітних режимів зйомки фото та відео. Розглянемо на прикладі камери від Kodak деякі з них.

1.1.2.1. «360° сферичний» або «режим віртуальної реальності».

Режим, який використовує два об'єктива і «зшиває» зображення таким чином, що вихідний формат дозволяє зняти повну картину того, що відбувається навколо. Режим віртуальної реальності умовно візуалізовано на рисунку 1.4 [4].

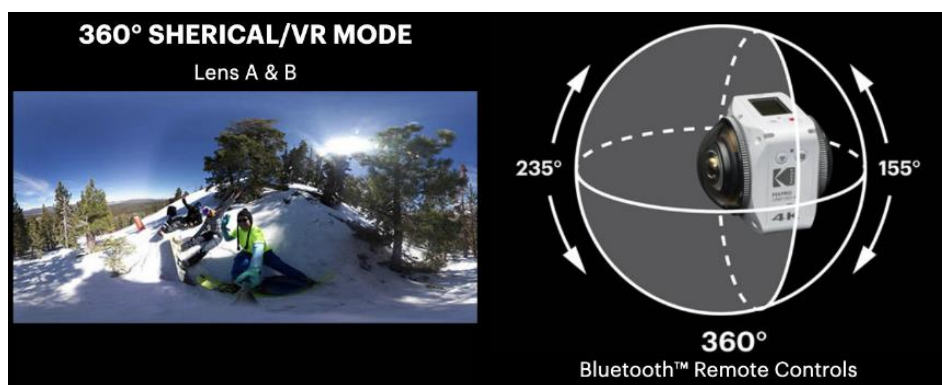


Рисунок 1.4 – «360° сферичний» або «режим віртуальної реальності»

1.1.2.2. «197° ультраширокий фронтальний режим».

Цей режим використовує один об'єктив. Відео та фото, які дозволяє зробити цей режим представляють фото та відео звичайного формату у формі прямокутника зі співвідношенням сторін 16:9. Ультраширокий фронтальний режим умовно візуалізовано на рисунку 1.5 [4].



Рисунок 1.5 – Ультраширокий фронтальний режим

1.1.2.3. Режим «Купола»

Режим, що також використовує лише один об'єктив і дозволяє знімати відео з надзвичайно широким кутом огляду, які потім можуть бути переглянуті та оброблені в додатку або іншому спеціальному програмному забезпеченні. Режим купола умовно візуалізовано на рисунку 1.6 [4].

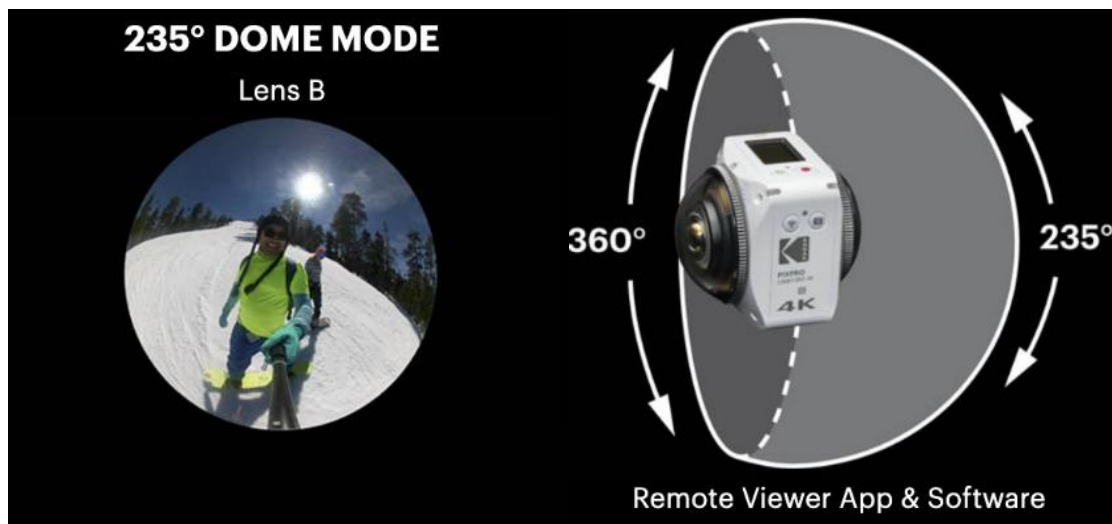


Рисунок 1.6 – Режим купола

1.1.3 Огляд технології, що закладена у камерах з великим кутом зору

Найпопулярніші камери 360 мають дві лінзи, які розташовані навпроти одна одної. Кожна лінза «риб'яче око» має кут огляду понад 180 градусів. Отже, в результаті поля зору двох лінз перекриваються в певній області. Ці області, що перекриваються, накладаються одна на одну. Потім вони зшиваються за допомогою програмного забезпечення для створення 360-градусних зображень. Програмне забезпечення, яке використовується для з'єднання 360-градусних зображень, враховує такі фактори, як спотворення об'єктива та паралакс. Потім він об'єднує зображення або відео, щоб створити цілісне панорамне зображення, яке потім можна переглядати на комп'ютері чи мобільному пристрої.

Точний метод, який використовують кругові камери, може відрізнитися залежно від конкретного пристрою. Але основний процес передбачає використання

двох або більше лінз типу «риб'яче око» для захоплення огляду сцени на 360 градусів. Раніше нам доводилося отримувати ці окремі записи з обох камер, а потім з'єднувати їх за допомогою складного програмного забезпечення. Але сьогодні виробники камер випускають 360 камери з програмним забезпеченням із набагато простішим керуванням для цієї мети. Деякі 360-градусні камери використовують навіть більше об'єктивів, ніж дві, щоб отримати відео ще вищої якості. Але найпопулярніші мають лише дві лінзи. Відео з обох об'єктивів зшивається автоматично за допомогою програми, яка постачається разом із камерою. Тож нам не потрібно використовувати складне програмне забезпечення для зшивання відео вручну.

Існують відео, де користувач тримає кругову камеру за допомогою невидимої палиці для селфі. Цей ефект створюється за допомогою сліпої зони в 360-градусних камерах. Під час використання двох об'єктивів камери поля зору двох об'єктивів перекриваються в певній області. Але є область, де немає огляду з обох камер, де лінзи перекриваються (цю область показано за допомогою засобу 3D-перегляду нижче). Ця область створює сліпу пляму, і об'єкти, які присутні в цій області не фіксуються камерою.

Сліпу пляму можна помітити, лише якщо камера знаходиться дуже близько до пейзажу. Так само, як людина не здатна помітити сліпу пляму в своїх очах (через обробку мозку), людина не може помітити сліпу пляму в 360-градусних камерах. Це пов'язано з тим, що програмне забезпечення в цих камерах може зшивати відзнятий матеріал, щоб виглядало, ніби сліпої зони немає. Компанії, що займаються 360 камерами, використовують цю функцію в своїх інтересах, представивши невидиму палицю для селфі [5].

360-градусні камери руками, формують дуже дивні зображення через сліпу зону. Ділянки, де лінзи перекриваються, часто є проблематичними, оскільки процес зшивання іноді може створити помітне спотворення кінцевого результату. Більшість камер 360 використовують розширені алгоритми зшивання, щоб мінімізувати ці проблеми. Але їх не можна повністю уникнути. довга палиця для

селфі – практичне вирішення цієї проблеми. Цікаво спостерігати, як ці компанії використовують практичні обмеження продукту для своєї реклами.

Для отримання потрібного кадру, треба перекадрувати необхідні частини відзнятого матеріалу. Це робиться за допомогою програми або програмного забезпечення, яке постачається разом із круговою камерою. Крім рефреймінгу, у цих програмах є стільки функцій, таких як відстеження об'єктів і надстабільне відео [5].

Продуктивність 360-градусних камер за слабого освітлення може бути низькою через кілька факторів, зокрема розмір і якість датчика зображення, розмір діафрагми об'єктивів і необхідність фіксувати широкий діапазон рівнів освітлення в одній сцені. Більшість 360-градусних камер споживчого класу мають дуже маленький датчик у кожній моделі камери через менший розмір модуля камери, який вони повинні вмістити. Крім того, об'єктиви, які використовуються в 360-градусних камерах, часто мають меншу діафрагму, ніж традиційні об'єктиви камери, що може обмежити кількість світла, що потрапляє в камеру. Це призводить до більш темного та шумного зображення в умовах слабого освітлення. Тому, як наслідок, він має дуже низьку точність кольору, особливо в умовах слабого освітлення [5].

Добре освітлене середовище має важливе значення для зйомки якісних зображень і відео за допомогою камер 360. 360 камери, які використовуються для високобюджетних комерційних проєктів, мають більші датчики, але вони також дуже дорогі. Оскільки 360-градусні камери мають захоплювати як яскраві, так і тіньові області на зображеннях досить часто, камери забезпечують точне керування рівнями яскравості експозиції камери тощо. Нам потрібно використовувати ці елементи керування, щоб отримати найкращі зображення з 360-градусних камер, особливо в умовах слабого освітлення.

Перспектива від третьої особи – популярна функція 360-градусних камер, яка дозволяє глядачеві побачити себе чи інших людей на сцені з зовнішньої точки зору. Щоб досягти цього, 360-градусні камери використовують палиці для селфі, які дозволяють розташувати камеру на відстані від об'єкта зйомки, водночас знімаючи

повний круговий огляд. Завдяки широкому огляду 360-градусних камер створити вид від третьої особи дуже легко та виглядає так само, як у комп'ютерній грі. Багато мотоблогерів використовують камери 360, щоб отримати зображення від третьої особи для більш драматичного ефекту у відео.

Електронна стабілізація зображення (EIS) працює шляхом обрізання кадру, а потім переміщення кадру, щоб нейтралізувати моменти камери. Оскільки камери 360 захоплюють усі 360 переглядів, їй не потрібно обрізати кадр, оскільки вона має багато місця для переміщення в кадрі та нейтралізації тремтіння камери. Тож він дає надзвичайно стабільні знімки навіть без зовнішніх стабілізаторів камери. Деякі передові 360 камери також використовують оптичну стабілізацію зображення, але більшість популярних 360 камер використовують лише EIS [5].

Відстеження об'єктів є важливою функцією 360-градусних камер, яка дозволяє камері автоматично стежити за певним об'єктом або об'єктом сцени та фокусуватися на ньому, коли він рухається. Ми можемо створити ілюзію камери, яка змінює свій кут, щоб стежити за об'єктом, але насправді камера не рухається. Однак відстеження об'єктів у 360-градусних камерах може бути складним через широкий кут огляду камери та сліпі зони. Крім того, програмне забезпечення камери має бути в змозі точно розпізнавати та відстежувати об'єкт, що може бути складно за умов слабого освітлення чи високого контрасту. Але в пристойно освітленому середовищі ця функція працює дуже добре.

Майбутнє 360-градусних камер багатообіцяюче, оскільки ці пристрої продовжують розвиватися та вдосконалюватися з точки зору технології та функціональності. Одним із поточних обмежень 360-градусних камер є їх низька продуктивність в умовах слабого освітлення. Однак у міру вдосконалення датчиків камер майбутні 360-градусні камери, ймовірно, запропонують кращу продуктивність за слабого освітлення. Багато сучасних 360-градусних камер невеликі та портативні, але майбутні моделі можуть стати ще компактнішими та легшими, що зробить їх більш доступними для щоденного використання. Майбутні камери 360 можуть запропонувати розширені функції, такі як визначення глибини

та обробка в реальному часі, що ще більше підвищить якість і функціональність ефекту занурення.

1.2 Принцип роботи катадіоптричної камери

Катадіоптричні камери досягають майже півсферичного поля зору за допомогою перспективної камери та катадіоптричної системи [7], яка являє собою комбінацію відбивних (катоптричних) і заломлюючих (діоптричних) елементів [6, 8]. Така система схематично показана в лівій частині рисунку 1.7 для параболічного дзеркала. В правій частині рисунку 1.7 показано одне всенаправлене зображення, отримане описаною параболічною катадіоптричною камерою [9].

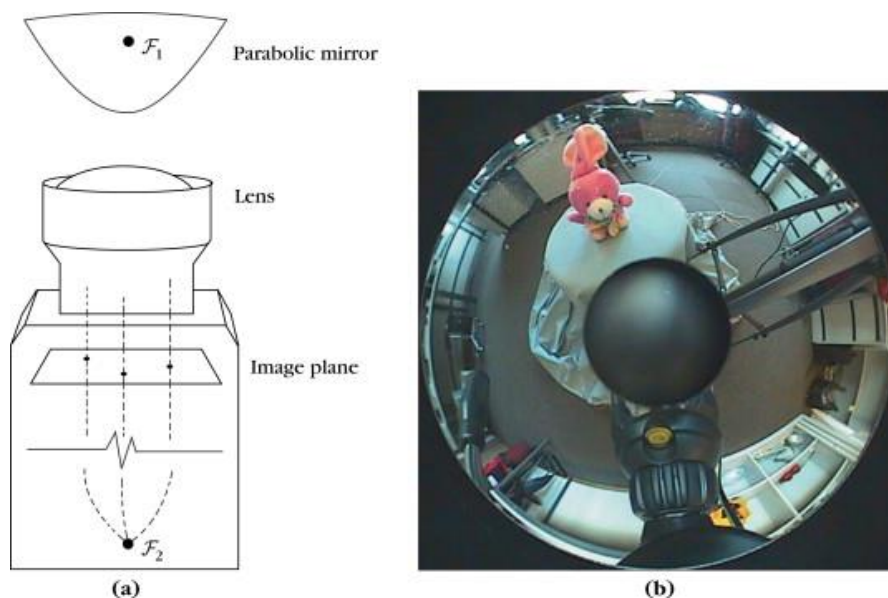


Рисунок 1.7 – Всеспрямована система з параболічним дзеркалом

На рисунку 1.7. можна побачити параболічне дзеркало розміщене в параболічному фокусі F_1 ; інший фокус F_2 знаходиться на нескінченності, натхненний [6]. В правій частині рисунку 1.7 можна побачити всенаправлене зображення, отримане параболічною катадіоптричною камерою, що схематично показано в лівій частині рисунку 1.7.

Формування катадіоптричного зображення вивчали в [6, 10, 11]. Особливий інтерес представляють центральні катадіоптричні системи, оскільки вони мають

єдину ефективну точку зору. Ця властивість важлива не тільки для полегшення аналізу знятої сцени, але й для виконання мультиракурсної 3D-реконструкції. Зображення, отримані перспективною камерою зі світла, відбитого катадіоптричною системою, нелегко аналізувати, оскільки лінії в 3D-просторі проектуються на конічні перерізи [6, 12, 13]. Уніфікуючу модель для катадіоптричної проективної геометрії для центральних катадіоптричних камер створили Гейер і Данілідіс у 2001 році [6]. Будь-яка центральна катадіоптрична проекція еквівалентна композиції двох відображень на сфері. Перше відображення являє собою центральну сферичну проекцію з центром сфери, що падає на фокус дзеркала і не залежить від форми дзеркала. Друге відображення – це проекція точки на головній осі кулі на площину, перпендикулярну до цієї осі. Однак положення точки проекції на вісь кулі залежить від форми дзеркала. Модель, розроблена Гейером і Данілідісом [6], включає перспективні, параболічні, гіперболічні та еліптичні проекції. Катадіоптрична проективна структура дозволяє нам отримати ефективний і простий аналіз сцени з всенаправлених зображень, знятих катадіоптричними камерами.

Тепер ми більш детально опишемо проективну модель для параболічного дзеркала, де друге відображення представляє проекцію північного полюса на площину, яка включає екватор. Зокрема, друге відображення відоме як стереографічна проекція, і воно конформне

Розглянемо переріз параболоїда в катадіоптричній системі з параболічним дзеркалом. Це показано на рисунку 1.8. Усі точки параболі рівновіддалені від фокуса f і директриси d . l і бути перпендикулярною до параболічної осі. Якщо коло має центр l і радіуса, що дорівнює подвоєній фокусній відстані параболоїда, то коло і параболі двічі перетинають пряму l , а директриса дотична до кола. Північний полюс N кола – це точка, діаметрально протилежна перетину кола і директриси. Точка P проектується на коло з його центра, що дає $P1$. Це еквівалентно проективному представленню, де проективний простір (набір променів) представлений у вигляді кола. Видно, що $P2$ – стереографічна проекція точки $P1$ на пряму l від північного полюса N , де $P1$ – точка перетину променя P і кола.

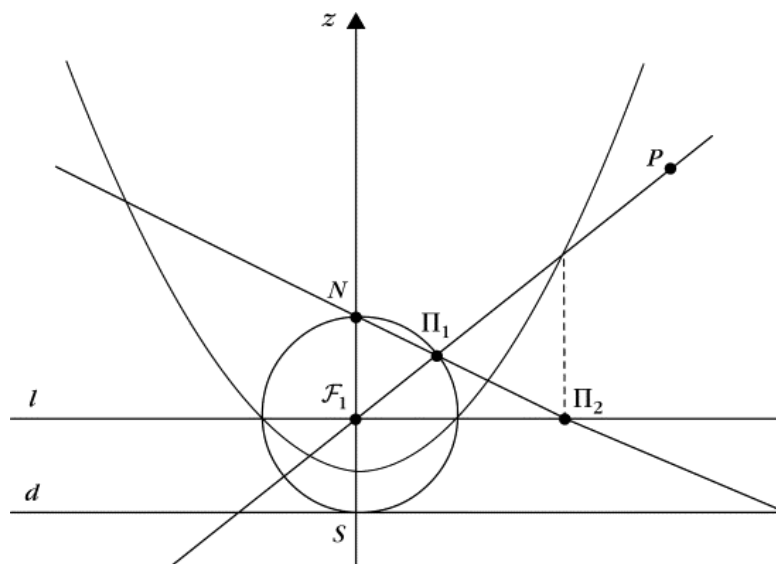


Рисунок 1.8 – Поперечний переріз відображення всенаправленого зображення на сфері

Таким чином, ми можемо зробити висновок, що параболічна проекція точки P дає точку Π_2 , яка є колінеарною з Π_1 і N . Розширюючи це міркування до трьох вимірів, проекція параболічним дзеркалом еквівалентна проекції на сферу (Π_1) з подальшим стереографічним проекція (Π_2) [7]. Прямим наслідком цього результату є те, що параболічна катадіоптрична проекція є конформною, оскільки вона представляє композицію двох конформних відображень [6].

Для інших типів дзеркал у катадіоптричних системах положення точки проекції у другому відображенні є функцією ексцентриситету ϵ конічника (див. теорему 1, встановлену Гейєром і Данілідісом [6]). Для гіперболічних дзеркал з $\epsilon > 1$ та еліптичних дзеркал з $0 < \epsilon < 1$ точка проекції лежить на головній осі сфери, між центром сфери та північним полюсом. Перспективну камеру також можна розглядати як вироджений випадок катадіоптричної системи з коником ексцентриситету $\epsilon = \infty$. У цьому випадку точка проекції для другого відображення збігається з центром сфери.

2 ЗАСОБИ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ

Основним фактором при виборі засобів розробки є наявність можливості керування паралельними обчисленнями, які пропонує відеокарта. Програмний продукт, який виконує обробку відео з користю для продуктивності та швидкодії за допомогою паралельних обчислень. Через це вибір постав між засобами розробки, які мають функціонал, що дозволяє отримати переваги паралельних обчислень, а саме Java з OpenGL та JavaScript з WebGL.

2.1 Огляд засобів розробки програмного продукту для обробки зображення

Для початку було зроблено огляд мов програмування в парі з сумісними їм графічними бібліотеками, які дозволяють отримати переваги паралельних обчислень для обробки графічної інформації, а також відповідних їм середовищ розробки.

2.1.1 Огляд мов програмування з бібліотеками, що дозволяють виконувати паралельні обчислення

В цьому підпункті будуть розглянуті Java, а точніше бібліотека з додатковими спрощеннями Java (Processing) з Open Graphics Library та JavaScript з Web Graphics Library.

Java виступає як широко використовувана об'єктно-орієнтована мова програмування та програмна платформа, яка присутня на багатьох пристроях, таких як ноутбуки, мобільні пристрої, ігрові консолі та медичні пристрої. Її правила та синтаксис ґрунтуються на мовах C та C++.

Помітна перевага використання Java для розробки програмного забезпечення полягає в її переносимості. Після кодування програми Java на ноутбучі її

перенесення на мобільний пристрій стає простим процесом. Мова, задумана в 1991 році Джеймсом Гослінгом з Sun Microsystems (пізніше придбаної Oracle), спочатку була спрямована на можливість «написати один раз і запустити будь-де».

Важливо відрізнити Java від JavaScript. На відміну від JavaScript, який не потребує компіляції, код Java має пройти через цей процес. Крім того, хоча JavaScript обмежується веб-браузерами, Java демонструє можливість працювати на різних платформах.

Незважаючи на швидкий приплив нових і передових інструментів розробки програмного забезпечення, які витісняють раніше важливі продукти, довговічність Java заслуговує на увагу. Понад два десятиліття з моменту свого заснування Java залишається основною мовою для розробки програмного забезпечення. Розробники продовжують віддавати перевагу Java замість таких альтернатив, як Python, Ruby, PHP, Swift, C++ та інші, що робить його вирішальним навиком на конкурентному ринку праці. [15].

Processing – це мова програмування з відкритим кодом і середовище розробки, створене на основі мови програмування Java. Він спеціально розроблений для художників, дизайнерів та інших творчих професіоналів, які хочуть створювати інтерактивну графіку, анімацію та інші візуальні програми.

Processing був створений у 2001 році Беном Фраєм і Кейсі Рісом, які на той час були студентами Массачусетського технологічного інституту (MIT). Вони хотіли створити мову програмування, яку було б легко вивчити та використовувати, але водночас була достатньо потужною для створення складних інтерактивних програм [16].

Однією з головних цілей Processing є зробити програмування більш доступним для людей, які не є професійними розробниками програмного забезпечення. Мова має простий синтаксис, який легко читати та писати, і вона надає велику кількість попередньо створених функцій і бібліотек, які можна використовувати для створення складних візуальних ефектів без необхідності писати багато коду з нуля.

Processing також надає інтерактивне середовище розробки (IDE), яке полегшує написання, тестування та налагодження коду. IDE містить такі функції, як підсвічування коду, автозавершення та візуальний налагоджувач, що полегшує пошук і виправлення помилок у вашому коді.

Оскільки Processing побудовано на основі Java, він повністю сумісний з бібліотеками Java і може використовуватися для створення повноцінних програм Java. Це означає, що Processing не обмежується лише створенням візуальних програм; його також можна використовувати для створення керованих даними програм, ігор та іншого програмного забезпечення.

У Processing є велика та активна спільнота користувачів, які створили широкий спектр проєктів, використовуючи цю мову. Веб-сайт Processing надає безліч ресурсів для вивчення мови, включаючи підручники, довідкові посібники та велику колекцію прикладів проєктів.

Отже, якщо підсумувати Processing – це бібліотека низького рівня для анімації та графічного інтерфейсу з відкритим кодом, побудована на Java з додатковими спрощеннями, такими як додаткові класи, псевдоніми математичних функцій і операцій.

JavaScript – це динамічна, об'єктно-орієнтована [18] прототипна мова програмування. Реалізація стандарту ECMAScript зазвичай використовується для створення сценаріїв на веб-сторінках. Це забезпечує взаємодію з користувачами на стороні клієнта, дозволяючи контролювати браузері, асинхронний обмін даними з серверами та змінювати структуру та зовнішній вигляд веб-сторінок. JavaScript класифікується як прототипна сценарна мова в рамках парадигми динамічного набору тексту. Окрім своєї прототипної природи, JavaScript також містить елементи інших парадигм програмування, наприклад імперативного та часткового функціонального програмування. Крім того, він демонструє пов'язані архітектурні властивості, такі як динамічна та слабка типізація, автоматичне керування пам'яттю, успадкування прототипів і розпізнавання функцій як об'єктів першого класу. Актуальною на даний момент є версія ES2020, що вийшла у червні 2020 року [19].

JavaScript, наразі, є однією з найпопулярніших мов програмування в інтернеті [20]. В перші роки існування, більшість професійних програмістів скептично ставилися до мови, цільова аудиторія якої складалася з програмістів-аматорів [21]. AJAX, технологія, яка набула популярності, революціонізувала можливість динамічного відтворення сторінок і відкрила нові можливості завдяки динамічному завантаженню та вставці даних у документи. Поява AJAX змінила сприйняття мови серед професійної спільноти, а подальша адаптація відповідно до стандартів ES6+ привела до появи численних цінних функцій, необхідних для ефективного програмування. Як наслідок, з'явилося безліч практичних навичок JavaScript, що охоплюють тестування та налагодження, що призвело до створення різноманітних бібліотек і фреймворків. Крім того, застосування JavaScript вийшло за межі браузера [22].

OpenGL є двовимірним і тривимірним графічним API, який може похвалитися широким використанням на різних обчислювальних платформах. Незалежність від операційної системи та віконної системи, а також прозорість для мережі, OpenGL дає розробникам можливість створювати високоефективне та візуально захоплююче графічне програмне забезпечення. Він обслуговує різні ринки, включаючи САПР, створення контенту, енергетику, розваги, розробку ігор, виробництво, медицину та віртуальну реальність [23].

Реєстр OpenGL містить специфікації основного API та shading language; специфікації схвалених Khronos і виробником розширень OpenGL; заголовні файли, що відповідають специфікаціям; і відповідна документація, включаючи специфікації, розширення та заголовки для API GLX, WGL і GLU [24].

Реєстр OpenGL є частиною об'єднаного реєстру OpenGL для OpenGL, OpenGL ES і OpenGL SC, який включає реєстр XML API зарезервованих нумерантів і функцій. Поточна версія OpenGL 4.6.

OpenGL – (англ. Open Graphics Library – відкрита графічна бібліотека). На фундаментальнішому рівні OpenGL служить специфікацією – документом, що описує набір функцій разом із їхньою точною поведінкою. Виробники апаратного забезпечення, керуючись цими специфікаціями, розробляють реалізації, які, по

суті, є бібліотечними функціями, що відповідають вимогам, викладеним у специфікації OpenGL. Ці реалізації створено для використання апаратних можливостей, коли це можливо. У випадках, коли апаратне прискорення обмежено, функції виконуються через програмне забезпечення. Виробники проходять спеціальні тести на відповідність, щоб підтвердити свою реалізацію як офіційну реалізацію OpenGL. Отже, розробникам програмного забезпечення потрібно лише ознайомитися з функціями, детально описаними в специфікації, залишаючи фактичну реалізацію розробникам обладнання.

Реалізації OpenGL знаходять застосування в різних операційних системах, включаючи Linux, MacOS, Microsoft Windows і численні UNIX-подібні системи. Ігрові платформи, такі як Sony PlayStation, також виграють від ефективних реалізацій OpenGL. Для платформ, які не підтримують цю специфікацію від виробників, доступні різні програмні реалізації OpenGL. Відкрита бібліотека Mesa виділяється як комплексний OpenGL-сумісний API програмування. Незважаючи на те, що Mesa є неофіційною реалізацією через відповідні витрати на ліцензування для офіційного визнання, Mesa залишається повністю сумісною зі специфікацією OpenGL.

Web Graphics Library, WebGL, або бібліотека веб-графіки – це API JavaScript для відтворення інтерактивної 2D- і 3D-графіки у веб-браузерах без необхідності використання додаткових плагінів. Він використовує потужність графічних процесорів (GPU), щоб увімкнути високоефективне апаратно-прискорене відтворення графіки у веб-додатках. WebGL базується на OpenGL ES, широко використовуваному стандарті для графіки вбудованих систем, і надає аналогічні можливості веб-браузерам. Його можна використовувати в елементах HTML `<canvas>`. Ця відповідність дає змогу API використовувати переваги апаратного прискорення графіки, яке забезпечує пристрій користувача.

WebGL дозволяє розробникам створювати візуально насичену та складну графіку безпосередньо у веб-браузерах. Він підтримує як 2D, так і 3D візуалізацію, що дозволяє створювати захоплюючі візуальні враження. Використовуючи обчислювальну потужність графічного процесора, WebGL може справлятися з

інтенсивними обчислювальними завданнями, що робить його ідеальним для додатків, які вимагають рендеринга графіки в реальному часі, таких як ігри, моделювання та візуалізація даних. WebGL забезпечує інтерактивний досвід користувача, дозволяючи розробникам реагувати на введення користувача в реальному часі. Ця інтерактивність необхідна для веб-ігор, інтерактивної візуалізації даних та інших цікавих веб-додатків. WebGL підтримується основними веб-браузерами, включаючи Chrome, Firefox, Safari та Edge, забезпечуючи стабільну продуктивність і відтворення на різних платформах і пристроях. WebGL легко інтегрується з JavaScript, дозволяючи розробникам маніпулювати графічними елементами, обробляти взаємодії користувачів і створювати динамічний вміст у веб-додатках. WebGL є особливо потужним для створення тривимірної графіки, що робить його популярним вибором для 3D-ігор, віртуальної реальності (VR) і моделювання в Інтернеті.

Таким чином, WebGL дозволяє розробникам створювати візуально привабливі та інтерактивні веб-додатки, використовуючи можливості сучасних графічних процесорів. Його здатність створювати та обробляти складну 2D– і 3D-графіку безпосередньо у веб-браузерах робить її фундаментальною технологією для веб-розробників, зацікавлених у розширенні меж веб-візуального досвіду.

Підтримка WebGL є у всіх сучасних браузерах, однак пристрій користувача також має мати апаратне забезпечення, яке підтримує ці функції [26].

API WebGL 2 підтримує більшу частину набору функцій OpenGL ES 3.0; функціонал надається через інтерфейс WebGL2RenderingContext.

WebGL – це кросплатформний безкоштовний відкритий веб-стандарт для низькорівневого API 3D-графіки на основі OpenGL ES, доступного для ECMAScript через елемент HTML5 Canvas. Розробники, знайомі з OpenGL ES 2.0, розпізнають WebGL як API на основі шейдерів, що використовує GLSL із конструкціями, семантично подібними до базового API OpenGL ES. Він дуже близький до специфікації OpenGL ES, з деякими поступками щодо того, що розробники очікують від мов із керуванням пам'яттю, таких як JavaScript. WebGL 1.0 відкриває набір функцій OpenGL ES 2.0; WebGL 2.0 відкриває API OpenGL ES 3.0.

WebGL приносить 3D без плагінів у Інтернет, реалізований прямо у веб-переглядачі. Основні постачальники браузерів Apple (Safari), Google (Chrome), Microsoft (Edge) і Mozilla (Firefox) є членами робочої групи WebGL [27]. Структуру програмної системи зображено на рисунку 2.1 [28].

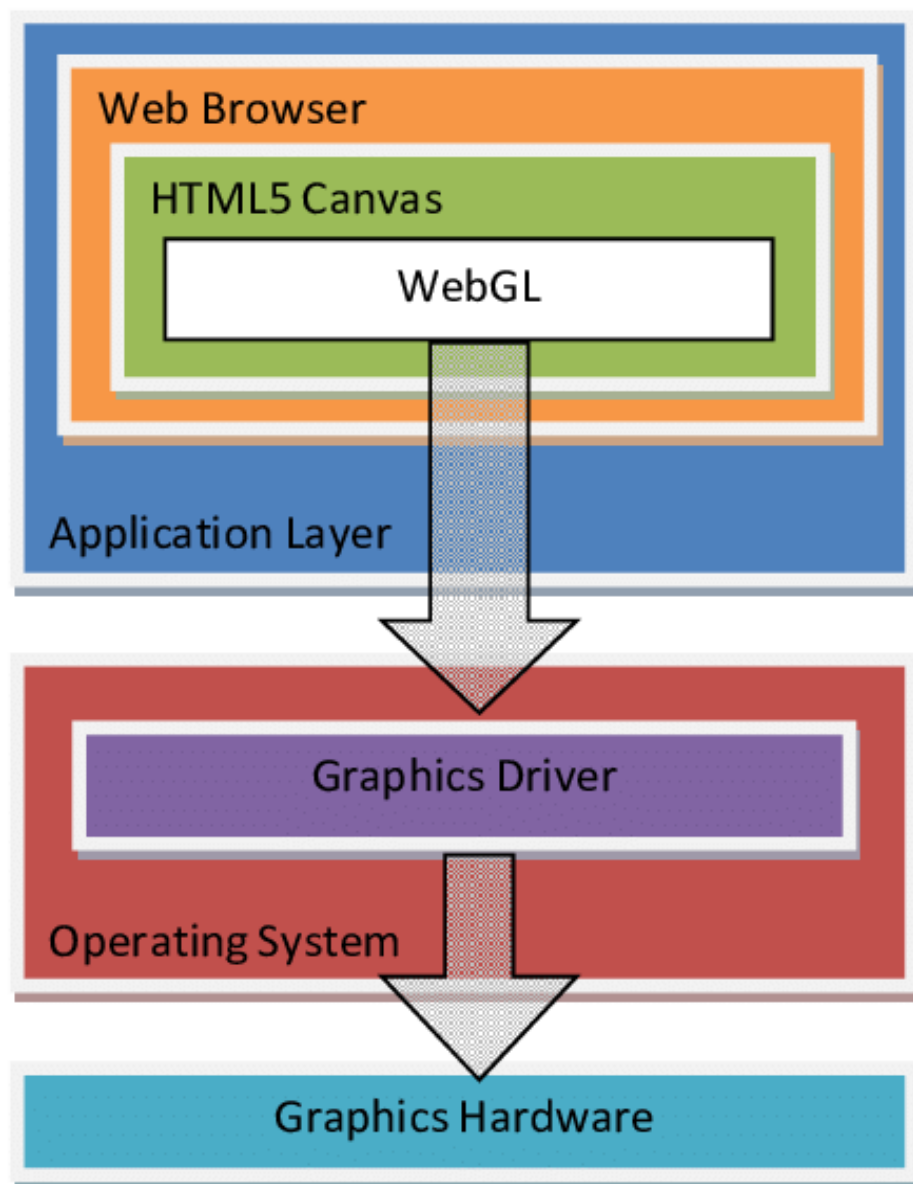


Рисунок 2.1 – Структура програмної системи

У програмі WebGL дані зазвичай завантажуються на графічний процесор із власною системою координат. Координати в такій системі координат обмежені $(1,1,1)$ і $(-1,-1,-1)$. Координати x , y і z кожної вершини мають бути від $-1,0$ до $1,0$. Координати за межами цього діапазону не будуть видимі. Такі координати зазвичай називаються координатами кліпу. На рисунку 2.2 зображено графічна репрезентація вищезазначеного діапазону [29].

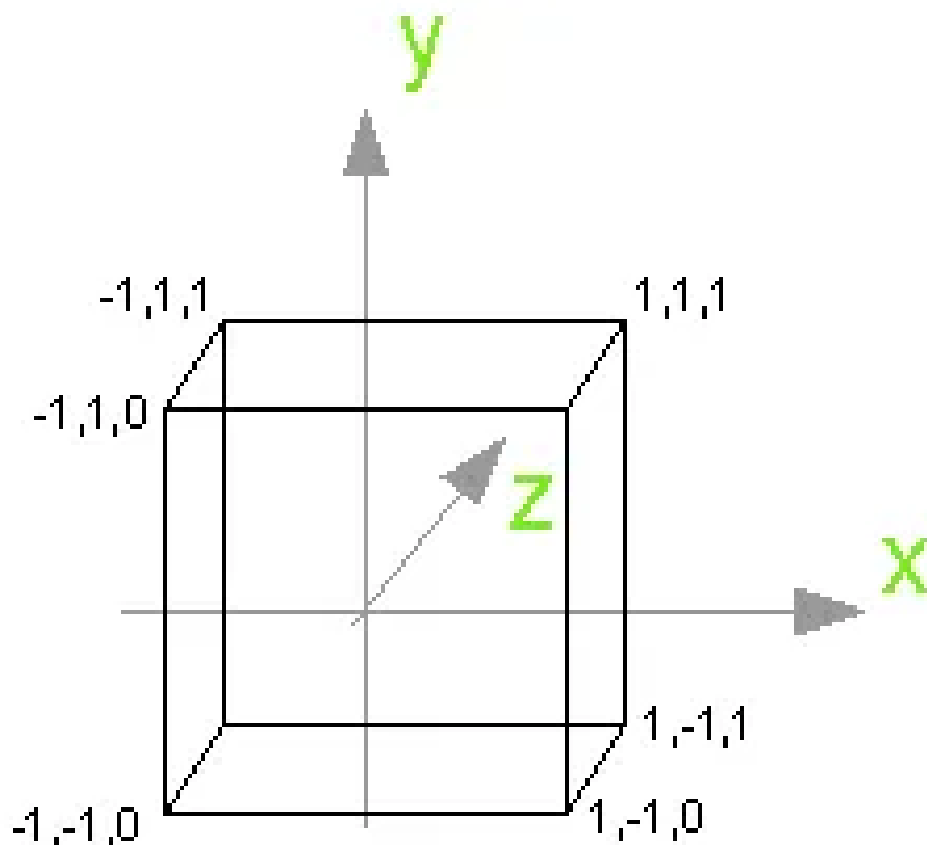


Рисунок 2.2 – Діапазон координат всередині якої відображається інформація. Їх також часто називають нормалізованими координатами пристрою (NDC). Така система координат використовується у вершинному шейдері. Вершинний шейдер – це етап у конвеєрі візуалізації, який обробляє окремі вершини, які потім передаються растеризеру (растеризатор перетворює вершини на пікселі на екрані). Проекція в координати кліпу здійснюється за допомогою матриці проекції. Існує дві основні проекції: ортогографічна проекція та перспективна проекція. Матриця ортогографічної проекції визначає кубоподібний квадрат усеченої форми, де кожна вершина за межами цього прямокутника обрізана. Програми CAD зазвичай працюють з такою проекцією. Перспективна проекція зазвичай використовується в іграх, де симулюється реальне життя, а об'єкти, розташовані далі, здаються набагато меншими. На рисунку 2.3 можна побачити як влаштовано перспективну та ортогографічну проекцію [29].

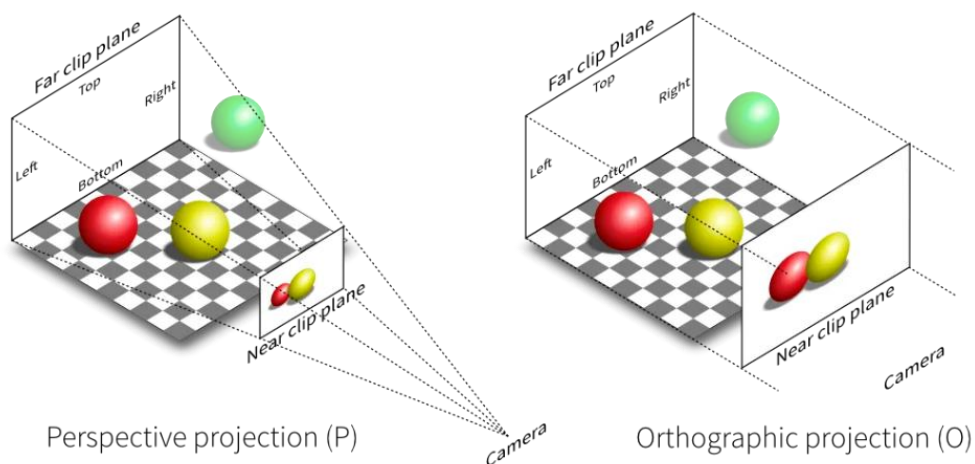


Рисунок 2.3 – Наочне порівняння проєкцій різних типів

На останньому кроці координати кліпу повинні бути перетворені на екранні координати. Цей процес перетворює координати з $-1,0$ і $1,0$ на діапазон координат, визначений видимим полотном. Полотно має свою систему координат. Верхній лівий кут екранної системи координат – $(0,0)$, а нижній правий – (ширина, висота). Графічне зображення вищеприписаної системи координат можна побачити на рисунку 2.4 [29].

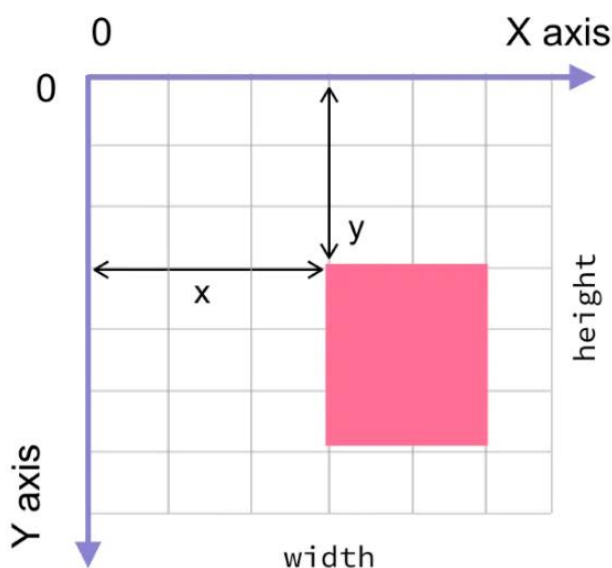


Рисунок 2.4 – Екранна системи координат

OpenGL Shading Language (GLSL) – це мова високого рівня з синтаксисом на основі мови програмування C. Він був створений OpenGL ARB (OpenGL Architecture Review Board), щоб надати розробникам більш прямий контроль над графічним конвеєром без використання мови асемблера ARB або мов, специфічних для апаратного забезпечення [30].

Шейдери використовують GLSL (OpenGL Shading Language), спеціальну мову OpenGL із синтаксисом, подібним до C. GLSL виконується безпосередньо графічним конвеєром. Існує кілька типів шейдерів, але для створення графіки в Інтернеті зазвичай використовуються два: вершинні шейдери та фрагментні (піксельні) шейдери. Вершинні шейдери перетворюють позиції форми в координати тривимірного малюнка. Фрагментні шейдери обчислюють відтворення кольорів фігури та інших атрибутів [31].

GLSL не такий інтуїтивно зрозумілий, як JavaScript. GLSL строго типізований, і тут багато математики, пов'язаної з векторами та матрицями. Це може стати дуже складним – дуже швидко.

Вершина – це точка в тривимірній системі координат. Вершини можуть і зазвичай мають додаткові властивості. Тривимірна система координат визначає простір, а вершини допомагають визначити форми в цьому просторі

Шейдер – це функція необхідна для відображення графічних елементів на екрані. Шейдери працюють на GPU (графічному процесорі), оптимізованому для таких операцій. Використання графічного процесора для роботи з шейдерами звільняє частину роботи від центрального процесора. Це дозволяє центральному процесору зосередити свою обчислювальну потужність на інших завданнях, наприклад на виконанні коду.

Вершинні шейдери маніпулюють координатами в тривимірному просторі та викликаються один раз для кожної вершини. Метою вершинного шейдера є налаштування змінної `gl_Position` – це спеціальна, глобальна та вбудована змінна GLSL. `gl_Position` використовується для збереження позиції поточної вершини [31].

Функція `void main()` – це стандартний спосіб визначення змінної `gl_Position`. Усе всередині `void main()` буде виконано вершинним шейдером. Вершинний шейдер дає змінну, яка містить спосіб проектування положення вершини в 3D-просторі на 2D-екран. Функція називається однаково для вершинного та фрагментного шейдера

Фрагментні (або текстурні) шейдери визначають кольори RGBA (червоний, зелений, синій, альфа) для кожного пікселя, що обробляється – один фрагментний

шейдер викликається один раз для кожного пікселя. Метою фрагментного шейдера є налаштування змінної `gl_FragColor`. `gl_FragColor` – це вбудована змінна GLSL, як `gl_Position` [31].

Результатом обчислень є змінна, що містить інформацію про колір RGBA.

2.1.2 Огляд середовищ розробки

В цьому підрозділі будуть оглянуті два середовища розробки: середовище розробки Processing та Visual Studio Code, які використовуватимуться для розробки на мові програмування Processing та JavaScript відповідно.

Середовище розробки Processing (PDE) полегшує написання програм Processing. Програми пишуться в текстовому редакторі і запускаються натисканням кнопки Виконати. У Processing комп'ютерна програма називається ескізом. Ескізи зберігаються в Sketchbook, який є папкою на вашому комп'ютері [32].

На ескізах можна малювати дво- і тривимірну графіку. За замовчуванням рендерер призначений для малювання двовимірної графіки. P3D-рендерер дає змогу малювати тривимірну графіку, що включає керування камерою, освітленням і матеріалами. P2D-рендерер є швидким, але менш точним рендерером для малювання двовимірної графіки. І P2D, і P3D рендерери прискорюються, якщо ваш комп'ютер має OpenGL-сумісну відеокарту.

Можливості Processing розширені бібліотеками та інструментами. Бібліотеки дозволяють ескізам робити щось, що виходить за межі основного коду Processing. Існують сотні бібліотек, наданих спільнотою Processing, які можна додати до ескізів, щоб увімкнути нові речі, як-от відтворення звуків, комп'ютерне бачення та робота з розширеною 3D-геометрією. Інструменти розширюють PDE, щоб полегшити створення ескізів, надаючи інтерфейси для таких завдань, як вибір кольорів.

У Processing є різні режими програмування, що дозволяє розгортати ескізи на різних платформах і програмувати різними способами. Режим Java є типовим. Інші

режими програмування можна завантажити, вибравши «Додати режим...» у меню у верхньому правому куті PDE.

Середовище розробки Processing (PDE) складається з простого текстового редактора для написання коду, області повідомлень, текстової консолі, вкладок для керування файлами, панелі інструментів із кнопками для типових дій і серії меню. Параметри меню змінюються від режиму до режиму. Два базові вікна середовища розробки Processing зображенні на рисунку 2.5 [32].

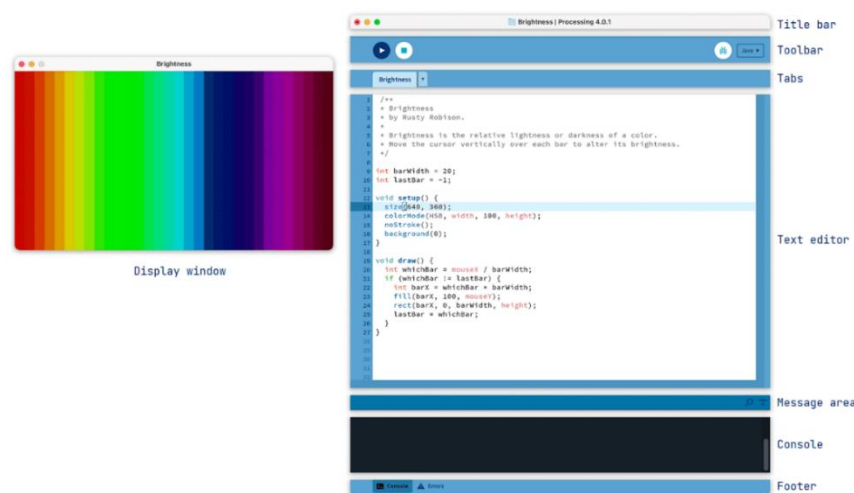


Рисунок 2.5 – Інтерфейс середовища розробки Processing

Код пишеться в текстовому редакторі. Він має функції для вирізання/вставки та пошуку/заміни тексту. Область повідомлень надає відгук під час збереження та експорту, а також відображає помилки. На консолі відображається текстовий вивід шляхом обробки ескізів, включаючи повні повідомлення про помилки та текстовий вивід ескізів за допомогою функцій `print()` і `println()` (зверніть увагу, що консоль добре працює для випадкових повідомлень, але не призначена для високошвидкісного виведення в реальному часі). Кнопки на панелі інструментів можуть запускати та зупиняти програми.

Середовище розробки Processing (PDE) має широкі можливості налаштування. Найпоширеніші параметри можна змінити у вікні «Параметри», розташованому в меню «Файл» у Windows і Linux і в меню «Обробка» у macOS. Повний список налаштувань зберігається у файлі "preferences.txt". Цей файл можна відкрити та редагувати безпосередньо лише тоді, коли Processing не запущено. Ви

можете знайти розташування цього файлу на вашому комп'ютері, прочитавши нижній лівий кут вікна налаштувань.

Processing має три вбудовані рендери екрана. Засіб візуалізації за замовчуванням призначений для малювання двовимірних фігур. P2D – це швидший, але менш точний рендерер для малювання двовимірних фігур. P3D – для тривимірної геометрії; він також може керувати камерою, освітленням і матеріалами. Рендери P2D і P3D прискорюються, якщо ваш комп'ютер має відеокарту, сумісну з OpenGL. Функція `smooth()` впливає на кількість згладжування для кожного рендерера. Перегляньте посилання на `smooth()` для отримання додаткової інформації.

Рендеринг, який використовується для кожного ескізу, визначається за допомогою функції `size()`. Якщо засіб візуалізації явно не визначено у `size()`, він використовує засіб візуалізації за замовчуванням, як показано в наступній програмі

Щоб змінити рендерер, додайте третій параметр до `size()`.

Було докладено значних зусиль, щоб код обробки працював однаково в різних рендерерах, але наразі існують деякі невідповідності, які пояснюються в довідці. Для отримання додаткової інформації перегляньте довідковий запис `size()`.

З випуском Processing 4.0 рендерер FX2D доступний як окрема бібліотека. Це корисно для 2D-графіки на великих дисплеях або дисплеях з високою роздільною здатністю, і зазвичай набагато швидше, ніж рендерер за замовчуванням, хоча деякі функції все ще є «експериментальними». Встановіть його, вибравши Sketch → Імпорт бібліотеки → Додати бібліотеку та вибравши JavaFX зі списку.

Processing використовує систему координат з початком координат у верхньому лівому куті. Якщо ваш ескіз має ширину 320 пікселів і висоту 240 пікселів, координата (0, 0) є верхнім лівим пікселем, а координата (320, 240) – нижнім правим. Останній видимий піксель у нижньому правому куті екрана знаходиться в позиції (319, 239), оскільки пікселі малюються праворуч і нижче координати. Початок координат та додатній напрям осей координат зображено на рисунку 2.6.

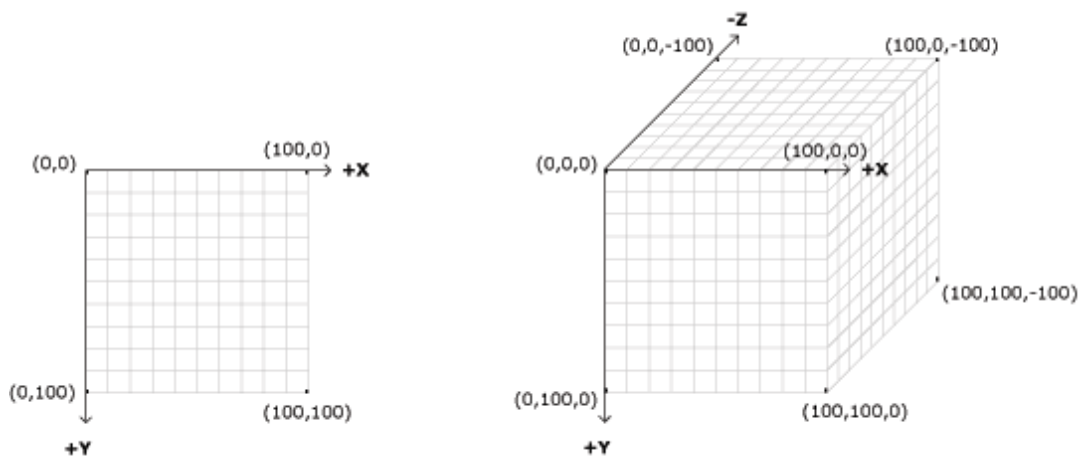


Рисунок 2.6 – Різновиди систем координат, що використовуються середовищем розробки Processing

Використовуючи тривимірну систему координат P3D, z-координата дорівнює нулю на поверхні зображення, а від’ємні z-значення рухаються назад у просторі. Під час малювання в 3D камера розташована в центрі екрана [32].

Налагоджувач обробки – це інструмент для діагностики проблем із ескізом. Увімкніть його, щоб призупинити ескіз під час виконання та переходити код по одному рядку за раз. Налагоджувач можна увімкнути через меню «Файл» («Налагодження» > «Увімкнути налагоджувач») або клацнувши піктограму налагоджувача – метелика у верхньому правому куті PDE.

Коли налагоджувач увімкнено, програма працює як звичайно, але зупиняється на «точках зупинки». Щоб створити точку зупину, встановіть курсор на рядок, де потрібно призупинити ескіз, і виберіть Налагодження > Перемкнути точку зупину. Комбінація клавіш Command-B. Щоб видалити точку зупину, знову виберіть Перемкнути точку зупину. Коли додається точка зупину, номер рядка замінюється символом: <>.

Запуск ескізу в режимі налагодження призводить до призупинення ескізу в будь-якій точці зупинки. Під час паузи поточні значення змінних відображаються на окремій панелі. Ви можете перейти до наступної точки зупинки, вибравши «Продовжити», або переходити рядок за рядком коду за допомогою «Кроку». Покрокове виконання працює лише в межах поточної функції.

Visual Studio Code – це спрощений редактор коду з підтримкою таких операцій розробки, як налагодження, виконання завдань і керування версіями. Він має на меті надати саме ті інструменти, які потрібні розробнику для швидкого циклу створення коду та налагодження, і залишає складніші робочі процеси для більш повнофункціональних IDE, таких як Visual Studio IDE. [33] Visual Studio Code, який також зазвичай називають VS Code, [34] розроблений Microsoft для Windows, Linux і macOS. [35]

В опитуванні розробників Stack Overflow 2023 серед 86 544 респондентів Visual Studio Code назвали найпопулярнішим інструментом середовища розробника, при цьому 73,71% повідомили, що вони ним користуються. Опитування також виявило, що Visual Studio Code частіше використовують ті, хто вчиться програмувати, ніж професійні розробники (78% проти 74%). [36].

2.2 Порівняння мов програмування використаних при розробці програмного продукту

Основним фактором при виборі засобу розробки була наявність можливості керування паралельними обчисленнями, які має запропонувати відеокарта. Програмний продукт, який би виконував обробку відео міг би з користю для продуктивності та швидкодії використати паралельні обчислення. Через це вибір постав між засобами розробки, які мають функціонал, що дозволяє отримати переваги паралельних обчислень, а саме Processing з OpenGL та JavaScript з WebGL.

Вибір між Processing з OpenGL і JavaScript із WebGL для програми обробки відео залежить від кількох факторів, зокрема конкретних вимог та цільової платформи для програми. Давайте розберемо переваги та недоліки для обох варіантів.

Processing відома своєю простотою та легкістю використання. Якщо ви новачок у програмуванні чи обробці відео, «Processing» може забезпечити м'який

курс навчання. Processing дозволяє швидко створювати прототипи візуальних програм. Його інтуїтивно зрозумілий синтаксис і вбудовані функції можуть прискорити процес розробки. Навколо Processing існує активна спільнота, що означає, що доступна обширна документація, навчальні посібники та підтримка спільноти. Processing з OpenGL забезпечує високоефективне відтворення графіки. Якщо програма буде вимагати складних візуалізацій або симуляцій, OpenGL може використовувати апаратне прискорення для швидшої обробки. Кросплатформеність: програми Processing можуть з незначними змінами в коді працювати на різних платформах, включаючи Windows, macOS та Linux.

Варто також зауважити недоліки цього засобу розробки: хоча OpenGL може значно підвищити продуктивність, рівень оптимізації може вимагати більше зусиль порівняно з WebGL. Якщо програма має бути переважно веб-орієнтованою, обробка може бути не оптимальним вибором для веб-розгортання. WebGL більше підходить для веб-додатків.

JavaScript з WebGL може запропонувати такі переваги: WebGL розроблено спеціально для веб-браузерів. Якщо ваша програма повинна працювати у веб-середовищі, JavaScript із WebGL є очевидним вибором. WebGL використовує потужність GPU, забезпечуючи високоефективну обробку графіки. Він може ефективно виконувати складні завдання обробки відео. JavaScript у поєднанні з WebGL може надати дуже інтерактивний і динамічний інтерфейс користувача, який може бути вирішальним для певних програм. WebGL – це веб-стандарт, який підтримується основними браузерами. Це забезпечує стабільну роботу на різних платформах і пристроях.

Даний засіб розробки має недоліки, які варто взяти до уваги: сумісність із веб-переглядачами. Незважаючи на те, що WebGL широко підтримується, старі чи менш поширені веб-переглядачі можуть мати обмежену або взагалі не підтримувати WebGL, що може обмежити охоплення вашої програми. Оскільки програми WebGL працюють у веб-переглядачі, слід пам'ятати про заходи безпеки, особливо якщо ваша програма містить конфіденційні дані або взаємодію з користувачем.

На початку розробки було обрано саме Processing через легкість швидкого прототипування. Це дозволяє перевірити початкові напрями дослідження та різні шляхи розробки. При цьому це середовище розробки дозволяє розробку повноцінного програмного продукту, власно що і являється основною метою переддипломної практики.

Попри це після розробки початкового прототипу засіб розробки було змінено, адже сьогодні найвищою популярністю користуються саме веб-додатки. Було обрано JavaScript з WebGL.

JavaScript є універсальною і широко використовуваною мовою програмування, здебільшого відомою через її ролі у веб-розробці. Це – високорівнева мова програмування, що дозволяє розробникам створити динамічні, інтерактивні, та адаптивні веб-сторінки. JavaScript може бути одразу інтегрований з HTML та CSS, що забезпечує чудовий досвід користувача та створення функціональних сайтів. Це універсальна мова, яка може програмувати front end та серверну частину веб-додатків, може бути обраною для створення мобільних застосунків і навіть повноцінних програм для ПК. JavaScript підтримує об'єктно-орієнтовану та функціональну парадигми розробки, що робить його гарним вибором для широкого спектру різноманітних програм. З значними екосистемами бібліотек, JavaScript вважається дуже гарною технологією в сучасному світі розробки програмного забезпечення.

Отже, оскільки мені важлива була стандартизованість та кросплатформеність JavaScript із WebGL став більш підходящим вибором через веб-сумісність, високу продуктивність та інтерактивність.

2.3 Методи проєціювання поверхні сфери на площину

Задачі проєціювання поверхні сфери на площину не є чимось новим. Більше того, ця задача несвідомо почала своє існування, коли в 6 ст. до н.е. Піфагор висунув теорію про кулястість землі. Так деякі мапи, які частково зображають поверхню нашої планети на площині датуються 13 століттям.

На сьогоднішній день існує багато способів вирішення такої задачі, при цьому варто зауважити, що усунути спотворення, яке виникає при такій трансформації не представляється можливим. Розглянемо декілька методів проєціювання поверхні сфери на площину, що використовуються у картографії, які б могли бути корисними при створенні математичної моделі.

2.3.1 Циліндричний метод

Нормальна циліндрична проєкція – це будь-яка проєкція, у якій меридіани відображаються на рівних вертикальних лініях, а кола широти (паралелі) – у горизонтальні лінії [37].

Відображення меридіанів на вертикальних лініях можна візуалізувати, уявивши циліндр, вісь якого збігається з віссю обертання Землі. Цей циліндр обертається навколо Землі, проєктується на неї, а потім розгортається.

За геометрією побудови циліндричні виступи простягаються на відстані схід-захід. Величина розтягування однакова на будь-якій вибраній широті на всіх циліндричних проєкціях і визначається секансом широти як кратним масштабу екватора. Різні циліндричні проєкції відрізняються один від одного виключно їх простяганням з півночі на південь.

Розтягнення північ-південь дорівнює розтягуванню схід-захід: шкала схід-захід відповідає шкалі північ-південь: конформна циліндрична або Меркатора; це надмірно спотворює області у високих широтах.

Розтягнення з півночі на південь зростає з широтою швидше, ніж розтягування зі сходу на захід: циліндрична перспективна (або центральна циліндрична) проєкція; непридатний, оскільки викривлення ще гірше, ніж у проєкції Меркатора.

Розтягнення з півночі на південь зростає з широтою, але менш швидко, ніж розтягування схід-захід: наприклад, циліндрична проєкція Міллера.

Відстані північ-південь ні розтягнуті, ні стиснуті: рівнопрямокутна проєкція або «plate carrée».

Стиснення з півночі на південь дорівнює косинусу широти (величині, зворотній розтягуванню зі сходу на захід): рівноплоща циліндрична. Ця проекція має багато іменованих спеціалізацій, що відрізняються лише константою масштабування, наприклад ортографічна проекція Галла–Пітерса або Галла (неспотворена на паралелях 45°), Бермана (неспотворена на паралелях 30°) і циліндрична рівна площа Ламберта (неспотворена на паралелях). екватор). Оскільки ця проекція масштабує відстань з півночі на південь за величиною, зворотною розтягуванню зі сходу на захід, вона зберігає площу за рахунок форм.

У першому випадку (Mercator) масштаб схід-захід завжди дорівнює масштабу північ-південь. У другому випадку (центральный циліндричний) масштаб північ-південь перевищує масштаб схід-захід всюди від екватора. Кожен інший випадок має пару січних ліній – пару однакових широт протилежного знака (або екватора), на яких масштаб схід-захід збігається зі масштабом північ-південь.

Звичайні циліндричні проекції відображають всю Землю як кінцевий прямокутник, за винятком перших двох випадків, коли прямокутник простягається нескінченно у висоту, зберігаючи незмінну ширину. Графічна інтерпретація циліндричного методу зображено на рисунку 2.7 [37].

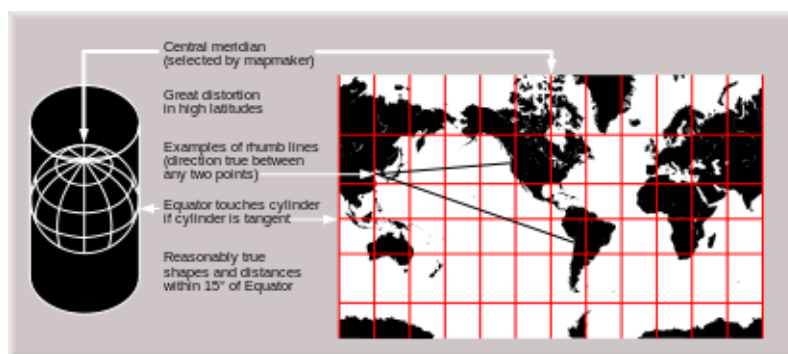


Рисунок 2.7 – Циліндричний метод

2.3.2 Конічний метод

Термін «конічна проекція» використовується для позначення будь-якої проекції, в якій меридіани відображаються на рівновіддалені лінії, що радіально виходять від вершини, а кола широти (паралелі) відображаються в дуги кіл з центром у вершині. [38]

Під час створення конічної карти виробник карти довільно вибирає дві стандартні паралелі. Ці стандартні паралелі можна візуалізувати як січну, де конус перетинає земну кулю, або, якщо виробник карт двічі обирає ту саму паралель, як дотичну лінію, де конус дотичний до земної кулі. Отримана конічна карта має низькі спотворення в масштабі, формі та площі поблизу стандартних паралелей. Відстані вздовж паралелей на північ від обох стандартних паралелей або на південь від обох стандартних паралелей розтягнуті; відстані вздовж паралелей між стандартними паралелями стискаються. Коли використовується одна стандартна паралель, відстані вздовж усіх інших паралелей розтягуються. Графічна інтерпретація конічного методу зображено на рисунку 2.8 [37].

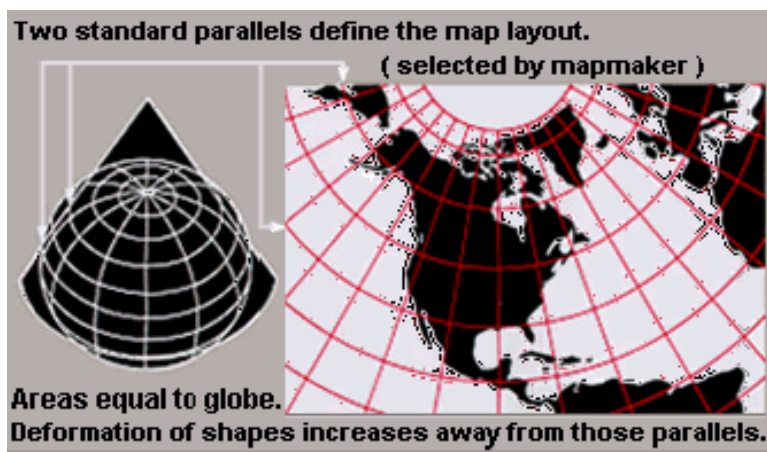


Рисунок 2.8 – Конічний метод

2.3.3 Гномонічний метод

Гномонічна проекція, також відома як центральна проекція або прямолінійна проекція, – це перспективна проекція сфери з центром проекції в центрі сфери на будь-яку площину, яка не проходить через центр, найчастіше на дотичну площину. При гномонічній проекції кожне велике коло на сфері проектується на пряму на площині (велике коло – це геодезична на сфері, найкоротший шлях між будь-якими двома точками, аналогічний прямій на площині). [39] У більш загальному вигляді можна взяти гномонічну проекцію будь-якої n -вимірної гіперсфери на гіперплощину.

Проекція є n -вимірним узагальненням тригонометричного дотичного, який відображається з кола на пряму, і, як і у випадку з дотичною, кожна пара

антиподальних точок на сфері проектується в одну точку на площині, тоді як точки на площині, що проходить через центр сфери і паралельна площині зображення, проектується на нескінченні точки; часто проекцію розглядають як взаємну відповідність між точками на півкулі та точками на площині, і в цьому випадку будь-яка кінцева частина площини зображення представляє частину півкулі. [40]

Гномонічна проекція – азимутальна (радіально-симетрична). У центрі проєктованого зображення спотворення форми не відбувається, але спотворення швидко зростає від нього.

Гномонічна проекція виникла в астрономії для побудови сонячних годинників і картографування небесної сфери. Він зазвичай використовується як проекція географічної карти та може бути зручним у навігації, оскільки курси великого кола нанесені у вигляді прямих ліній. Прямолінійні фотооб'єктиви здійснюють перспективну проекцію світу на площину зображення; це можна розглядати як гномонічну проекцію сфери зображення (абстрактна сфера, що вказує напрямки кожного променя, що проходить через камеру, змодельовану як отвір-обскура). Гномонічна проекція використовується в кристалографії для аналізу орієнтації ліній і площин кристалічних структур. Він використовується в структурній геології для аналізу орієнтації площин розломів. У комп'ютерній графіці та комп'ютерному представленні сферичних даних відображення куба – це гномонічна проекція сфери зображення на шість граней куба.

У математиці простір орієнтацій неорієнтованих прямих у 3-вимірному просторі називається реальною проєктивною площиною і зазвичай зображується або «проєктивною сферою», або її гномонічної проекцією. Коли кут між прямими використовується як міра відстані, цей простір називається еліптичною площиною. Гномонічна проекція 3-сфери одиничних кватерніонів, точки якої представляють тривимірні повороти, призводить до векторів Родрігеса. Гномонічна проекція гіперболоїда з двох аркушів, що розглядається як модель гіперболічної площини, називається моделлю Бельтрамі–Клейна. Графічна інтерпретація гномонічного методу зображено на рисунку 2.9.

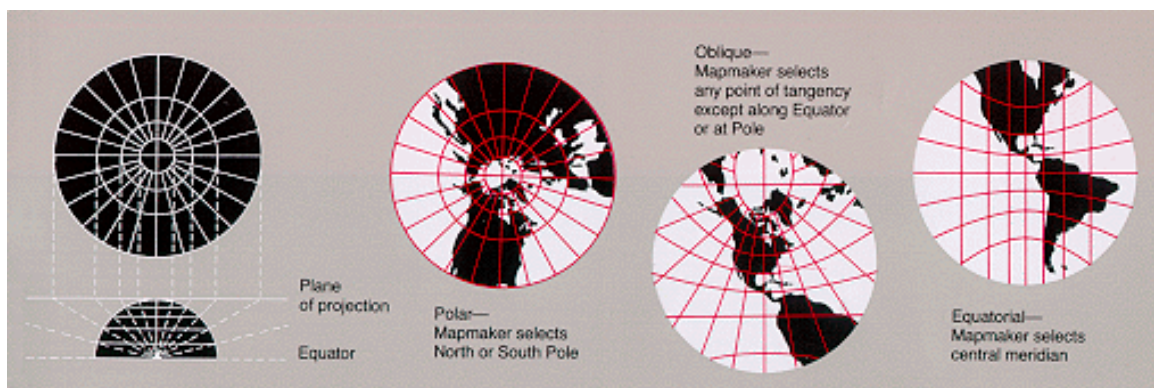


Рисунок 2.9 – Гномонічний метод

У картографії картографічна проекція – це будь-який із широкого набору перетворень, які використовуються для представлення вигнутої двовимірної поверхні земної кулі на площині [41, 42, 43]. У картографічній проекції координати, часто виражені як широта та довгота, місць на поверхні земної кулі перетворюються на координати на площині [44, 45]. Проекція є необхідним етапом у створенні двовимірної карти і є одним з істотних елементів картографії.

Усі проекції кулі на площину обов'язково певним чином і певною мірою спотворюють поверхню [46]. Залежно від призначення карти одні спотворення допустимі, а інші ні; тому існують різні картографічні проекції, щоб зберегти одні властивості кулеподібного тіла за рахунок інших властивостей. Вивчення картографічних проекцій полягає насамперед у характеристиці їх спотворень. Немає обмежень на кількість можливих картографічних проекцій [47]. Більш загально, проекції розглядаються в кількох областях чистої математики, включаючи диференціальну геометрію, проєктивну геометрію та многовиди.

Незважаючи на буквальне значення назви, проекція не обмежується перспективними проекціями, наприклад тими, що є результатом відкидання тіні на екран, або прямолінійного зображення, створеного камерою-обскурою на плоскій плівковій пластині. Натомість будь-яка математична функція, яка чітко й плавно перетворює координати з кривої поверхні на площину, є проекцією. Небагато прогнозів у практичному використанні є перспективними.

Більшість цієї статті припускає, що поверхня, яку потрібно відобразити, є сферою. Землю та інші великі небесні тіла, як правило, краще моделювати як

сплюснені сфероїди, тоді як маленькі об'єкти, такі як астероїди, часто мають неправильну форму. Поверхні планетних тіл можна картографувати, навіть якщо вони надто неправильні, щоб добре моделювати їх за допомогою сфери чи еліпсоїда [48]. Тому, у більш загальному плані, картографічна проекція – це будь-який метод сплюснення безперервної кривої поверхні на площині.

Найбільш відомою картографічною проекцією є проекція Меркатора [47]. Ця картографічна проекція має властивість бути конформною. Однак протягом 20-го століття його критикували за розширення регіонів, розташованих далі від екватора [48]. На відміну від цього, проекції рівних площ, такі як синусоїдальна проекція та проекція Галла–Пітерса, показують правильні розміри країн відносно один до одного, але спотворюють кути. Національне географічне товариство та більшість атласів віддають перевагу картографічним проекціям, які є компромісом між площею та кутовими спотвореннями, таким як проекція Робінсона та трипельна проекція Вінкеля [47, 49].

Вище описані методи слугуватимуть базисом при розробці математичної моделі.

3 ОПИС ПРОГРАМНО-АПАРATНОЇ РЕАЛІЗАЦІЇ КАТАДІОПТРИЧНОЇ СИСТЕМИ

У розділі описано задачу, що поставлена перед розробкою, розробку математичної моделі, шлях розробки, який пройдено перед кінцевою реалізацією математичної моделі та безпосередньо програмну реалізацію. В доповнення розробленого програмного продукту було також виготовлено макет катадіоптричної системи, про що також буде написано у даному розділі.

3.1 Розробка математичної моделі обробки зображення

Сферичне дзеркало відображає все навколо. Стояла задача вивести це зображення в звичному для людини форматі фото\відео, а саме у вигляді прямокутника. Одна камера може фіксувати тільки півсферу дзеркала катадіоптричної системи. Розгляньмо точку зору камери, див. Рисунок 3.1.

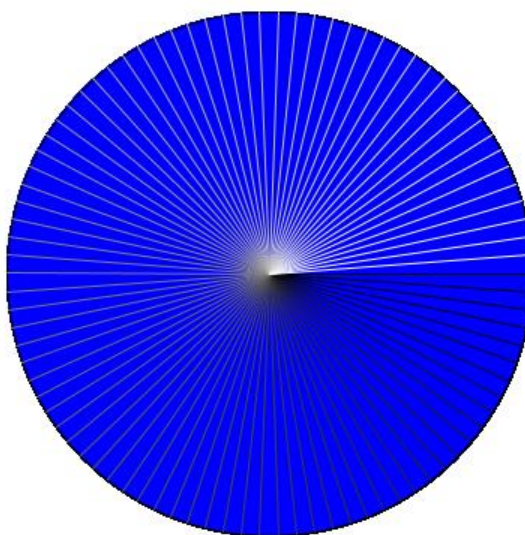


Рисунок 3.1 – Точка зору камери на сферичне дзеркало

Камера фіксує відображення сфери, що містить зображення 360 градусів середовища навколо осі направленості камери. Цей кут змінюється рівномірно, а

отже, зміна координати обробленого зображення по осі абсцис буде також лінійною.

З віссю ординат відображення змінюється нелінійно. Розгляньмо розріз сфери по одній з ліній що зображена на рисунку 3.1. На розрізі показано промені що надходять в камеру з відображення на сфері, а також нормалі поверхні сфери у відповідних місцях.

На рисунку 3.2 показано промені що відбиваються від сфери у напрямку камери і нормалі у відповідних точках на сфері.

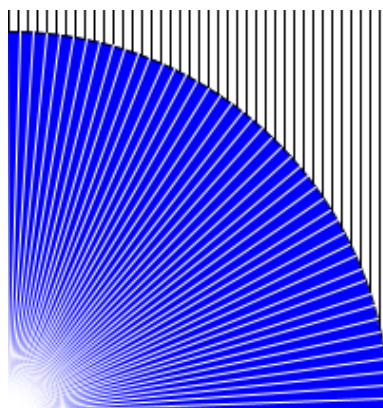


Рисунок 3.2 – Промені до камери з відображення на сфері

Ми знаємо що кут падіння променя світла в однорідному середовищі дорівнює куту відбивання, отже ми можемо відслідкувати звідки відбитий промінь прийшов, тобто, провести зворотнє трасування променів, що й показано на рисунку 3.3.

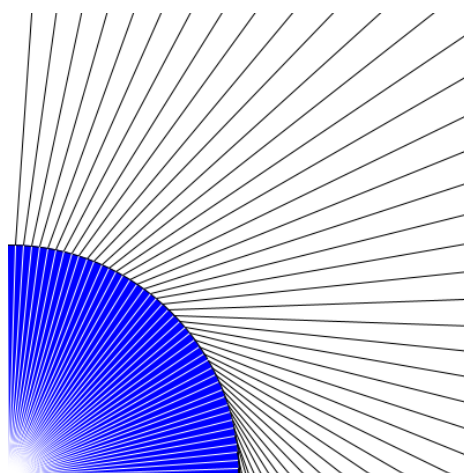


Рисунок 3.3 – Зворотнє трасування відбитих променів

Для того щоб зрозуміти за яким законом нелінійно буде змінюватись вісь ординат для створення обробленого зображення без додаткових штучних

спотворень на наступному рисунку буде показано розріз площини зображення, а також діапазон відбитих променів, які нас цікавлять. Нас цікавлять відбиті промені, що не будуть містити камери, а також частин корпусу, та візуальної інформації, що була відбита набагато нижче лінії горизонту. Зеленим умовно показано діапазон, що цікавить нас, червоним – той діапазон, який не увійде до обробленого зображення, див. Рисунок 3.4.

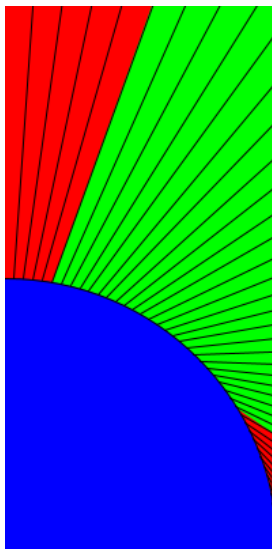


Рисунок 3.4 Діапазон кутів, що становлять інтерес

В результаті було отримано функцію залежності відбитого кута від координати на осі ординат обробленого зображення. Графік залежності показано на рисунку 3.5. Координата текселя змінюється за законом косинуса, що залежить від функції оберненого тангенса.

$$r = \cos((\tan^{-1}(y) + \pi * 0.5) * 0.5) * 0.5, \quad (1)$$

де r – відстань від початку координат,

y – координата по осі ординат обробленого зображення

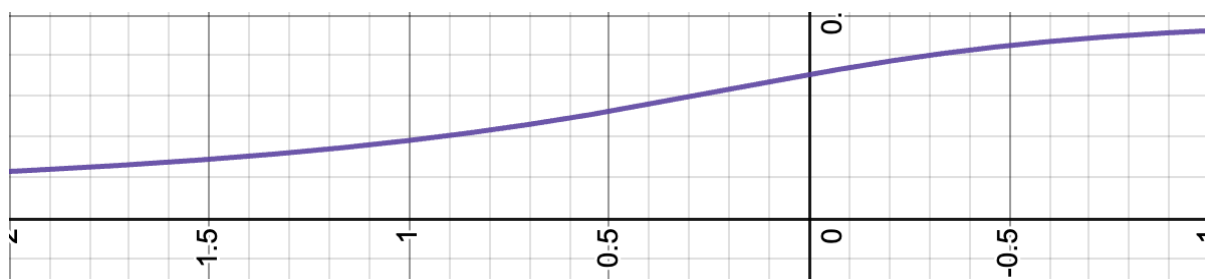


Рисунок 3.5 – Графік залежності відбитого кута від координати на осі ординат обробленого зображення

3.2 Шлях розробки програмного продукту для обробки зображення

Для перевірки концепції програми, що дозволяє переглядати результат сферично-панорамної зйомки катадіоптричної системи на екрані пристрою було запропоновано метод текстуровання для накладення отриманого з катадіоптричної системи, а саме відображення навколишнього середовища на дзеркальній сфері на прямокутну область, що є звичним способом перегляду інформації, зокрема медіаконтенту на екрані користувацького пристрою початкового зображення. За початковою задумкою, такий спосіб накладання відображення зі сфери на прямокутну область дозволив би зекономити обчислювальні можливості пристрою передаючи невеликий буфер інформації графічному процесору, який би безпосередньо виконував операцію текстуровання.

Початкова перевірка концепції дозволила зрозуміти, що це створення такої програми є доцільним, адже не зовсім зрозуміле початкове відображення зі сфери тепер є цілком зрозумілим для людського сприйняття.

При цьому також стають очевидними недоліки такого способу проєціювання відображення зі сфери певні спотворення, яких би можна уникнути використовуючи інший спосіб проєціювання зображення отримане за допомогою катадіоптричної системи на прямокутну область.

Наступним було запропоновано спосіб текстуровання, який би враховував спотворення по осі ординат.

Варто зазначити, що для мінімізації спотворень при використанні вищезазначених методів необхідно скоротити відстань між точками текстури таким чином, виходить, що чим більше точок тим менше спотворення. При цьому неможливо повністю уникнути спотворень, адже є обмеження на: спотворення буде зменшуватись, допоки відстань між точками текстур більша ніж відстань між сусідніми пікселями — найменшою складовою зображення. Програма для обчислень використовує засоби розробки, що дозволяють залучати обчислювальні можливості графічного процесора пристрою, на якому виконується програма, а

саме паралельні обчислення на які, на відміну від центрального процесора, що розрахований на послідовні обчислення здатен графічний процесор. Вище зазначений підхід має місце на існування, адже передаючи інформацію один раз, або з тривалою періодичністю це дійсно може зекономити ресурс обчислювальних можливостей пристрою. Осільки ж у даному випадку, при якому передачу буферу необхідно виконувати безперервно, і можливо, протягом тривалого часу, при цьому уникаючи затримки, адже необхідно здійснювати обробку відеопотоку в реальному часі даний підхід не є оптимальним. Враховуючи це, стає зрозумілим, що передавати і опрацьовувати об'ємні буфери інформації стає не вигідним в порівнянні з опрацюванням кожного пікселя прямокутної області, на яку накладається зображення зі сфери, за певним законом, уникаючи передачу масивного буферу інформації, що і призвело до наступного і остаточного шляху програмної реалізації поставленої задачі.

Обробка зображення в шейдері стала наступною на черзі для перевірки концепції шляху розробки. В цьому методі буфер інформації що передається буде складати лише чотири вертекси, які потрібні для формування прямокутної області на яку буде накладено зображення та текстури, що міститиме поточне зображення отримане з катадіоптричної системи. Зазначимо, що оскільки місце розташування прямокутної області відносно області виведення графічної інформація веб-застосунку (canvas) змінюватись не буде, то вертекси, що представляють собою інформацію про розташування геометричної фігури передаються лише один раз на початку програми. В разі необхідності змінити положення прямокутної області відносно canvas треба буде передати оновлену інформацію, але оскільки передбачено, що положення прямокутної області фіксоване, така можливість не була імплементована. На відміну від вертексів у випадку відеопотоку текстура, що міститиме поточне зображення отримане з камери буде змінюватись з часом, а саме з кожним новим кадром, частота оновлення, яких дорівнює 30 кадрів на секунду буде змінюватись і частина буфер інформації, що міститиме текстуру, що в свою чергу міститиме поточне зображення отримане з катадіоптричної системи.

Скорочення буферу інформації, що передається дозволило відчутно оптимізувати роботу програми. Оскільки обчислення, що визначають координату знаходження одиниці інформації кольору(кольору пікселя) не здійснюватиметься заздалегідь – ці обчислення тепер відбуватимуться у фрагментному шейдері для кожного пікселя окремо, і на відміну від попереднього способу, в якому ця інформація отримувалась послідовним чином перед виконанням коду шейдерів, в цьому способі обчислення будуть відбуватися одночасно, що дозволить скоротити час на обробку кожного нового кадру і таким чином створити враження безперервного відтворення відео, зробивши час затримки, що витрачається на обробку кадра мінімально можливим.

Праворуч від необробленого зображення, що знаходиться в лівому верхньому кутку знаходиться наближення обробленого зображення. Оскільки оброблене зображення є зашироким для зручного перегляду було прийнято рішення зробити наближення області інтересу. Область наближення визначається місцезнаходженням мишки користувача та обмежено областю обробленого зображення, тобто, таким чином, щоб можна було вільно переміщати мишку, але при цьому зображення області інтересу наближало лише оброблене зображення. На рисунку 3.3.1 наведено блок-схему частини коду, яка відповідає, щоб вищенаведені умови виконувались. На відміну від обчислень, що визначають координату знаходження одиниці інформації кольору, які проводяться відеопроцесором – дана частина коду виконується центральним процесором комп'ютера або пристрою, на якому вони здійснюються. В цьому порівнянні і є фундаментальна різниця між паралельними і послідовними обчисленнями, адже коли центральний процесор прораховує, чи виконується умова, для обчислень йому необхідна суміжна інформація. На противагу, під час здійснень обчислень кольору пікселя відеопроцесору не потрібно зберігати інформацію про суміжні пікселі.



Рисунок 3.6 – Блок схема коду, що визначає область наближення

Розмір області на яку виводиться оброблене зображення визначається шириною(червона лінія зі стрілками, див. рисунок 3.7) вікна: оброблене зображення займає всю ширину вікна браузера, в якому відкрито програму. Висота(синя лінія зі стрілками) області виведення обробленого зображення визначається за формулою довжини дуги кола де довжина дуги дорівнює ширині області виведення(червона лінія зі стрілками) і відповідає довжині дуги кола(червоне кільце) з радіусом, що дорівнює половині радіуса дзеркальної сфери(синя лінія) на зображення отриманого з катадіоптричної системи. Висота(синя лінія зі стрілками) дорівнює радіусу, який можна знайти по даній формулі довжини кола і відповідає радіусу дзеркальної сфери(синя лінія) на

зображення отриманого з катадіоптричної системи. На рисунку 3.7 показано схематичне зображення інтерфейсу користувача.

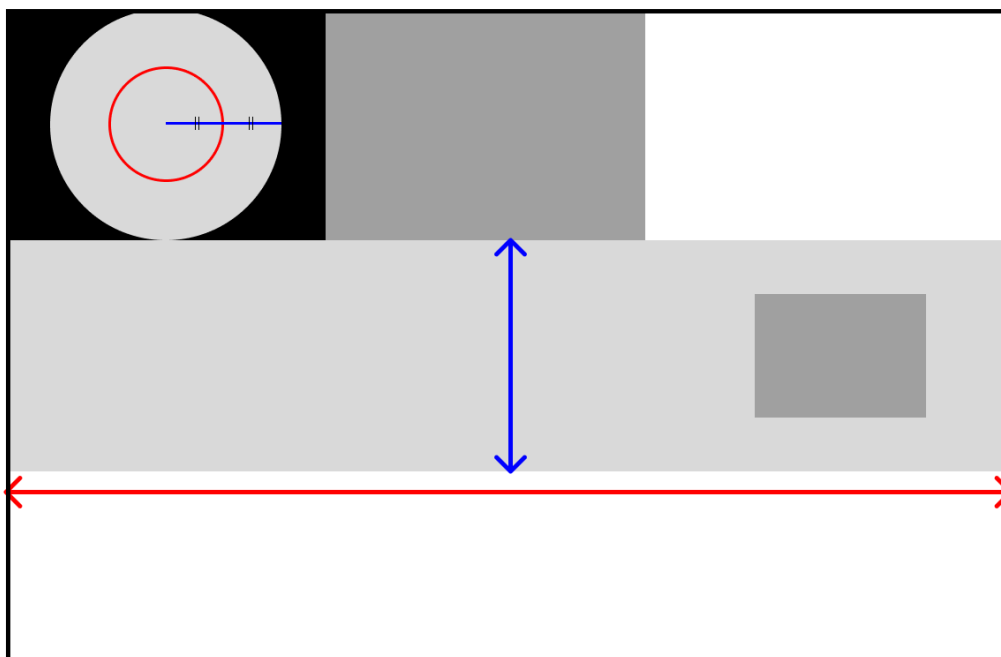


Рисунок 3.7 – Схематичне зображення інтерфейсу користувача

3.3 Програмна реалізація продукту по обробці зображення

Програма запускається з доступу до веб-камери та налаштування елементів відео. Встановлено слухачі подій для рухів миші. Функція `getWebCam` отримує доступ до веб-камери за допомогою `navigator.getUserMedia`.

Відеоелемент встановлюється як вихідний об'єкт, а відеокадри використовуються як текстури в WebGL. Відеокадри зв'язуються як текстура, а текстурований прямокутник малюється на полотні WebGL. Полотно малюється на двовимірному полотні (`canvas2`) із масштабованими розмірами, забезпечуючи інтерактивний досвід на основі рухів миші. Контексти WebGL і Canvas ініціалізуються, шейдери компілюються, а необхідні атрибути та уніформи встановлюються. Визначено функції компіляції шейдера (`createProgram`) і створення текстури (`createTexture`). Вершинні та фрагментні шейдери надаються як рядки (`vertexShaderSource` і `fragmentShaderSource`). За допомогою цих шейдерів створюється та зв'язується програма шейдерів. Функції `rectVertices` і `rectTexCoords`

генерують дані координат вершини та текстури для прямокутника. Ці функції визначають геометрію та відображення відеотекстури на прямокутник. Функція малювання відповідає за очищення полотна WebGL, прив'язку відео текстури та малювання текстурованого прямокутника. Функція drawLoop безперервно викликає малювання за допомогою requestAnimationFrame, створюючи цикл візуалізації. Функція зворотного виклику оновлює змінні mouseX і mouseY на основі рухів миші. Положення елемента div регулюється відповідно до рухів миші. Блок-схема зображена на рисунку 3.8.

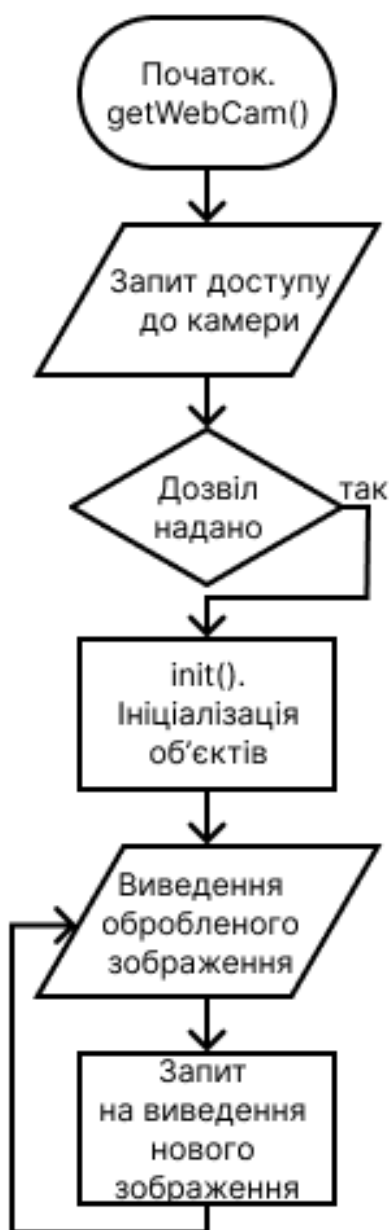


Рисунок 3.8 – Блок-схема програмного продукту

Оглянемо класи, які застосовуються, та об'єкти, що присутні в кодї програмного продукту.

3.3.1 Клас Model

Клас представляє 3D-модель у графічному контексті WebGL. Він обробляє дані координат вершини, нормалі та текстури, надаючи методи для ініціалізації буфера та візуалізації. Він містить базову функціональність для малювання моделі у вигляді суцільних трикутників і каркаса за допомогою лінійних смуг. Конструктор ініціалізує буфери для даних координат вершин, нормалей і текстури. Це просто та ефективно. Клас надає методи для буферизації даних вершин, нормалей і координат текстури. Він також містить окремі методи для малювання моделі як суцільних трикутників (`draw()`), так і каркасів (`drawStroke()`). Клас обробляє створення буфера, зв'язування та завантаження даних належним чином. Буфери створюються один раз під час ініціалізації та оновлюються, коли надаються нові дані. Дані буфера завантажуються ефективно за допомогою `gl.STREAM_DRAW` для динамічних даних. Це підходить для випадків, коли дані часто змінюються, як у моєму випадку, коли потрібно декілька десятків разів на секунду оновлювати зображення з камери, і передавати його як текстуру для обробки.

3.3.2 Клас ShaderProgram

Клас є базовим представленням шейдерної програми в контексті візуалізації графіки, з використанням WebGL. Він інкапсулює пов'язані з шейдером атрибути та забезпечує метод використання програми шейдера. Конструктор класу ініціалізує різні атрибути, пов'язані з властивостями шейдерної програми. Клас має властивості екземпляра для зберігання атрибутів і однорідних розташувань. Назви є описовими, що дозволяє легко зрозуміти їх призначення. Клас містить єдиний метод, `use()`, який має чітке призначення та дотримується принципу єдиної відповідальності. Клас інкапсулює програму шейдера та пов'язані з нею атрибути, сприяючи інкапсуляції та модульності. Клас можна легко повторно

використовувати в різних частинах програми завдяки його інкапсульованому дизайну. Клас припускає існування об'єкта `gl`, що вказує на залежність від зовнішнього контексту. Важливо переконатися, що об'єкт `gl` належним чином ініціалізований і доступний там, де використовується клас `ShaderProgram`.

В коді присутні нижче перелічені об'єкти.

1. `Gl` – об'єкт класу `WebGLRenderingContext`, що надає доступ до інтерфейсу OpenGL ES 2.0, який в свою чергу дозволяє відображати 2D та 3D графіку в HTML елементі `canvas`. Використовується для обробки початкового зображення та виведення обробленого зображення.

2. `ShProg` – об'єкт класу `shaderProgram`, описаного вище.

3. `Ctx` – об'єкт класу `CanvasRenderingContext2D`, що є частиною програмного інтерфейсу `Canvas`, який в свою чергу дозволяє відображати 2D елементи такі як геометричні фігури, текст та зображення. Використовується для виведення області наближення обробленого зображення.

4. `VideoSurface` – об'єкт класу `Model`, описаного вище.

3.4 Виготовлення прототипу катадіоптричної системи

Для ефективного тестування та подальшого використання програмного продукту необхідно було зробити макет катадіоптричної системи, яка б розкрила потенціал розробленої програми. Для розробки макету потрібно три речі: камера, сферичне дзеркало та корпус.

Для катадіоптричної системи обрано бюджетну веб-камеру, що може бути під'єднана до комп'ютера за допомогою USB-кабеля, а саме камера Defender c-090. Також варто зауважити, що живиться камера через USB-кабель, а для підключення камери не потрібна попередня установка драйвера. Ця камера має наступні характеристики:

- кут огляду 54°;
- баланс білого автоматичний;

- фокусування ручне;
- роздільна здатність зображення 640x480 із застосуванням інтерполяції до 3200x2400 пікселів;
- маса 80 грам;
- розміри 65x80x40 мм.

Як можна побачити з вищенаведених характеристик [50] обрана камера має ручне фокусування. Цей функціонал є зручним при використанні камери в катадіоптричній системі. На рисунку 3.9 можна побачити зовнішній вигляд камери Defender c-090 [50].



Рисунок 3.9 – Камера Defender c-090

Корпус був розроблений таким чином, щоб задовольнити наступні вимоги:

- фіксація дзеркала;
- фіксація камери;
- відносна нерухомість камери, щодо дзеркала;
- розташування дзеркала під камерою;
- мінімізація площі, що блокує відображення середовища.

Для створення макета катадіоптричної системи було спроектовано корпус (див. рисунок 3.10). На рисунку можна побачити вид з боків зібраного макета. Було взято сферичне дзеркало радіусом 4 см. Для того, щоб отримати повне зображення сфери з мінімальною кількістю зайвої інформації камеру було розташовано на відстані 23 см від сфери.

Нижня планка у формі диска радіусом 11 см, що знаходиться на одному рівні з отвором для USB дроту тримає дзеркальну сферу і не дає торкатися сфері нижньої

поверхні, це стало можливим завдяки внутрішньому отвору радіусом 3 см. Таким чином сферичне дзеркало захищено від впливу поверхні на якій розташовується катадіоптрична система.

Верхня планка також у формі диска радіусом 8 см, що знаходиться під камерою і тримає її за рахунок внутрішнього отвору радіусом 2 см, а також планки у формі ромба, що тримається на бокових пластинах з отвором 2 см для кабеля і який остаточно фіксує камеру.

Схематичне зображення складових можна побачити на рисунку 3.10.

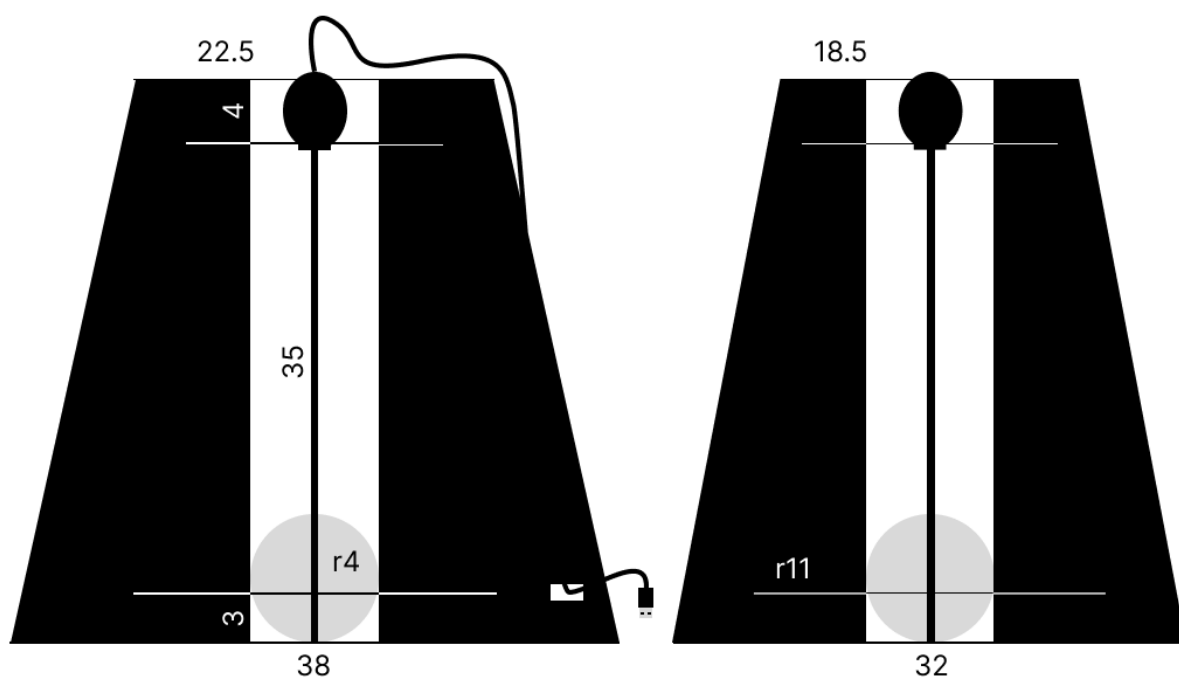


Рисунок 3.10 – Макет. Вид з боків. Схематичне зображення

3.5 Тестування катадіоптричної системи

Для початку тестування було налаштовано фокусні відстань, щоб камера бачила відображення на сфері чітко, без розмиття. Також було виставлено і зафіксовано камеру таким чином, щоб сферичне дзеркало знаходилося точно по центру кадра. На рисунку 3.11 показано як виглядав макет катадіоптричної системи після проведених налаштувань.



Рисунок 3.11 – Макет катадіоптричної системи

На рисунку 3.12 наведено знімок екрана під час виконання програми. В нижній частині знімку можна побачити власне оброблене зображення. Одразу можна помітити, що вертикальні лінії, які є вертикальними і паралельні в реальному житті являються такими і на обробленому зображенні.

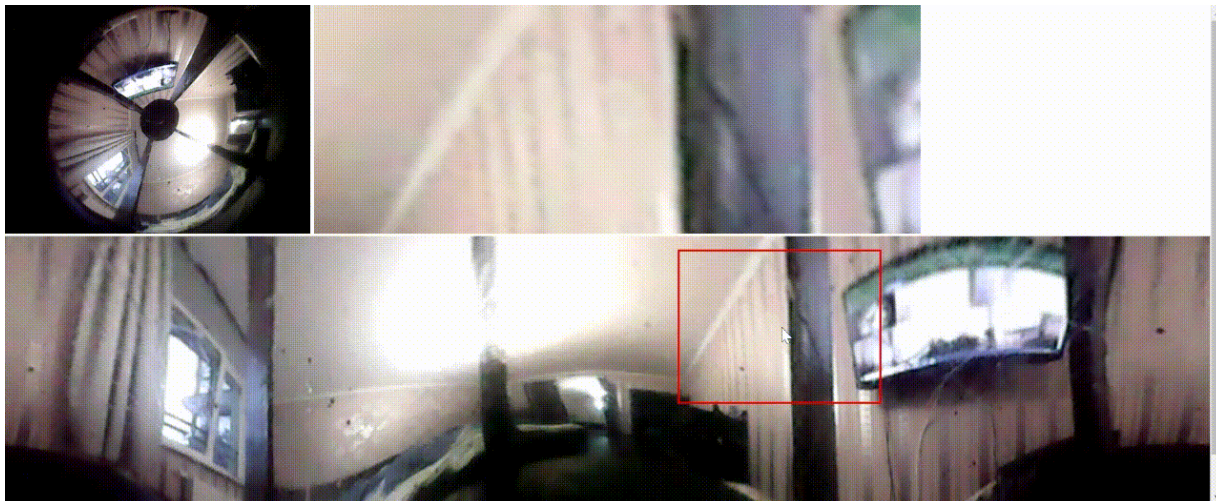


Рисунок 3.12 – Знімок екрана під час виконання програми

Для тестування було зроблено декілька шахових візерунків різних масштабів, що б допомогли оцінити спотворення. Приклад основного візерунку, що була використана для оцінки спотворень показано на рисунку 3.13, сторона квадрата 10 сантиметрів.

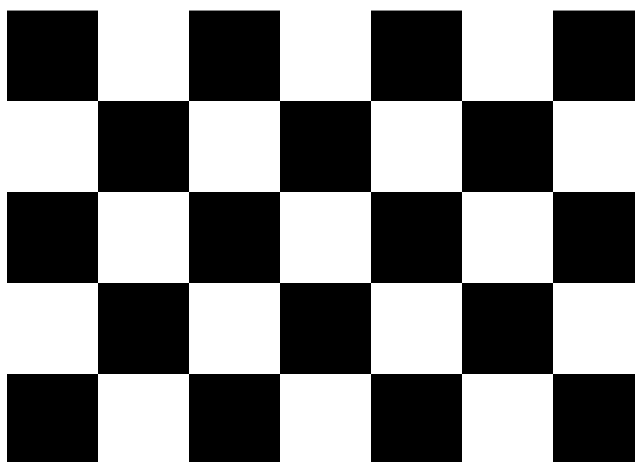


Рисунок 3.13 – Шаховий візерунок для оцінки спотворень

У випадку з шаховими візерунками менших розмірів оцінити спотворення важко, адже квадрати виглядають однаково. Єдиний висновок, що можна зробити – це те, що прямокутна решітка не зберігає форму прямокутника і дещо спотворена. Тест з решіткою менших розмірів можна побачити на рисунку 3.14.

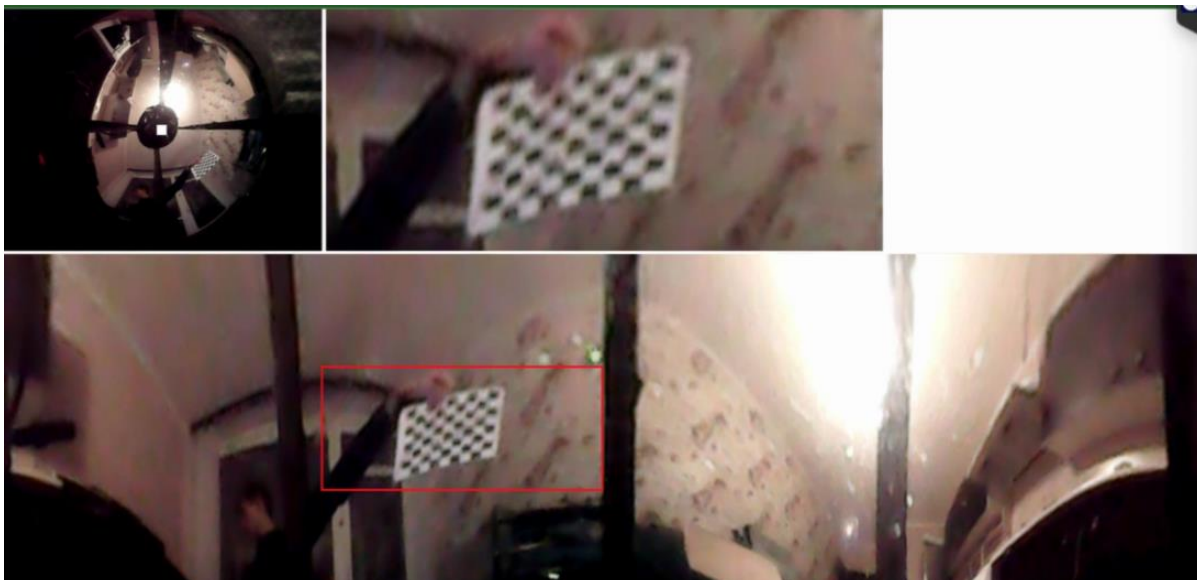


Рисунок 3.14 – Тест з решіткою менших розмірів

Було протестовано кожну умовну чверть зображення. Зображення спотворюється однаково незалежно від того в якій області зображення об'єкт знаходиться. В ході тестування було пропрацьовано два випадки: коли решітку розташовано вертикально, та з напрямом нормалі площини решітки до дзеркала. На рисунках 3.15, 3.16 можна побачити відповідні випадки.

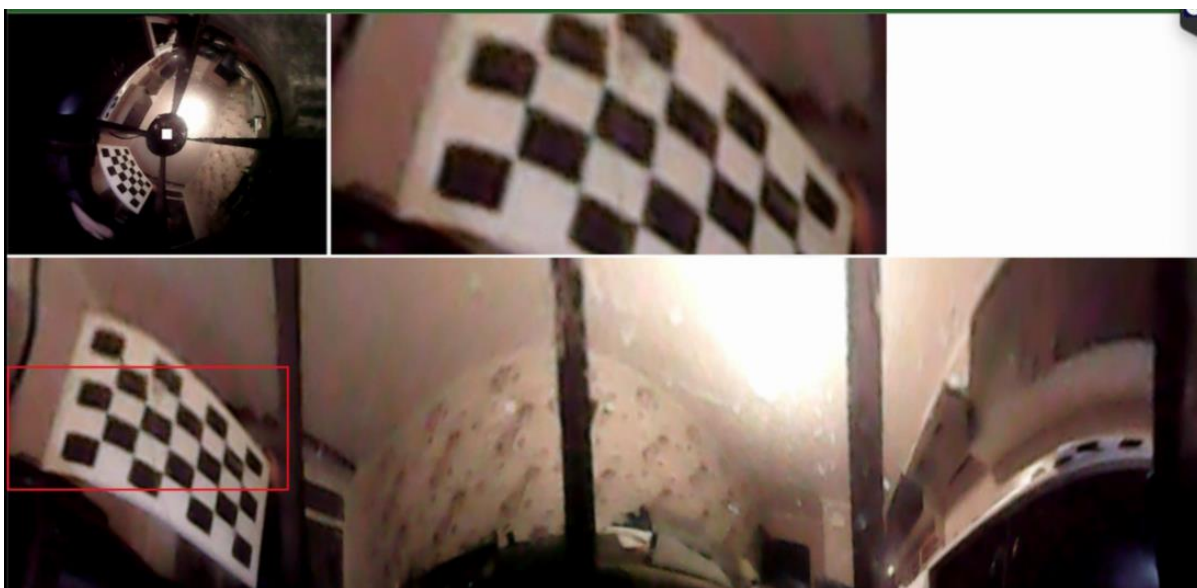


Рисунок 3.15 – Решітку розташовано умовно вертикально

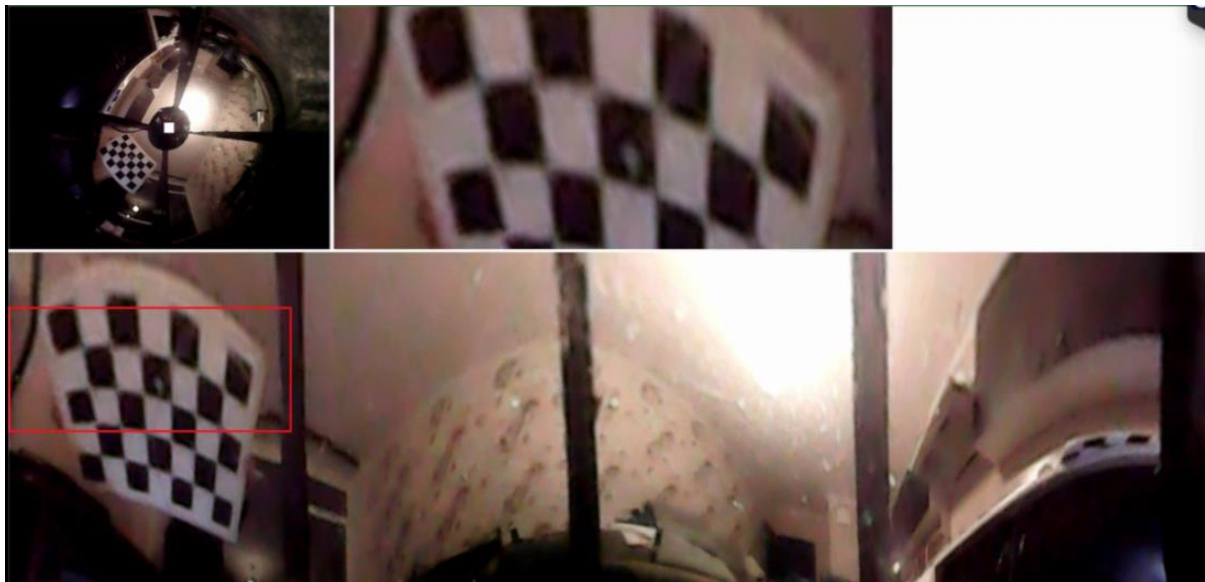


Рисунок 3.16 – Вигляд з напрямом нормалі площини решітки умовно до дзеркала

З рисунку 3.15 можна побачити, що вертикальні лінії паралельні між собою, вони не паралельні іншим вертикальним лініям на зображенні, тому що тестувальний візерунок розташовано з незначним відхиленням. Горизонтальні лінії тепер криві, але при цьому виглядають паралельно.

З рисунку 3.16 можна зробити інші спостереження: вертикальна лінії не паралельні і розходяться в різному напрямку, при цьому якщо продовжити їх в напрямку горизонту, то видно, що лінії мають точку сходу. Таке спостереження пояснюється правилом перспективи: об'єкти ближче здаються більшими ніж об'єкти на задньому плані. Горизонтальні ж лінії криві та не паралельні одна одній. Найближча крива має найбільшу кривизну, а найдальша здається прямою.

4 ОГЛЯД ІНТЕРФЕЙСУ КОРИСТУВАЧА. СИСТЕМНІ ВИМОГИ

Для того, щоб користуватися програмним продуктом, в ідеальному випадку, необхідно мати розроблену в дипломній роботі катадіоптричну систему. Встановити її на горизонтальну поверхню, наприклад стіл, або закріпити систему до стелі або подібної поверхні. Підключити систему через USB-A дріт до комп'ютера або ноутбука. Камера, що використовується катадіоптричною системою не потребує драйвера, тому інших підготовчих кроків виконувати не потребується. Можна переходити до безпосереднього огляду інтерфейсу користувача.

4.1 Огляд інтерфейсу користувача застосунку

Після відкриття розробленого веб-додатку потрібно схвалити запит дозволу на доступ поточної вкладки браузера до зображення з камери. На рисунку 4.1 зображено приклад вигляду запиту.

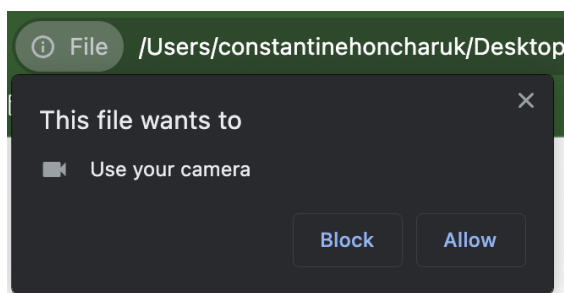


Рисунок 4.1 – Запит дозволу на використання додатком камери

Після цього на екрані з'явиться зображення з камери в реальному часі. Перш за все користувачу необхідно впевнитися, що камера направлена у напрямку центру сфери. І хоча конструктор передбачає фіксацію камери у напрямку сфери іноді може трапитися так що камера буде направлена у напрямку точки на поверхні сфери, але не на її центр. Для того, щоб це зробити необхідно сумістити лінзу камери, що відображається у сферичному дзеркалі, з місцем, що позначено

невеликим білим квадратом на необробленому зображенні, що можна побачити у верхній лівій частині вікна повертаючи камеру. Після налаштування правильного положення камери можна користуватися програмою за призначенням.

Для зрозумілості вербального огляду інтерфейсу користувача погляньмо на рисунок 4.2.

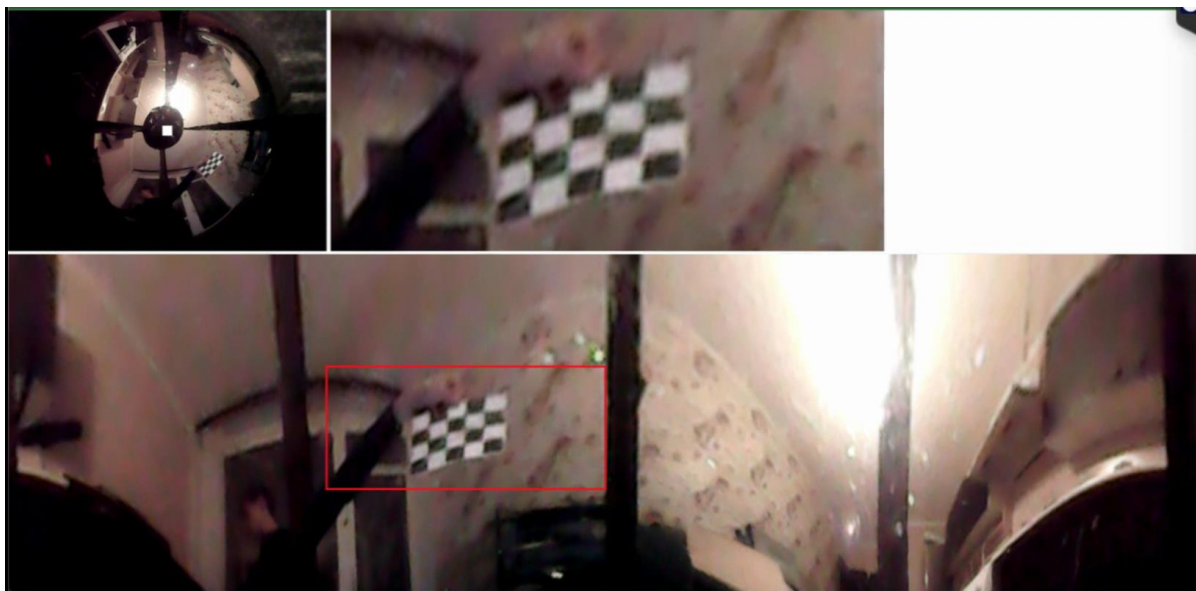


Рисунок 4.2 – Інтерфейс користувача програмного продукту

Інтерфейс візуально зрозуміло розподілено на дві частини верхню і нижню. Нижня частину інтерфейсу містить оброблене програмою зображення. Верхня містить(зліва на право) початкове зображення з камери та наближення обраної області обробленого зображення.

Користувач може обрати область наближення переміщуючи курсор миші по обробленому зображенню. Геометричний центр прямокутника відповідає положенню миші, коли прямокутник повністю знаходиться над обробленим зображенням, тобто відповідні сторони прямокутника не виходять за межі відповідних сторін обробленого зображення. Курсор миші переміщується по екрану без обмежень. Коли курсор миші виходить за межі обробленого зображення прямокутник залишається на місці, де востаннє фіксувався курсор миші над зоною детекції.

По центру початкового зображення можна побачити білий прямокутник. Цей прямокутник не є частиною зображення з камери, натомість це частина інтерфейсу

користувача, що у разі потреби може допомогти користувачу правильно виставити камеру для коректного перетворення початкового зображення на оброблене. Білий прямокутник визначає геометричний центр зображення де програма очікує побачити частину катадіоптричної системи у вигляді сферичного дзеркала. Центр сфери повинен знаходитися в місці розташування білого прямокутника, іншими словами камера повинна бути віддзеркалена якраз в геометричному центрі зображення, таким чином, щоб білий прямокутник блокував її відображення, не даючи змогу користувачу побачити її.

4.2 Мінімальні системні вимоги програми

Мінімальні системні потреби розробленої програми здебільшого визначені системними вимогами браузера, при цьому обов'язкова підтримка WebGL девайсом, на якому запущено програмний продукт.

Нижче, в таблиці 4.1 можна більш детально ознайомитися з системними вимогами в залежності від операційної системи, що використовується.

Таблиця 4.1 – Мінімальні системні вимоги відповідно до ОС, що використовується

Операційна система	Мінімальні системні вимоги
Windows	Windows 10+ або Windows Server 2016+. Процесори що містять набір інструкцій SSE3
macOS	macOS Catalina 10.15 чи пізніші версії
Linux	64-bit Ubuntu 18.04+, Debian 10+, openSUSE 15.2+, Fedora Linux 32+.
Android	Android 7.0 Nougat чи пізніші версії
iOS	iOS 15 чи пізніші версії

Більшість сучасних браузерів підтримують WebGL. Це включає Google Chrome, Mozilla Firefox, Apple Safari та Microsoft Edge.

Переконайтеся, що ви використовуєте останню версію веб-переглядача, щоб скористатися перевагами останніх функцій і вдосконалень WebGL.

Для візуалізації WebGL покладається на графічне обладнання пристрою. Як наслідок, можливості WebGL залежать від можливостей відеокарти.

Хоча WebGL може працювати на різноманітному обладнанні, потужніші графічні карти забезпечать кращий досвід роботи зі складною 3D-графікою.

WebGL підтримується кількома операційними системами, зокрема Windows, macOS, Linux і деякими мобільними операційними системами.

WebGL може бути вимкнено в деяких браузерах через проблеми з безпекою чи продуктивністю. Переконайтеся, що WebGL увімкнено в налаштуваннях вашого браузера.

Вимоги до запуску додатків WebGL залежать від комбінації факторів, пов'язаних із технологією, продуктивністю, безпекою та запланованою взаємодією з користувачем. Давайте дослідимо деякі причини цих вимог.

Залежність від апаратного забезпечення графіки. WebGL використовує графічний процесор (GPU) пристрою для відтворення 3D-графіки. Можливості WebGL безпосередньо пов'язані з можливостями GPU. Потужніші графічні процесори можуть працювати зі складнішою та деталізованою графікою, забезпечуючи кращий досвід користувача.

Візуалізація 3D-графіки може потребувати інтенсивних обчислень. WebGL розроблено, щоб використовувати переваги апаратного прискорення, яке забезпечують сучасні графічні процесори, що дозволяє плавно та ефективно відтворювати складні сцени. Це підвищення продуктивності особливо важливо для таких програм, як ігри та симуляції.

WebGL працює в середовищі веб-браузера, і безпека є головним пріоритетом. Вживаються заходи безпеки, щоб запобігти потенційним експлойтам або атакам через WebGL. Браузери можуть вимикати певні функції або накладати обмеження, щоб забезпечити безпеку даних користувачів і цілісність веб-перегляду.

WebGL реалізовано як JavaScript API у веб-браузерах. Різні браузери можуть мати різні варіанти реалізації та підтримки WebGL. Постачальники веб-переглядачів постійно оновлюють і вдосконалюють свої реалізації WebGL, а використання сучасного оновленого браузера забезпечує сумісність із найновішими функціями й оптимізацією.

Підсумовуючи, вимоги до додатків WebGL формуються сукупністю факторів, щоб забезпечити баланс між продуктивністю, безпекою, сумісністю та здатністю забезпечити переконливу взаємодію з користувачем у межах обмежень середовища веб-браузера.

Веб-сайт get.webgl.org надає інформацію, щодо підтримки браузером WebGL, і перегляду основної інформації що надає WebGL.

5 РОЗРОБКА СТАРТАП-ПРОЄКТУ

5.1 Опис ідеї проєкту

В таблиці 1 проаналізовано:

- зміст ідеї;
- можливі напрямки застосування;
- основні вигоди, що може отримати користувач товару (за кожним напрямком застосування).

Аналіз даних пунктів дає можливість визначитись з базовими потенційними ринками, в межах яких потрібно шукати групи потенційних клієнтів

Таблиця 5.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Створення програмно-апаратного комплексу: катадіоптрична оптична система, яку можна застосовувати в системах відео-спостереження, безпілотних системах та онлайн-конференціях	Відеоспостереження	Катадіоптрична камера з надвеликим кутом огляду
	Безпілотні системи	Катадіоптрична система за більш привабливою ціновою пропозицією
	Відео-конференції	Вирішення проблеми обмеженості часу роботи існуючих камер з надвеликим кутом зору.

Після визначення потенційного ринку, основним завданням є аналіз потенційних техніко-економічних переваг ідеї порівняно із пропозиціями конкурентів.

В таблиці 5.2 показано чим дана ідея відрізняється від існуючих аналогів та замінників.

Таблиця 5.2. Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				W	N	S
		Мій проект	Insta360	GoPro MAX	Kodak PIXPRO			
1	Кут огляду камери більше 180 градусів	€	€	€	€		+	
2	Режими зйомки	Відсутнє	€	€	€	+		
3	Програмне забезпечення для обробки зображення	€	€	€	€		+	
4	Наявність місця під акумулятор	Відсутнє	€	€	€			+
5	Інтеграція з комп'ютером в режимі реального часу	€	Відсутнє	Відсутнє	Відсутнє			+
6	Вартість	1000 грн	25999 грн	22999 грн	18000 грн		+	+

В таблиці 2 літери W, N, S розшифровуються як:

W – слабка сторона;

N – нейтральна сторона;

S – сильна сторона.

5.2 Технологічний аудит стартапу

В межах даного підрозділу буде проведено аудит технології, за допомогою якої можна реалізувати ідею проекту. Для визначення технологічної здійсненності ідеї проекту необхідно дати відповідь на наступні питання:

- за якою технологією буде виготовлено товар згідно ідеї проекту?
- чи існують такі технології, чи їх потрібно розробити/доробити?
- чи доступні такі технології авторам проекту?

Результати проведеної роботи занесені в таблиці 5.3.

Таблиця 5.3. Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Розробка програмно-апаратного комплексу, для	Камера кругового огляду	Наявна	Доступна
2	сферично-панорамної зйомки	Катадіоптрична камера	Наявна	Доступна
Обрана технологія реалізації ідеї проекту: катадіоптрична камера, оскільки дана технологія більш дешева при запровадженні, а також в такій системі легко можна інтегрувати інші апаратні рішення: змінювати камеру тощо.				

5.3 Аналіз ринкових можливостей запуску стартап-проекту

Наступним кроком буде визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту. Даний пункт дозволить спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Проведемо аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (табл. 5.4).

Таблиця 5.4. Попередня характеристика потенційного ринку стартап-проекту

№	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	На східно-європейському ринку немає пропозицій стаціонарних камер.
2	Загальний обсяг продаж, грн/ум. од	1000 грн/ ум. од
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Технічні виклики залежні від галузі використання
5	Специфічні вимоги до стандартизації та сертифікації	Технічні вимоги за ДСТУ, конструкторська документація, експлуатаційна документація, регуляторні провадження

Потенціал цього стартапу полягає в його унікальній пропозиції катадіоптричних камер і програмного забезпечення, призначеного для інтеграції в автономні системи та системи спостереження.

Використання катадіоптричної технології виділяє стартап на ринку. Цей інноваційний підхід забезпечує переваги з точки зору розміру, ваги та поля зору, потенційно пропонуючи унікальну торгову пропозицію порівняно з традиційними системами камер.

Стартап має намір вийти на ринки, що розвиваються, зокрема в індустрії автономних транспортних засобів і секторі відеоспостереження. Оскільки ці ринки продовжують зростати, існує значний потенціал для зростання попиту на передові системи камер і програмне забезпечення.

Універсальність катадіоптричних камер дозволяє застосовувати їх у різних галузях, крім автономних транспортних засобів і спостереження. Стартап може досліджувати партнерства та співпрацю, щоб адаптувати свою технологію до різних секторів, розширюючи охоплення ринку.

Пропонуючи послуги з налаштування на основі конкретних галузевих потреб, стартап може обслуговувати різноманітні сегменти ринку. Ця гнучкість може відкрити нові джерела доходу та надати рішення для нішевих програм.

Співпраця з визнаними гравцями у суміжних галузях може прискорити вихід на ринок і підвищити довіру. Партнерство з ключовими гравцями в автомобільній промисловості, технологічному секторі або секторі відеоспостереження може надати доступ до ресурсів, каналів розподілу та ширшої клієнтської бази.

Відданість постійним дослідженням і розробкам гарантує, що стартап залишається в авангарді технологічних досягнень. Цей проактивний підхід має вирішальне значення для підтримки конкурентної переваги та адаптації до мінливих вимог ринку.

Однак реалізація цього потенціалу вимагатиме прийняття стратегічних рішень, ефективних зусиль з маркетингу та продажів, постійних інновацій і проактивного підходу до вирішення проблем. Використовуючи свої сильні сторони, усуваючи слабкі сторони та орієнтуючись на зовнішні фактори, стартап має можливість утвердитися як ключовий гравець у розвитку автономних технологій і технологій спостереження.

Надалі визначимо потенційні групи клієнтів, їх характеристики, та формується орієнтовний перелік вимог до товару для кожної групи (табл. 5). У таблиці 5.5 сформована характеристика потенційних клієнтів стартап-проекту: потреба, що формує ринок, цільова аудиторія(цільові сегменти ринку) та вимоги споживачів до товару.

Таблиця 5.5. Характеристика потенційних клієнтів стартап-проекту

№	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Вимоги споживачів до товару
1	Системи відеоспостереження з великим кутом зору	Виробники систем відеоспостереження	Запит на безперебійну роботу системи
2	Фіксація візуальної інформації безпілотними системами	Виробники безпілотних систем	Збільшення кута огляду без потреби встановлення додаткового обладнання
3	Проведення відео-конференцій між офісами	Підприємства, що працюють в офісах	Камера з великим кутом огляду, що інтегрується з комп'ютером

Наступним кроком проведемо аналіз ринкового середовища: складемо таблицю факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (табл. 5.6, 5.7).

Таблиця 5.6. Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Технічні виклики в галузі систем відеоспостережень	Подолати ряд технічних проблем, щоб забезпечити безперебійну роботу в умовах, що залежать від середовища в якому встановлено систему відеоспостереження	Розробка живлення від акумулятора та запису відео на випадок втрати зв'язку із комп'ютером. Розробка конструктивних рішень, що дозволяють експлуатацію системи в незалежності від місця монтажу камери

Продовження таблиці 5.6

2	Технічні виклики у безпілотних системах	Стартап повинен буде переконатися, катадіоптричні камери, які він розробляє інтегруватимуться з безпілотними системами та системами відеоспостереження	Розробка кросплатформеного програмного забезпечення, який легко інтегрувати у системах, що функціонують на різних операційних системах
---	---	--	--

Таблиця 5.7. Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Фінансовий	Розробка з нуля системи фіксації візуальної інформації для специфічної системи відеоспостереження або безпілотних систем потребує значних фінансових вливань	Зосередження стартапу на розробці та виробництві універсальних та недорогих катадіоптричних систем, які з незначними конструктивними та\або технічними допрацюваннями можна використовувати в системах відеоспостереження та безпілотних системах, робить його економічно ефективним рішенням

Продовження таблиці 5.7

2	Програмне забезпечення	Існує мала кількість ПЗ, яке б забезпечувало обробку зображення отриманого з катадіоптричної системи.	Стартап може запропонувати програмне забезпечення для обробки зображення отриманого з камери, таким чином вирішуючи питання першочергової обробки для отримання зображення у прямокутному форматі, що зручне для подальшої обробки, що специфічна для системи, в якій використовується катадіоптрична камера.
3	Конструктивні та технічні труднощі	Розробка з нуля системи фіксації візуальної інформації може викликати технічні труднощі	Стартап може запропонувати консультування з конструктивних та технічних питань, що стосуються катадіоптричних камер

Визначимо загальні риси конкуренції на ринку (табл. 5.8).

Таблиця 5.8. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Тип конкуренції – Олігополія	Даний товар з'явився відносно недавно на ринку, тому його виробляє лише декілька компаній, які конкурують між собою	Одним із варіантів дій компанії є реклама із усіх можливих джерел інформації, яка буде містити прилипливий слоган, також приваблювання покупців різними акціями та подарунками.

Продовження таблиці 5.8

2. За рівнем конкурентної боротьби – міжнародний	При аналізі ринку не було виявлено місцевих виробників	Особливість даного товару в тому, що він є недорогим та простим в запровадженні рішенням.
3. За галузевою ознакою – внутрішньогалузева	Дана продукція є вузьконапрямленою і використовується тільки з однією ціллю	Дослідження потенційно дадуть змогу отримати найпривабливіший товар на ринку, а саме зменшення габаритів камери та збільшення роздільної здатності зображення
4. Конкуренція за видами товарів: товарно-видова	Товар має на меті виконання однієї задачі, але існує декілька варіантів його використання.	За мету компанії взяти створення товару, що виділяється із представлених на ринку за показником ціна/якість, а також тим, що пропонує стаціонарне рішення, серед існуючих мобільних рішень.
5. За характером конкурентних переваг – цінова	Однією з основних переваг майбутнього проекту при аналізі ринку була його мала ціна	Максимально розкрити дану перевагу при рекламі та продажу, при цьому зауважуючи, що якість поступається конкурентам
6. За інтенсивністю – марочна	Кожна компанія рекламує його під власною маркою	Розвиток та популяризація власної марки шляхом реклами, якісного товару та праці

Після аналізу конкуренції проводиться більш детальний аналіз умов конкуренції в галузі (табл. 5.9).

Таблиця 5.9. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товаризації-ники
	Прямі конкуренти в галузі відсутні	Бар'єром входження в ринок є мала обізнаність потенційного покупця про компанію-виробника	Товар збирається безпосередньо в компанії, камера та дзеркало купуються, або замовляються окремо	Підприємства, що займаються впровадженням систем відеоспостереження або виробники безпілотних систем	Пропозиція недоступна для клієнтів, що не є бізнесами
Висновки	Можна зробити висновок, що на даний момент інтенсивність конкуренції незначна		Постачальники не диктують умови роботи на ринку. І хоча їх кількість невелика	У клієнта обмежений вибір пропозицій, якщо за якихось причин товар, ціна або сервіс його не влаштує.	Обмежень не має

Отже, стартап не має зіткнутися з конкуренцією з боку різних компаній, які вже пропонують камери з кутом огляду більше 180 градусів. Однак унікальна

зосередженість стартапу на стаціонарних камерах для систем відеоспостереження, безпілотних систем та при цьому його недороге та просте у використанні рішення, а також його консультаційні послуги можуть допомогти йому закріпитися на ринку. Даний висновок отриманий з особливостей самого товару.

На основі аналізу конкуренції, проведеного в таблиці 5.9, а також із урахуванням характеристик ідеї проекту, вимог споживачів до товару та факторів маркетингового середовища визначимо та обґрунтуємо перелік факторів конкурентоспроможності (табл. 5.10).

Таблиця 5.10. Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування
1	Ціна	Даний продукт за функціональними характеристиками поступається конкурентам у мінімізації викривлення зображення та роздільній здатності обробленого зображення, але за рахунок використання технологій відмінних від конкурентів можна досягти низької ціни. Це може бути особливо привабливим для підприємств і фізичних осіб, які бажають використувати дешеву камеру з кутом зору більше 180 градусів.
2	Пристосованість до конкретних потреб	Пропонуючи консультаційні послуги, щоб допомогти підприємствам планувати та впроваджувати власні рішення для фіксування візуальної інформації стартап може пристосувати своє рішення до конкретних потреб кожного підприємства. Це може допомогти забезпечити сталість мережі в довгостроковій перспективі.

Продовження таблиці 5.10

3	Відносна легкість впровадження катадіопричної камери	Недороге та просте в розгортанні рішення стартапу може бути особливо привабливим для підприємств, які шукають універсальне рішення для систем відео спостереження та безпілотних систем, для яких важлива наявність зображення, що охоплює значну область навколишнього середовища для обробки візуальної інформації.
---	--	---

За визначеними факторами конкурентоспроможності виконаємо аналіз сильних та слабких сторін стартап-проекту.

Таблиця 5.11. Порівняльний аналіз сильних та слабких сторін проекту «Катадіопрична камера та програмний продукт для обробки зображень»

№	Фактор конкуренто-спроможності	Бали 1-20	Рейтинг у порівнянні з камерами з кутом зору більше 180 градусів						
			-3	-2	-1	0	1	2	3
1	Ціна	19							+
2	Пристосованість	15				+			
3	Легкість впровадження	18						+	

Перейдемо до фінального етапу ринкового аналізу можливостей впровадження проекту. Складемо матрицю аналізу сильних та слабких сторін, загроз та можливостей (табл. 5.12), на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (табл. 5.11). Даний аналіз також ще називають SWOT-матрицею.

Таблиця 5.12. SWOT– аналіз стартап-проекту

Сильні сторони: Використання катадіоптричних камер, забезпечує перевагу з точки зору кута огляду, що складає 360 градусів. Розробка програмного забезпечення, дозволяє інтегрувати розширений функціонал в системи камер, як наприклад, фокусування на рухомому об'єкті тощо. Катадіоптричні камери можуть бути універсальними та застосовуватися в системах відеоспостереження та безпілотних системах – ринку, що динамічно розвивається. Здатність інтегруватися в автономні системи та системи спостереження робить стартап ключовим гравцем на зростаючому ринку розумних і автоматизованих технологій.	Слабкі сторони: Залежність від технологічного прогресу: швидкий технологічний прогрес може призвести до того, що поточні пропозиції швидко застаріють, що потребуватиме постійних інвестицій у дослідження та розробки. Обмежена впізнаваність бренду: оскільки це стартап, бренд може бути недостатньо впізнаваним порівняно з усталеними конкурентами, що ускладнює завоювання довіри на ринку.
---	--

Продовження таблиці 5.12

Можливості: Зростаючий попит на автономні транспортні засоби та системи відкриває значні можливості для інтеграції катадіоптричних камер. Розширення ринку відеоспостереження, викликане потребою в посилених заходах безпеки, створює потенційний шлях для продуктів стартапу. Співпраця з відомими гравцями в технологічній або автомобільній промисловості та виробниками БПЛА може надати можливості для зростання, використовуючи їхні ресурси та присутність на ринку. Пропонування варіантів налаштування для конкретних галузевих потреб може відкрити нові джерела доходу та задовольнити ринкові ніші.	Загрози: Галузь може мати сильну конкуренцію з боку визнаних гравців та інших стартапів, які пропонують аналогічні або альтернативні технології. Суворі правила в автомобільному секторі та секторі спостереження можуть створити проблеми для інтеграції продукту та виходу на ринок. Оскільки камери та програмне забезпечення є невід’ємною частиною автономних систем спостереження, загрози кібербезпеці можуть викликати серйозне занепокоєння.
--	--

Стартап, що спеціалізується на катадіоптричних камерах і інтегрованому програмному забезпеченні для автономних систем і систем спостереження, демонструє багатообіцяючі переваги, такі як інноваційні технології, досвід у розробці програмного забезпечення, універсальність і потенціал інтеграції. Однак серед викликів є потреба в навчанні ринку, високі початкові витрати, залежність від технологічних досягнень і обмежене впізнавання бренду.

Можливості полягають у зростаючій автономній промисловості, зростаючому ринку відеоспостереження, потенційному партнерстві та співпраці та наданні послуг з налаштування. З іншого боку, загрози включають конкурентне середовище, регуляторні проблеми, ризики кібербезпеки та вразливість до економічних спадів.

Щоб процвітати, стартап повинен використовувати свої сильні сторони, щоб отримати вигоду від ринкових можливостей, усунути недоліки та орієнтуватися в потенційних загрозах. Це може включати стратегічні партнерства, цільовий маркетинг для підвищення впізнаваності бренду, безперервні інновації, щоб випереджати технологічні тенденції, а також проактивні заходи для вирішення проблем регулювання та кібербезпеки.

Визначені альтернативи проаналізуємо з точки зору строків та ймовірності отримання ресурсів (табл. 5.13).

Таблиця 5.13. Альтернативи ринкового впровадження стартап-проекту

№	Альтернатива ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Перехід на міжнародний рівень	Ресурси менш ймовірно будуть отримані у міжнародних інвесторів	Від 1 до 3 місяців
2	Зміншення габаритів системи та створення стандартизованих рішень під різні випадки користування катадіоптричною камерою, що збільшить попит серед клієнтів	Ресурси більш ймовірно будуть отримані у міжнародних інвесторів	Від 6 до 12 місяців

Із запропонованих альтернатив обидві не мають проблем з постачанням ресурсів для реалізації проекту, тому було через більшу імовірність отримання ресурсів було обрано другу альтернативу, пов'язану з вдосконаленням катадіоптричної камери. Адже потреба в компактних камерах, що забезпечують кут огляду більше 180 градусів тільки зростає. Здебільшого це відбувається через війну росії проти України, яка показала, ефективність БПЛА та інші безпілотних систем.

5.4 Розроблення ринкової стратегії проекту

Для розроблення ринкової стратегії першим кроком визначимо стратегію охоплення ринку. Для цього зробимо опис цільових груп потенційних споживачів (табл. 5.14).

Таблиця 5.14. Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Громади та організації у віддалених або недостатньо охоплених районах	Доволі висока, особливо для тих хто вже має сонячні панелі	Попит відповідно також є високим	Помірна через невелику кількість компаній	Відносно простий вхід у даний сегмент ринку

Продовження таблиці 5.14

2	Некомерційні організації, які працюють над подоланням цифрового розриву	Середня готовність	Організації можуть бути зацікавлені в консультаційних послугах стартапу та продажу обладнання, технологій.	Низька конкурентність через малий попит сегменту	Легкий вхід у сегмент
3	Уряди та державні установи	Висока готовність	Залежить від планів уряду на виділення фінансування даного проекту (Висока, якщо бюджет виділено)	Середня, через великий потенційний заробіток та продажу товару оптом.	Найскладніший вхід у сегмент порівняно з попередніми через високі стандарти регулювання

За результатами аналізу потенційних груп споживачів в якості цільової групи було обрано одну групу , а саме некомерційні організації, які працюють над подоланням цифрового розриву. Через присутність значних фінансових впливань.

Для роботи в обраних сегментах ринку сформуємо базову стратегію розвитку (табл. 5.15).

Таблиця 5.15. Визначення базової стратегії розвитку

№	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1	Концентрація на потребах одного цільового сегменту, виділення товару серед конкурентів за рахунок його відмінностей	Задоволення потреб вибраного цільового сегменту краще, ніж конкуренти за рахунок відмінних властивостей продукту	Виробники безпілотних систем, підприємства, що займаються впровадженням систем відеоспостереження	Стратегія спеціалізації; стратегія диференціації
2	Надання товару важливих з точки зору споживача відмінних властивостей із концентрацією на одному.	Впровадження відмінного товару на ринок, який містить всі важливі з точки зору споживача властивості	Виробники безпілотних систем, підприємства, що займаються впровадженням систем відеоспостереження	Стратегія диференціації

З огляду на цільові групи потенційних споживачів, то базова стратегія розвитку обрана з оглядом на задоволення потреб вибраного цільового сегменту краще, ніж конкуренти за рахунок відмінних властивостей продукту. Тобто стратегія являє собою деякий симбіоз стратегій спеціалізації та диференціації.

ВИСНОВКИ

1. Проведено огляд існуючих камер з кутом огляду більше 180 градусів. Показано, що на ринку є попит на камери з надвеликим кутом огляду, зокрема для таких застосунків, як системи відеоспостереження, безпілотні система та стаціонарні камери для онлайн-конференцій. Проведено вибір засобів розробки – обрано середовище розробки Visual Studio Code, мову програмування JavaScript та графічну бібліотеку WebGL, що дало змогу реалізувати програмний продукт для обробки зображення з катадіоптричної камери, що використовує переваги паралельних обчислень завдяки потужності графічного процесора комп'ютера.

2. Проведено розробку математичної моделі задачі обробки зображення та її реалізацію на основі мов програмування JavaScript та GLSL, що дало змогу отримати систему, яка обробляє зображення: переводить зображення у вигляді кола у прямокутний формат. Отримане зображення виводиться з роздільною здатністю 640x480 пікселів зі частотою 30 кадрів на секунду або більше. Користувачу надано змогу наближати область зображення, яка його цікавить. Розроблений програмний продукт кросплатформений, доступний на Windows, MacOS, Linux, Android, iOS.

3. Проведено виготовлення прототипу катадіоптричної системи, яка складається з камери Defender c090, сферичного дзеркала та корпусу, що надало можливість спостерігати за навколишнім середовищем і отримувати зображення з роздільною здатністю 640x480 пікселів.

4. Проведено тестування розробленого програмно-апаратного комплексу. Тестування показало, що чим ближче об'єкт знаходиться до верхньої або нижньої границі зображення – тим більше спотворення зображення цього об'єкта. Для того, щоб мінімізувати спотворення зображення, об'єкт спостереження варто розташовувати якомога далі від верхньої та нижньої грані зображення. Тестування на шаховому візерунку показало, що вертикальні лінії не спотворюються і зображені за законами перспективи, а спотворення горизонтальних ліній уникнути не вдалося. Загальні спотворення зображення не заважають розпізнаванню об'єктів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Optical design fundamentals for infrared systems: веб-сайт. URL: <https://web.archive.org/web/20211114231047/https://books.google.com/books> (дата звернення: 30.11.2023)
2. Камера insta360 x3: веб-сайт. URL: https://www.insta360.com/ru/product/insta360-x3#x3_specs (дата звернення: 30.11.2023)
3. Камера GoPro MAX 360: веб-сайт. URL: https://www.bhphotovideo.com/c/product/1503547-REG/gopro_chdhz_201_max_360_action_camera.html/specs (дата звернення: 30.11.2023)
4. Камера Kodak PIXPRO Orbit 460 4K: веб-сайт. URL: <https://kodakpixpro.com/cameras/360-vr/orbit360-4k/> (дата звернення: 30.11.2023)
5. Які технології використовують 360-градусні камери: веб-сайт. URL: <https://flixabit.net/how-do-360-cameras-work-capturing-world-in-every-direction/> (дата звернення: 30.11.2023)
6. Fan K.-W., Liu S., Sinha P. Scalable data aggregation for dynamic events in sensor networks, in: Proceedings of the Fourth International Conference on Embedded Networked Sensor Systems, 2006.
7. Krishnamachari B., Estrin D., Wicker S. The impact of data aggregation in wireless sensor networks, in: Proceedings of the International Workshop Distributed Event-Based Systems, 2002.
8. Bandyopadhyay S., Coyle E. An energy efficient hierarchical clustering algorithm for wireless sensor networks, in: Proceedings IEEE INFOCOM, 2003.
9. Aghajan H., Cavallaro A. Multi-Camera Networks Principles and Applications, 2019. 242-243 p.
10. Heinzelman W.B., Chandrakasan A.P., Balakrishnan H., An application-specific protocol architecture for wireless microsensor networks, IEEE Transactions on Wireless Communications, 2000. 660-670 p.

11. Younis O., Fahmy S. HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks, IEEE Transactions on Mobile Computing, 2004. 366-379 p.

12. Gupta I., Riordan D., Sampalli S., Cluster-head election using fuzzy logic for wireless sensor networks, in: Proceedings of the Third Annual Communication Networks and Services Research Conference, 2005.

13. Mhatre V., Rosenberg C., Kofman D., Mazumdar R., Shroff N. A minimum cost heterogeneous sensor network with a lifetime constraint, IEEE Transactions on Mobile Computing, 2022. 99 p.

14. Камера Defender c-090: веб-сайт. URL: <https://ua.defender-global.com/catalog/webcams/c-090-0-3mp-black> (дата звернення: 30.11.2023)

15. Java programming language: веб-сайт. URL: <https://www.ibm.com/topics/java> (дата звернення: 30.11.2023)

16. Introduction to processing java: веб-сайт. URL: <https://www.geeksforgeeks.org/introduction-to-processing-java/> (дата звернення: 30.11.2023)

17. Програмне середовище Processing: веб-сайт. URL: <https://processing.org/> (дата звернення: 30.11.2023)

18. ECMAScript 2023 language specification. Чинний від 2023-06-14. Вид. офіц. Geneva, 2023. 840 с. URL: https://ecma-international.org/wp-content/uploads/ECMA-262_14th_edition_june_2023.pdf (дата звернення: 11.12.2023).

19. JavaScript ES2020 features with simple examples: веб-сайт. URL: <https://betterprogramming.pub/javascript-es2020-features-with-simple-examples-d301dbef2c37> (дата звернення: 30.11.2023)

20. 15 найпопулярніших мов програмування за версією GitHub: веб-сайт. URL: <https://habr.com/ru/articles/310262/> (дата звернення: 30.11.2023)

21. JavaScript: веб-сайт. URL: <http://www.crockford.com/javascript/javascript.html> (дата звернення: 30.11.2023)
(дата звернення: 30.11.2023)

22. Основы JavaScript. MDN Web Docs: веб-сайт. URL: <https://developer.mozilla.org/> (дата звернення: 30.11.2023)

23. OpenGL: веб-сайт. URL: <https://www.khronos.org/opengl/> (дата звернення: 30.11.2023)

24. Реєстр OpenGL: веб-сайт. URL: https://registry.khronos.org/OpenGL/index_gl.php (дата звернення: 30.11.2023)

25. OpenGL. OpenGL ARB Meeting Notes - March 9-10: веб-сайт. URL: www.opengl.org (дата звернення: 30.11.2023)

26. WebGL API: веб-сайт. URL: https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API (дата звернення: 30.11.2023)

27. WebGL: веб-сайт. URL: <https://www.khronos.org/webgl/> (дата звернення: 30.11.2023)

28. Hardware design of WebGL: веб-сайт. URL: https://www.researchgate.net/figure/Hardware-Design-of-WebGL_fig1_264397674 (дата звернення: 30.11.2023)

29. Convert world to screen coordinates in WebGL: веб-сайт. URL: <https://olegvaraksin.medium.com/convert-world-to-screen-coordinates-and-vice-versa-in-webgl-c1d3f2868086> (дата звернення: 30.11.2023)

30. Kessenich J., Baldwin D., Rost R. THE OPENGL SHADING LANGUAGE. 2023. 244 с. URL: <https://registry.khronos.org/OpenGL/specs/gl/GLSLangSpec.1.10.pdf> (дата звернення: 30.11.2023).

31. GLSL Shaders: веб-сайт. URL: https://developer.mozilla.org/en-US/docs/Games/Techniques/3D_on_the_web/GLSL_Shaders (дата звернення: 30.11.2023)

32. Processing development environment: веб-сайт. URL: <https://processing.org/environment/> (дата звернення: 30.11.2023)

33. Visual studio code: веб-сайт. URL: <https://code.visualstudio.com/docs> (дата звернення: 30.11.2023)

34. VS code run code: веб-сайт. URL: <https://www.alphr.com/vs-code-run-code/> (дата звернення: 30.11.2023)

35. Microsoft shocks the world with visual studio code: веб-сайт. URL: <https://techcrunch.com/2015/04/29/microsoft-shocks-the-world-with-visual-studio-code-a-free-code-editor-for-os-x-linux-and-windows/> (дата звернення: 30.11.2023)

36. Most popular technologies. Developer's survey: веб-сайт. URL: <https://survey.stackoverflow.co/2023/#technology-most-popular-technologies> (дата звернення: 30.11.2023)

37. Lambert J., Tobler W. Notes and comments on the composition of terrestrial and celestial maps. Redlands, 2011. 100 p.

38. Conic projection: веб-сайт. URL: <https://web.archive.org/web/20161212000553/progonos.com/furuti/MapProj/Normal/ProjCon/projCon.html> (дата звернення: 30.11.2023)

39. Williams, C.E.; Ridd, M.K. Great Circles and the Gnomonic Projection. The Professional Geographer, 1960. 14-16 p.

40. Snyder, J.P. Map Projections – A Working Manual. U.S. Geological Survey Professional Paper. Vol. 1395. Washington, D.C: United States Government Printing Office, 1987. 164-168 p.

41. Lambert, J., Tobler, W. Notes and comments on the composition of terrestrial and celestial maps. Redlands, 2021.

42. Richardus, P., Adler, R. map projections. New York, NY: American Elsevier Publishing Company, 1972

43. Robinson, A., Randall S., Morrison, J., Muehrcke, P. Elements of Cartography (fifth ed.). Wiley, 1985

44. Snyder, J.P., Voxland, P.M. "An album of map projections". Album of Map Projections. U.S. Geological Survey Professional Paper. Vol. 1453. United States Government Printing Office, 2022.

45. Ghaderpour, E. Some equal-area, conformal and conventional map projections: a tutorial review. Journal of Applied Geodesy, 2016. 197-209 p.

46. Monmonier, M. How to lie with maps (3rd ed.). The University of Chicago Press, 2018

47. Snyder, J. P. Flattening the earth: two thousand years of map projections. University of Chicago Press, 1993
48. Hargitai, H., Wang, J., Stooke, P.J., Karachevtseva, I., Kereszturi, A., Gede, M. Map Projections in Planetary Cartography, Lecture Notes in Geoinformation and Cartography, Springer International Publishing, 2018. 177-202 p.
49. Singh, Ishveena. Which is the best map projection? Geoawesomeness, 2018
50. Камера Defender c-090: веб-сайт. URL: <https://ua.defender-global.com/catalog/webcams/c-090-0-3mp-black> (дата звернення: 30.11.2023)

ДОДАТОК А

КАЛІБРАЦІЯ ОБ'ЄКТИВУ ДЛЯ СФЕРИЧНО-ПАНОРАМНОЇ ЗЙОМКИ НА ОСНОВІ КАТАДІОПТРИЧНОЇ КАМЕРИ

Тези доповіді на Міжнародній інтернет – конференція «SCIENCE 2023:
RESEARCH AND INNOVATION» 1-2 грудня 2023 року у м. Філадельфія, США.

Аркушів 5

UDC 004.4

1. Technical science

CALIBRATION OF LENS FOR SPHERICAL PANORAMIC IMAGING BASED ON
CATADIOPTIC CAMERA

Honcharuk K.H.

student at Institute of Nuclear and Thermal Energy
NTUU “Igor Sikorsky Kyiv Polytechnic Institute”

A. Demchyshyn

Assoc. Professor, PhD

NTUU “Igor Sikorsky Kyiv Polytechnic Institute”

Frantsevich Institute for Problems of Materials Science NASU

Annotation: topicality of this project lies in its relevance for advancing the capabilities of panoramic imaging technologies. This project addresses a specific niche within computer vision and optics, giving answers to challenges associated with processing images captured by catadioptric optical system. The goal of this project is to create a device and software that allows a user to view a 360 degrees scene on a flat screen in real time without distortion. The practical value of this project lies in its potential to improve the functionality and performance of catadioptric optical systems, providing positive impact for the range of applications and industries that rely on accurate panoramic imaging.

Key words: catadioptric camera, spherical panoramic shooting, image processing.

Catadioptric cameras achieve an almost hemispherical field of view with a perspective camera and a catadioptric system [1], which represents a combination of reflective (catoptric) and refractive (dioptric) elements [2, 3]. The prototype that was made during the research consists of a spherical mirror and a camera Defender c-090 (see fig.1). The goal of this project is to create a device and software that allows a user to view a 360 degrees scene on a flat screen in real time without distortion. One camera without modification is able to register only half of spherical mirror.



Figure1 – Prototype of the catadioptric system

The angle from which reflection is coming, is not distorted by a sphere, thus a rate of change of horizontal coordinate of final, rectangular image is linear.

Unlike horizontal coordinate, a rate of change of vertical coordinate is non-linear due to reflection and refraction of rays of incoming light. At figure 2, that illustrates cut of a sphere by one of lines from the previous picture. At figure 2 black lines are rays, that after reflection coming towards camera's lens, white lines are normals of surface of the sphere at corresponding places.

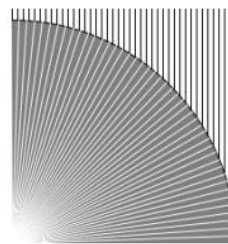


Figure 2 – Reflected rays coming towards the camera

We know that the first law of reflection states that when light is reflected from a surface, the angle of incidence is always equal to the angle of reflection, where both the angles are measured from the path of the normal to the surface and in the same conditions[]. Thus, a ray can be retraced to investigate where the ray of light has come from. Figure 3 illustrates normals of a surface (white lines) and light rays (black lines), we can consider that they represent rays which come towards camera from figure 2 before reflection.

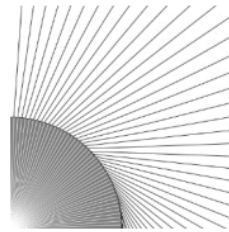


Figure 3 – Reversed ray tracing

We need to understand which range of angles of retraced rays is of interest to us in order to calculate the function which will represent non-linear rate of change of vertical coordinate of processed image without unintentional distortions. These rays contain useful visual information, that means don't contain camera reflection and rays coming way below the horizon line. In the Figure 4 light grey area is to be of interest and is included to final image after processing, on the other hand, dark grey is not included, although are processed.

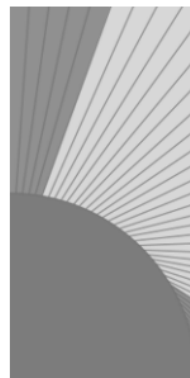


Figure 4 – Range of angles of interest

Distance between rays on left edge of figure 4 is constantly growing from bottom to top, judging by a rate of change of this distance we can calculate overall rate of change – r , of vertical coordinate and was figured out that it's changing by cosine law which depends on tangent of vertical coordinate(y). Graph can be seen in figure 5.

$$r = \cos((\tan^{-1}(y) + \pi * 0.5) * 0.5) * 0.5,$$

r – a rate of change of vertical coordinate of processed image,

y – change of vertical coordinate of input image



Figure 5 – Graph represents a rate of change of vertical coordinate of an image
Interface of the software that implements above stated math model illustrated on figure 6.



Figure 6 – User interface of the software

The software is running in browser and leverages JavaScript, as a programming languages that can be interpreted by browser and WebGL as library that gives tools to leverage parallel processing abilities of graphical processing unit[4]. Unlike CPU, GPU allows pixel-processing at the same time independently from one another, that leads to

better performance. The application is running 30 frames per second or faster on devices that support WebGL.

During the research following goals was accomplished.

1. Conducted a market analysis for cameras with viewing angles exceeding 180 degrees, identifying a demand for ultra-wide-angle lenses in video surveillance, unmanned systems, and fixed cameras for online conferencing. Selected Visual Studio Code, JavaScript, and WebGL for development, leveraging parallel computing for image processing from a catadioptric camera.

2. Developed a cross-platform software product using JavaScript and GLSL to process images from a catadioptric camera, delivering a resolution of 640x480 pixels at a speed of 30 frames per second or higher. The system is compatible with Windows, MacOS, Linux, Android, and iOS.

3. Manufactured a prototype catadioptric system comprising a Defender c090 camera, spherical mirror, and housing, enabling observation and image capture with a resolution of 640x480 pixels.

4. Tested the hardware and software complex, revealing image distortion near the upper or lower borders. Minimized distortion by recommending placement of observed objects away from these edges. Vertical lines adhered to perspective laws, while horizontal lines exhibited unavoidable distortion. Despite general distortions, object recognition remained unaffected.

References

1. C. Geyer, K. Daniilidis, Catadioptric projective geometry, *International Journal of Computer Vision*, 45 (3) (2001) 223–243.
2. S. Nayar, Catadioptric omnidirectional camera, in: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997.
3. E. Hecht, A. Zajac, *Optics*. Addison-Wesley, 1997.
4. WebGL. URL: <https://www.khronos.org/webgl/>