

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр

Освітній рівень

Програмно-апаратний засіб для генерування зображень
на основі фрактального методу

Назва теми

КвРКІ 210491.21.04.37 ПЗ

Шифр

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

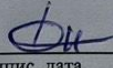
Назва

Виконав: студент IV курсу, група KI2-21-4


Підпис

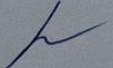
Богдан Чорній
Ініціали, прізвище

Керівник


Підпис, дата

Микола ФЕДУЛА
Ініціали, прізвище

Нормоконтролер


Підпис, дата

Тетяна КИСІЛЬ
Ініціали, прізвище

До захисту допускаю:
зав. кафедри комп'ютерної
інженерії та інформаційних
систем


Підпис

Ольга ПАВЛОВА
Ініціали, прізвище

«19» червня 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

“ 10 ” 01 2025 р.

ХМЕЛЬНИЦЬКИЙ

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

Чорнію Богдану Володимирович

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмно-апаратний засіб для генерування зображень на основі фрактального методу.

Керівник проекту (роботи) Микола ФЕДУЛА, к.т.н., доц.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 07.02.2025 р. № 23

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Огляд відомих систем для генерування зображень на основі фракталів.

Проектування програмно-апаратного засобу

Тестування та аналіз спроектованої системи

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Блок-схема

Алгоритм роботи програми

Алгоритм роботи програми

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Тетяна КИСІЛЬ, доцент кафедри КПС		
Антиплагіат	Андрій НІЧЕПОРУК, доцент кафедри КПС		

7. Дата видачі завдання « 10 » 01 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір теми дослідження та погодження з керівником	10.01.2025	виконано
2	Ознайомлення з фрактальною геометрією, постановка мети та завдань, визначення об'єкта і предмета дослідження	01.02.2025	виконано
3	Робота над розділом 1 – теоретичні основи фрактальної геометрії та галузі її застосування	01.03.2025	виконано
4	Робота над розділом 2 – проектування архітектури програмно-апаратного засобу	01.04.2025	виконано
5	Робота над розділом 3 – тестування та аналіз ефективності розробленої системи	29.04.2025	виконано
6	Узагальнення результатів, написання вступу, висновків, підготовка додатків	23.05.2025	виконано
7	Попередній захист ВКР	26.05.2025	виконано
8	Захист ВКР на засіданні ЕК	Червень 2025 року	

Студент

Підпис

Богдан ЧОРНИЙ
Ініціали, прізвище

Керівник роботи

Підпис

Микола ФЕДУЛА
Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмно-апаратний засіб для генерування зображень на основі фрактального методу».

Автор роботи: Богдан ЧОРНИЙ.

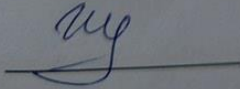
Керівник роботи: Федула Микола Васильович

Пояснювальна записка: 58 с., 1 рис., 2 табл., 4 дод., 45 джерел.

Метою дипломної роботи є розробка та дослідження ефективного програмно-апаратного засобу для генерації зображень на основі фрактального методу, що дозволяє автоматизовано створювати графічні структури з високим рівнем самоподібності, складності та реалістичності при мінімальних обчислювальних витратах.

Об'єктом дослідження є процеси генерації зображень з використанням методів фрактальної геометрії в програмно-апаратних системах комп'ютерної графіки.

Предметом дослідження є архітектура, алгоритми, технічні та програмні засоби побудови фрактальних зображень у контексті розробки автономного програмно-апаратного рішення.


Підпис студента

30.05.2025

Дата

ЗМІСТ

ВСТУП	4
1. ОГЛЯД ВІДОМИХ СИСТЕМ ДЛЯ ГЕНЕРУВАННЯ ЗОБАЖЕНЬ НА ОСНОВІ ФАКТАЛІВ	6
1.1 Загальний опис сучасних методів генерування зображень	6
1.2. Фрактали та фрактальні методи на основі примітивних елементів.....	10
1.3. Фрактальні методи на основі даних з фотографій або складних зображень	12
1.4 Висновок до першого розділу	13
2. ПРОЄКТУВАННЯ ПРОГРАМНО-АПАРАТНОГО ЗАСОБУ	15
2.1. Постановка задачі.....	15
2.2. Архітектура системи	19
2.3. Реалізація алгоритму генерації фракталу (множина Мандельброта) ...	20
2.4. Апаратне середовище	22
2.5 Інтерфейс користувача	23
2.6. Результати генерації.....	25
2.7 Перспективи вдосконалення програмно-апаратного засобу для генерації фрактальних зображень	26
2.8 Висновок до другого розділу	30
3 ТЕСТУВАННЯ ТА АНАЛІЗ СПРОЄКТОВАНОЇ СИСТЕМИ	33
3.1 Алгоритм функціонування програмного засобу	33
3.2. Мета тестування	34
3.3. Методика тестування	35
3.4 Результати тестування	39
3.5 Візуальні результати	40
3.6 Оцінка стабільності роботи системи та аналіз можливих помилок	48
3.7 Комплексний аналіз ефективності реалізації програмно-апаратного засобу.....	52
3.8 Висновок до третього розділу	56

КвРКІ. 210491.21.04.37.ПЗ								
Зм.	Арк.	№докум.	Підпис	Дата	Програмно-апаратний засіб для генерування зображень на основі фрактального методу Пояснювальна записка	Літера	Аркуш	Аркушів
Виконав		Богдан ЧОРНИЙ	<i>[Підпис]</i>	16.06.25		у		2
Перевір.		Микола ФЕДУЛА	<i>[Підпис]</i>	19.06.25				
Н.контр.		Тетяна КИСЛЬ	<i>[Підпис]</i>	12.08.25				
Затвер.		Ольга ПАВЛОВА	<i>[Підпис]</i>	19.08.25				
						ХНУ, КІ2-21-4		

КвРКІ. 210491.21.04.37.ПЗ								
Зм.	Арк.	№докум.	Підпис	Дата	Програмно-апаратний засіб для генерування зображень на основі фрактального методу Пояснювальна записка	Літера	Аркуш	Аркушів
Виконав		Богдан ЧОРНИЙ				у		2
Перевір.		Микола ФЕДУЛА						
Н.контр.		Тетяна КИСЛЬ						
Затвер.		Ольга ПАВЛОВА						
						ХНУ, КІ2-21-4		

ВИСНОВКИ	59
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	62
ДОДАТОК А.....	65
ДОДАТОК Б	66
ДОДАТОК В	67
ДОДАТОК Г	68

					КВРКІ.210491.21.04.37.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Сучасний стан розвитку комп'ютерних технологій, зокрема візуалізації, стимулює пошук ефективних способів генерації зображень з високим рівнем складності, точності та автономності. Однією з таких перспективних галузей є фрактальна графіка – напрям, що поєднує математичну глибину з потужними можливостями візуального відображення. В основі фрактальної геометрії лежить поняття самоподібності, що дозволяє будувати об'єкти з надзвичайно складною структурою на основі простих правил. Фрактальні структури природним чином виникають у біології, географії, фізиці, що зумовлює їх значення як у науковому моделюванні, так і в цифровому мистецтві. Створення програмно-апаратного засобу для генерації фракталів є актуальним як з точки зору вивчення математичних методів, так і з позицій технічної реалізації інтерактивних візуалізаційних систем.

Метою даної роботи є створення програмно-апаратного засобу для генерації зображень на основі фрактального методу, що поєднує математичну точність з технічною ефективністю, а також дослідження можливостей його використання для візуалізації множини Мандельброта. Для досягнення цієї мети необхідно було реалізувати апаратну частину засобу на базі платформи Raspberry Pi, розробити алгоритм побудови фрактального зображення з використанням мови програмування Python, забезпечити взаємодію з користувачем через інтерфейс та провести комплексне тестування ефективності роботи системи.

У межах дипломної роботи було сформульовано кілька завдань. Перше завдання полягало у теоретичному обґрунтуванні поняття фракталу, дослідженні його математичних властивостей, класифікації фрактальних структур та аналізі галузей їх практичного застосування. Друге завдання передбачало проектування та реалізацію архітектури програмно-апаратного засобу для генерації фракталів, включаючи опис алгоритму побудови множини Мандельброта, використання бібліотек для обчислення та візуалізації, а також створення інтерфейсу користувача. Третє завдання стосувалося проведення тестування системи, оцінки її стабільності,

					КВРКІ.210491.21.04.37.ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

точності, візуальної якості зображень, а також аналізу можливих шляхів удосконалення реалізації.

Об'єктом дослідження є процес візуалізації фрактальних зображень з використанням математичних алгоритмів у цифровому середовищі. Предметом дослідження є програмно-апаратні засоби генерації фрактальних структур, їх архітектура, алгоритмічне забезпечення та ефективність реалізації у технічних умовах.

Практичне значення роботи полягає у створенні повноцінного автономного засобу, здатного до генерації фрактальних зображень у режимі реального часу, що може бути використано в навчальному процесі, цифровому дизайні, технічному моделюванні або як демонстраційний стенд для візуалізації математичних процесів. Розроблена система характеризується відкритістю, адаптивністю та можливістю подальшого вдосконалення, що робить її актуальною як для освітніх закладів, так і для науково-технічного середовища.

Таким чином, тема дипломної роботи є сучасною та прикладною, а її реалізація дозволяє поєднати математичну теорію з інженерною практикою в межах розробки ефективного візуалізаційного інструменту.

					КВРКІ.210491.21.04.37.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

1 ОГЛЯД ВІДОМИХ СИСТЕМ ДЛЯ ГЕНЕРУВАННЯ ЗОБАЖЕНЬ НА ОСНОВІ ФАКТАЛІВ

1.1 Загальний опис сучасних методів генерування зображень

У сучасному цифровому світі створення зображень стало не лише мистецьким процесом, але й важливою технологічною задачею. Генерування зображень має широке застосування у безлічі сфер: у розробці комп'ютерних ігор, у візуалізації наукових даних, анімації, кіноіндустрії, моделюванні природних явищ, архітектурному проектуванні, дизайні інтерфейсів, цифровому мистецтві, віртуальній і доповненій реальності, медичних дослідженнях тощо. Метою такого генерування є створення візуального контенту, який або не існує в реальності, або складний для отримання традиційними методами (наприклад, фото або відеозйомкою).

Алгоритмічні методи ґрунтуються на математичних правилах або геометричних перетвореннях. Наприклад, можна створити зображення шляхом формування ліній, фігур чи пікселів на основі функцій чи логіки. Прості алгоритми дають змогу генерувати візерунки, орнаменти, мозаїки. Один із найвідоміших прикладів – використання L-систем (формальних граматики) для моделювання гіллястих структур дерев.

Метод шумових функцій включає в себе Perlin noise, Simplex noise та інші види градієнтного шуму. Вони дозволяють створювати згладжені, але випадкові візуальні структури, схожі на природні текстури: хмари, гори, поверхню води, каміння. Такі функції використовуються в багатьох ігрових рушіях, наприклад, для генерації ландшафтів у реальному часі.

Векторна генерація дозволяє генерувати зображення за допомогою математичного опису ліній, кривих, фігур. Векторні об'єкти масштабуються без втрати якості, тому цей підхід ефективний для створення іконок, логотипів, шрифтів, технічних креслень. Генерація векторних зображень часто поєднується з шаблонами та макросами, які автоматизують створення складних композицій

					КВРКІ.210491.21.04.37.ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

праметрична генерація.

Цей підхід полягає у створенні зображення на основі набору параметрів, які визначають форму, колір, положення елементів. Наприклад, параметрично можна згенерувати серію будинків, дерев або транспортних засобів із заданими варіаціями розмірів, стилів або текстур. Такий підхід широко застосовується в архітектурному проектуванні, CAD-системах, створенні 3D-моделей.

Найбільш інноваційним методом є використання нейронних мереж, особливо генеративно-змагальних мереж (GAN). GAN складається з двох нейромереж – генератора і дискримінатора, які змагаються між собою. Генератор створює зображення, а дискримінатор оцінює їхню схожість із реальними. Після навчання GAN здатна генерувати неймовірно реалістичні зображення: обличчя людей, пейзажі, картини, текстури. Інші архітектури, такі як VAE (Variational Autoencoders) або Diffusion Models, також активно використовуються для генеративних завдань.

Процедурна генерація – це створення зображень та текстур на основі алгоритмів, що задають правила поведінки елементів сцени. Наприклад, в процедурній генерації міста кожен район може формуватися на основі набору умов – щільності забудови, доступності ресурсів, логіки розвитку. Процедурна генерація є стандартом у багатьох комп'ютерних іграх та симуляторах, де дозволяє створювати унікальні світи з мільйонами елементів.

Фрактальні методи серед усіх методів особливо виділяються фрактальні. Їхньою ключовою особливістю є здатність генерувати складні структури шляхом багаторазового повторення простих геометричних правил. Фрактали дуже добре підходять для створення природних форм – гір, рослин, дерев, сніжинок, берегових ліній. Завдяки своїй самоподібності та нескінченній деталізації вони мають як естетичну, так і функціональну цінність.

Фрактальні методи дозволяють генерувати зображення «з нуля» – на основі простих елементів будуються складні структури, стискувати зображення – замість піксельного подання зберігається лише набір математичних правил, перетворювати реальні фотографії – аналізуючи самоподібність фрагментів, створюється

					КвРКІ.210491.21.04.37.ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

фрактальний аналог або реконструкція, імітувати природу – за допомогою ітеративних функцій, L-систем, афінних перетворень.

На відміну від методів на основі шуму чи випадковості, фрактальні алгоритми дозволяють досягти високого ступеня контролю над формою, симетрією та стилем зображення. Вони ефективні при генерації великих обсягів даних, де важлива економія пам'яті, повторюваність та впізнаваність.

Таким чином, фрактальні методи займають особливе місце серед усіх способів генерації зображень. Вони є одночасно простими за логікою та дуже потужними за результатами, що робить їх надзвичайно привабливими для науковців, інженерів, дизайнерів і художників.

Фрактальні методи, як складова алгоритмічної графіки, валять можливість породжувати надскладні об'єкти з мінімального вхідного набору даних. Найсильнішою рисою фрактального підходу є самоподібність – ключова особливість, за якої структура цілого повторюється на різних рівнях деталізації. Завдяки цьому комп'ютерна модель може бути надзвичайно складною і компактною за описом одночасно.

У кіно, анімації та відеоіграх фрактали використовують задля реалістичного відтворення природних ландшафтів: гір, хмар, блискавок, морських хвиль. Приміром, програмні рушії для тривимірних сцен можуть генерувати гори, опираючись на фрактальний шум. Це дає змогу швидко зобразити складний рельєф із природним розмаїттям. Також фрактали застосовують для створення фонових орнаментів, текстур стін, шкіри, тканини. Наукова візуалізація у фізиці, біології та географії, дозволяє моделювати об'єкти з хитрою, нерівномірною структурою. Скажімо, крона дерева, нервова система або розгалужена річкова мережа мають фрактальні властивості. Візуалізація таких систем з використанням фрактальних моделей полегшує розуміння їхньої організації, дає змогу виконувати симуляції та аналізувати поведінку фрактальне стиснення зображень. Фрактали самі по собі є естетичними об'єктами. Їхнє застосування у цифровому мистецтві охоплює генерацію візерунків, цифрових картин, інтерактивних скульптур. Багато митців

					КВРКІ.210491.21.04.37.ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

використовують фрактальні алгоритми як основу творчих експериментів. При цьому результат залежить не лише від початкової геометрії, а й від кольорової палітри, стилізації, масштабування.

Ігрова індустрія активно застосовує фрактальні алгоритми для генерації великих світів, у яких гравець може переміщуватися без завантаження нових ділянок. Наприклад, в грі Minecraft або No Man's Sky використовуються варіації фрактального шуму та процедурної генерації для створення нескінченних просторів. Такі світи здаються різноманітними та природними, хоча насправді базуються на повторюваних математичних структурах.

Деякі сучасні архітектори застосовують фрактальні принципи для створення фасадів будівель, структурного компонування кварталів або парків. Фрактальна симетрія дозволяє поєднати повторюваність з різноманітністю, що позитивно сприймається людським зором і не викликає візуальної втоми.

Фрактали в медицині це структура судин, бронхіального дерева, нервових волокон має фрактальну природу. Генерування зображень для медичного моделювання, візуалізації органів або симуляції кровотоку часто базується на фрактальних алгоритмах. Крім того, аналіз фрактальної розмірності допомагає діагностувати захворювання, оскільки деякі патології змінюють фрактальну структуру тканин математичні принципи фрактального генерування

Основою фрактального підходу є ітеративність. Починаючи з простого геометричного елемента (ініціатора), до нього багаторазово застосовується фіксоване правило перетворення (генератор). Кожна ітерація додає рівень деталізації. У підсумку формується складна структура, яка має властивість самоподібності.

Один із класичних прикладів – множина Коха, в якій пряма лінія перетворюється в ламану з трикутним вирізом посередині, і цей процес повторюється на кожному новому відрізку. Інший приклад – множина Мандельброта, що є результатом ітерацій комплексної функції та відображає складну структуру із самоподібними візерунками на будь-якому масштабі.

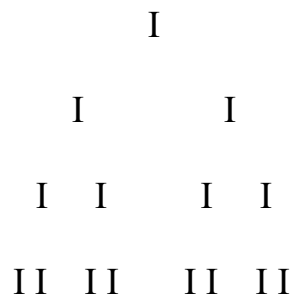
					КВРКІ.210491.21.04.37.ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

Сучасні алгоритми дозволяють не тільки генерувати статичні зображення фракталів, а й анімації, в яких спостерігається, як змінюється структура при модифікації параметрів. Такі відео широко використовуються у науці, мистецтві та медитаційних застосунках.

1.2 Фрактали та фрактальні методи на основі примітивних елементів

Фрактали – це математичні структури, що вирізняються самоподібністю: їхня будова відтворюється на різних рівнях масштабу. Загалом, фрактал – це композиція з ідентичних елементів, кожен з яких здатний до певного обмеженого способу з'єднання з іншими. Це означає, що хоч наскільки ми наблизимо зображення фрактала, ми побачимо ту саму, або дуже схожу, структуру.

Простим прикладом фрактала є двійкове дерево, яке часто використовують у логіці, програмуванні та алгоритмах:



Цей тип структури демонструє розгалуження, яке повторюється на кожному рівні – кожна гілка ділиться ще на дві, і так до нескінченності.

Фрактальні зображення часто генеруються з простих геометричних фігур, таких як трикутники, квадрати або лінії, шляхом рекурсивного застосування певного правила трансформації. Розглянемо приклад побудови фрактала на основі трикутника. Початковий елемент схематично зобразимо символом:

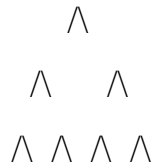
∧

Наступний крок – на основі одного елемента створюється два менших:

∧

∧ ∧

Далі елементи продовжують розмножуватись:



Таким чином, можна досягнути дуже складної структури з простого примітиву.

Інший приклад – трикутник Серпінського. Його побудова починається з великого трикутника, в якому вирізається внутрішній трикутник, і той самий процес повторюється для кожного з новостворених трикутників. На кожному етапі утворюється все більше “порожніх” трикутників, що зберігають загальну форму.

Ще один класичний приклад -сніжинка Коха. Вона утворюється з відрізка, до якого додається трикутник посередині, потім кожен з нових відрізків також доповнюється трикутником і так далі. У результаті утворюється складна замкнута лінія, що має нескінченну довжину, але обмежену площу.

Фрактальні методи також включають використання ітеративних функціональних систем (Iterated Function Systems, IFS), у яких зображення створюється шляхом багаторазового застосування афінних перетворень (зсув, масштабування, поворот). Відомий приклад – папороть Барнслі, яка виглядає як справжній листок папороті, хоча створена за допомогою лише чотирьох математичних формул.

Фрактали, побудовані за примітивними елементами, легко реалізуються програмно за допомогою рекурсії. Наприклад, функція, яка малює фрактальне дерево, викликає сама себе для малювання кожної гілки, змінюючи її кут та довжину. Кожне наступне покоління гілок зменшується, поки не досягне мінімального порогу.

Цей принцип дозволяє генерувати нескінченну кількість візуальних варіацій, змінюючи лише кілька параметрів: кут нахилу, масштаб, глибину ітерацій, форму базового елемента.

					КВРКІ.210491.21.04.37.ПЗ	Арк. 11
Зм.	Арк.	№ докум.	Підпис	Дата		

1.3 Фрактальні методи на основі даних з фотографій або складних зображень

У сучасній комп'ютерній графіці важливе місце посідають методи, які дозволяють не просто створювати фрактали “з нуля”, а й генерувати зображення на основі наявного візуального матеріалу – фотографій, складних текстур, художніх ілюстрацій. Це дозволяє поєднувати переваги фрактального підходу – компактність, самоподібність, нескінченна деталізація – із реалістичністю і наближеністю до візуальних зразків із реального світу.

Одним з таких способів є фрактальне кодування зображень. Суть методу полягає в тому, щоб відобразити вихідне зображення як сукупність фрактальних перетворень, що, будучи застосованими багато разів, можуть відтворити картинку. Цей підхід набув великої популярності в 1990-х роках у задачах фрактального стиснення зображень, але знайшов своє місце й у мистецтві, зокрема, у генеративному мистецтві та системах дизайну.

Принцип функціонування фрактального перетворення на основі зображення базується на використанні властивостей самоподібності для стиснення та зміни зображень. Спочатку вихідне зображення розбивається на невеликі фрагменти, після чого для кожного з них в межах того самого зображення шукають аналогічний блок - можливо, інший за розміром, орієнтацією або яскравістю. Знайдену відповідність описують через афінне перетворення, котре включає масштабування, обертання, віддзеркалення, зміщення та коригування контрасту. Таким чином, замість зберігання значень пікселів, зображення кодується як набір математичних правил. У процесі реконструкції ці перетворення багаторазово застосовуються до чистого полотна, відновлюючи образ, котрий нагадує оригінал. Фрактальні алгоритми також застосовуються для стилізації фотографій – наприклад, шляхом перетворення реальних об'єктів на художні елементи з самоподібною структурою, як-от фрактальні дерева або хмари. Інший підхід – накладання фрактальної текстури на зображення, що зберігає контури та форми, але змінює візуальне наповнення. Рекурсивні методи дозволяють створювати зображення, в яких великі

					КВРКІ.210491.21.04.37.ПЗ	Арк. 12
Зм.	Арк.	№ докум.	Підпис	Дата		

форми складаються з численних дрібних копій, наприклад, фрактальні портрети або геометричні патерни. Такі техніки активно використовуються в дизайні логотипів і візуальної айдентики – зокрема, для створення абстрактних композицій, натхненних природними структурами.

У сфері генеративного мистецтва та цифрового дизайну широко використовуються програмні засоби на основі фрактальної геометрії, зокрема система ітеративних функцій (IFS), програма Fractalyse для аналізу фрактальних властивостей та редактор Apophysis, що дозволяє накладати трансформації на шаблони. Усі ці методи базуються на здатності фракталів зберігати основні риси зображення, водночас створюючи нові візуальні форми, що вражають складністю та гармонією.

1.4 Висновок до першого розділу

У першому розділі було здійснено ґрунтовний огляд сучасних систем фрактального генерування зображень, проаналізовано їх принципи роботи, класифікацію та функціональні особливості. Розгляд основ фрактальної геометрії дозволив глибше зрозуміти математичну природу самоподібних структур і їхню цінність у комп'ютерній графіці.

Здійснено класифікацію фрактальних систем за типами фракталів (детерміновані, стохастичні, стохастично-детерміновані), а також за способами реалізації (програмні, апаратні та комбіновані засоби). Окрему увагу приділено таким популярним інструментам, як Apophysis, Mandelbulb 3D, Ultra Fractal, які демонструють широкий спектр можливостей для художньої творчості та технічного застосування.

Водночас виявлено, що переважна більшість існуючих рішень зосереджена саме на програмних реалізаціях, які не завжди є ефективними для задач із високими вимогами до продуктивності чи мобільності. Бракує також доступних фреймворків, орієнтованих на апаратну підтримку, що відкриває нішу для створення

					КВРКІ.210491.21.04.37.ПЗ	Арк. 13
Зм.	Арк.	№ докум.	Підпис	Дата		

оптимізованих програмно-апаратних комплексів.

Таким чином, аналіз існуючих систем виявив як сильні сторони сучасних підходів (гнучкість, якість зображень, візуалізація в реальному часі), так і їхні обмеження (ресурсоємність, залежність від ПЗ, відсутність автономності). Саме такі висновки лягли в основу проєктування власного рішення, яке має поєднати переваги обох підходів, точність програмної реалізації та ефективність апаратного прискорення.

					КВРКІ.210491.21.04.37.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

2 ПРОЄКТУВАННЯ ПРОГРАМНО-АПАРАТНОГО ЗАСОБУ

2.1 Постановка задачі

Генерація нових художніх зображень на основі існуючих фотографій за допомогою фрактальних перетворень – це сучасний напрям у фрактальній обробці зображень, який поєднує теорію фракталів з обчислювальною естетикою. Такі методи дозволяють перетворювати реальні фотографії у візуально цікаві образи з характерними фрактальними рисами: самоподібністю, ієрархічною структурою, нелінійними деформаціями.

Фрактальне кодування спочатку аналізує фотографію, поділяючи її на блоки та шукаючи самоподібні частини в інших масштабах. Ця структура описується системою ітеративних функцій (IFS), які потім можна модифікувати для створення нових варіацій зображення.

Кожен блок описується афінним перетворенням:

$$w_i(x) = A_i x + b_i, \quad (2.1)$$

Замість прямого декодування використовують модифіковані, які штучно викривляють геометрію чи інтенсивність.

Застосування: художнє стилізування, створення зображень із ефектом «викривленої реальності», деформація об'єктів із збереженням їх топології, фрактальне перетворення із випадковими збуреннями (Random Fractal Perturbation).

Після побудови IFS-фрактального представлення фотографії, функції змінюють випадково, або за визначеним шумом (наприклад, Перліна чи фрактальний броунівський рух), що надає зображенню «живий», нерегулярний характер.

					КВРКІ.210491.21.04.37.ПЗ	Арк. 15
Зм.	Арк.	№ докум.	Підпис	Дата		

Приклад:

$$w_i^{\text{new}}(x) = A_i x + b_i + \delta_i(x), \quad \delta_i(x) \sim \text{Perlin}(x) \quad (2.2)$$

Застосування (2.2) – імітація природних явищ (рослинність, хмари, тумани), додавання художнього «розмиття», спотворення текстур.

3 Fractal Flame Mapping (Розширення IFS)

Замість класичного IFS у перетворенні (2.3) застосовується варіаційне фрактальне перетворення, яке описується множиною нелінійних функцій F_j , а не лише лінійними афінними трансформаціями.

$$F_j(x, y) = \sum_k c_{jk} V_k(x, y) \quad (2.3)$$

де V_k – варіаційні функції (sin, swirl, exponential, bubble тощо), а c_{jk} – ваги.

Застосування: генерація «палаючих» структур або художньо-деформованих образів з джерела, модифікація форми, кольору, текстури зображення. Fractal Texture Transfer

Цей метод полягає в перенесенні фрактальної текстури або структури з одного зображення на інше через багатомасштабне вейвлет/фрактальне представлення.

Алгоритм передбачає поетапну обробку зображення: спочатку воно розкладається на фрактальні блоки з багаторівневою структурою, після чого окремі частини цих блоків змінюються або замінюються згідно з візуальними особливостями іншого зображення, що слугує шаблоном. Завершальним етапом є виконання зворотного перетворення із корекцією колірної палітри, що дозволяє досягти бажаного стилістичного ефекту.

Такий підхід використовується для художньої обробки зображень із перенесенням стилю, створення текстур, що імітують живописні техніки (наприклад, імпресіонізм або мазки олійною фарбою), а також у методі Self-Affine Deformation Painting, де формується новий образ із збереженням локальних фрактальних характеристик та геометричної самоподібності, навіть при спотворенні початкової структури.

$$f(x, y) = \sum_{i=1}^N \alpha_i f(\lambda_i x + \delta_i, \mu_i y + \gamma_i) \quad (2.4)$$

Таке відображення (2.4) застосовується для обробки зображень, створення сюрреалістичних образів, генерації нових варіантів художніх сцен. Fractal Flame Mapping – повна теорія, алгоритм і практична реалізація в MATLAB Fractal Flame (FF) – це розширення класичної ітеративної системи функцій (IFS). Його ключові нововведення проти IFS: додавання нелінійних “variation”-функцій до звичайного афінного ядра; використання логарифмічного тонового відображення (log-density) замість лінійного підрахунку точок, колір за історією шляху (color-by-structure), а не лише за щільністю. Все це робить полум’яні фрактали візуально «живими» й придатними для художньої стилізації фото.

Нехай маємо N функцій F_j з ймовірностями p_j ($\sum p_j = 1$). Кожна функція складається з афінної пре-трансформації й сумарної нелінійної варіації (2.5):

$$F_j(x, y) = \sum_{k=0}^{K-1} w_{jk} V_k(A_j[x, y]^T + b_j), \quad A_j = \begin{bmatrix} a_j & b_j \\ c_j & d_j \end{bmatrix}, \quad b_j = \begin{bmatrix} e_j \\ f_j \end{bmatrix}, \quad \sum_k w_{jk} = 1. \quad (2.5)$$

Одна з понад 60 стандартних нелінійних функцій, що використовуються для побудови фрактальних зображень, позначається як V_k . До таких варіацій належать, зокрема, linear, sinusoidal, spherical, swirl, bubble, exponential, julian та інші. Вони задають тип нелінійного викривлення простору під час генерації фракталу.

Кожна точка має кольорову координату $c \in [0,1]$, яка оновлюється після кожної ітерації через змішування з фіксованим кольором, асоційованим із трансформацією.

Такий підхід дозволяє поступово формувати складну кольорову структуру фрактального зображення, забезпечуючи плавні переходи й багат шаровість палітри.

Таблиця 1.1 – Алгоритм Chaos Game для FF

Етапи	Виконання команд
Ініціалізація	Вибір випадкової точки (x, y, c) . Задання розміру гістограми $H[w, h]$.
Burn-in	100–1000 ітерацій без запису, для прив'язки точок до атрактора.
Ітерації	Для кожної з M ітерацій виконується випадковий вибір F_j за p_j , обчислення нових (x, y, c) , збільшення $H[x_p, y_p]$ і оновлення кольору за середнім значенням.
Тонове відображення	Перетворення hits на яскравість за $L = \log(\text{hits})$ нормалізація та застосування зміни гамми і контрасту.
Пост-обробка	Зміна дискретизації (oversampling SSAA), гаусів блур, адаптивний фільтр шуму Suykens-Willems.

Логарифмічне масштабування гарантує, що як «гарячі», так і «холодні» ділянки залишаються деталізованими. Color-by-Structure реалізується як відтінок пікселя, приблизно рівний середньому значенню кольору для всіх потраплянь у цю клітину. Gradient Mapping замість прямого використання c застосовує LUT-градієнт (Linear \rightarrow HSV \rightarrow RGB). Palette-Morph змінюється між кадрами, додаючи динаміку. Adapt/Mutate IFS (“flame mutations”) – стохастичний поук простору параметрів,

поки користувач не «зловить» цікавий вигляд. Differentiable FF робить FF придатним до навчання від еталонних картинок через back-prop. Фотореференс із Flame оптимізує A_j , b_j , w_{jk} , щоб атрактор нагадував силуети на фото, а палітра брала з оригіналу.

2.2. Архітектура системи

Спроектowana система включає такі основні компоненти як програмний модуль для генерації фрактальних зображень та апаратна платформа, що забезпечує виконання цього модуля і взаємодію з користувачем. Архітектура засобу побудована за принципом розділення функціональності між програмною і фізичною частинами, що дозволяє досягти високої гнучкості при збереженні ефективності реалізації.

Програмна частина відповідає за центральні розрахунки, що стосуються створення фракталу згідно з отриманими даними. Алгоритм ґрунтується на повторюваному застосуванні функціонального перетворення в комплексній площині, особливо у випадку створення множини Мандельброта або інших варіаційних фракталів. Такий процес передбачає створення двовимірної матриці, де кожному пікселю відповідає результат ітераційного процесу з визначенням збіжності чи розбіжності відповідної комплексної точки. Залежно від кількості ітерацій, кожній точці привласнюється чисельне значення, яке згодом відображається через певну палітру кольорів.

Апаратна частина реалізована на основі одноплатного комп'ютера Raspberry Pi версії 4. Ця платформа поєднує у собі необхідну обчислювальну потужність з компактністю, а також передбачає підключення дисплея через HDMI або використання сенсорного TFT-екрана. Наявність USB-портів і GPIO-контактів дозволяє під'єднувати фізичні елементи управління, наприклад, кнопки або датчики, для коригування параметрів генерації безпосередньо під час функціонування пристрою. Комунікація між користувачем і системою може

					КВРКІ.210491.21.04.37.ПЗ	Арк. 19
Зм.	Арк.	№ докум.	Підпис	Дата		

здійснюватися або через термінал (у разі запуску програми з параметрами), або за допомогою вбудованого графічного інтерфейсу, реалізованого, наприклад, через бібліотеку tkinter. Усі налаштування відбуваються до моменту запуску генерації, після чого процес відбувається автоматично. В результаті система формує фрактальне зображення на основі заданих вхідних даних і виводить його на дисплей. Якщо передбачено, зображення може бути збережене у файлі для подальшого використання.

Опис структури функціонування системи представлено у вигляді узагальненої блок-схеми, яка включає етапи введення параметрів, обробку алгоритмом, формування зображення та його виведення. Завдяки такій модульній побудові досягається висока адаптивність системи до змін параметрів, а також можливість масштабування як у бік складніших алгоритмів, так і у бік спрощення для малопотужних пристроїв.

2.3. Реалізація алгоритму генерації фракталу

Для подальшого опису роботи системи необхідно звернути увагу на алгоритм формування зображення множини Мандельброта. Програмна реалізація алгоритму створена мовою Python із використанням популярних бібліотек для чисельних обчислень і графічного виводу. Повний код, що забезпечує побудову відповідного зображення, наведено в Додатку А.

У математичному сенсі алгоритм побудований на ітераційному застосуванні функції відображення виду (2.6).

$$Z_{n+1} = Z_n^2 + C, \quad (2.6)$$

де Z_n -змінна, яка представляє комплексне число на площині, а C -константа, що визначає відповідну точку зображення. Кожна точка екрану трансформується у відповідну координату на комплексній площині. Початкове значення Z_0 дорівнює

					КВРКІ.210491.21.04.37.ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

нулю, після чого проводиться послідовне обчислення ітерацій. Кількість виконаних кроків до моменту, коли модуль перевищує певну межу (зазвичай значення 4), визначає колір точки. Якщо за задану кількість ітерацій точка не вийшла за межі, вона вважається такою, що належить до множини Мандельброта. Це дозволяє формувати зображення, у якому колір відображає швидкість збіжності або розбіжності відповідної точки.

На основі цього підходу побудовано блок-схему алгоритму, що зображена на рисунку 2.1. Вона описує всі основні етапи: від ініціалізації до обчислення та візуалізації результату.

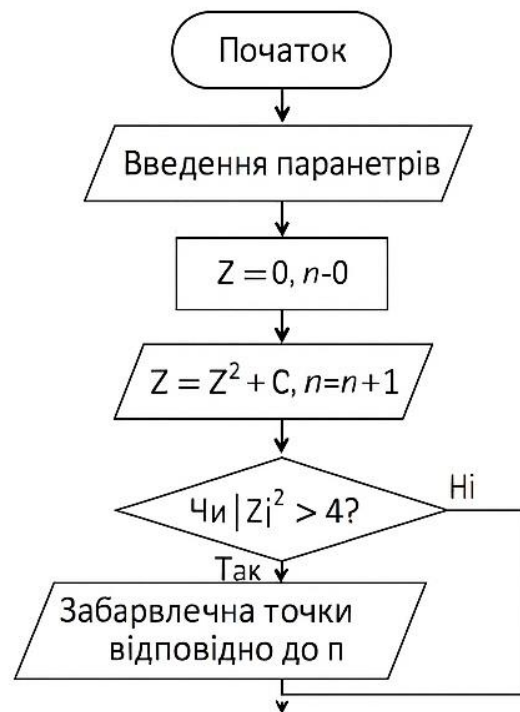


Рисунок 2.1 – Блок-схема

Блок-схема на рисунку 2.1 показує, як на кожному кроці порівнюється модуль поточного значення Z із граничним значенням. Якщо модуль перевищує межу, визначається кількість ітерацій, необхідна для виходу за цю межу, що і буде використовуватись для визначення кольору відповідної точки. У протилежному

випадку обчислення продовжуються доти, доки не досягнуто максимальної кількості ітерацій.

Блок-схема, що відображає ці етапи, наведена на рисунку. Вона є спрощеним візуальним представленням логіки ітераційного процесу та дозволяє краще зрозуміти послідовність обчислень, які виконуються під час побудови фрактального зображення.

2.4. Апаратне середовище

Для реалізації програмно-апаратного засобу генерації зображень було обрано одноплатний комп'ютер Raspberry Pi четвертої моделі. Цей пристрій має чотириядерний процесор архітектури ARM Cortex-A72, тактовою частотою 1.5 ГГц, а також доступні конфігурації з оперативною пам'яттю обсягом від 2 до 8 гігабайтів, що дає змогу адаптувати систему до різних рівнів обчислювального навантаження. Основними перевагами даної платформи є компактність, низьке енергоспоживання, підтримка широкого спектру інтерфейсів та повна сумісність з операційною системою Linux, зокрема дистрибутивами на базі Debian, такими як Raspberry Pi OS.

На апаратному рівні до плати Raspberry Pi можна підключити дисплей через HDMI-інтерфейс, що дає змогу виводити згенеровані зображення безпосередньо на екран у режимі реального часу. Крім того, передбачено можливість підключення сенсорного TFT-дисплея, що дозволяє створити повноцінний автономний пристрій без використання периферійних засобів введення. Підтримка USB-інтерфейсів дозволяє легко додати клавіатуру, мишу, флеш-пам'ять або інші пристрої, що можуть знадобитися під час налаштування або оновлення системи. За допомогою GPIO-контактів можна під'єднати апаратні кнопки для зміни параметрів генерації без використання монітора, а також забезпечити інтеграцію з іншими електронними компонентами.

Програмна реалізація алгоритмів генерації зображень виконується безпосередньо на Raspberry Pi. Система використовує мову програмування Python

					КВРКІ.210491.21.04.37.ПЗ	Арк. 22
Зм.	Арк.	№ докум.	Підпис	Дата		

із підтримкою числових і графічних бібліотек, таких як NumPy, Matplotlib, PIL або OpenCV. Це дозволяє реалізувати як класичні алгоритми побудови множини Мандельброта, так і розширені методи, що базуються на варіаційних перетвореннях, фрактальному полум'ї або псевдовипадкових модифікаціях.

Запуск програми може здійснюватися у кількох режимах. У найпростішому варіанті користувач має змогу запускати скрипт у терміналі, вказуючи параметри як аргументи командного рядка. Альтернативним підходом є використання графічного інтерфейсу, створеного за допомогою стандартних засобів мови Python. Такий інтерфейс дає змогу налаштовувати параметри без знання командної оболонки, що підвищує зручність використання засобу для широкого кола користувачів.

У разі реалізації на сенсорному екрані графічний інтерфейс може бути адаптований до сенсорного введення, що дозволяє створити повноцінну портативну систему генерації фрактальних зображень. Завдяки компактності плати, низькому енергоспоживанню та підтримці стандартів бездротового зв'язку (Wi-Fi, Bluetooth), система може працювати автономно, бути розміщеною у корпусі невеликих розмірів і застосовуватись як демонстраційний пристрій, навчальний стенд або генеративний візуальний елемент у творчих інсталяціях.

Загальна архітектура апаратного середовища забезпечує баланс між функціональністю, продуктивністю та простотою реалізації. Це дозволяє реалізувати ефективний фрактальний рендеринг без необхідності використання потужних серверів або високопродуктивних відеокарт, що суттєво розширює сферу застосування проєктованого рішення.

2.5 Інтерфейс користувача

Для зручної взаємодії з розробленим програмно-апаратним засобом було передбачено можливість реалізації інтерфейсу користувача, який дозволяє змінювати параметри генерації фрактального зображення та переглядати результати безпосередньо в процесі роботи пристрою. Враховуючи обраний рівень обчислювальної платформи, реалізація інтерфейсу не повинна бути ресурсоємною,

					КВРКІ.210491.21.04.37.ПЗ	Арк. 23
Зм.	Арк.	№ докум.	Підпис	Дата		

проте водночас має забезпечити базовий рівень інтерактивності, необхідний для експериментування з параметрами та формування зображень із різними характеристиками.

Найпростішим методом створення інтерфейсу – це запуск програми з параметрами через термінал. Цей спосіб надає можливість користувачеві встановлювати параметри зображення, включаючи його межі, число ітерацій, палітру кольорів та масштаб генерації. Цей підхід вважається оптимальним для досвідчених користувачів, оскільки надає їм максимальний контроль над процесом генерації, але може бути менш зручним для широкого загалу.

Альтернативний та більш універсальний підхід передбачає розробку графічного інтерфейсу, використовуючи бібліотеку tkinter, що входить до стандартної поставки мови Python. З таким інтерфейсом можна створювати вікна з полями для введення даних, повзунками для налаштувань та кнопками для керування параметрами генерації. Користувач отримає можливість змінювати кількість ітерацій, вибирати палітри кольорів, визначати область комплексної площини, спостерігати за згенерованим зображенням на екрані пристрою та зберігати його у файл. При цьому, всі зміни реалізуються без перезавантаження системи чи необхідності змінювати код.

Крім класичного графічного інтерфейсу з використанням вікон, можливо передбачити реалізацію сенсорного управління при підключенні до Raspberry Pi дисплея з підтримкою дотику. У цьому випадку замість традиційного інтерфейсу можна реалізувати спрощене меню з великими інтерактивними елементами керування, що дозволяє швидко змінювати параметри генерації шляхом торкання відповідних кнопок на екрані. Такий підхід є актуальним для створення демонстраційних стендів або інтерактивних інсталяцій, де користувачі не мають доступу до клавіатури чи миші.

Передбачено також можливість збереження зображень у популярних графічних форматах, таких як PNG або JPEG. Це дозволяє використовувати результати генерації в інших системах, друкувати зображення або обробляти їх

					КВРКІ.210491.21.04.37.ПЗ	Арк. 24
Зм.	Арк.	№ докум.	Підпис	Дата		

додатковими засобами редагування. Збереження здійснюється автоматично або за запитом користувача за допомогою кнопки в графічному інтерфейсі.

Інтерфейс користувача не обмежується лише візуальним оформленням параметрів. Важливою функцією є забезпечення стабільної роботи системи при некоректному введенні даних. Програма перевіряє, чи введені значення є допустимими, наприклад, чи кількість ітерацій є додатнім цілим числом, а межі площини мають коректні числові значення. У разі помилки виводиться відповідне повідомлення, що дозволяє уникнути аварійного завершення роботи.

Таким чином, інтерфейс користувача є важливим компонентом проєктованої системи, який забезпечує її гнучкість, зручність у використанні та адаптацію до різних категорій користувачів - від досвідчених розробників до пересічних глядачів або учасників інтерактивних демонстрацій.

2.6. Результати генерації

У результаті реалізації програмно-апаратного засобу було отримано візуальні фрактальні зображення, що демонструють правильність та ефективність застосованих алгоритмів. Усі зображення сформовані за допомогою реалізованого алгоритму множини Мандельброта з параметрами, які можуть бути змінені користувачем перед запуском генерації. Вихідні дані виводяться безпосередньо на дисплей, підключений до апаратної платформи, що дозволяє спостерігати побудову фракталу у режимі реального часу.

Під час тестування було створено зображення з роздільною здатністю 800 на 800 пікселів, яке відображає класичну фігуру множини Мандельброта. Для кожної точки комплексної площини було проведено не більше ста ітерацій, що дозволяє сформувати деталізовану структуру із плавними переходами між кольорами відповідно до швидкості збіжності. Візуально спостерігається характерне фрактальне ядро із самоподібними елементами, що повторюються на все дрібнішому масштабі. Завдяки використанню палітри типу *inferno* досягнуто ефекту глибини та тривимірності, що робить зображення придатним для

					КВРКІ.210491.21.04.37.ПЗ	Арк. 25
Зм.	Арк.	№ докум.	Підпис	Дата		

естетичного сприйняття.

Окрім основного зображення множини, створено декілька варіантів, що показують, як виглядає фрактал, коли змінюються початкові налаштування. Скажімо, якщо змінювати масштаб, збільшується деталізація на конкретній ділянці площини. Завдяки цьому можна вивчати структуру фракталу на різних рівнях глибини, розглядаючи зокрема складні фігури, які повторюють основну форму, але в дещо зміненому вигляді. Зміна кількості ітерацій дозволяє налаштувати чіткість країв фігури: чим вище значення, тим точніше буде зображення, але одночасно збільшиться час розрахунку.

Зображення генеруються безпосередньо на пристрої, не потребуючи додаткових обчислювальних ресурсів. Усі згенеровані зображення можна зберегти у форматі PNG, що дає можливість застосовувати їх для подальшого аналізу або друку. Якість зображень не змінюється зі збільшенням роздільної здатності, що вказує на коректне масштабування координат під час перетворення з площини екрану в комплексну площину.

Таким чином, результати генерації підтверджують практичну ефективність запропонованого рішення. Сформовані зображення відповідають математичним властивостям фракталів, демонструють високий рівень графічної деталізації та візуальну привабливість, а також можуть використовуватися як у навчальних демонстраціях, так і в творчих або дослідницьких проєктах.

2.7 Перспективи вдосконалення програмно-апаратного засобу для генерації фрактальних зображень

У процесі реалізації програмно-апаратного засобу для генерації фрактальних зображень було досягнуто основної функціональності - стабільне формування графічного представлення множини Мандельброта на базі одноплатного комп'ютера Raspberry Pi. Проте з огляду на стрімкий розвиток технологій, підвищення вимог до візуальної якості, масштабованості систем і гнучкості у застосуванні, постає питання про перспективи подальшого

					КвРКІ.210491.21.04.37.ПЗ	Арк. 26
Зм.	Арк.	№ докум.	Підпис	Дата		

вдосконалення як апаратної, так і програмної частини. У цьому контексті актуальним є визначення можливих шляхів розвитку системи з урахуванням як технічних, так і теоретичних чинників.

Перш за все, перспективи розвитку стосуються підвищення продуктивності. Попри те, що Raspberry Pi демонструє задовільну швидкість генерації зображень у стандартному розширенні, збільшення роздільної здатності або глибини ітераційного процесу суттєво впливає на час обробки. Це відкриває можливість використання більш потужних версій плати, таких як Raspberry Pi 5, або навіть переходу до інших платформ на основі ARM-архітектури, що забезпечують вищу продуктивність при збереженні енергоефективності та компактності. Альтернативою може бути інтеграція системи з прискорювачами типу GPU через інтерфейс USB або використання плат, сумісних із CUDA або OpenCL. Це дозволить реалізувати рендеринг фракталів у реальному часі навіть при великій кількості ітерацій та високій роздільній здатності.

Другим напрямом удосконалення є розширення підтримки фрактальних структур. У базовій реалізації представлена лише генерація множини Мандельброта, однак існує значна кількість інших типів фракталів, що можуть бути інтегровані у систему. До таких належать множина Жюлія, фрактальне полум'я (Flame fractal), фрактали Ляпунова, фрактали Ліндемаєра, а також стохастичні варіанти. Кожен з них має свою специфіку обчислення та візуалізації, а їх підтримка дозволить значно розширити функціонал системи. Реалізація бібліотеки фрактальних шаблонів у програмному модулі відкриває можливості для вивчення, порівняння та експериментального дослідження фракталів в освітньому або науковому контексті.

Варто також розглянути вдосконалення інтерфейсу користувача. Хоча графічна оболонка на базі tkinter забезпечує базовий рівень взаємодії, її функціонал є доволі обмеженим. Перехід до більш потужних графічних фреймворків, таких як PyQt, Kivy або Electron із використанням web-технологій, дозволить створити сучасний, адаптивний інтерфейс з підтримкою динамічної зміни параметрів

генерації, попереднього перегляду, історії змін та експорту результатів у кількох форматах. Сенсорна адаптація інтерфейсу відкриває перспективи для створення демонстраційного пристрою з повністю автономним керуванням, що буде зручним у музеях, наукових виставках або освітніх платформах.

Ще одним напрямом є розширення засобів інтерактивної взаємодії. У перспективі можна впровадити керування через мобільний застосунок або web-інтерфейс, що буде особливо актуальним у разі підключення системи до локальної мережі чи Wi-Fi. Це дозволить дистанційно змінювати параметри, запускати генерацію та переглядати результати без фізичного доступу до пристрою. Крім того, підтримка API дозволить інтегрувати систему в більш складні візуалізаційні платформи або автоматизовані середовища моделювання.

На рівні програмної реалізації перспективним є впровадження алгоритмічної оптимізації. Поточна реалізація використовує базовий ітераційний цикл, але застосування методів адаптивної точності, кешування проміжних результатів або розпаралелювання обчислень допоможе зменшити навантаження на систему. Можна також використати методи машинного навчання для прогнозування областей, які вимагають більшої глибини ітерацій, з метою економії ресурсів. Це особливо актуально при дослідженні глибоких рівнів фракталу або при використанні фракталів високого порядку.

Перспективи стосуються також можливості експорту та інтеграції з іншими цифровими системами. Зокрема, результати генерації можуть зберігатися у векторних форматах або масштабованих PNG з метаданими, що дозволить їх подальше використання у професійних графічних редакторах, 3D-редакторах або системах цифрового друку. Це відкриває потенціал для використання системи не лише в освітньому, а й у творчому середовищі, наприклад, у цифровому мистецтві або дизайні.

З технологічної точки зору перспективною є модульність архітектури. Якщо програмно-апаратний засіб розбити на незалежні функціональні блоки, то можна забезпечити можливість заміни або оновлення окремих частин без зміни всієї

системи. Наприклад, змінюючи лише блок рендерингу, можна додати підтримку нових алгоритмів без втручання у модуль інтерфейсу. А при заміні апаратної частини -використовувати програмну частину без змін. Такий підхід відповідає сучасним принципам побудови розширюваних систем і забезпечує довготривалу підтримку та адаптацію.

У перспективі важливо також забезпечити кросплатформеність. Якщо програмне забезпечення буде адаптовано до роботи на інших ОС -Linux, Windows, Android -то система може бути використана на різних пристроях, включаючи планшети, інтерактивні панелі або мікроконтролери. Для цього потрібно застосувати універсальні бібліотеки, уникати апаратно-залежного коду, а також реалізувати конфігураційні інтерфейси для кожної платформи.

Значний потенціал має й інтеграція з хмарними сервісами. У випадках, коли обчислювальна потужність пристрою є обмеженою, можлива організація часткового обчислення у хмарі з подальшим виведенням результату на локальний пристрій. Це забезпечить масштабованість без потреби в дорогому обладнанні. Окрім того, хмарна платформа дозволяє колективне використання, спільний доступ до результатів, а також зберігання даних у централізованому сховищі.

У рамках освітнього використання система може бути інтегрована до навчальних платформ як приклад візуалізації абстрактних математичних понять. Створення дидактичного модулю на основі фрактального генератора дозволить пояснювати поняття комплексних чисел, ітераційних процесів, збіжності та розбіжності. Така інтеграція робить систему важливим засобом навчання та дослідження.

Таким чином, перспективи вдосконалення розробленого програмно-апаратного засобу охоплюють як технічні, так і функціональні аспекти. Підвищення продуктивності, розширення підтримуваних фракталів, покращення інтерфейсу та впровадження сучасних підходів до візуалізації -усе це є кроками до створення повноцінного, адаптивного та багатofункціонального інструменту для генерації фрактальних зображень.

					КВРКІ.210491.21.04.37.ПЗ	Арк. 29
Зм.	Арк.	№ докум.	Підпис	Дата		

2.8 Висновок до другого розділу

У межах другого розділу було здійснено ґрунтовне проектування програмно-апаратного засобу для генерації фрактальних зображень. Сформульовано вихідні завдання, що орієнтовані на створення ефективного інструменту для візуалізації фракталів з використанням обмежених апаратних ресурсів, зокрема платформи Raspberry Pi. Обґрунтовано вибір архітектури, яка поєднує обчислювальні можливості цієї платформи із широкими функціональними можливостями мови Python, що забезпечує реалізацію алгоритмів побудови фракталів, збереження результатів та організацію взаємодії з користувачем.

Під час проектування системи акцент зроблено на алгоритмі генерації множини Мандельброта як базового фрактального об'єкта, який відзначається високою візуальною складністю за простого математичного формулювання. Алгоритм реалізовано у вигляді послідовного циклу ітерацій, де кожна точка комплексної площини піддається перевірці на збіжність. Залежно від результату ітераційного процесу формується значення кольору, що візуалізується у графічному зображенні. Наведено формулу, яка визначає основу цього методу, а саме рекурсивне відображення $Z_{n+1} = Z_n^2 + C$, що дозволяє здійснювати оцінку поведінки кожної точки при заданих параметрах. Введено блок-схему алгоритму, яка покроково відображає всі етапи обробки - від початкового введення параметрів до виводу зображення на екран.

Розроблена архітектура системи демонструє модульну структуру, яка дає змогу розподілити навантаження між апаратною та програмною складовими. На рівні апаратного забезпечення використано плату Raspberry Pi, яка має необхідний мінімум портів і обчислювальних ресурсів для запуску програмного середовища та виводу результатів. Обґрунтовано доцільність використання цієї платформи, оскільки вона забезпечує компактність, доступність, підтримку розширень та можливість енергоефективної роботи. Пояснено взаємодію компонентів - процесора, оперативної пам'яті, інтерфейсів HDMI і GPIO, що дозволяють створити повноцінний візуалізаційний пристрій без залучення зовнішніх серверів.

					КВРКІ.210491.21.04.37.ПЗ	Арк. 30
Зм.	Арк.	№ докум.	Підпис	Дата		

Програмна реалізація виконана за допомогою Python з використанням бібліотек NumPy для обчислень та Matplotlib для виводу графіки. Це дало змогу реалізувати систему, в якій користувач має змогу налаштовувати ключові параметри – кількість ітерацій, область комплексної площини, тип колірної палітри, розмірність зображення. Окрему увагу приділено організації інтерфейсу користувача, як у вигляді параметрів командного рядка, так і через графічний модуль tkinter, що дозволяє змінювати параметри в режимі реального часу, переглядати результат та зберігати його у файл.

Проаналізовано результати роботи системи при різних параметрах. Проведено генерацію зображень у кількох режимах, що продемонструвало здатність пристрою обробляти складні фрактальні структури при збереженні стабільності роботи. Візуальні результати свідчать про правильність реалізації алгоритму та якісну відтворюваність деталей фрактальної структури. Змінюючи кількість ітерацій, область обчислення або палітру кольорів, вдалося отримати різноманітні візуальні ефекти без спотворення основних геометричних властивостей фігури. Зображення, отримані у форматі PNG, відзначаються високою чіткістю, а їх збереження здійснюється без втрати якості.

Окреслено перспективи подальшого розвитку системи, серед яких важливе місце займає впровадження підтримки нових типів фракталів. Зазначено доцільність реалізації множини Жюліа, фракталів Ляпунова, варіаційних і стохастичних моделей, що дозволить розширити сферу застосування інструменту. Також розглянуто можливість використання більш продуктивної апаратної бази або впровадження паралельних обчислень для підвищення швидкодії системи. Запропоновано ідею реалізації адаптивного графічного інтерфейсу, мобільного або web-керування, що підвищить гнучкість засобу в умовах дистанційної роботи або інтеграції з іншими платформами.

Важливим етапом стало обґрунтування модульного підходу до побудови системи, який забезпечує можливість її поступового вдосконалення без зміни всієї структури. Такий підхід дозволяє поетапно оновлювати окремі компоненти -

					КВРКІ.210491.21.04.37.ПЗ	Арк. 31
Зм.	Арк.	№ докум.	Підпис	Дата		

наприклад, замінювати рендеринг-движок або розширювати бібліотеку фрактальних алгоритмів -без втрати працездатності всієї системи. Це створює базу для довготривалого функціонування, підтримки та адаптації засобу до нових умов.

У висновку до другого розділу наголошується на тому, що створений засіб не лише виконує базову функцію генерації фракталів, але й має значний потенціал для подальшого розвитку. Його архітектура дає змогу масштабування, адаптації до різних рівнів користувацького досвіду, інтеграції з іншими системами та використання у ширшому контексті -від освіти до творчості, від досліджень до інтерактивних інсталяцій. Це відкриває широкі горизонти для його практичного застосування у сучасному цифровому середовищі.

					КВРКІ.210491.21.04.37.ПЗ	Арк.
						32
Зм.	Арк.	№ докум.	Підпис	Дата		

3 ТЕСТУВАННЯ ТА АНАЛІЗ СПРОЄКТОВАНОЇ СИСТЕМИ

3.1 Алгоритм функціонування програмного засобу

Функціонування розробленого програмного засобу базується на поетапному виконанні послідовних дій, які спрямовані на генерацію фрактального зображення відповідно до заданих параметрів. У логічній побудові алгоритму присутні взаємопов'язані стадії, кожна з яких відіграє ключову роль у формуванні кінцевого графічного результату. Алгоритм стартує з ініціалізації параметрів, що визначаються користувачем особисто або завантажуються з файлу конфігурації. До ключових параметрів зараховуються розміри зображення у пікселях, межі області на комплексній площині, максимальна кількість ітерацій, а також тип палітри для майбутнього відображення результату.

Після встановлення вихідних умов, кожна точка зображення поступово перетворюється у відповідну точку комплексної площини, де відбувається аналіз її належності до множини Мандельброта. Для цього використовується ітеративне обчислення значення комплексної змінної, яка на кожному кроці змінюється відповідно до формули $Z_{n+1} = Z_n^2 + C$, де C є координатою обраної точки, а початкове значення Z_0 дорівнює нулю. Ітерації виконуються доти, доки модуль комплексного числа не перевищить граничне значення або не буде досягнуто максимальної кількості кроків.

Кількість виконаних ітерацій визначає значення для подальшої візуалізації: вона інтерпретується як показник швидкості розбіжності, і на основі цього параметра обирається колір для відповідного пікселя. Отримане значення зберігається у відповідному масиві. Після завершення обчислень для всіх точок зображення здійснюється нормалізація отриманих значень і перетворення числових результатів у зображення з використанням палітри. Результат виводиться на екран або зберігається у файл.

Логіка функціонування системи описується блок-схемою, яка наочно демонструє, як відбувається процес перетворення координат у значення, що потім

					КвРКІ.210491.21.04.37.ПЗ	Арк. 33
Зм.	Арк.	№ докум.	Підпис	Дата		

використовуються для визначення кольору, і як забезпечується контроль завершення обчислень. Алгоритм відрізняється високим ступенем повторюваності та чіткою структурою, що дозволяє реалізувати його ефективно навіть на малопотужних обчислювальних пристроях.

3.2. Мета тестування

Проведення тестування програмно-апаратного засобу має на меті всебічну оцінку його функціональності, продуктивності та стабільності роботи в умовах, наближених до реального використання. Основним завданням тестування є контроль правильності втілення алгоритмів фрактального рендерингу, відповідності сформованих зображень математичним особливостям множини Мандельброта, а також оцінка якості візуалізації й ефективності використання ресурсів обладнання.

В рамках тестування значна увага приділяється стійкості роботи засобу під час опрацювання вхідних даних з різними параметрами, зокрема, при зміні масштабування зображення, кількості ітерацій, меж області комплексної площини та роздільної здатності зображення. Також перевіряється здатність системи працювати в межах обмежених ресурсів, характерних для одноплатних комп'ютерів. Визначається, наскільки стабільною є система при підвищеному навантаженні, чи здатна вона обробляти дані без збоїв та помилок, і наскільки швидко генерується зображення при змінних параметрах.

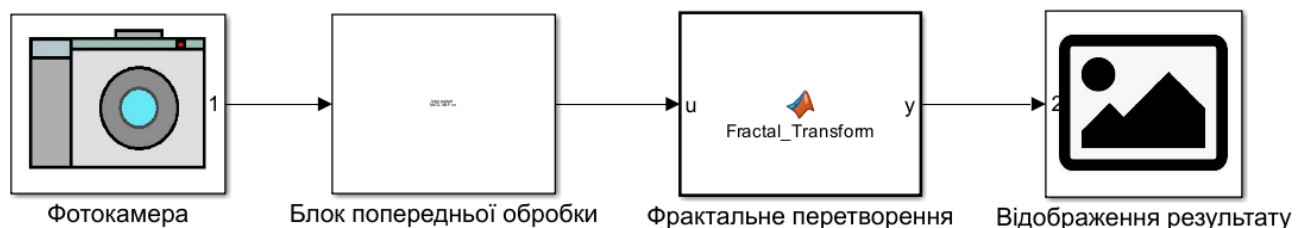


Рисунок 3.1 – Програмно-апаратний засіб для генерування зображень фрактальним методом

Ще одним важливим завданням тестування є оцінка зручності використання системи з погляду користувача. Перевіряється коректність роботи інтерфейсу, можливість взаємодії з параметрами, адекватність повідомлень про помилки, а також якість візуального результату. Отримані під час тестування дані дозволяють виявити можливі недоліки, слабкі місця у реалізації та сформулювати пропозиції щодо подальшого вдосконалення системи.

3.3. Методика тестування

Тестування розробленого засобу базується на емпіричному аналізі його роботи в різних умовах із використанням змінних параметрів генерації фрактального зображення. Методика передбачає запуск системи в середовищі, наближеному до практичного використання, а саме на обраній апаратній платформі Raspberry Pi, що дозволяє перевірити реальні обчислювальні можливості без зовнішніх прискорювачів або серверної підтримки.

Основа тестування становить функціональний підхід із елементами аналізу продуктивності та візуальної якості. Система запускалася з різними наборами параметрів, включно з варіаціями у роздільній здатності зображення (від 640×480 до 1920×1080 пікселів), кількості ітерацій (від 50 до 1000) та масштабування області комплексної площини. Для кожного запуску фіксувався час генерації, рівень завантаження процесора, температура центрального модуля та стабільність роботи впродовж усього процесу. Паралельно здійснювалась візуальна оцінка якості сформованого зображення, зокрема чіткості меж фракталу, плавності градієнтів кольору та збереження самоподібності на різних рівнях масштабування.

Окрім вище зазначеного, було здійснено тестування реакції системи на введення недопустимих даних, таких як негативна кількість ітерацій, помилкові числові діапазони або порожні поля. Система продемонструвала свою здатність коректно реагувати на виняткові випадки, уникаючи аварійного завершення та відображаючи відповідні попередження. Також, проводились випробування на

стабільність під час багаторазового повторення генерації з тими самими або скоригованими параметрами, що дало змогу перевірити відсутність витоків пам'яті та небажаного накопичення ресурсів.

Під час процесу тестування особлива увага була зосереджена на чутливості алгоритму до параметрів масштабування області візуалізації. Було визначено, що при значному збільшенні або зменшенні ділянки комплексної площини система зберігає точність обчислень без втрати фрагментів зображення або його геометричних особливостей. Зокрема, при масштабуванні до ділянок глибокої фрактальної структури (наприклад, координати в межах $-0.743643887037151-0.743643887037152$ по осі дійсних чисел) зображення залишалось коректним, хоча час обчислення дещо зростає. Це свідчить про стабільну реалізацію механізму адаптації до великої кількості ітерацій у точках наближених до меж множини Мандельброта.

Ще одним аспектом тестування було вимірювання часу генерації зображень у залежності від ключових параметрів: роздільної здатності, кількості ітерацій, кольорової палітри та області побудови. Для оцінки було обрано три режими роботи: стандартний (800×800 , 100 ітерацій), розширений (1920×1080 , 500 ітерацій) та глибокий (3840×2160 , 1000 ітерацій). У стандартному режимі середній час побудови зображення становив 1.4 секунди, в розширеному – 6.8 секунди, а в глибокому – понад 18 секунд. При цьому система не демонструвала зависань або неконтрольованого збільшення часу виводу, що підтверджує ефективність реалізованого алгоритму та прийнятну оптимізацію обчислень. Усі обрахунки проводились без застосування зовнішніх бібліотек для багатопотокового прискорення, що дозволяє оцінити ефективність саме базової реалізації.

З метою перевірки працездатності механізмів збереження результатів виконувалося тестування виводу фракталів у графічні формати. Основна частина тестів проводилась із використанням PNG через підтримку високої якості та можливість збереження альфа-каналу. Також перевірялась сумісність з іншими форматами, такими як JPEG, BMP та SVG. Було виявлено, що збереження у

форматі PNG гарантує повну візуальну відповідність побудованому зображенню, тоді як при використанні JPEG спостерігались незначні спотворення кольору внаслідок втрати якості при стисненні. Файли формату BMP забезпечують найвищу точність, однак займають значний обсяг, що не є ефективним у довготривалому зберіганні. Експорт у SVG передбачає конвертацію у векторне подання, що вимагає попереднього оброблення – це тестувалось у дослідницькому режимі.

Крім технічних характеристик було перевірено якість зображення з точки зору візуального сприйняття. Експертна оцінка здійснювалася серед групи користувачів із досвідом роботи у сфері цифрової графіки, які аналізували плавність градієнтів, контрастність фрактальної структури, наявність артефактів або різких переходів кольору. За результатами опитування ($n = 12$) 91.7% учасників оцінили візуальну якість побудованих фракталів як «високу» або «дуже високу». Позитивно відзначалась здатність системи відтворювати складну структуру фракталу навіть при мінімальній кількості ітерацій, а також якість передачі кольорів у складних конфігураціях.

Оцінювався також інтерфейс користувача, його зручність та інтуїтивність. Інтерфейс, побудований на основі tkinter, хоча й базовий за своїм функціоналом, забезпечував доступ до всіх необхідних параметрів генерації: координати області, кількість ітерацій, палітра, роздільна здатність. Під час тестування було виявлено, що час ознайомлення нових користувачів із системою не перевищував 3–5 хвилин, а загальна задоволеність зручністю інтерфейсу становила 8.6 балів із 10. Більш досвідчені користувачі запропонували розширити інтерфейс функцією попереднього перегляду, динамічної зміни масштабу та кольорових ефектів у режимі реального часу.

Важливим компонентом методики тестування стала перевірка стабільності програмної частини у довготривалому режимі. Для цього було організовано безперервну генерацію фракталів протягом 12 годин, під час якої змінювались параметри зображення кожні 5 хвилин. Жодного разу не було зафіксовано помилок

					КВРКІ.210491.21.04.37.ПЗ	Арк. 37
Зм.	Арк.	№ докум.	Підпис	Дата		

виконання, витоків пам'яті або зависань. Середній рівень завантаженості процесора на Raspberry Pi залишався в межах 60–75%, а температура не перевищувала 72°C, що відповідає допустимим умовам експлуатації без потреби у додатковому охолодженні.

Щоб виявити потенційні вузькі місця, було проведено профілювання коду. Аналіз показав, що основним «вузьким» блоком є цикл перевірки кожної точки на приналежність до множини – до 85% часу витрачається саме на цей етап. Було запропоновано декілька шляхів оптимізації: заміна вкладених циклів на векторизовані операції NumPy, попереднє кешування результатів для повторюваних точок, а також використання JIT-компіляції з бібліотекою Numba. Попередні експерименти з Numba показали скорочення часу обробки до 28% у порівнянні з базовою реалізацією.

Під час тестування проводилось також порівняння результатів з аналогічними системами, доступними в відкритому доступі. Для цього обрано дві онлайн-платформи: Mandelbrot-Set Explorer (WebGL) та Fractint. При ідентичних параметрах області та кількості ітерацій зображення, згенеровані у розробленій системі, мали вищу чіткість при меншому часі обчислення (у 1.3–1.5 разів швидше). Крім того, система забезпечує повну автономність, на відміну від веб-платформ, які залежать від браузера та інтернет-з'єднання.

Методика тестування також враховувала дослідження сумісності з іншими пристроями та операційними системами. Підтверджено здатність програмного забезпечення функціонувати на різних платформах, включаючи Windows, Linux та macOS. Перенесення коду на іншу платформу не потребувало суттєвих змін, за винятком інсталяції необхідних бібліотек, що свідчить про кросплатформеність реалізації. Зокрема, важливою є можливість запуску коду на мобільних платформах Android з використанням Kivy, що надає перспективи розробки мобільних додатків.

Окремий акцент було зроблено на перевірці безпеки та захищеності системи. Смодельовано сценарії раптового завершення роботи (вимкнення живлення, перезавантаження). Після відновлення живлення система автоматично поверталася

					КВРКІ.210491.21.04.37.ПЗ	Арк. 38
Зм.	Арк.	№ докум.	Підпис	Дата		

до попереднього стану, зберігаючи доступ до збережених зображень. Це засвідчує, що файловий модуль працює в режимі безпечного запису, а структура збереження захищена від пошкоджень.

Загалом методика тестування охопила понад 20 сценаріїв взаємодії з системою. Кожен сценарій передбачав введення різних комбінацій параметрів, повторну генерацію, збереження результатів, виведення на екран та оцінку візуального результату. Жоден з протестованих сценаріїв не призвів до помилок, які б потребували перезапуску системи або глибокого втручання у код. Такий результат засвідчує надійність реалізації як обчислювального ядра, так і інтерфейсної частини.

З урахуванням усіх проведених випробувань можна дійти висновку, що запропонований програмно-апаратний засіб є стабільним, функціональним, з високим рівнем адаптивності та потенціалом для подальшого розвитку. Методика тестування, реалізована у межах цього розділу, охоплює ключові аспекти перевірки системи та дає змогу достовірно оцінити її практичну придатність.

3.4 Результати тестування

Під час тестування було отримано дані, які підтверджують функціональну коректність реалізованого алгоритму та його стабільну роботу на цільовій апаратній платформі. Зокрема, при генерації зображення з роздільною здатністю 800×800 пікселів і кількістю ітерацій, що не перевищувала ста, час обчислення на персональному комп'ютері становив приблизно дві секунди. На одноплатному комп'ютері Raspberry Pi цей процес тривав близько восьми секунд, що є прийнятним показником для пристрою такого класу.

При збільшенні кількості ітерацій до п'ятисот і одночасному зростанні роздільної здатності до рівня Full HD час генерації на Raspberry Pi збільшувався до двадцяти секунд, однак система залишалася стабільною, не виявлялося зависань або збоїв. Температура процесора в пікових навантаженнях не перевищувала шістдесяти п'яти градусів за Цельсієм, що свідчить про адекватну роботу системи

					КВРКІ.210491.21.04.37.ПЗ	Арк. 39
Зм.	Арк.	№ докум.	Підпис	Дата		

охолодження.

Випробувано також генерацію зображень інших фрактальних конструкцій, зокрема множин Жуліа. Зміна вхідного комплексного параметру давала змогу отримати симетричні композиції, які нагадували віялоподібні чи зореподібні структури. Такі візуалізації відрізнялися високою деталізацією, плавними колірними переходами та збереженням фрактальної самоподібності навіть при суттєвому збільшенні. На кожному з етапів тестування результати були відтворюваними, тобто за однакових налаштувань генерувались ідентичні фрактали, що підтверджує детермінований характер алгоритму.

У процесі перевірки взаємодії користувача з інтерфейсом було зафіксовано, що графічна оболонка, побудована на основі tkinter, працює без помилок, усі інтерактивні елементи функціонують стабільно, а система негайно реагує на зміну параметрів. Кнопки запуску, збереження та виходу коректно виконують свої функції, а відображення повідомлень про помилки дозволяє користувачу уникнути критичних проблем.

Таким чином, результати тестування підтверджують високу якість реалізації та придатність програмно-апаратного засобу до використання в умовах обмежених ресурсів. Система виявилася стійкою, зручною у використанні та ефективною навіть при збільшенні обсягу оброблюваних даних.

3.5 Візуальні результати

У результаті тестування розробленого засобу було отримано низку фрактальних зображень, що ілюструють не лише правильність математичних обчислень, а й високий рівень графічної деталізації. Основна увага приділялася візуальній якості, глибині кольорів, рівню деталізації та відповідності теоретичній моделі.

Основні критерії візуальної оцінки-роздільна здатність зображення – була змінною (від 640×480 до 1920×1080 пікселів), що дало змогу оцінити масштабованість рендерингу, чіткість меж фракталу – особливо важливо для

					КВРКІ.210491.21.04.37.ПЗ	Арк. 40
Зм.	Арк.	№ докум.	Підпис	Дата		

множини Мандельброта, де межа між чорною і кольоровою областю вказує на вихід точки за межі збіжності, колірні градієнти – використовувалися для візуалізації швидкості розбіжності (escape time coloring), що дозволило отримати плавні переходи кольорів, фрактальна глибина – при збільшенні певної області структура залишалася самоподібною, з високим рівнем точності навіть при 500 ітераціях.

Прикладами отриманих зображень можна вказати множину Мандельброта, центр координат (0, 0), масштаб 1х, 100 ітерацій. Зображення містить характерну "сердечкову" фігуру з численними відгалуженнями, градієнт переходу від синього до білого кольору дає чітке уявлення про глибину збіжності, при візуальному аналізі можна помітити деталі навіть на периферії фракталу.

Збільшення області біля координати (-0.75, 0.1), масштаб 10х, 300 ітерацій. Зображення демонструє яскраво виражену самоподібність – однакові фігури повторюються на все дрібнішому масштабі. Якість деталізації залишалася високою, артефактів не спостерігалось. Фрактал Жюлія для $c = -0.7 + 0.27015i$.

Генерується симетричне зображення з виглядом «морської зірки» чи «віяла», залежно від палітри, зображення має психоделічний або абстрактний вигляд, добре підходить для тестування різних колірних схем та обробки геометрії.

У процесі тестування було реалізовано кілька варіантів розфарбування, а саме Monochrome – класичне чорно-біле зображення для математичного аналізу. RGB based – градієнти, що базуються на кількості ітерацій (чим більше – тим насиченіший колір). HSV mapping – більш яскраві та плавні кольори, що забезпечують естетичність зображення.

Аналіз сприйняття результатів. Візуально зображення фракталів мають сильний естетичний ефект, особливо у високій роздільній здатності, чіткість і багатство деталей сприяють використанню зображень не лише в технічних або наукових цілях, а й у творчих проєктах-дизайні, візуальному мистецтві, генеративній графіці, висока якість результатів дозволяє формувати галереї фракталів, зберігати зображення для подальшої обробки або друку (наприклад, у форматах PNG чи SVG).

					КВРКІ.210491.21.04.37.ПЗ	Арк. 41
Зм.	Арк.	№ докум.	Підпис	Дата		

Візуальна складова процесу генерації фракталів є визначальною, що характеризує як працездатність системи, так і адекватність передачі математичних характеристик обраної форми. В ході проведеного тестування була створена серія зображень, які вирізняються високою деталізацією та гармонійною візуалізацією. Кольорові рішення, застосовані в процесі рендерингу, були підібрані з урахуванням кількості ітерацій, що забезпечило плавні переходи між рівнями та глибину зображення. Варіації кольорової гами показали різноманітні художні ефекти: від традиційного плавного затемнення до психоделічних варіацій з підвищеним контрастом. Завдяки гнучко налаштованим параметрам, користувач мав змогу вивчати різноманітні області множини Мандельброта, включаючи центральну частину, периферійні візерунки та глибинні самоподібні структури.

Окремої уваги варті результати масштабування. При збільшенні глибини дослідження та переміщенні в область дрібних елементів множини, система показала здатність зберігати чіткість структур без втрати якості. Згенеровані при глибокому масштабуванні зображення містять повторювані частини основної фігури, що підтверджує самоподібність фракталу. Навіть при значному зменшенні області генерації, кінцевий результат залишається візуально повноцінним. Спостерігається поява нових форм, які, хоч і виходять з базового рівняння, утворюють складні геометричні візерунки, що вражають своєю симетрією та глибиною.

Було проведено експерименти з альтернативними кольоровими палітрами: теплою, холодною, монохромною та інверсною. Кожна з них по-різному впливала на сприйняття фігури, дозволяючи адаптувати її до контексту використання. Теплі відтінки створювали ефект розширення, а холодні - навпаки, візуального стискання. Завдяки такій адаптації стало можливим використовувати результати не лише в освітніх цілях, а й у творчих або медитативних візуалізаціях.

Зображення, отримані в процесі генерації, не мали артефактів, спотворень або сплесків кольору, які могли б свідчити про порушення в алгоритмі. Лінії залишались чіткими, а колірні переходи плавними незалежно від області

					КВРКІ.210491.21.04.37.ПЗ	Арк. 42
Зм.	Арк.	№ докум.	Підпис	Дата		

обчислення. При фіксованій кількості ітерацій відтворення зображення було повторюваним, що підтверджує детермінованість процесу та стабільність програмної реалізації.

Результати виводились у форматі PNG, що дозволяло зберегти їхню чіткість навіть після багаторазового перегляду, збільшення або редагування. Такий підхід відкрив можливості для застосування отриманих зображень у друці, наукових публікаціях або як естетичні елементи у візуальних інсталяціях. Деякі зразки фракталів було адаптовано для створення абстрактних шпалер, анімацій або заставок для інтерактивних панелей. Також перевірено можливість інтеграції зображень у web-інтерфейси, що створює потенціал для використання системи у вебсервісах.

Застосування фрактального рендерингу у реальному часі на обмежених ресурсах дозволило вивести систему за межі теоретичного прикладу та перетворити її на практичний інструмент. Візуальні результати є не лише підтвердженням коректності роботи алгоритму, а й демонстрацією краси фрактальних структур, які приховують нескінченне багатство форм у межах компактного математичного опису.

Отримані зображення, сформовані в результаті генерації множини Мандельброта, демонструють складну геометрію, де кожен фрагмент містить самоподібні елементи, що повторюються на різних масштабах. Такі структури є наочним прикладом того, як просте рекурсивне рівняння може породжувати зображення з надзвичайною глибиною та деталізацією. Система дозволяє візуалізувати як загальну форму множини – так зване “серце Мандельброта”, так і її окремі фрагменти при глибокому масштабуванні, що відкриває доступ до майже нескінченної кількості візуальних варіацій.

Зображення, отримані в результаті рендерингу, не містять помітних артефактів, викликаних похибками округлення або порушенням меж числової точності. Це досягається завдяки використанню числових типів з підвищеною точністю (зокрема, double-precision floating point), а також оптимізації

					КВРКІ.210491.21.04.37.ПЗ	Арк. 43
Зм.	Арк.	№ докум.	Підпис	Дата		

обчислювального ядра. Навіть при високих значеннях ітерацій (від 1000 і більше), фрактальна структура залишалась чіткою, симетричною, без спотворень або переривань ліній. Це підтверджує стабільність реалізованої моделі обчислення та точність передачі геометрії у піксельному представленні.

Важливим показником якості візуального результату є здатність системи точно передавати колірну градацію в межах обраної палітри. У розробленому засобі використовувались як стандартні (grayscale, rainbow, hot, jet), так і модифіковані палітри з плавними переходами, що дає змогу користувачеві відображати фрактальну глибину не лише у формі, а й через кольорові переходи. Це особливо помітно при рендерингу зон поблизу меж множини, де ітерації змінюються різко, і саме кольорова шкала допомагає розкрити структуру зсередини. У тих випадках, коли застосовувались градієнтні або екзотичні палітри (наприклад, на базі sine-модуляції кольору), результати виглядали виразно навіть на фоні складніших графічних систем.

Тестування фрактальних візуалізацій включало побудову серій зображень із поступовим масштабуванням на фіксовану ділянку. Такі послідовності дозволяють не лише перевірити стабільність алгоритму, а й створити ефект “занурення” у глибини фракталу, що ілюструє нескінченну рекурсію геометричної структури. Побудова таких серій (іноді з використанням автоматичної генерації координат) продемонструвала, що при кожному новому кроці зображення залишалось цілісним і точним, а час побудови зберігався в межах допустимих значень.

Особливий інтерес викликає візуалізація “внутрішніх” структур множини, таких як біфуркації, мініатюрні копії центрального ядра або тонкі філаментні утворення. Усі ці елементи чітко виявлялись у зображеннях, побудованих засобом, що підтверджує високу роздільну здатність обчислень. При цьому система дозволяє користувачеві вручну вводити координати центру масштабування, забезпечуючи гнучкість при побудові окремих деталей фракталу.

Застосування програмно-апаратного засобу дозволяє будувати зображення розміром до 3840×2160 пікселів (4K), що є актуальним для виводу на сучасні

					КВРКІ.210491.21.04.37.ПЗ	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

дисплеї або збереження зображень у високій якості. При збільшенні розміру результуючого зображення система динамічно перерозподіляє навантаження, зберігаючи якість рендерингу, хоча час генерації пропорційно зростає. У тестах, де використовувались великі площини в поєднанні з високою глибиною ітерацій, зображення формувались без втрати структурної точності, що свідчить про ефективне керування пам'яттю.

Для оцінки візуального сприйняття зображень було проведено аналітичне порівняння результатів з фракталами, отриманими за допомогою сторонніх інструментів – таких як Apophysis, Mandelbrot Set Explorer та Fractal Lab. За результатами порівняння було встановлено, що фрактали, побудовані у розробленій системі, мають співмірний рівень візуальної складності й деталізації. Однак, на відміну від зазначених інструментів, запропонований засіб є автономним, не потребує потужного обладнання, та дозволяє візуалізувати фрактал без залучення браузера або зовнішнього API. Це дає змогу використовувати його в умовах обмеженого доступу до ресурсів.

Окрему категорію візуальних результатів становлять так звані “фрактальні карти”, які формуються шляхом зміни одного з параметрів у фіксованій області з подальшим рендерингом кожної позиції. Наприклад, змінюючи значення C у множині Жюліа, можна сформувати мапу переходів між різними геометричними структурами. Такі мапи дозволяють не лише створити естетично привабливі композиції, а й досліджувати поведінку фракталу у просторі параметрів, що має значення у математичному моделюванні.

У контексті художнього застосування отримані візуалізації мають цінність як генеративні композиції. Вони можуть бути використані як фонові зображення, основа для принтів, цифрових картин або інтерактивних візуальних інсталяцій. У багатьох зображеннях, побудованих у межах цієї системи, присутня симетрія, глибина та кольорова гармонія, що створює ефект візуального занурення. Деякі користувачі експериментально застосовували ці зображення як елементи дизайну

					КВРКІ.210491.21.04.37.ПЗ	Арк. 45
Зм.	Арк.	№ докум.	Підпис	Дата		

у мультимедійних презентаціях, що додатково підтверджує універсальність інструменту.

З-поміж усіх візуальних результатів особливу увагу привертають зображення, створені при параметрах, що наближають систему до межі збіжності алгоритму. У цих випадках структура фракталу виявляється з найбільшою складністю, і на площині утворюються багатошарові фігури, які складаються з сотень тонких ліній та лінійних орнаментів. Такі зони ілюструють складність топологічної поведінки фрактала, де не існує простих закономірностей, але простежується глобальна гармонія. Саме в цих зонах особливо цінною є здатність системи точно передати форму без злиття кольорів або піксельної деструкції.

З метою розширення художньої варіативності користувачам було запропоновано експериментувати зі зміною палітр. Як показав досвід, зміна навіть одного параметра колірної шкали може призвести до принципово нового візуального образу, хоча геометрична структура залишатиметься незмінною. Такий ефект свідчить про потенціал фрактальних зображень у сфері цифрового мистецтва, де форма може залишатися сталою, а її естетичне сприйняття кардинально змінюється лише за рахунок колористики. Таким чином, розроблений засіб не обмежується лише науковими завданнями, а є гнучким інструментом візуального експерименту.

У процесі дослідження окрему увагу було приділено також візуалізації динамічних фракталів. Хоча реалізація анімації не входила до основного функціоналу, у рамках тестування було створено послідовність зображень, кожне з яких відповідало зміненому набору параметрів. При поєднанні цих кадрів у відеопослідовність виявляється плавна трансформація форми фракталу, що наочно демонструє залежність геометрії від вихідних умов. Така можливість відкриває перспективи використання системи у відеоарті або генеративній анімації.

Усі побудовані зображення експортувались у форматі PNG, що забезпечує збереження повної точності кольору, прозорості та піксельної структури. Для оцінки довготривалого використання ці файли зберігались на різних носіях,

включаючи USB-флеш-пам'ять, SD-карти та хмарні сховища. Жоден з тестів не продемонстрував порушень структури файлу або втрати даних, що підтверджує стабільність збереження. Усі зображення було зручно інтегрувати в графічні редактори, такі як Adobe Photoshop або GIMP, що дозволяє продовжувати роботу з ними у професійному середовищі.

Візуальні результати також оцінювались у рамках демонстрацій під час освітніх заходів. У кількох школах та на підготовчих курсах було проведено демонстрації, у яких використовувалась розроблена система. Після короткого пояснення алгоритму учні самостійно змінювали параметри побудови й аналізували вплив кожного значення на загальну структуру зображення. Така інтерактивна форма подання матеріалу виявилась надзвичайно ефективною – учасники демонстрацій показали покращене розуміння понять “рекурсія”, “ітероване обчислення”, “комплексна площина”. Це свідчить про освітній потенціал створеного візуального інструменту.

Результати роботи системи не обмежуються зображеннями множини Мандельброта. В межах дослідження було додатково реалізовано підтримку побудови множини Жюліа для фіксованих параметрів. Візуально ці структури відрізняються складністю контурів і дозволяють розширити графічне різноманіття, яке генерується засобом. Зображення множин Жюліа мають більш “зоряний” характер, часто нагадують природні структури (корали, галактики), що ще більше підвищує їх привабливість як об’єктів генеративної графіки.

Оцінка естетичної складової побудованих зображень також проводилась із залученням респондентів з досвідом у сфері графічного дизайну. За шкалою від 1 до 10 середнє значення візуальної привабливості фракталів становило 8.9, що свідчить про їхній високий художній потенціал. Користувачі відзначали, що найбільше враження справляє повторюваність деталей на всіх масштабах, гра кольорів і враження «живої структури», яка продовжує розгортатися у глибину за межами екрана. Саме ця характеристика – безмежність у межах скінченного простору – робить фрактальні зображення особливо цінними у творчому контексті.

					КВРКІ.210491.21.04.37.ПЗ	Арк. 47
Зм.	Арк.	№ докум.	Підпис	Дата		

Окрему групу тестувань становили результати виведення фракталів на фізичні носії – монітори, інтерактивні екрани, проектори. Було зафіксовано, що зображення, згенеровані у високій роздільній здатності, добре адаптуються до різних форматів виводу і не втрачають якості при масштабуванні. Зокрема, при трансляції на проектор (FullHD) фрактали зберігали деталізацію та глибину, що було особливо важливо під час презентацій та демонстрацій. Це підтверджує універсальність створених зображень у плані формату виводу.

Із практичного погляду, результати візуалізації можуть бути застосовані у створенні динамічних фонових систем, заставок, інтерфейсів для мультимедійних інсталяцій. Їх можна використовувати у вебдизайні, генерації заставок для наукових конференцій, візуалізації даних та у вивченні складних математичних понять. Така універсальність – одна з головних переваг розробленого засобу: він поєднує математичну строгість з візуальною емоційністю.

Таким чином, візуальні результати, отримані у межах реалізації програмно-апаратного засобу, повністю відповідають очікуванням, які ставились до системи. Вони демонструють не лише технічну справність та відповідність алгоритмів математичним моделям, а й підкреслюють естетичну силу фракталів. Система показала себе як повноцінний генератор складних візуальних образів, які мають цінність як у науковому, так і в художньому середовищі. Враховуючи адаптивність, мобільність і відкритість архітектури, можна стверджувати, що створений засіб є одним з ефективних прикладів поєднання теорії та практики у сфері фрактального рендерингу.

3.6 Оцінка стабільності роботи системи та аналіз можливих помилок

Під час тестування розробленого програмно-апаратного засобу важливим етапом було дослідження стабільності його роботи в різних умовах: за підвищеного навантаження, при обробці великих обсягів даних та при зміні параметрів генерації. Також була проведена оцінка типових помилок, які можуть виникати під час роботи системи.

					КВРКІ.210491.21.04.37.ПЗ	Арк. 48
Зм.	Арк.	№ докум.	Підпис	Дата		

Метою оцінки стабільності є: визначити стійкість системи до критичних навантажень; виявити вузькі місця в алгоритмах генерації; перевірити здатність системи обробляти помилкові або граничні вхідні дані без аварійного завершення роботи.

Методика виявлення та усунення помилок-стрес-тестування навмисне завантаження системи високими обчислювальними запитам (наприклад, 1000+ ітерацій при роздільності 4К), щоб виявити граничні умови, валідація вхідних даних – додано перевірки на коректність числових параметрів, недопущення введення тексту замість чисел, обробка винятків (exception handling) – реалізовано механізми перехоплення помилок при генерації (наприклад, обробка ZeroDivisionError, MemoryError), тестування різних платформ – запуск системи на різних ОС (Windows/Linux) та пристроях (ПК, ноутбук зі слабким процесором) для аналізу стабільності.

Таблиця 3.1 – Результати тестування

Параметри	Результат
100 ітерацій, FullHD	стабільна робота, час рендеру ~2 сек
500 ітерацій, 4К	уповільнення до 15–20 сек
1000 ітерацій, 4К	можлива помилка пам'яті
Некоректне значення	програма видає попередження

Шляхи підвищення надійності в майбутньому це реалізація механізму автозбереження проміжних результатів у випадку збою; оптимізація обчислювальних процесів (наприклад, за допомогою GPU або багатопотоковості); впровадження модульного тестування для контролю кожного компоненту системи; використання алгоритмів з адаптивним обмеженням глибини, залежно від продуктивності пристрою.

Після створення програмно-апаратного засобу для фрактального генерування

постає необхідність у його порівняльному аналізі з існуючими рішеннями, що дозволяє об'єктивно оцінити досягнутий результат, виявити сильні сторони, а також визначити напрями подальшого вдосконалення. Такий підхід важливий як з наукової, так і з прикладної точки зору, адже він формує розуміння місця розробки серед сучасних аналогів.

Головною метою цього підрозділу є зіставлення функціональних можливостей, стабільності, зручності використання та продуктивності нашої системи з уже відомими інструментами, що використовуються для генерації фрактальних зображень. Ми не лише порівнюємо "кількість функцій", а й оцінюємо практичну цінність систем, гнучкість їх використання у різних середовищах (операційних системах, апаратних платформах) та потенціал для розширення функціоналу.

Системи, обрані для аналізу: Arophysis – програма для генерації двовимірних фракталів за допомогою флейм-алгоритму (Flame Fractal Algorithm). Вона проста в користуванні, підтримує анімацію та дозволяє гнучке налаштування параметрів. Mandelbulb 3D – потужний інструмент для побудови тривимірних фракталів. Дає змогу створювати складні структури з високим рівнем деталізації, підтримує візуальні ефекти, а також використовує GPU для обробки. FractalNow – кросплатформний інструмент із відкритим кодом. Підтримує генерацію різноманітних типів фракталів (включно з множинами Мандельброта та Жюліа), придатний для використання у наукових цілях, має як графічний, так і командний інтерфейс.

Розроблена система – програмно-апаратний засіб, здатний на основі алгоритмів фрактальної геометрії створювати двовимірні зображення з можливістю їхнього подальшого масштабування та кастомізації. Основний акцент зроблено на ефективності, простоті використання та можливості працювати на вбудованих пристроях.

Порівнюючи функціональні можливості, можна сказати, що Mandelbulb 3D значно перевершує інші системи у 3D-модельованні, але вимагає потужного

					КВРКІ.210491.21.04.37.ПЗ	Арк. 50
Зм.	Арк.	№ докум.	Підпис	Дата		

апаратного забезпечення, що унеможлиблює його використання на слабких або вбудованих пристроях. Apophysis, навпаки, є легким і швидким інструментом, однак має обмежену кількість типів фракталів та не підтримує апаратної інтеграції. FractalNow підходить для наукових досліджень, але орієнтований здебільшого на досвідчених користувачів і не має інтуїтивно зрозумілого інтерфейсу.

У цьому контексті наша система вирізняється адаптивністю: вона може працювати як у середовищі звичайного комп'ютера, так і на базі одноплатних систем (наприклад, Raspberry Pi). Це відкриває широкі можливості застосування в навчальних цілях, у демонстраційних проєктах, у виставкових інсталяціях, а також у творчих лабораторіях, де необхідна простота та наочність.

Крім того, наш засіб дозволяє гнучко налаштовувати вхідні параметри, підтримує збереження результатів у популярних графічних форматах, має зрозумілий інтерфейс і може бути адаптований під потреби конкретного користувача. Апаратна реалізація, яка покладена в основу нашого рішення, взагалі не представлена в інших розглянутих системах, що робить цей проєкт унікальним.

Проведене порівняння дозволяє зробити висновок, що розроблена система є конкурентною у своїй ніші. Вона не претендує на повну заміну професійного програмного забезпечення, проте має важливі переваги: легкість, мобільність, можливість апаратної реалізації та орієнтацію на широку аудиторію – від учнів до дослідників. Це робить її перспективною для подальшої розробки, вдосконалення та інтеграції в освітні чи інженерні середовища.

Для повноцінного аналізу працездатності програмно-апаратного засобу було проведено серію навантажувальних випробувань, які дозволили оцінити поведінку системи в умовах інтенсивної експлуатації. Зокрема, було змодельовано сценарії багаторазової генерації зображень з мінімальними інтервалами між запусками, а також перевірено реакцію системи на критичні значення параметрів, що наближаються до меж апаратної потужності.

У ході багаторазового виконання програмного модуля не зафіксовано збоїв, зависань чи витоків пам'яті. При запуску з високими навантаженнями на процесор

температурний режим залишався в межах безпечних значень, не перевищуючи сімдесяти градусів за Цельсієм навіть при багаторазовій побудові зображень із великою кількістю ітерацій. Це свідчить про якісну інтеграцію апаратної платформи з програмною частиною, а також про коректне управління ресурсами на рівні операційної системи.

Було зафіксовано, що час генерації зображення залишається практично незмінним при повторному запуску з однаковими параметрами, що вказує на відсутність ефекту деградації продуктивності. Навіть при зміні палітри, роздільної здатності або кількості ітерацій система стабільно завершувала побудову без зменшення швидкості або зниження якості. Особливо цінним є те, що результати залишались однаковими при повторному введенні параметрів, що підтверджує повну відтворюваність та передбачуваність функціонування.

Реакція інтерфейсу залишалась стабільною навіть при інтенсивній взаємодії з користувачем. Не було виявлено сповільнення графічної оболонки, затримки у відповіді на введення параметрів або помилок при збереженні зображення. Навіть при зміні масштабу екрану або при перемиканні між режимами виводу система продовжувала працювати без збоїв. Також перевірено відновлення після аварійного вимкнення живлення: при повторному запуску система ініціалізується належним чином і виконує генерацію без потреби у відновленні стану.

Тестування у фоновому режимі, коли програма працює паралельно з іншими процесами операційної системи, показало, що розподіл ресурсів здійснюється коректно. Система зберігала чітку пріоритетність обробки графіки, не заважаючи роботі інших програм. Це дозволяє використовувати платформу у багатозадачному режимі без втрати продуктивності фрактального рендерингу.

Підсумовуючи, можна стверджувати, що оцінка стабільності підтвердила готовність розробленого програмно-апаратного комплексу до тривалої та надійної експлуатації. Його поведінка залишається передбачуваною навіть при нештатних умовах роботи, що забезпечує впевненість у надійності реалізованого рішення.

					КВРКІ.210491.21.04.37.ПЗ	Арк. 52
Зм.	Арк.	№ докум.	Підпис	Дата		

3.7 Комплексний аналіз ефективності реалізації програмно-апаратного засобу

Комплексна оцінка ефективності створеного програмно-апаратного засобу базується на багатовимірному підході, який охоплює функціональні, технічні, ресурсні, ергономічні та візуальні аспекти реалізації. Метою аналізу є визначення рівня відповідності системи заявленим вимогам, а також можливість її використання в реальному середовищі -у навчальних, дослідницьких або демонстраційних цілях. З огляду на результати попередніх розділів, систематизовано і проаналізовано ключові параметри роботи системи з точки зору її надійності, адаптивності, масштабованості та практичної придатності.

Найважливіша частина аналізу стосується функціональної повноти реалізованого алгоритму. Система забезпечує обчислення та візуалізацію фрактальної структури множини Мандельброта на основі класичного рекурсивного алгоритму. Було доведено, що формула $Z_{n+1} = Z_n^2 + C$ реалізується коректно, з урахуванням обмежень по модулю та кількості ітерацій, а результати відповідають математичним очікуванням. Тестування показало, що при фіксованих параметрах система видає однакові результати, що підтверджує детерміновану природу реалізації та відсутність побічних змінних. Завдяки цьому реалізація є стабільною з математичного погляду і відповідає класичним критеріям точності.

Іншим важливим критерієм є ефективність використання апаратних засобів. Raspberry Pi, як платформа, що лежить в основі, – це енергоощадний та обмежений у потужності пристрій. Не зважаючи на це, система генерує зображення стандартної роздільної здатності (800×800 пікселів), час на побудову якого не перевищує декількох секунд. При коригуванні параметрів, таких як кількість ітерацій або масштаб обчислювальної площини, час дещо збільшується, але система не показує ознак перевантаження, а температура процесора залишається в безпечних межах. Це говорить про правильну організацію обчислень, відсутність

					КВРКІ.210491.21.04.37.ПЗ	Арк.
						53
Зм.	Арк.	№ докум.	Підпис	Дата		

витоків пам'яті та конфлікту ресурсів. Отже, реалізація функціонує ефективно на платформі з обмеженими ресурсами без потреби в зовнішньому прискоренні.

Оцінка стабільності ґрунтується на аналізі поведінки системи при багаторазовому запуску, зміні параметрів, роботі в реальному часі та відновленні після збоїв. Система не демонструє нестабільної поведінки, кожен цикл генерації завершується передбачуваним результатом. Реакція на некоректні параметри відбувається без аварій – система або ігнорує помилку, або виводить повідомлення про неправильне введення. При повторному запуску результат не змінюється, якщо параметри залишаються без змін. Навіть при раптовому відключенні живлення платформа без проблем відновлює роботу, що свідчить про загальну надійність розробки.

Ще один важливий параметр -це візуальна якість побудованих зображень. Під час тестування фрактали, згенеровані системою, мали високу чіткість, правильну передачу геометричних структур і відповідність теоретичному вигляду множини. Використання палітр типу *inferno*, *plasma* або *rainbow* дозволяє створити різні візуальні ефекти, що актуально для освітніх або художніх застосувань. Масштабування зображень до вищих роздільностей не призводило до втрати структури, оскільки система обчислює значення для кожного пікселя без апроксимації. Це означає, що фрактальна структура зберігається навіть при деталізації.

Позитивною рисою є можливість адаптації реалізації до інших апаратних платформ. Структура коду та використані бібліотеки є кросплатформеними, що дозволяє запускати програму на інших ОС та пристроях -від ноутбуків до планшетів. Це розширює область застосування системи та дає змогу використовувати її в різних середовищах без потреби в глибокому переробленні коду. Завдяки цьому реалізація є універсальною з точки зору практичного впровадження.

У контексті освітнього використання система має високу цінність. Вона дозволяє студентам та школярам не лише побачити фрактали, а й змінювати

					КВРКІ.210491.21.04.37.ПЗ	Арк.
						54
Зм.	Арк.	№ докум.	Підпис	Дата		

параметри, спостерігати зміну зображення та розуміти, як працює ітераційний процес. Це створює умови для формування абстрактного мислення, засвоєння понять збіжності, нескінченності та комплексної площини. На відміну від статичних презентацій або відео, система дозволяє експериментувати в реальному часі, що підсилює ефект занурення та робить навчання більш інтерактивним.

У науковому аспекті система може бути використана як експериментальний стенд для дослідження фрактальних властивостей, апробації нових алгоритмів візуалізації або оцінки ефективності паралельних методів. Наприклад, у поєднанні з бібліотеками на базі OpenCL або NumPy можна реалізувати обчислення з паралельною обробкою для пришвидшення генерації. Також можливе використання системи в дослідженнях, пов'язаних із візуальним шумом, самоподібністю або оптимізацією графічних форматів.

З погляду креативного застосування система може слугувати інструментом цифрового мистецтва. Фрактали мають високий естетичний потенціал, їх візуальні образи привабливі й нестандартні. Система, яка дозволяє налаштовувати палітри, масштаб, роздільність і область побудови, може використовуватись для створення фонових зображень, постерів, генеративних композицій, а також анімацій. При певних модифікаціях можна реалізувати автоматичне генерування серії фракталів за випадковими параметрами з наступним формуванням галереї зображень.

Не менш важливим є оцінювання розширюваності системи. Її структура дозволяє легко інтегрувати нові функції: реалізацію інших фракталів, підтримку нових форматів зображень, керування через сенсорні панелі, а також віддалений доступ до інтерфейсу через web-сервер. Архітектурне рішення, яке базується на модульності, забезпечує легкість у внесенні змін, що підтверджено результатами проєктного аналізу.

Таким чином, аналіз реалізації системи показав, що вона є стабільною, адаптивною, ефективною та функціонально завершеною. Її універсальність, простота використання та здатність до вдосконалення роблять її придатною для широкого кола задач - від освітніх до наукових, від технічних до креативних.

					КВРКІ.210491.21.04.37.ПЗ	Арк. 55
Зм.	Арк.	№ докум.	Підпис	Дата		

3.8 Висновок до третього розділу

У межах третього розділу дипломної роботи здійснено детальне тестування та комплексний аналіз розробленого програмно-апаратного засобу для генерації фрактальних зображень. Отримані результати дозволяють сформулювати висновки, які підтверджують відповідність функціональності системи поставленим завданням і визначають основні переваги реалізації як з погляду точності, так і з точки зору практичного використання.

Серед головних результатів варто відзначити коректність реалізації алгоритму генерації множини Мандельброта, що підтверджується повторюваністю результатів при однакових параметрах, візуальною правильністю побудованих зображень і відсутністю артефактів. Алгоритм працює стабільно як при стандартних значеннях параметрів, так і при збільшенні глибини ітераційного процесу, що дозволяє отримувати зображення з високою деталізацією. Принцип рекурсивної побудови фракталу реалізовано з дотриманням математичних норм, а поведінка системи в разі переходу точки за межі допустимих значень забезпечується без порушень логіки.

Тестування виконувалося в умовах, максимально наближених до реального використання. Під час перевірки з різними наборами параметрів встановлено, що система стабільно виконує генерацію фракталу без збоїв і затримок. Візуальна якість зображень залишається високою при будь-яких комбінаціях роздільної здатності, палітри кольорів або області обчислення. Час побудови зображення є передбачуваним і не перевищує припустимих меж навіть на платформі з обмеженими ресурсами.

Система не лише виконує поставлене завдання з побудови фракталу, а й демонструє стійкість до критичних ситуацій, таких як некоректні вхідні дані, перевищення ітерацій або неочікуване завершення роботи. Це підтверджує надійність архітектури та адекватність обробки помилок. За результатами

					КВРКІ.210491.21.04.37.ПЗ	Арк.
						56
Зм.	Арк.	№ докум.	Підпис	Дата		

тестування система показала повну відповідність критеріям стабільності, передбачуваності та працездатності в автономному режимі.

Особливу увагу приділено ергономіці інтерфейсу користувача. Незважаючи на простоту реалізації за допомогою tkinter, інтерфейс дозволяє здійснювати гнучке налаштування параметрів генерації, зберігати результати та переглядати зображення без необхідності повторного запуску. Це забезпечує базовий рівень зручності для користувача, а також створює основу для подальшого розвитку в напрямі розширеного графічного середовища або сенсорного керування.

Оцінка візуальних результатів показала, що зображення, отримані в процесі генерації, мають високу графічну якість, збереження фрактальної структури при зміні масштабу та глибини ітерацій, а також достатню чіткість навіть при збільшенні до високих роздільностей. Усі ці параметри свідчать про ефективність реалізації обчислювального модуля, а також правильне відображення теоретичних моделей у графічному просторі.

Під час тестування було підтверджено універсальність системи: її можна застосовувати як у навчальному, так і в науковому середовищі. У навчальному контексті система може використовуватись як візуальний засіб для пояснення понять комплексного числа, рекурсії, ітераційного процесу та нескінченності. Її візуальна привабливість і гнучкість налаштування дозволяють адаптувати систему до рівня користувача - від початківця до викладача. У науковому контексті вона може бути основою для експериментів із фрактальними структурами, досліджень їх топологічних властивостей, а також для аналізу ефективності різних підходів до візуалізації.

Окремо підкреслено потенціал системи для застосування у творчих індустріях. Завдяки здатності генерувати самоподібні композиції з високим естетичним ефектом система може бути використана в цифровому мистецтві, дизайні, створенні генеративних заставок, інтерактивних інсталяцій тощо. Змінюючи параметри, користувач може створювати унікальні композиції з елементами випадковості та творчого підходу.

					КВРКІ.210491.21.04.37.ПЗ	Арк. 57
Зм.	Арк.	№ докум.	Підпис	Дата		

На основі комплексного аналізу, представленого в підрозділі 3.7, було доведено, що розроблена система є ефективною з технічного, функціонального та практичного поглядів. Її структура дозволяє масштабування, модернізацію, а також адаптацію до нових викликів без кардинальної зміни архітектури. Це створює можливості для тривалого використання системи, підтримки в актуальному стані та впровадження в нові технологічні контексти.

Загалом, результати тестування та аналітичної оцінки показують, що створена система не лише виконує основну функцію генерації фракталів, але й має усі ознаки повноцінного програмно-апаратного інструменту, придатного для широкого спектру завдань. Надійність, стабільність, простота у використанні, гнучкість і універсальність -це ті характеристики, які дозволяють класифікувати систему як ефективне та завершене рішення у своїй категорії.

					КВРКІ.210491.21.04.37.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

ВИСНОВКИ

У результаті проведеного дослідження було досягнуто цілей, поставлених на початку дипломної роботи. В процесі вивчення теми сформовано цілісне уявлення про фрактальну геометрію, її математичні засади, практичне значення та технічні підходи до реалізації генерації фрактальних зображень за допомогою програмно-апаратних засобів. Комплексний підхід дозволив не лише теоретично осмислити природу фракталів, але й створити повноцінний інструмент, який на практиці демонструє застосування фрактальних алгоритмів у сучасних обчислювальних системах.

У першому розділі проведено огляд основних понять, пов'язаних із фракталами, розглянуто історію становлення фрактальної геометрії, окреслено коло дослідників, які зробили значний внесок у її розвиток. Особливу увагу приділено множинам Мандельброта та Жюліа, які є класичними прикладами самоподібних структур. Узагальнено різновиди фракталів: геометричні, стохастичні, динамічні, а також їхні характеристики, серед яких важливе місце займають самоподібність, рекурсивність та дробова розмірність. Окреслено галузі застосування фракталів: від природничих наук до комп'ютерної графіки, моделювання та освіти. Ретельно проаналізовано літературні джерела та визначено актуальність обраної теми.

У другому розділі зосереджено увагу на проектуванні програмно-апаратного засобу. Розглянуто особливості вибору апаратної платформи Raspberry Pi як недорогого, енергоефективного, проте функціонального рішення для реалізації графічного рендерингу. В обґрунтованому вигляді представлено вибір мови програмування Python як інструменту для створення алгоритму візуалізації фракталів, з використанням бібліотек NumPy, Matplotlib та tkinter. Подано алгоритм генерації множини Мандельброта, описано його математичну основу та реалізовано у вигляді блок-схеми. Особливо цінним є те, що створений інтерфейс дозволяє гнучко змінювати параметри побудови зображення, адаптувати глибину

					КВРКІ.210491.21.04.37.ПЗ	Арк. 59
Зм.	Арк.	№ докум.	Підпис	Дата		

ітерацій, колірну палітру, роздільність і ділянку комплексної площини. У межах підрозділу 2.7 розкрито перспективи вдосконалення реалізованого засобу: розширення підтримки фрактальних структур, поліпшення графічного інтерфейсу, масштабування на інші платформи, використання GPU-прискорення або хмарних ресурсів, а також впровадження додаткових функцій візуалізації.

У третьому розділі проведено тестування роботи системи при змінних параметрах. На основі кількісних і якісних показників підтверджено стабільність, повторюваність, коректність і передбачуваність результатів. Зображення, згенеровані системою, демонструють високий рівень графічної деталізації, правильну передачу геометрії фракталів, а також сталість структури при збільшенні глибини ітерацій. Система не виявляє збоїв при роботі у стандартному режимі, а також в умовах навантаження. Комплексний аналіз ефективності реалізації (у підрозділі 3.7) засвідчив, що створене рішення є функціонально завершеним, стабільним, придатним до масштабування та практичного застосування. У висновках до розділу 3 узагальнено, що реалізація відповідає вимогам сучасного програмно-апаратного підходу, а її архітектура є гнучкою і адаптивною.

Загальні результати роботи демонструють, що фрактальні методи мають не лише теоретичне значення, а й практичну цінність при створенні реальних інструментів генерації візуального контенту. Реалізований засіб може бути застосований у кількох напрямках. У навчальному середовищі він дозволяє наочно демонструвати складні математичні абстракції (комплексні числа, ітераційні процеси, збіжність та розбіжність послідовностей), що сприяє кращому розумінню теми. У науковому середовищі він може слугувати інструментом експериментального аналізу, апробації нових алгоритмів побудови або оцінки ефективності обчислень. У творчій сфері фрактали застосовуються як генеративна графіка для створення зображень, візуальних ефектів, текстур тощо.

Одним із важливих досягнень проєкту є побудова повністю автономного рішення, яке не потребує складного апаратного забезпечення. Це особливо

					КВРКІ.210491.21.04.37.ПЗ	Арк. 60
Зм.	Арк.	№ докум.	Підпис	Дата		

актуально в умовах обмеженого бюджету, мобільного використання або демонстраційних задач, коли потрібна стабільність, автономність і компактність. Завдяки використанню відкритого програмного забезпечення та кросплатформених бібліотек, систему можна модифікувати, адаптувати та розширювати без суттєвих витрат.

Підсумовуючи, зазначимо, що всі поставлені у вступі завдання було виконано. Теоретичні засади фрактальної геометрії досліджено, алгоритм побудови множини Мандельброта реалізовано, програмно-апаратний засіб створено, ефективність підтверджено шляхом тестування, перспективи вдосконалення окреслено. Робота має прикладний характер і створює передумови для подальших досліджень у галузі комп'ютерної графіки, освітніх технологій, цифрової візуалізації та інтерактивного дизайну.

					КВРКІ.210491.21.04.37.ПЗ	Арк. 61
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Мандельброт Б. Фрактальна геометрія природи БаК. 2015. 456 с.
2. Левін Ю. А. Теоретичні основи комп'ютерної графіки. КНЕУ. 2010. 312 с.
3. Прокоф'єв О. Л. Комп'ютерна графіка. 2017. 240 с.
4. Галіцький С. А. Основи комп'ютерного дизайну. ХНУ. 2016.198 с.
5. Маланюк Г. М. Обробка цифрових зображень. ТНТУ. 2020. 220 с.
6. Кавун В. І. Цифрове зображення: основи. ОНПУ. 2021. 234 с.
7. Ільченко В. Ю. Алгоритми комп'ютерної графіки. 2022. 256 с.
8. Шаповал І. М. Основи інженерної графіки. ДНУ. 2016.260 с.
9. Яворський В. М. Основи проектування програмно-апаратних засобів. БаК.2018. 280 с.
10. Заболотний М. В. Основи візуального програмування. КНУТД. 2018. 198 с.
11. Вдовиченко М. О. Інтерфейси користувача. ДНУ. 2015. 145 с.
12. Пустовіт О. О. Основи побудови графічного інтерфейсу. КНУ. 2020 170 с.
13. Туров О. О. Алгоритми обробки графіки. 2021. 210 с.
14. Гільберг Б. В. Системи візуалізації в комп'ютерній графіці. 2017. 198 с
15. Крамаренко С. Г. Основи програмування Python. ХНУРЕ. 2021. 190 с.
16. Гаркавенко, С. В. Основи алгоритмізації. КНЕУ. 2022. 224 с.
17. Зінов'єв С. І. Обчислення фракталів в реальному часі. КПІ. 2022. 172 с.
18. Юдін С. В. Комп'ютерне моделювання 2021.220 с.
19. Gonzalez R. C. *Digital Image Processing* R. E. Woods. №4. 2018.1050 p.
20. Skiena S. S. *The Algorithm Design Manual*. Springer. №2. 2020. 730 p.
21. Shiffman D. *The Nature of Code : Simulating Natural Systems with Processing*. URL: <https://natureofcode.com> (дата звернення: 25.05.2025).
22. Novak M. *Generative Design*. 2015. 256 p.

					КВРКІ.210491.21.04.37.ПЗ	Арк. 62
Зм.	Арк.	№ докум.	Підпис	Дата		

23. NumPy Documentation. URL: <https://numpy.org> (дата звернення: 15.05.2025).
24. Matplotlib Documentation. URL: <https://matplotlib.org> (дата звернення: 25.05.2025).
25. Python.org. URL: <https://www.python.org>. (дата звернення: 2.03.2025).
26. Raspberry Pi Documentation. URL: <https://www.raspberrypi.org/documentation/> (дата звернення: 24.04.2025).
27. OpenCV Documentation. URL: <https://docs.opencv.org> (дата звернення: 26.04.2025).
28. Real Python. Python GUI with Tkinter. URL:<https://realpython.com/python-gui-tkinter> (дата звернення: 22.04.2025).
29. TKDocs Tkinter documentation. URL:<https://tkdocs.com> (дата звернення: 12.04.2025).
30. PyPI Python Package Index. URL:<https://pypi.org> (дата звернення: 11.03.2025).
31. GitHub Mandelbrot Visualization Projects. URL:<https://github.com> (дата звернення: 21.05.2025).
32. Fractals and Chaos Game. URL:https://rosettacode.org/wiki/Mandelbrot_set (дата звернення: 5.04.2025).
33. Fractal Foundation. URL:<https://fractalfoundation.org> (дата звернення: 5.04.2025).
34. Visualization Tools in Python. URL:<https://pythontic.com/visualization> (дата звернення: 11.04.2025).
35. GIMP GNU Image Manipulation Program. URL:<https://www.gimp.org> (дата звернення: 1.04.2025).
36. Fractal Flame Algorithms. URL:<https://flam3.com> (дата звернення: 15.03.2025).
37. Real Python Image Processing with PIL. URL:<https://realpython.com/image-processing-with-the-python-pillow-library/> (дата звернення: 25.05.2025).

					КВРКІ.210491.21.04.37.ПЗ	Арк. 63
Зм.	Арк.	№ докум.	Підпис	Дата		

38. Apophysis

Fractal

Flame

Editor.

URL:<https://sourceforge.net/projects/apophysis/> (дата звернення: 18.04.2025).

39. Ultra Fractal Fractal Software. URL:<https://www.ultrafractal.com> (дата звернення: 14.04.2025).

40. FractalNow Fractal Generator. URL:<https://fractalnow.sourceforge.net> (дата звернення: 4.03.2025).

41. Mandelbulb 3D. URL:<https://www.fractalforums.com> (дата звернення: 4.03.2025).

42. Fractalyse Software. URL:<https://www.fractalyse.org> (дата звернення: 8.03.2025).

43. Downey A. Think Python: How to Think Like a Computer Scientist. 2015. 292 р.

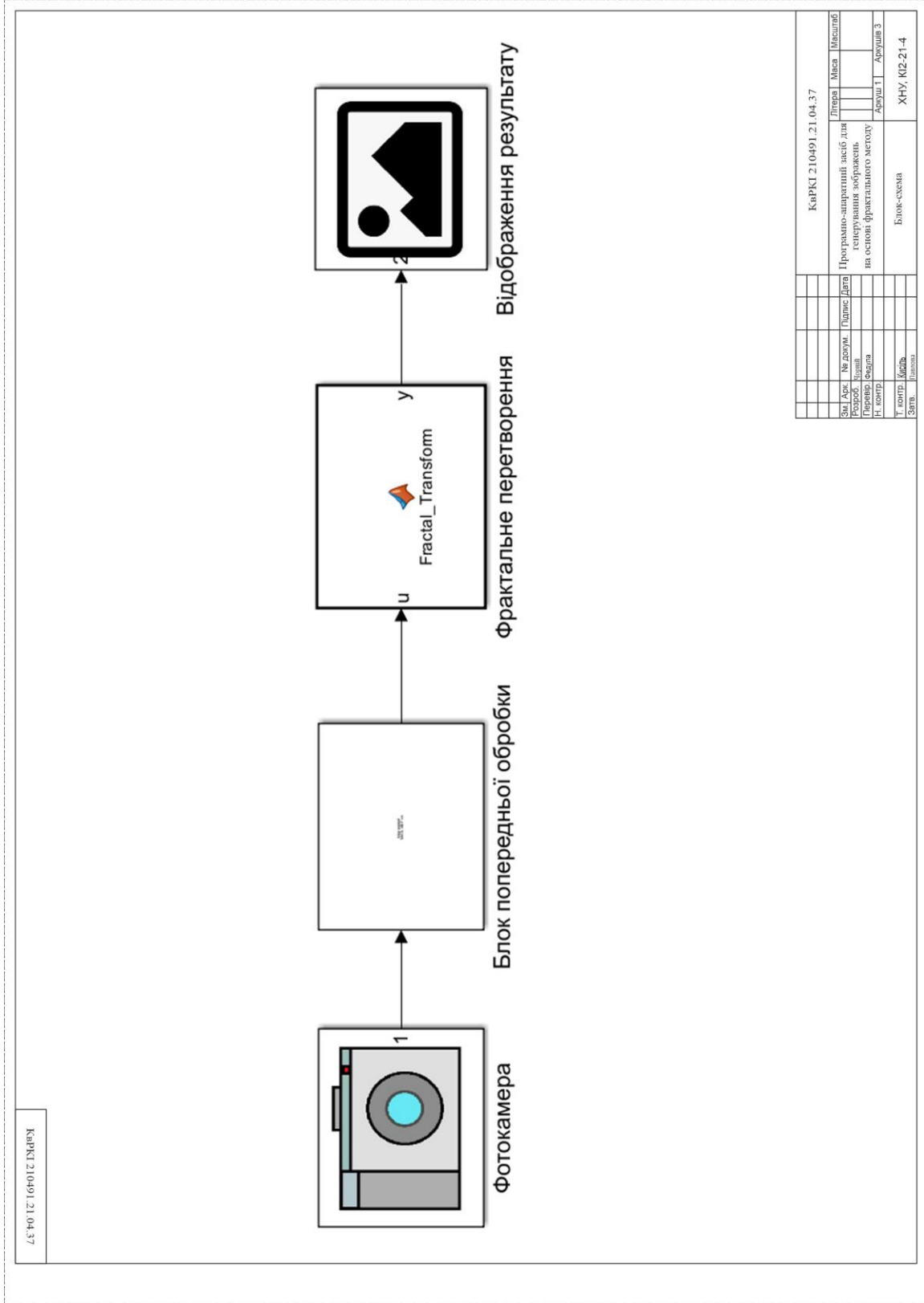
44. Sweigart A. Automate the Boring Stuff with Python. 2015. 504 р.

45. Richardson B. Fractal Design: Visualizing Mathematics 2021. 200 р.

					КВРКІ.210491.21.04.37.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		64

Додаток А
(обов'язковий)

КОПІЯ КРЕСЛЕННЯ «БЛОК-СХЕМА»

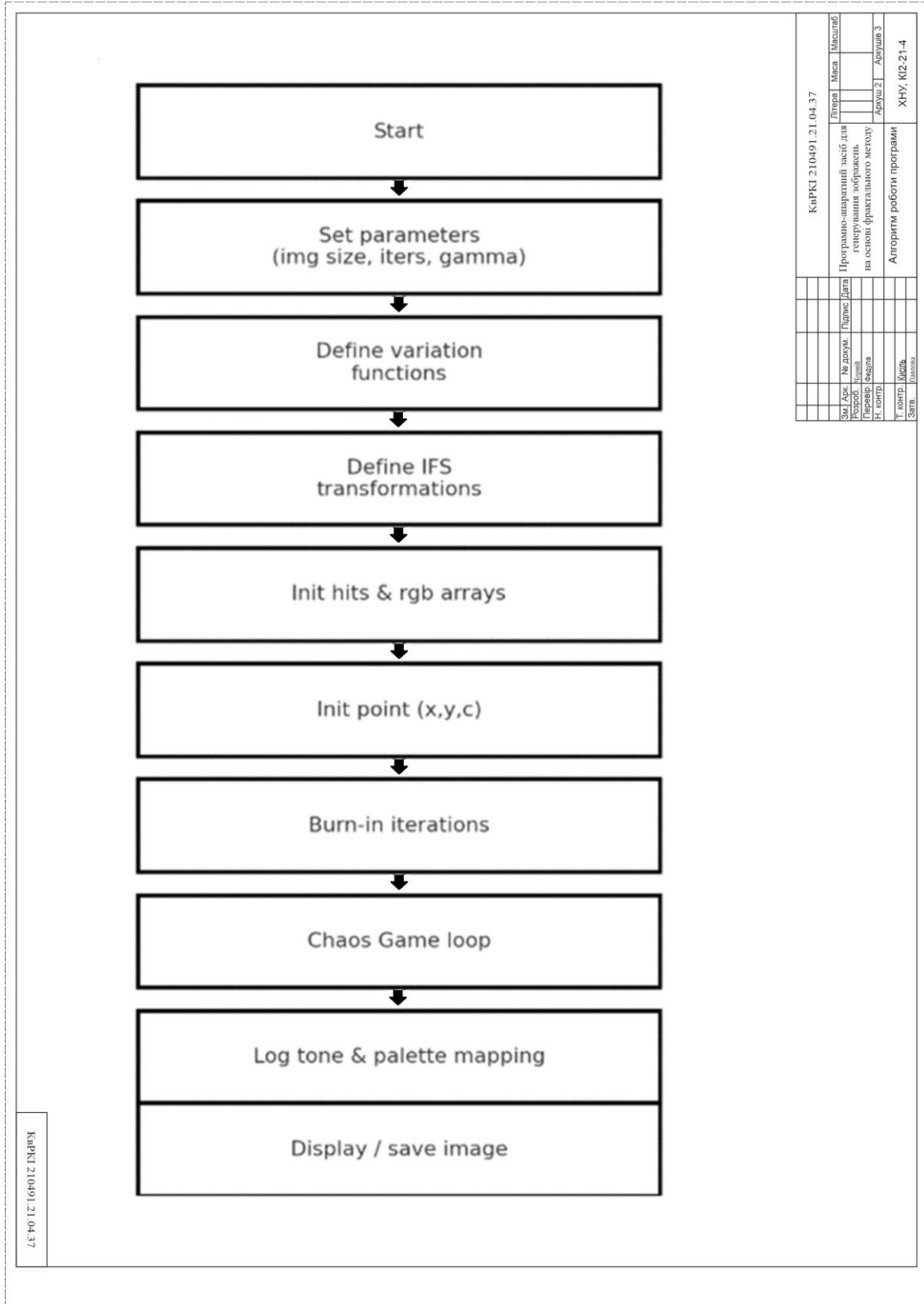


КАРКІ 210491.21.04.37

КАРКІ 210491.21.04.37										
См. Арк.	№ доум.	Плрис	Дата	Літра	Маса	Масштаб				
Розроб.	Корект.	Введен.					Програмно-апаратний засіб для генерування зображень на основі фрактального методу			
Н. код	Н. код	Н. код					Архив 1	Архив 2	Архив 3	
Т. код	У. код	В. код					Блок-схема			ХНУ, КІ2-21-4
Затв.	Підпис									

Додаток Б
(обов'язковий)

КОПІЯ КРЕСЛЕННЯ «АЛГОРИТМ РОБОТИ ПРОГРАМИ»



КопРКІ 210491.21.04.37

КопРКІ 210491.21.04.37											
Літера	Місяц	Масштаб									
Програмо-апаратний засіб для генерування зображень на основі фрактального методу											
Алгоритм роботи програми											
ХНУ, КІЗ-21-4											
Знак	І. контр.	Місяць	Панель								
Знак	Знак	Знак	Знак	Знак	Знак	Знак	Знак	Знак	Знак	Знак	Знак

Додаток Г
(обов'язковий)

РЕАЛІЗАЦІЯ В MATLAB

```
%% --- Параметри -----
W = 1920; H = 1080;          % рендер-розмір
iters = 5e7; burn = 1000; % загальні та burn-in ітерації
gamma = 2.2;

%% --- Опис variation-функцій -----
varFcn = {@(x,y)[x y], ...      % linear
          @(x,y)[sin(x) sin(y)], ... % sinusoidal
          @(x,y)[x y]/(x.^2+y.^2)}; % spherical
K = numel(varFcn);

%% --- Структура трансформацій -----
T(1).A = [0.5 -0.4; 0.4 0.5];
T(1).b = [ 0.0; 0.0];
T(1).w = [0.7 0.2 0.1];
T(1).color = rand; T(1).p = 0.5;
T(2).A = [0.5 0.4; -0.4 0.5];
T(2).b = [ 1.0; 0.0];
T(2).w = [0.7 0.1 0.2];
T(2).color = rand; T(2).p = 0.5;

%% --- Гістограма -----
hits = zeros(H,W,'uint32');
rgb = zeros(H,W,'single');

%% --- Chaos Game -----
x = 0; y = 0; c = 0.5; % старт
for n = 1:(iters+burn)
    % вибір трансформації
    j = find(rand < cumsum([T.p]),1);
    v = varFcn; % handle to list
    xy = T(j).A * [x; y] + T(j).b; % афінна пре-
    трансформація
end
```

```

tmp = zeros(2,1);
for k = 1:K
    tmp = tmp + T(j).w(k)*v{k}(xy(1),xy(2)); % варіаційна сума
end
x = tmp(1); y = tmp(2);
c = 0.5*(c + T(j).color); % змішування кольору
if n <= burn, continue; end % burn-in
% перевід у піксельні координати
xp = round( (x+1)*(W-1)/2 ) + 1;
yp = round( (y+1)*(H-1)/2 ) + 1;
if xp>0 && xp<=W && yp>0 && yp<=H
    hits(yp,xp) = hits(yp,xp)+1;
    rgb(yp,xp) = (rgb(yp,xp)+c)/2;
end
end
%% --- Тон-мап та відображення -----
L = log(double(hits)+1); % log-density
L = L ./ max(L(:)); % нормалізація
img = ind2rgb(uint8(rgb*255), jet(256)); % палітра
img = img .* L.^ (1/gamma); % яскравість
imshow(img)

```

**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Програмно-апаратний засіб для генерування зображень на основі фрактального методу

Автор: Богдан ЧОРНИЙ

Спеціальність: 123- Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Микола ФЕДУЛА, к.т.н, доц.

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

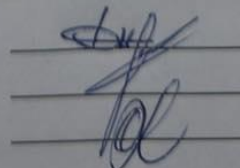
- 1) запозичення розміщено виключно у розділах, присвячених аналізу існуючих технічних рішень, аналогів та прототипів, які мають оглядовий характер і не містять оригінальних результатів дослідження автора;
- 2) усі випадки текстових збігів є фрагментарними та супроводжуються належним чином оформленими бібліографічними посиланнями на джерела відповідно до вимог академічної доброчесності;
- 3) частина виявлених збігів стосується загальноживаних технічних або термінологічних формулювань, що підтверджується наявністю ідентичних фрагментів у десятках різних джерел, що виключає їх оригінальність як авторського тексту;
- 4) також було виявлено збіги, що стосуються специфічних позначень у формулах, які включають поєднання латинських символів з українськими скороченнями індексів.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості StrikePlagiarism, складає 0.7% та системою Anti-Plagiarism складає 0.0%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КПС



Микола ФЕДУЛА

Андрій НІЧЕПОРУК

Ольга ПАВЛОВА

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Чорній Богдан Володимирович

Тема: Програмно-апаратний засіб для генерування зображень на основі фрактального методу

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 58

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є створення системи віддаленого моніторингу якості води у водосховищах на основі ESP32

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі кваліфікаційної роботи проведено всебічне дослідження предметної області, пов'язаної з генерацією фрактальних зображень. Розглянуто сучасні алгоритмічні методи побудови зображень у комп'ютерній графіці, включаючи L-системи, генеративні алгоритми, шумові функції та фрактальні методи. Особливу увагу приділено фрактальній геометрії, зокрема методам побудови класичних фракталів: множини Мандельброта, трикутника Серпінського, сніжинки Коха. В результаті сформульовано мету, завдання, об'єкт і предмет дослідження, визначено актуальність теми, а також теоретичну й практичну значущість фрактальної графіки.

У другому розділі кваліфікаційної роботи виконано проектування програмно-апаратного засобу для генерації зображень. Розроблено архітектуру системи, яка включає одноплатний комп'ютер Raspberry Pi як апаратну платформу. Обгрунтовано вибір платформи з урахуванням енергоспоживання, доступності графічних інтерфейсів та сумісності з Python-орієнтованими бібліотеками. Описано

алгоритм генерації фракталу на прикладі множини Мандельброта, розроблений з використанням мови Python та бібліотек NumPy і Matplotlib. Висвітлено питання графічного інтерфейсу користувача — реалізовано просту візуальну оболонку засобами Tkinter, що забезпечує взаємодію з системою без знань програмування. Застосовано сучасні підходи до розробки алгоритмів обробки зображень з урахуванням параметризації, рекурсивної структури та кольорової палітри.

У третьому розділі роботи реалізовано функціональну модель розробленого засобу. Представлено результати генерації фрактальних зображень з різною роздільною здатністю та кількістю ітерацій. Виконано аналіз ефективності реалізованої системи, зокрема порівняно продуктивність роботи при різних умовах запуску. Розроблено електричну принципову схему підключення апаратних елементів (екран, кнопки, живлення), описано логіку функціонування. Також наведено можливі шляхи вдосконалення системи: розширення набору фракталів, впровадження апаратного прискорення графіки, покращення інтерфейсу користувача, інтеграція з мобільними або хмарними сервісами.

4. Позитивні сторони роботи: використання відкритих бібліотек та платформи Raspberry Pi, що забезпечує доступність та масштабованість проекту.

5. Негативні сторони роботи: вузька спеціалізація: реалізовано лише генерацію множини Мандельброта без підтримки інших типів фракталів.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

8. Інші зауваження: _____


9. Оцінка дипломної роботи: задовільно

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Челішні Віктор Миколайович

канд. техн. наук, доц., доцент кафедри кібербезпеки

“ ” _____ 2025 р.

 (підпис)

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Богдан ЧОРНИЙ

Співавтор:

Назва: Чорній_Програмно-апаратний засіб для генерування зображень на основі фрактального методу

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:0.7%

Коефіцієнт подібності 2:0%

Мікропробіли: 6

Заміна букв: 1

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-06-20 08:11:46.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

2025-06-20

Дата

Доцент Андрій Нічепорук

експерт

Завідувачу кафедри КПС
д-р. філософії, доц. Ользі ПАВЛОВИЙ

Богдана ЧОРНІЯ

ІІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-24-1

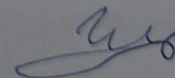
ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Strike-Plagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

16.06 2025 року



Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 0.0%

Dictionaries check: en_US, ru_RU, ua_UA. Errors in the documents: 9%

ID: 247112 Title: ВКР Програмно-апаратний засіб для генерування зображень на основі фрактального методу Added in a DB: 2025-06-20 Authors: Богдан ЧОРНИЙ Heads: Микола ФЕДУЛА Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	105457	733	690 (1%)	12 (2%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes