

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем
Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки

«До захисту в ЕК»

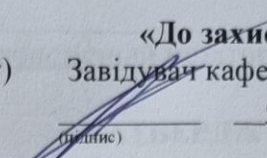
Директор інституту (декан факультету)


Андрій Форсюк
(ім'я та прізвище)

«12» лютого 2024р.

«До захисту допущено»

Завідувач кафедри


Сергій Грибков
(ім'я та прізвище)

«12» лютого 2024р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА

зі спеціальності 122 «Комп'ютерні науки»

(код і назва спеціальності)

освітньо-професійної програми Комп'ютерний еколого-економічний моніторинг
на тему: Дослідження та аналіз методів SDF-графіки в візуалізації результатів комп'ютерно-екологічного моніторинга

Виконав: здобувач 2 курсу, групи КМ-2-4М

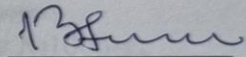
Жебрак Артем Миколайович

(прізвище, ім'я, по батькові повністю)


(підпис)

Керівник Литвинов Валерій Андроникович

(прізвище, ім'я та по батькові повністю)


(підпис)

Консультанти

(ім'я та прізвище)

(підпис)

(ім'я та прізвище)

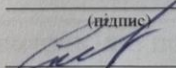
(підпис)

(ім'я та прізвище)

(підпис)

Рецензент

Ярослав Світлич
(ім'я та прізвище)


(підпис)

Я як здобувач(ка) Національного університету харчових технологій розумію і підтримую політику університету з академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) незарядженої допомоги під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач


(підпис)

Київ — 2024р.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем
Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки
Освітній ступінь магістр
Спеціальність 122 «Комп'ютерні науки»
(код і назва)
Освітньо-професійна програма Комп'ютерний еколого-економічний моніторинг
(назва)

ЗАТВЕРДЖУЮ

Завідувач

кафедри Інформаційних технологій,
штучного інтелекту і кібербезпеки

Грибков С. В.
«19» грудня 2023 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Жебрак Артем Миколайович

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та аналіз методів SDF-графіки в візуалізації
результатів комп'ютерно-екологічного моніторингу

керівник роботи Литвинов Валерій Андроникович, професор, доктор технічних
наук,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від 19 грудня 2023 року №1006-к

2. Строк подання здобувачем роботи: 25.01.2024

3. Вихідні дані до роботи:

1. Інформація про методи та технології, які застосовувалися під час дослідження та
розробки.

2. Приклад візуалізації за допомогою SDF-графіки.

3. Літературні джерела та інші джерела інформації, що були використані під час
дослідження.

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

1. Аналіз видів візуальних результатів комп'ютерно-екологічного моніторингу і
визначення області доцільного застосування методів SDF-
графіки.

2. Дослідження впливу рівня деталізації на сприйняття та аналіз результатів
моніторингу.

3. Оцінка ефективності застосування методів SDF-графіки та визначення
можливості і шляхів її оптимізації для підвищення ефективності сприйняття та
аналізу результатів




5. Перелік графічного матеріалу:

1. Поле знакових відстаней (SDF) 2. Ray Marching

3. Створення фрагментного шейдеру для SDF-графіки 4. Прикладу візуалізації результатів моніторингу атмосферного повітря за допомогою SDF-графіки

5. Фрагменти коду

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Литвинов В.А., професор	 02.12.23	03.12.23
2	Литвинов В.А., професор	 05.01.24	06.01.24
3	Литвинов В.А., професор	 16.01.24	17.01.24

7. Дата видачі завдання: 19 грудня 2023 року

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підготовка та узгодження теми роботи	15.10.23 – 20.10.23	Виконано
2	Пошук літератури та джерел інформації	24.10.23 – 30.10.23	Виконано
3	Написання 1 розділу	05.11.23 – 20.11.23	Виконано
4	Написання 2 розділу	21.12.23 – 10.01.24	Виконано
5	Написання 3 розділу	10.01.24 – 20.01.24	Виконано

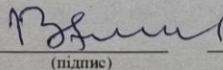
Здобувач


(підпис)

Жебрак А.М.

(прізвище та ініціали)

Керівник роботи


(підпис)

Литвинов В.А.

(прізвище та ініціали)

АНОТАЦІЯ

Дослідження та аналіз методів SDF-графіки в візуалізації результатів комп'ютерно-екологічного моніторингу є актуальною темою в сучасному науковому дослідженні. У даній роботі буде проведено аналіз методів SDF-графіки з метою їх застосування у візуалізації результатів комп'ютерно-екологічного моніторингу.

Робота буде складатися з аналізу та порівняння різних підходів до застосування SDF-графіки у візуалізації екологічних даних. На основі цього аналізу буде розроблено прототип системи візуалізації, який дозволить ефективно представляти та аналізувати дані з моніторингу довкілля.

Основною метою дослідження є розробка ефективного інструменту для візуалізації результатів комп'ютерно-екологічного моніторингу з використанням методів SDF-графіки. Досягнення цієї мети передбачає розробку інтерактивного інтерфейсу, що дозволить зручно взаємодіяти з візуалізованою інформацією та отримувати з неї корисні дані для аналізу.

Ключові аспекти дослідження включатимуть в себе аналіз ефективності, зручності використання та можливості розширення розробленого інструменту. Додатково, буде вивчено можливості оптимізації роботи з SDF-графікою для забезпечення швидкодії та відповідності вимогам реального часу.

Магістерська робота включає вступ, 3 розділи, висновок, список літератури та додатки. Обсяг основної частини становить 81 сторінок друкованого тексту. У роботі використано 29 ілюстрацій.

Ключові слова: SDF-ГРАФІКА, ВІЗУАЛІЗАЦІЯ ДАНИХ, КОМП'ЮТЕРНО-ЕКОЛОГІЧНИЙ МОНІТОРИНГ, ІНТЕРАКТИВНИЙ ІНТЕРФЕЙС.

ABSTRACT

The study and analysis of SDF-graphics methods in visualizing the results of computer-aided environmental monitoring is a relevant topic in modern scientific research. This paper will analyze the SDF-graphics methods for the purpose of their application in visualizing the results of computer-aided environmental monitoring.

The work will consist of analyzing and comparing different approaches to the use of SDF-graphics in the visualization of environmental data. Based on this analysis, a prototype visualization system will be developed that will allow for effective presentation and analysis of environmental monitoring data.

The main goal of the study is to develop an effective tool for visualizing the results of computer-based environmental monitoring using SDF-graphics. Achieving this goal involves the development of an interactive interface that will allow you to conveniently interact with the visualized information and obtain useful data for analysis.

The key aspects of the research will include an analysis of the efficiency, usability, and extensibility of the developed tool. Additionally, the possibilities of optimizing the work with SDF-graphics to ensure performance and compliance with real-time requirements will be studied.

The master's thesis includes an introduction, 3 chapters, a conclusion, a list of references and appendices. The volume of the main part is 81 pages of printed text. The work contains 29 illustrations.

Keywords: SDF-GRAPHICS, DATA VISUALIZATION, COMPUTER-AIDED ENVIRONMENTAL MONITORING, INTERACTIVE INTERFACE.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. АНАЛІЗ ВИДІВ ВІЗУАЛЬНИХ РЕЗУЛЬТАТІВ КОМП'ЮТЕРНО-ЕКОЛОГІЧНОГО МОНІТОРИНГУ І ВИЗНАЧЕННЯ ОБЛАСТІ ДОЦІЛЬНОГО ЗАСТОСУВАННЯ МЕТОДІВ SDF-ГРАФІКИ.....	10
1.1 Моніторинг довкілля	10
1.1.1 Види моніторингу	11
1.1.2 Способи відображення результатів моніторингу	13
1.2 SDF графіка.....	15
1.2.1 Поле знакових відстаней	16
1.2.2 Ray marching	21
1.2.3 Області доцільного застосування методів SDF-графіки.....	28
1.2.4 Постановка задач подальших досліджень.....	31
РОЗДІЛ 2. ДОСЛІДЖЕННЯ ВПЛИВУ РІВНЯ ДЕТАЛІЗАЦІЇ НА СПРИЙНЯТТЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ МОНІТОРИНГУ	34
2.1 Вплив рівня деталізації на сприйняття та аналіз результатів моніторингу.....	34
2.1.1 Фактори впливу візуальної деталізації результатів моніторингу .	35
2.1.2 Візуальна деталізації результатів моніторингу за допомогою SDF-графіки.....	36
2.2 Розробка та деталізація SDF-графіки для прикладів візуалізації	37
2.2.1 Створення фрагментного шейдеру для відображення SDF-графіки	38
2.2.2 Створення прикладу візуалізації результатів моніторингу атмосферного повітря за допомогою SDF-графіки	54
РОЗДІЛ 3. ОЦІНКА ЕФЕКТИВНОСТІ ЗАСТОСУВАННЯ МЕТОДІВ SDF-ГРАФІКИ ТА ВИЗНАЧЕННЯ МОЖЛИВОСТЕЙ І ШЛЯХІВ ЇХ ОПТИМІЗАЦІЇ ДЛЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ СПРИЙНЯТТЯ ТА АНАЛІЗУ РЕЗУЛЬТАТІВ.....	61
2.1 Оцінка ефективності SDF-графіки	62
3.1.1 Візуальна чіткість та зрозумілість.....	62
3.1.2 Динамічність та адаптивність	63
3.1.3 Взаємодія з користувачем	64
2.2 Оптимізація SDF-графіки.....	64
3.2.1 Алгоритмічні оптимізації.....	65

3.2.2 Оптимізація для мобільних платформ.....	66
3.2.3 Апаратне прискорення	67
2.3 Інтерактивність та користувацький досвід.....	68
3.3.1 Можливості взаємодії.....	69
3.3.2 Інтерфейс користувача	69
2.4 Адаптація до потреб користувача	70
3.4.1 Різні профілі користувачів	71
3.4.2 Багатовимірні дані	71
2.5 Можливості розширення та інтеграції.....	71
3.5.1 Розширення функціоналу	72
3.5.2 Інтеграція з геоданими	73
ВИСНОВКИ.....	74
БІБЛЮГРАФІЧНИЙ СПИСОК.....	75
ДОДАТОК А.....	76

ВСТУП

Актуальність теми.

У сучасних умовах розвитку галузі комп'ютерного еколого-економічного моніторингу, важливо досліджувати та вдосконалювати методи візуалізації результатів. Спеціалістам комп'ютерного еколого-економічного моніторингу, які володіють навичками програмування та аналізу економічних і екологічних процесів, важливо мати ефективні інструменти для візуалізації складних даних.

Зв'язок роботи з науковими програмами, планами, темами.

Дослідження відбувається в рамках програми комп'ютерного еколого-економічного моніторингу, спрямованої на вдосконалення технічних засобів та аналітичних методів цієї галузі.

Об'єкт дослідження.

Методи візуалізації результатів комп'ютерно-екологічного моніторингу.

Предмет дослідження.

Ефективність використання SDF-графіки для візуалізації різних аспектів екологічних даних.

Мета та задачі дослідження

Метою дослідження є визначення області застосування, можливостей та ефективності використання методів SDF-графіки у візуалізації результатів комп'ютерно-екологічного моніторингу.

Для цього поставлені наступні завдання:

- Проаналізувати види візуальних результатів комп'ютерно-екологічного моніторингу і визначити області доцільного застосування методів SDF-графіки.
- Дослідити вплив рівня деталізації на сприйняття та аналіз результатів моніторингу.
- Оцінити ефективність застосування методів SDF-графіки та визначити можливості і шляхи їх оптимізації для підвищення ефективності сприйняття та аналізу результатів

Методи дослідження

Використання комп'ютерних засобів для створення програмних застосунків, аналіз методів візуалізації даних.

Наукова новизна

Наукова новизна полягає в обґрунтуванні доцільності використання методів SDF-графіки для візуалізації результатів комп'ютерно-екологічного моніторингу з участю тривимірної графіки.

Практичне значення отриманих результатів

Результати дослідження сприятимуть розробці інструментів візуалізації для екологічних досліджень, що підвищить ефективність аналізу та прийняття рішень у галузі екології.

Структура та обсяг дослідження

Магістерська робота включає вступ, 3 розділи, висновок, список літератури та додатки. Обсяг основної частини становить 80 сторінок друкованого тексту. У роботі використано 29 ілюстрацій.

Апробація результатів магістерської роботи.

За результатами досліджень є одна публікація:

Жебрак А.М., Литвинов В.А. Відображення результатів комп'ютерно еколого-економічного моніторингу за допомогою методів SDF-графіки // Наукові праці П'ятої міжнар. наук.-практ. конф. «Сучасні тенденції розвитку інформаційних систем і телекомунікаційних технологій», 1 червня 2023 р. (Київ, Україна). — К.: НУХТ, 2023. — 114 с.

РОЗДІЛ 1. АНАЛІЗ ВИДІВ ВІЗУАЛЬНИХ РЕЗУЛЬТАТІВ КОМП'ЮТЕРНО-ЕКОЛОГІЧНОГО МОНІТОРИНГУ І ВИЗНАЧЕННЯ ОБЛАСТІ ДОЦІЛЬНОГО ЗАСТОСУВАННЯ МЕТОДІВ SDF-ГРАФІКИ.

1.1 Моніторинг довкілля

Основні ідеї системи моніторингу були розроблені експертами у 1971 році, а перші концепції введені в роботу Р. Манна у 1973 році. Наукові основи сучасного моніторингу довкілля були висвітлені в роботах академіка І. П. Герасимова і професора Ю. А. Ізраеля, які визначили принципи формування системи екологічного моніторингу.

Нарада в Найробі в 1974 році визначила основні положення глобальної системи моніторингу, зосереджуючись на попередженні змін у стані природи та загрозах для здоров'я людини. Детальне обговорення завдань моніторингу відбулося на міжнародному симпозіумі в Ризі у 1978 році.

Академік І. П. Герасимов і професор Ю. А. Ізраель внесли важливий внесок у формування наукових основ сучасного моніторингу навколишнього середовища. Це включало принципи створення системи екологічного моніторингу та розгляд міжнародних аспектів глобальної системи моніторингу.

Моніторинг довкілля сьогодні розглядається як аналітично-інформаційна система, що охоплює спостереження, оцінювання, прогнозування та забезпечення науково-інформаційної підтримки управлінських рішень.

Система моніторингу довкілля охоплює спостереження, збирання, оброблення, передавання, збереження та аналіз інформації про стан довкілля. Вона включає в себе оцінку фактичного стану довкілля, прогнозування його змін, а також надає науково-інформаційну підтримку у прийнятті управлінських рішень для запобігання негативним змінам та дотримання екологічної безпеки.

1.1.1 Види моніторингу

Види моніторингу за складовими довкілля:

1. Моніторинг атмосферного повітря:

- Спостереження за концентрацією різних забруднюючих речовин у повітрі.
- Вимірювання рівня шкідливих газів, таких як SO₂, NO, CO₂ і тд.
- Вивчення якості повітря щодо вмісту пилу та інших часток.

2. Моніторинг поверхневих вод суші:

- Визначення рівня забруднення води хімічними та біологічними речовинами.
- Аналіз водних організмів для визначення стану водного екосистеми.
- Слідкування за змінами температури та хімічного складу води.

3. Моніторинг морських вод і вод океанів:

- Вивчення солоності, температури та густини морських вод.
- Аналіз хімічного складу води та забруднення водойм морського середовища.

4. Моніторинг геологічного середовища:

- Визначення стану ґрунтів та гірських порід.
- Спостереження за геологічними процесами, такими як зсуви та обвали.

5. Моніторинг ґрунтів:

- Аналіз хімічного та фізичного складу ґрунту.
- Вивчення впливу сільськогосподарської діяльності та інших факторів на якість ґрунту.

Особливі види моніторингу довкілля:

1. Кліматичний моніторинг:

- Вимірювання температури, вологості, атмосферного тиску та інших кліматичних параметрів.
- Слідкування за змінами в кліматі та їхнім впливом на довкілля.

2. Радіаційний моніторинг:

- Вимірювання рівня радіації у різних джерелах та середовища.
- Спостереження за радіаційним фоном та виявлення радіаційних забруднень.

3. Біотичний моніторинг:

- Дослідження стану та змін в біотиці, включаючи популяції рослин і тварин.
- Вивчення впливу довкілля на живі організми та їхні екосистеми.

4. Еколого-гігієнічний моніторинг:

- Аналіз впливу довкілля на здоров'я людей.
- Визначення рівнів токсичних речовин, які можуть впливати на громадське здоров'я.

5. Моніторинг лісових екосистем:

- Вивчення росту, стану та руйнувань лісових екосистем.
- Моніторинг біорізноманіття та забруднення лісових ділянок.

6. Агроекологічний моніторинг:

- Вивчення впливу сільськогосподарської діяльності на довкілля.
- Моніторинг якості ґрунту та води в аграрних зонах.

7. Соціально-екологічний моніторинг:

- Дослідження соціальних наслідків екологічних проблем та забруднень.
- Вивчення сприйняття та реакції громадськості на екологічні питання.

8. Громадський екологічний моніторинг:

- Залучення громадськості до вивчення та контролю за станом довкілля.
- Сприяння участі громадян у прийнятті рішень, спрямованих на покращення екологічної ситуації.

За державним регламентом встановлено що в державному моніторингу довкілля визначають три основні види моніторингу: загальний (стандартний), оперативний (кризовий) та фоновий (науковий).

Загальний моніторинг, також відомий як стандартний, включає оптимальну кількість параметрів спостереження на пунктах, які об'єднані в інформаційно-технологічну мережу. Це дозволяє регулярно оцінювати та прогнозувати стан довкілля і приймати управлінські рішення на різних рівнях.

Оперативний моніторинг, або кризовий, вивчає спеціальні показники на цільовій мережі пунктів в реальному масштабі часу. Це важливо для об'єктів з підвищеним екологічним ризиком у конкретних регіонах, таких як зони надзвичайної ситуації чи райони аварій, для оперативного реагування та ліквідації кризових ситуацій.

Фоновий моніторинг, або науковий, передбачає високоточні спостереження за всіма аспектами довкілля, включаючи характер, склад і рух забруднювальних речовин, а також реакцію організмів на забруднення на різних рівнях, від окремих популяцій до біосфери в цілому. Цей тип моніторингу проводиться на базових станціях у природних і біосферних заповідниках, а також на інших природоохоронних територіях.

Система державного моніторингу довкілля в Україні розподілена на три рівні: локальний для окремих об'єктів, регіональний для адміністративно-територіальних одиниць і природних регіонів, та національний для всієї території країни.

1.1.2 Способи відображення результатів моніторингу

Відображення результатів моніторингу довкілля грає важливу роль у розумінні та комунікації впливу людської діяльності на навколишнє середовище. Для ефективної подачі інформації можна використовувати різноманітні методи візуалізації. Загальні способи відображення дозволяють не лише наочно представити дані, але й надати їм конкретний контекст для зрозумілого сприйняття.

Разом вони створюють комплексний образ стану довкілля, допомагаючи аудиторії зрозуміти, аналізувати та приймати рішення щодо екологічних викликів.

Загальні способи відображення результатів моніторингу довкілля:

1. Лінійні графіки:
 - Показують зміни в часі або просторі.
2. Стовпчасті діаграми:
 - Використовуються для порівняння значень між різними категоріями.
3. Кругові діаграми:
 - Відображають частки від загальної кількості.
4. Географічні карти:
 - Показують розподіл параметрів у просторі.
5. Інфографіка:
 - Спрощена візуалізація складних даних за допомогою графіки та ілюстрацій.
6. Теплові карти:
 - Відображення інтенсивності показників за допомогою кольорів на карті.
7. Радарні графіки:
 - Вказують на взаємозалежність різних параметрів.
8. Тривимірні моделі:
 - Подача даних у тривимірному просторі для більшого розуміння взаємозв'язків.
9. Анімації:
 - Відстеження динаміки змін з часом.
10. Спектральні діаграми:
 - Використовують різні кольори для відображення різних параметрів.
11. Схеми та графи:
 - Відображення взаємозв'язків між елементами системи.
12. Фотодокументація:
 - Зйомка та використання фотографій для візуального представлення стану довкілля.

13.Гістограми:

- Графічне представлення розподілу даних.

14.Контурні карти:

- Відображення ліній рівних значень на географічній карті.

15.Дашборди та інтерактивні засоби:

- Створення онлайн-дашбордів для аналізу та взаємодії з результатами.

1.2 SDF графіка

SDF графіка використовує концепцію "signed distance fields" (SDF) для представлення геометричної інформації об'єктів у сцені. SDF - це відображення простору, в якому кожна точка визначається її відстанню до найближчої поверхні та напрямком до цієї поверхні. Це значення може бути позитивним, якщо точка знаходиться поза поверхнею, і негативним, якщо точка знаходиться всередині об'єкта.

SDF графіка використовує це відображення для рендерингу об'єктів у сцені. Замість традиційних методів, які використовують расторові графічні об'єкти, SDF графіка використовує SDF для визначення того, як світло і тіні інтерактивно взаємодіють з об'єктами. Це дає можливість отримати більше точне та деталізоване зображення, особливо при рендерингу складних форм та об'єктів.

Реймарчинг (raymarching) - це техніка, яку часто використовують в SDF графіці. Замість традиційного трасування променів (ray tracing), де промінь відстежується вздовж шляху, реймарчинг просувається вздовж променя в напрямку його спрямування, вираховуючи значення SDF на кожному кроці. Це дозволяє більш ефективно працювати з SDF даними, особливо при обробці складних сцен.

SDF графіка може використовуватися як для створення 2D, так і для створення 3D графіки. В 2D випадку SDF графіка дозволяє відобразити складні форми та деталізовані об'єкти, використовуючи переваги SDF визначення

відстаней та напрямків. У 3D випадку SDF графіка може бути використана для рендерингу складних 3D-сцен з високою деталізацією та реалістичністю.

Критерії ефективності SDF графіки:

1. Чіткість та Зрозумілість:

- Якість графічних зображень та їх зрозумілість для користувачів.
- Чи можливо однозначно ідентифікувати екологічні показники та їх зміни.

2. Гнучкість та Адаптованість:

- Спроможність методів SDF-графіки адаптуватися до різних типів даних та різних областей моніторингу.

3. Візуальне сприйняття:

- Як добре методи SDF-графіки передають важливі екологічні зміни та їх взаємозв'язки.
- Чи можливо виявити важливі тенденції та аномалії на візуалізаціях.

4. Вплив Рівня Деталізації:

- Як різні рівні деталізації впливають на сприйняття та аналіз результатів моніторингу.
- Чи існує оптимальний рівень деталізації для конкретних завдань.

5. Швидкодія та Оптимізація:

- Час, необхідний для генерації та відображення візуальних результатів.
- Можливості оптимізації для підвищення швидкодії та зниження навантаження на обчислювальні ресурси.

1.2.1 Поле знакових відстаней

SDF або Signed Distance Field (поле знакових відстаней) - це математичний підхід до представлення геометричних об'єктів у комп'ютерній графіці та обчислювальній геометрії.

Поле знакових відстаней (SDF) представляє собою математичну техніку для опису геометричних об'єктів, що утримує інформацію про відстань від кожної точки у геометричному об'єкті до найближчої його поверхні. Не лише фіксує

відстані, але також зберігає напрямок до цієї найближчої поверхні та вказує, чи розташована точка всередині чи зовні геометричного об'єкта.

Кожній точці у SDF надається значення, яке може бути позитивним, негативним або рівним нулю. Позитивне значення позначає, що точка знаходиться за межами геометричного об'єкта, негативне вказує на її розташування всередині об'єкта, а значення нуль вказує на точність її розташування на самій поверхні, як видно на рисунку 1.1.

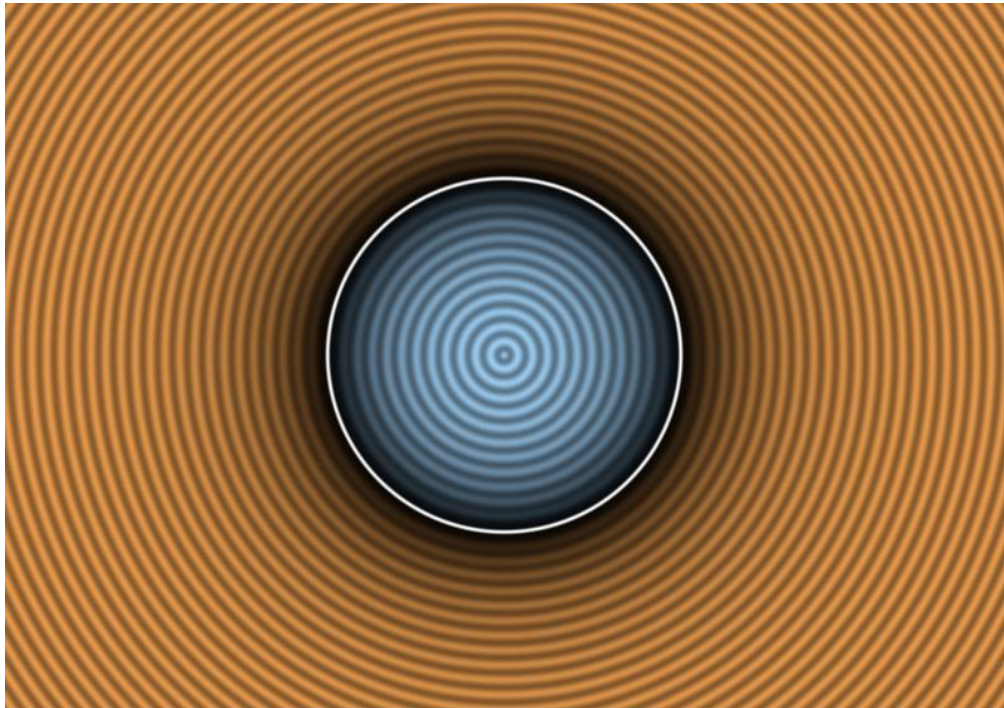


Рисунок 1.1 SDF коло

Існує можливість використання SDF для конструювання різноманітних геометричних фігур. Наприклад, для створення кола в 2D можна скористатися таким виразом:

$$\text{SDF}_{\text{кола}}(x,y)=\sqrt{x^2 + y^2} - \text{радіус} \quad (1.1)$$

Тут (x,y) - координати точки, а радіус визначається вами.

Аналогічно, для створення сфери в 3D можна використати аналогічний SDF вираз:

$$\text{SDF}_{\text{сфери}}(x,y,z)=\sqrt{x^2 + y^2 + z^2} - \text{радіус} \quad (1.2)$$

В функцію передається положення точки в просторі та інші аргументи, а вона в свою чергу повертає дистанцію до поверхні фігури.

Використовуючи подібні SDF функції, можна створювати різноманітні геометричні об'єкти, включаючи куби, конуси, тори, та інші. Це важливий інструмент для реалістичного моделювання об'єктів у комп'ютерній графіці та візуалізації тривимірних сцен.

Зараз розглянемо декілька інших SDF примітивів, які можна використовувати для конструювання різноманітних геометричних об'єктів у двовимірному просторі:

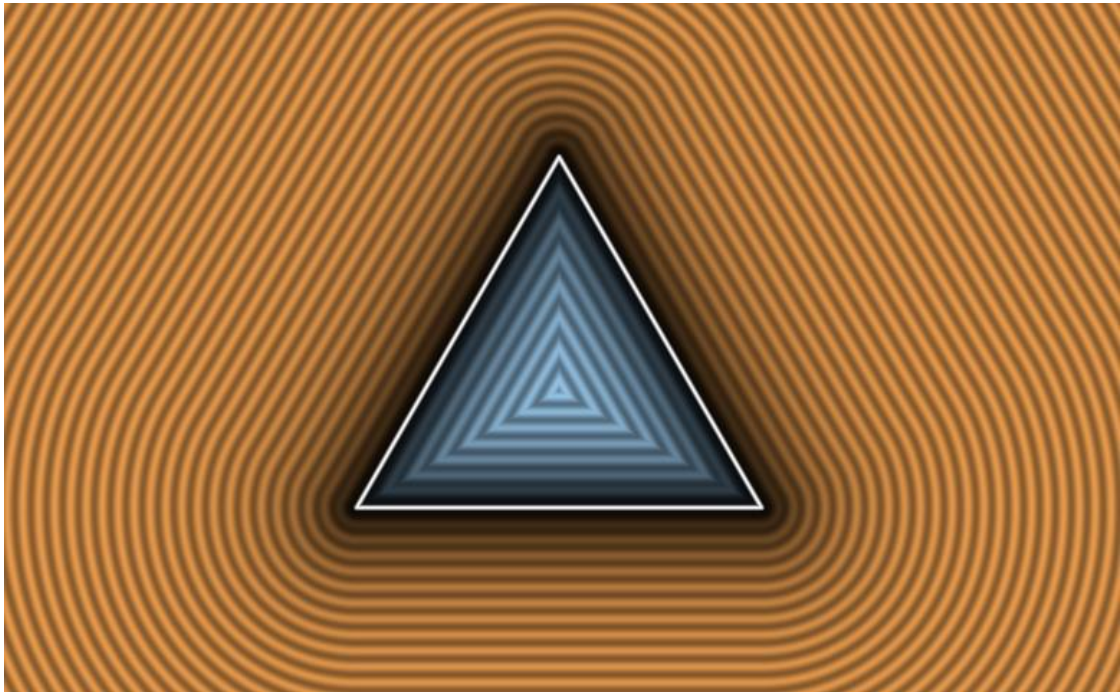


Рисунок 1.2 SDF трикутник

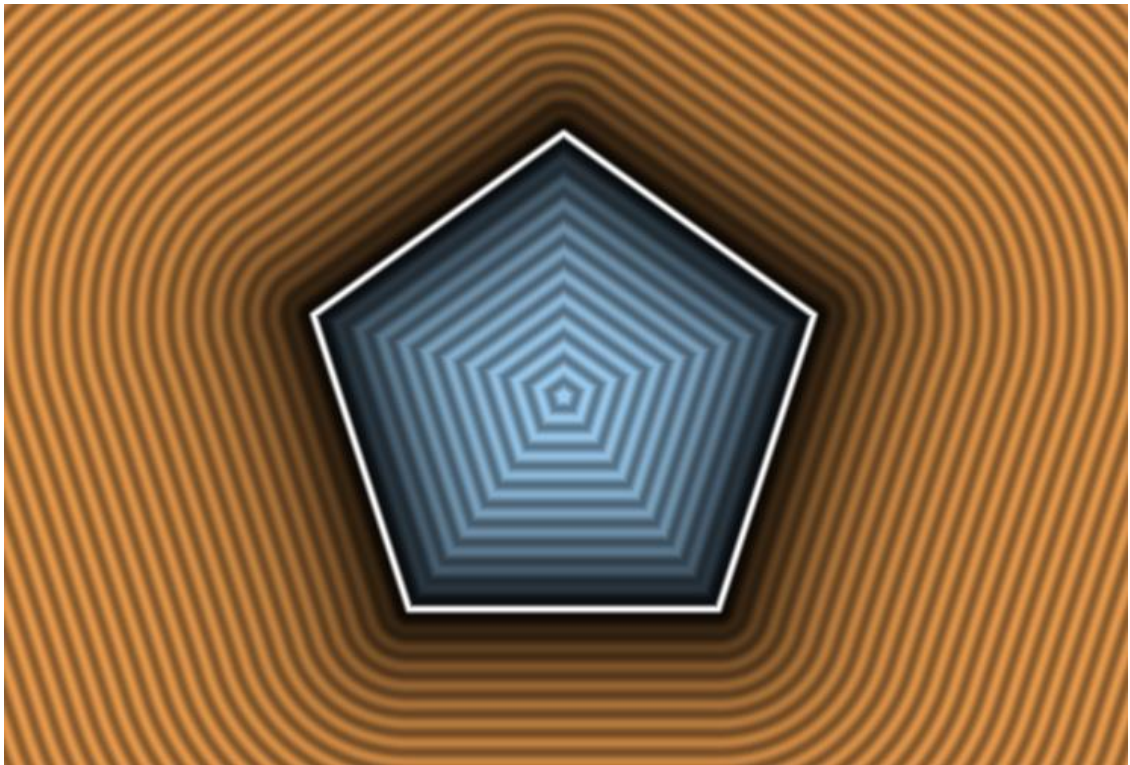


Рисунок 1.3 SDF п'ятикутник

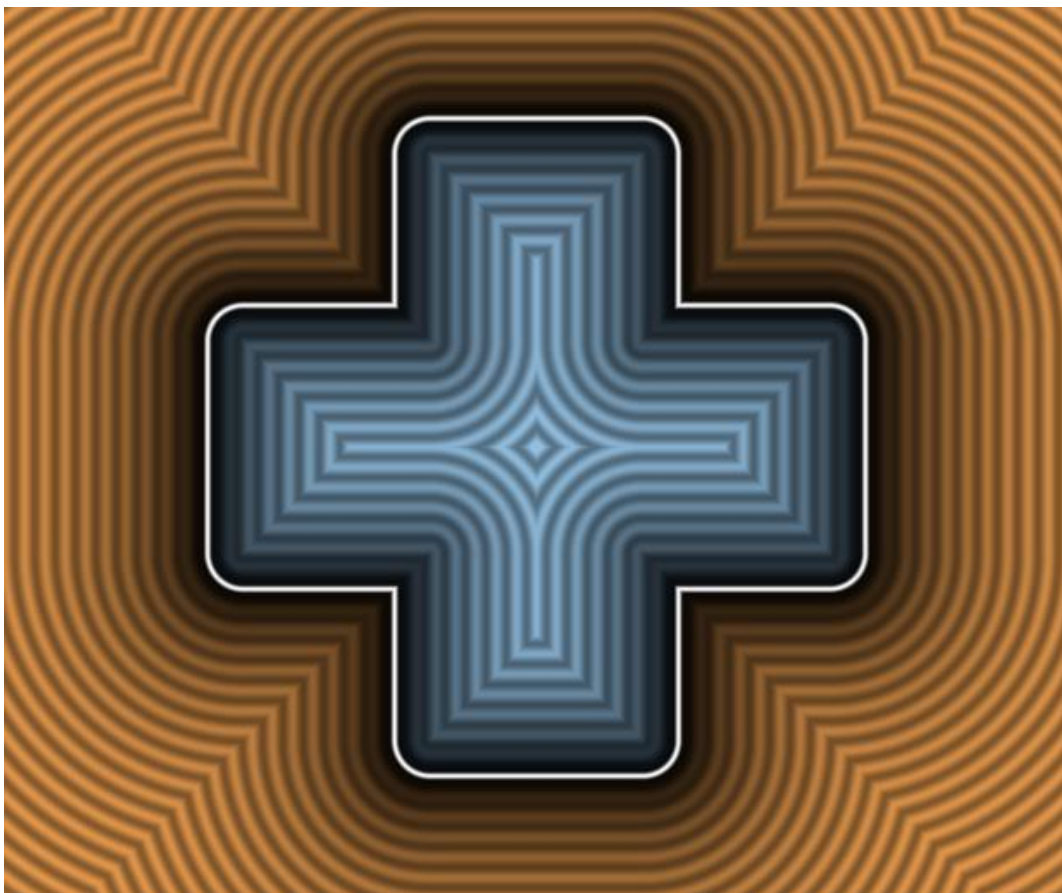


Рисунок 1.4 SDF крест



Рисунок 1.5 SDF прямокутник

Основні властивості SDF включають:

- **Знакова відстань:** відстань визначається з урахуванням напрямку, і має знак, що вказує, з якого боку точка розташована від об'єкта.
- **Компактність представлення:** SDF може бути представлений компактно за допомогою текстур або векторних полів, забезпечуючи ефективне збереження та передачу геометричної інформації.
- **Легкість обчислень:** Операції над SDF можуть виконуватися швидко та ефективно, що робить їх популярними для використання в реальному часі.
- **Можливість об'єднання та віднімання:** Об'єкти можуть бути комбіновані (об'єднані або відняті) за допомогою операцій на SDF, що дозволяє створювати складні форми.
- **Використання в рендерингу:** SDF часто використовується для рендерингу тривимірних об'єктів, забезпечуючи високий рівень деталізації та ефективність відображення.

- **Застосування в комп'ютерній графіці:** SDF широко використовується в комп'ютерній графіці для визначення колізій між об'єктами та для реалістичного освітлення.
- **Обробка комплексних форм:** SDF дозволяє легко представляти та обробляти складні форми, такі як криві та поверхні високого порядку.

Цей підхід допомагає уникнути обмежень традиційних методів відображення геометричних об'єктів та забезпечує ефективні рішення для багатьох завдань у комп'ютерній графіці та обчислювальній геометрії.

1.2.2 Ray marching

Для відображення 3D зображень, які використовують SDF, одним з найкращих методів є реймарчинг (ray marching). Реймарчинг є ефективним підходом, який дозволяє взяти віртуальний промінь та визначити його взаємодію з об'єктами, представленими у форматі SDF.

Вона базується на принципі просування променя (ray tracing), але використовує іншу стратегію обчислень. Ray marching здатен ефективно обробляти складні об'єкти, такі як фрактали чи об'єкти, описані математичними формулами, без необхідності їх аналітичної геометричної репрезентації. З'явився ray marching як ефективний метод рендерингу графіки в реальному часі.

Ray marching став особливо популярним у зв'язку з розвитком шейдерних технологій та здатністю використовувати високопродуктивні графічні карти для реалізації складних візуальних ефектів. Ця техніка дозволяє створювати складні та креативні візуальні сцени, що ілюструють математичні концепції та структури.

Ray marching використовує ітеративний процес для обчислення перетину променя з об'єктами сцени. Основна ідея полягає в тому, щоб крок за кроком просувати промінь через простір та обчислювати відстань від поточної точки променя до найближчого об'єкта сцени(рисунок 1.6).

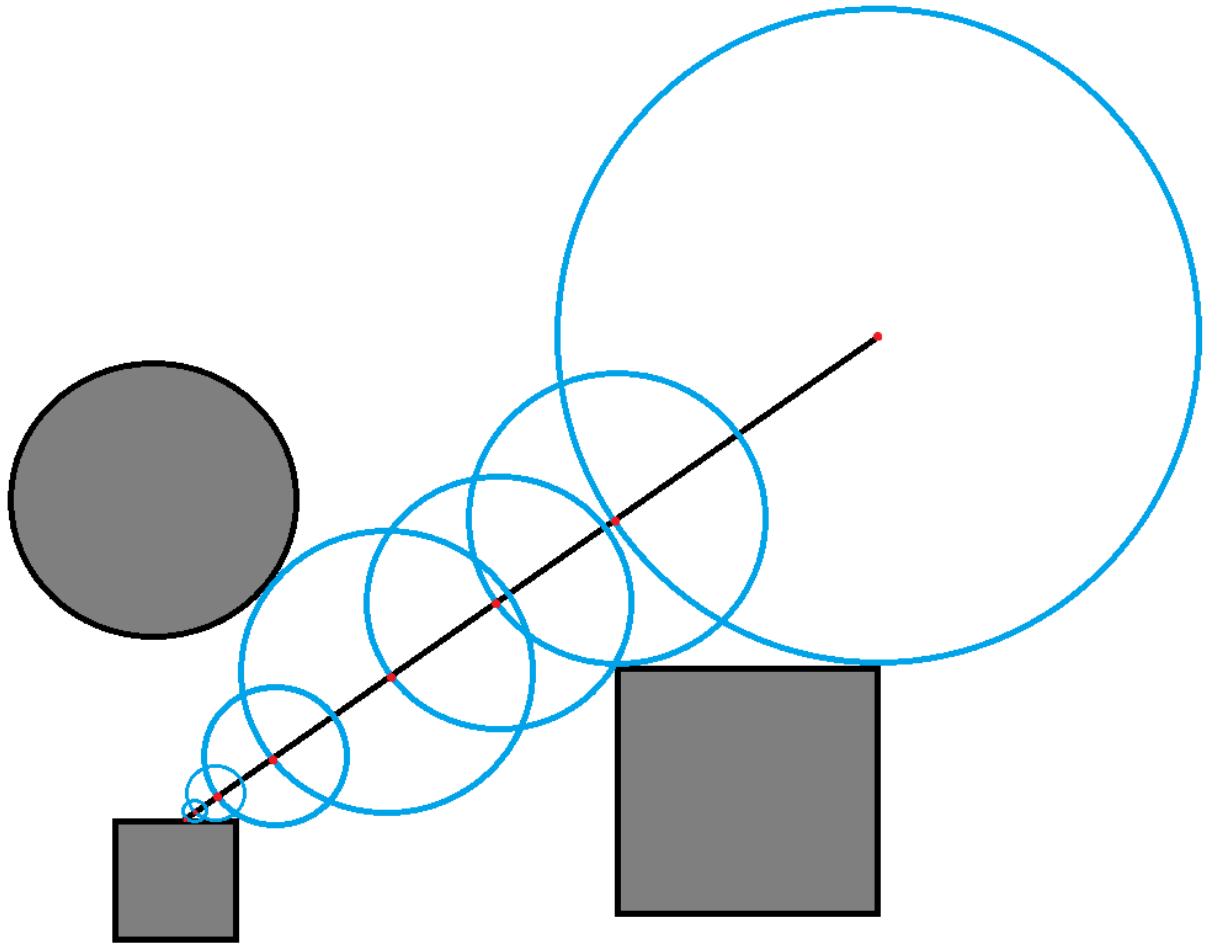


Рисунок 1.6 Проходження променя по сцені

На рисунку 1.6 червоними точками позначені шаги променя, а сині кола відповідають за найменшу дистанцію до об'єкта на сцені від точки на даній ітерації.

У контексті raymarching для кожного пікселя екрану створюється промінь (ray), який виходить з камери та прокладає шлях через сцену для визначення кольору пікселя(Рисунок 1.7).

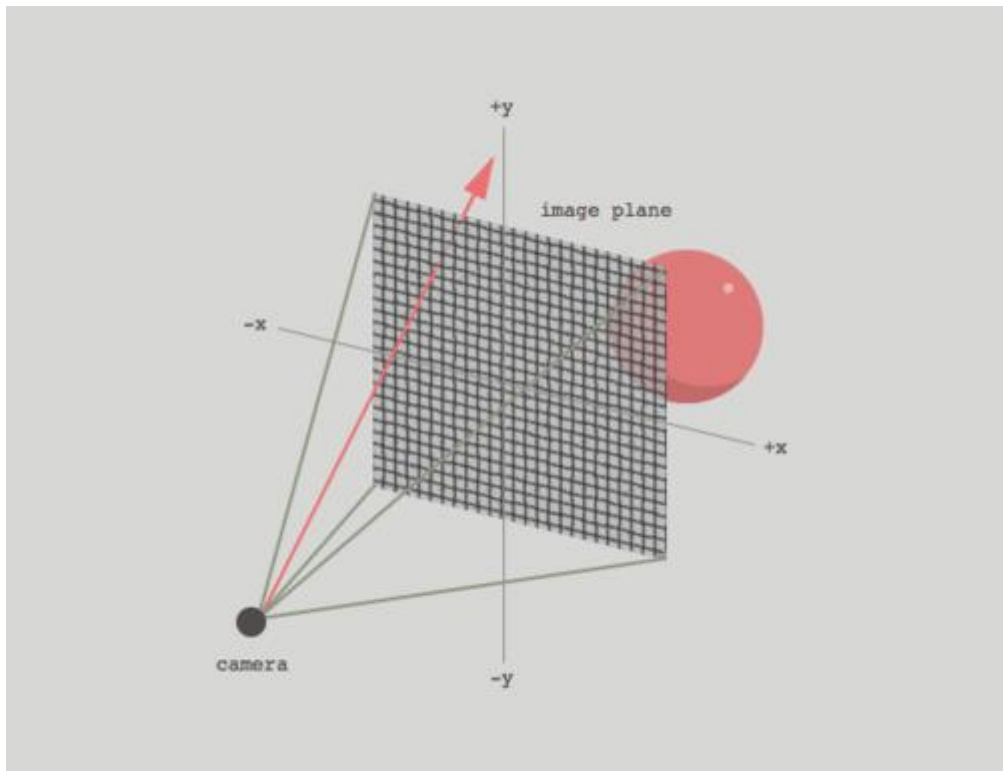


Рисунок 1.7 у праці[4]

Основні кроки роботи ray marching:

- **Ініціалізація променя:** Починаємо з початкової точки променя, яка зазвичай визначається позицією камери чи оглядача в сцені.
- **Ітерації:** Продовжуємо ітерації аж до досягнення обраної максимальної дистанції або до того моменту, коли промінь зіштовхується з об'єктом сцени.
- **Обчислення відстані:** На кожному кроці обчислюємо відстань від поточної точки променя до найближчого об'єкта сцени.
- **Пересування променя:** Промінь просувається вздовж свого напрямку на відстань, обчислену на попередньому кроці.
- **Перевірка зупинки:** Перевіряємо, чи промінь досягнув максимальної дистанції чи зіштовхнувся з об'єктом. Якщо так, ітерації завершуються.
- **Обчислення освітлення та колірив:** Після зупинки ітерацій обчислюються освітлення, колір та інші властивості для візуалізації поточної точки променя.

Цей процес повторюється для кожної точки на екрані, створюючи зображення сцени. Основна перевага ray marching полягає в його здатності ефективно обробляти складні об'єкти, такі як фрактали, а також в можливості використовувати програмування шейдерів для створення різноманітних візуальних ефектів.

Ray marching також має набір деяких інтересних можливостей:

Об'єднання об'єктів (Рисунок 1.8): Вміння об'єднувати об'єкти дозволяє створювати нові форми, об'єднуючи їх в єдиний об'єкт. Це корисно для створення складних структур та складних сцен.

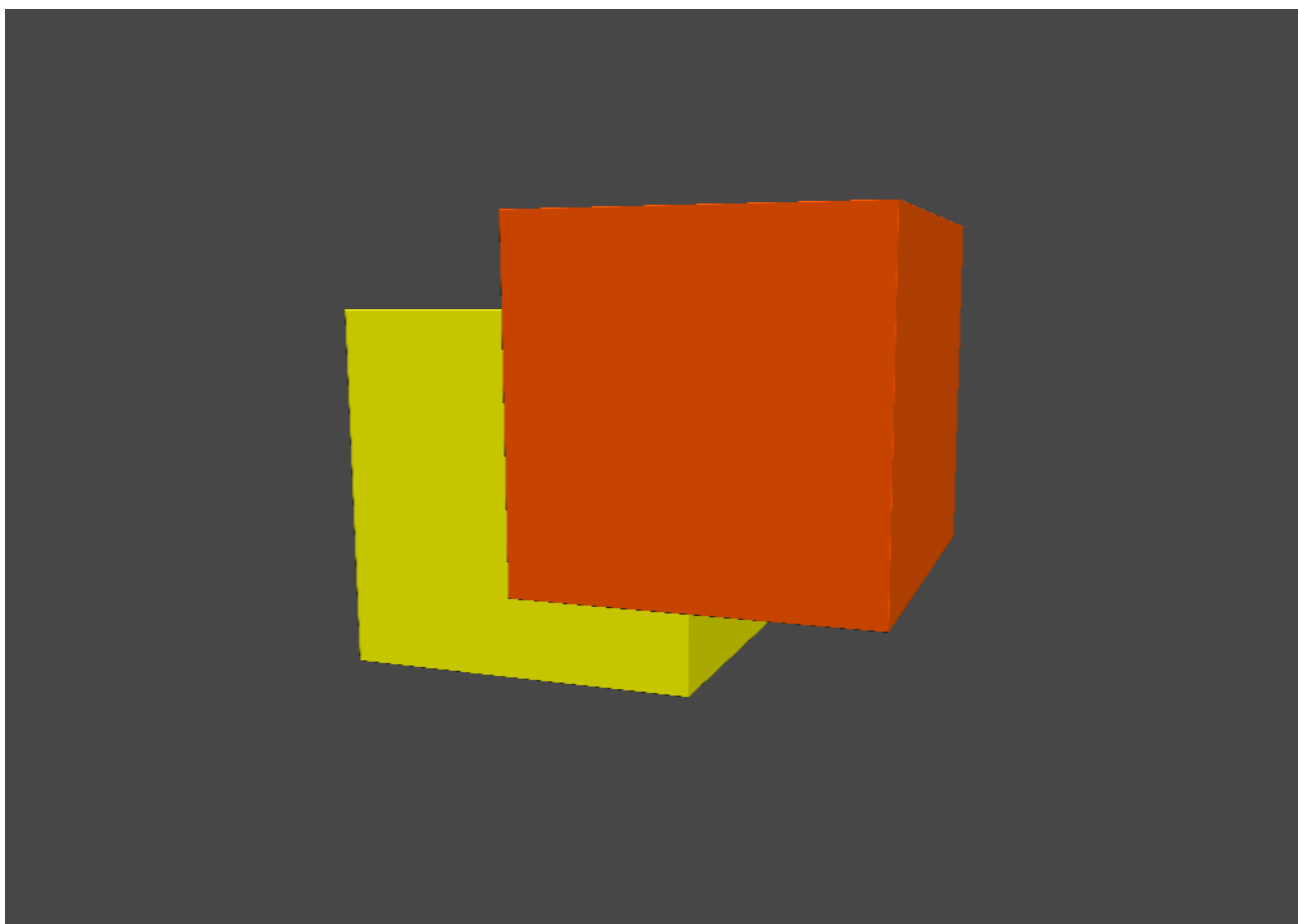


Рисунок 1.8 Об'єднання об'єктів

Віднімання об'єктів (Рисунок 1.9): Однією з унікальних особливостей реймарчингу є здатність віднімати один об'єкт від іншого. Це дозволяє створювати складні форми та відображати їхні взаємовідносини. Наприклад, ви можете відняти форму кулі від іншої, отримуючи тим самим складний виріз або вкладений об'єкт.

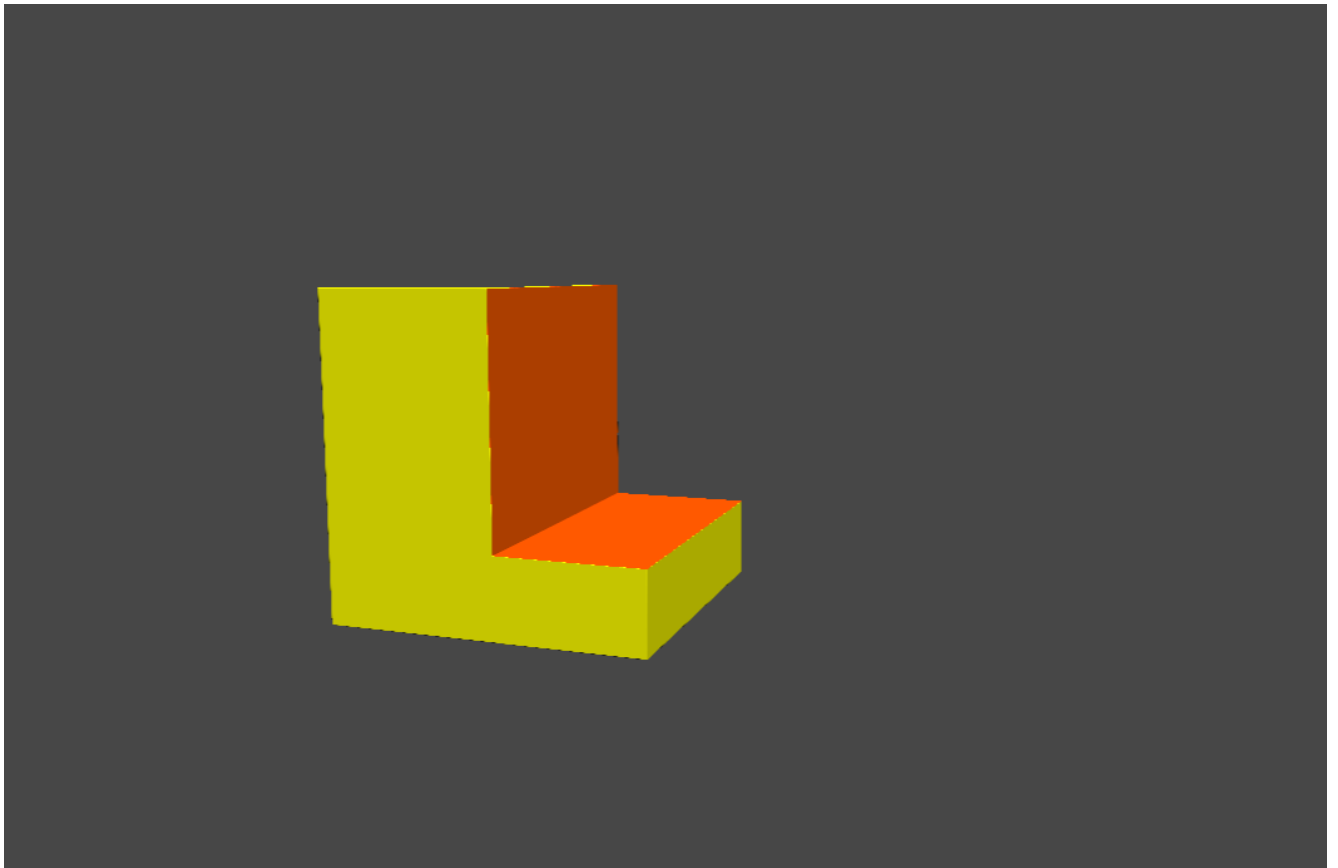


Рисунок 1.9 Віднімання об'єктів

Перетин об'єктів (Рисунок 1.10): Можливість визначення точок перетину двох об'єктів дозволяє створювати складні форми, які формуються в тільки тих місцях, де об'єкти перетинаються. Це дозволяє створювати складні та деталізовані візуальні ефекти.



Рисунок 1.10 Перетин об'єктів

Змішування об'єктів (Рисунок 1.11): Реймарчинг дозволяє змішувати властивості двох об'єктів, що взаємодіють між собою. Це включає в себе змішування кольорів, прозорості та інших візуальних характеристик. За допомогою змішування можна створювати плавні переходи між об'єктами, а також створювати цікаві ефекти в областях їх перетину.

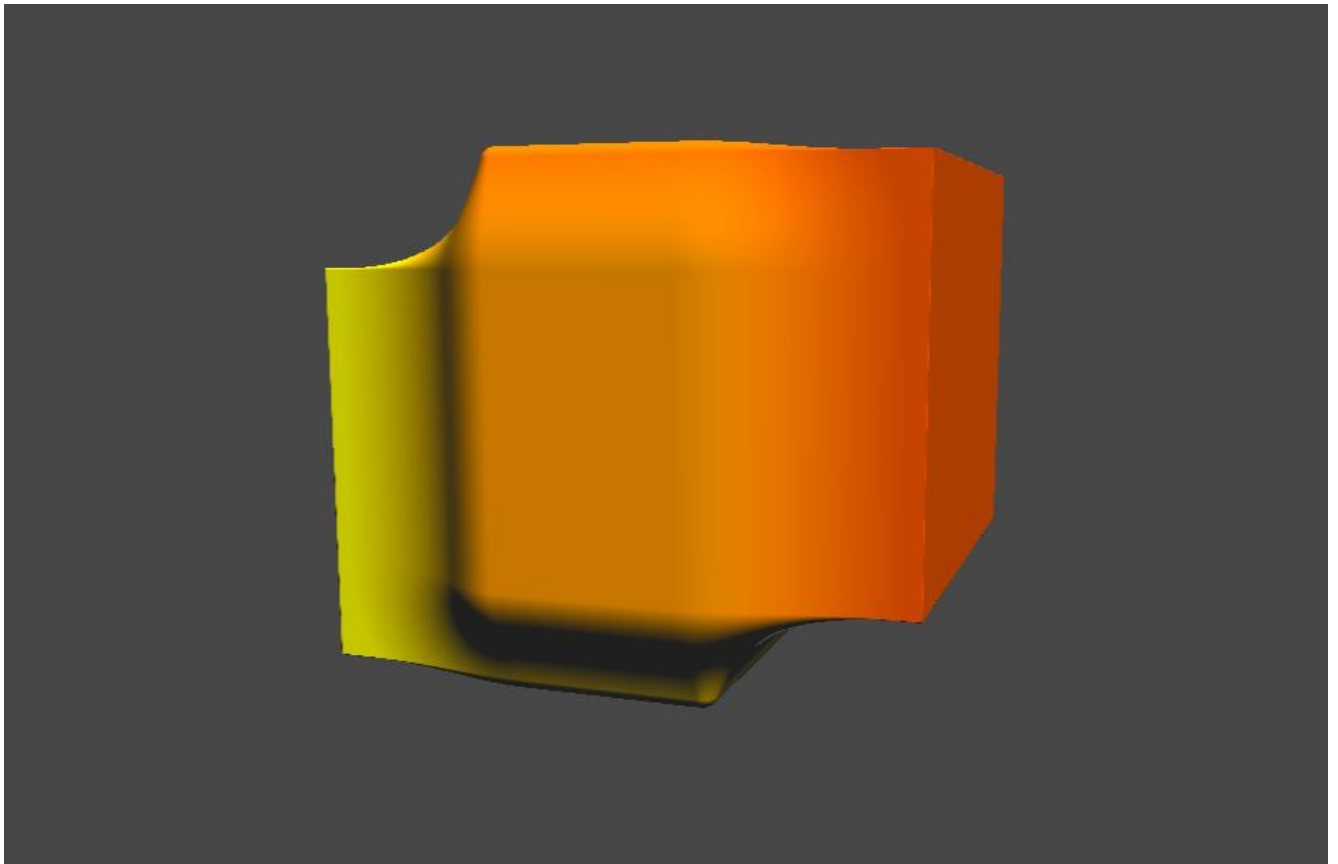


Рисунок 1.11 Змішування об'єктів

Основні переваги реймарчингу для відображення SDF-представленого 3D зображення включають:

- **Простота обчислень:** реймарчинг використовує промені для обходу сцени, роблячи обчислення відстаней та освітлення ефективними та швидкими.
- **Висока деталізація:** реймарчинг дозволяє отримати високу деталізацію, оскільки він може точно визначити взаємодію променя з геометричними об'єктами, представленими у формі SDF.
- **Динамічні сцени:** реймарчинг легко впроваджується для динамічних сцен, оскільки обчислення відстаней до об'єктів може здійснюватися на льоту.
- **Гнучкість управління освітленням:** застосування SDF у реймарчингу дозволяє легко управляти освітленням та відображенням тіней, що робить його ефективним для реалістичного візуалізації.
- **Можливість об'єднання та віднімання:** реймарчинг легко підтримує операції об'єднання та віднімання об'єктів, що дає можливість створювати складні геометричні форми.

- **Прозорість та туман:** реймарчинг легко обробляє ефекти прозорості та туману, що робить його ідеальним для сцен з різними матеріалами та атмосферними ефектами.
- **Підтримка складних ефектів:** метод реймарчингу дозволяє ефективно реалізувати тіні, реалістичне освітлення та відображення.
- **Поєднання з іншими ефектами:** реймарчинг легко поєднується з іншими ефектами, такими як відображення, заломлення та амбієнтна оклюзія, що розширює можливості візуалізації.

1.2.3 Області доцільного застосування методів SDF-графіки

SDF в першу чергу використовується для моделювання та відображення геометричних об'єктів у тривимірному просторі, його також можна застосовувати для графіків та інших типів відображення результатів моніторингу довкілля. Однак варто враховувати, що це може бути не завжди найбільш доцільним підходом і його використання може бути більше спрямоване на творчу подачу даних та інтерактивні можливості.

Застосування SDF-графіки може призводити до створення вражаючих та ефективних візуалізацій результатів 3D-моніторингу. Нижче подано кілька описових прикладів, які можуть виглядати особливо ефективно з використанням SDF-графіки:

1. Об'ємні хмари точок:

- **Опис:** візуалізація тривимірної просторової інформації, представленої у вигляді масиву точок у тривимірному просторі. Застосування SDF в контексті об'ємних хмар точок може покращити якість візуалізації та дозволити більш ефективно аналізувати геометричні та топологічні властивості об'єктів.
- **Переваги:** Точність та чіткість визначення об'ємних областей.

2. Контурні мапи з ефектом "глибини":

- Опис: SDF-графіка може надавати контурні мапи з додатковим ефектом "глибини", що дозволяє відокремлювати різні об'єкти та їх структури.
- Переваги: Забезпечення виділення та розрізнення між різними об'єктами в просторі.

3. Об'ємні теплові карти:

- Опис: Використання SDF-графіки для створення об'ємних теплових карт може показати зміни в параметрах чи характеристиках в об'ємі простору.
- Переваги: Інформативне відображення розподілу значень за об'ємом.

4. Об'ємні анімації з часовою "глибиною":

- Опис: Використання SDF-графіки для створення об'ємних анімацій, які враховують часовий аспект, може надати динаміку та рухливість результатам моніторингу.
- Переваги: Ефективна передача динамічних змін в часі, що полегшує сприйняття та аналіз змін.

5. Густина рослинності та лісового покриву:

- Опис: SDF-графіка може використовуватися для візуалізації густини рослинності або лісового покриву в 3D-просторі, надаючи точні та зрозумілі об'ємні представлення стану лісів чи рослинності.
- Переваги: Відокремлення та аналіз структури лісів за допомогою об'ємних графіків.

6. Об'ємні моделі ґрунту та рельєфу:

- Опис: SDF-графіка може допомагати в створенні об'ємних моделей рельєфу та ґрунту, враховуючи висотні різниці та структуру земельної поверхні.
- Переваги: Відображення геоморфологічних характеристик довкілля з високою точністю та реалізмом.

7. Об'ємна візуалізація водних ресурсів:

- **Опис:** SDF-графіка може служити для створення об'ємних моделей водних ресурсів, включаючи озера, річки та водойми, що дозволяє аналізувати їх розподіл та стан.
- **Переваги:** Зрозуміле та інформативне відображення гідрологічних об'єктів та їх взаємодій.

8. Візуалізація атмосферних явищ:

- **Опис:** Застосування SDF-графіки для візуалізації атмосферних явищ, таких як тумани, пил, або атмосферні течії.
- **Переваги:** Реалістичне та вражаюче відображення атмосферних умов для аналізу впливу на довкілля.

Можливості SDF в інших типах відображення:

- **Графіки:** Використання SDF для графіків може бути менш доцільним, оскільки існують спеціалізовані методи для представлення двовимірної інформації, такі як лінійні графіки та діаграми. Однак для створення графічних елементів у тривимірному просторі SDF може бути застосованим, наприклад, для позначення точок на графіці з третьою вимірювальною площиною.
- **Діаграми та інфографіка:** Інфографіка та діаграми зазвичай базуються на двовимірних візуалізаціях, і використання SDF для цих цілей може бути перебільшеним. Проте, для створення креативних тривимірних візуалізацій, особливо для інтерактивних інфографік, елементи SDF можуть додати унікальність та глибину.
- **Інтерактивні засоби:** В інтерактивних візуалізаціях SDF може додати інтерактивність та тривимірність, що сприяє кращому розумінню та взаємодії з даними про довкілля.
- **Географічні карти:** В контексті географічних карт SDF може забезпечити точне визначення відстаней до об'єктів на карті, а також надати глибину та

тривимірність для ландшафту та рельєфу. Це може бути особливо корисно при відображенні географічних характеристик у тривимірному просторі.

1.2.4 Постановка задач подальших досліджень.

Однією з першочергових задач використання SDF-графіки для візуалізації даних моніторингу довкілля є дослідження впливу рівня деталізації на сприйняття та аналіз результатів моніторингу. Це дослідження передбачає аналіз впливу різних рівнів деталізації на сприйняття та розуміння інформації, яку надає візуалізація за допомогою SDF-графіки. Буде розглянуто, як висока деталізація може покращити читабельність та точність візуалізації, а також як це може вплинути на витрати обчислювальних ресурсів.

Результатом другого розділу "Вплив рівня деталізації на сприйняття та аналіз результатів моніторингу" буде визначення оптимального рівня деталізації для ефективного використання SDF-графіки в контексті візуалізації даних моніторингу довкілля. При цьому будуть розглянуті фактори впливу візуальної деталізації на сприйняття результатів моніторингу та запропоновані практичні рекомендації щодо оптимального використання SDF-графіки для покращення візуалізації екологічних даних. Також, у межах розділу буде проведено розробку SDF-графіки для прикладів візуалізації, зокрема створено фрагментний шейдер для відображення SDF-графіки та розроблено приклад візуалізації результатів моніторингу атмосферного повітря.

Крім основного (цільового) напрямку застосування – візуалізація даних моніторингу довкілля - дослідження впливу рівня деталізації SDF-графіки є доцільним з наступних причин:

- **Оптимізація ресурсів:** Визначення оптимального рівня деталізації може допомогти оптимізувати використання ресурсів обчислювальної техніки, так що графічна якість залишається в прийнятних межах, але забезпечується ефективність в обробці SDF для створення деталізованих об'єктів у візуальному представленні довкілля.

- **Взаємодія з користувачем:** Рівень деталізації може впливати на сприйняття графічного вмісту користувачем. Дослідження може допомогти визначити оптимальний баланс між високою якістю графіки та продуктивністю системи, що важливо для комфортного використання та взаємодії.
- **Ефективність моніторингу довкілля:** Якщо SDF-графіка використовується для візуалізації даних моніторингу довкілля, то дослідження рівня деталізації може допомогти визначити, як точність та деталізація впливають на ефективність аналізу та сприйняття отриманих результатів.
- **Розробка оптимальних методів відображення:** Результати дослідження можуть призвести до розробки нових методів або оптимізації існуючих для відображення SDF-графіки з врахуванням рівня деталізації.
- **Сприяння науковому прогресу:** Дослідження впливу рівня деталізації в рамках SDF-графіки може внести вагомий внесок у науковий прогрес у галузі комп'ютерної графіки та обчислювальної графіки.

Проведені дослідження дадуть можливість глибше зрозуміти ефективність візуальної деталізації даних в контексті моніторингу довкілля. Ці наукові дослідження розглядатимуть вплив різних методів візуалізації на сприйняття та аналіз результатів моніторингу, зокрема у контексті використання SDF-графіки.

В ході дослідження буде враховано різноманітні аспекти, такі як роздільна здатність текстур, специфікація контурів, обсяг та складність геометрії та інтеграція з іншими технологіями. Аналіз впливу цих факторів дозволить визначити оптимальні методи візуалізації для забезпечення максимальної зрозумілості та використання інформації, отриманої в ході моніторингу довкілля.

Отримані результати дослідження стануть основою для розробки рекомендацій та покращень у використанні візуальної деталізації в екологічних дослідженнях. Це відкриє нові можливості для забезпечення належного розуміння

та ефективного управління довкіллями ресурсами, підвищуючи якість рішень у сфері екології.

Третій розділ буде піддаватися оцінці за такими ключовими критеріями:

1. Візуальне сприйняття:

- Як добре методи SDF-графіки передають важливі екологічні зміни та їх взаємозв'язки.
- Чи можливо виявити важливі тенденції та аномалії на візуалізаціях.

2. Швидкодія та Оптимізація:

- Час, необхідний для генерації та відображення візуальних результатів.
- Можливості оптимізації для підвищення швидкодії та зниження навантаження на обчислювальні ресурси.

Ці критерії дозволяють визначити ефективність застосування SDF-графіки в контексті екологічного моніторингу, зосереджуючись на якості візуалізацій, впливі рівня деталізації на сприйняття та аналіз, а також продуктивності методів для забезпечення оптимальної швидкодії та ефективності обчислень. Оцінка цих аспектів допоможе визначити, наскільки успішно та ефективно використовується SDF-графіка в контексті вивчення та аналізу даних моніторингу довкілля.

Хоча використання SDF в двовимірних візуалізаціях може виглядати надто складним та неефективним, у креативних проектах чи для вражаючих презентацій він може додати унікальний стиль та ефект. В той же час, його основне призначення все ж розкривається в тривимірному відображенні геометрії та інтерактивних застосуваннях що об'єднують геометричну глибину та тривимірний простір, можуть надати додатковий вимір та ефективно підсилити візуальне враження даних про довкілля.

Подальші дослідження в цьому напрямку можуть сприяти розвитку та вдосконаленню технологій візуалізації, що є актуальним напрямом в індустрії розвитку відеоігор, симуляції та візуалізації наукових даних.

РОЗДІЛ 2. ДОСЛІДЖЕННЯ ВПЛИВУ РІВНЯ ДЕТАЛІЗАЦІЇ НА СПРИЙНЯТТЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ МОНІТОРИНГУ

2.1 Вплив рівня деталізації на сприйняття та аналіз результатів моніторингу

Візуальна деталізація даних відіграє ключову роль у поліпшенні сприйняття та аналізу результатів моніторингу довкілля. Перш за все, графічне представлення інформації робить її більш доступною та зрозумілою для користувача. Графіки, діаграми та кольорові схеми дозволяють легше визначити зв'язки та тренди, спрощуючи інтерпретацію складних даних.

Далі, візуальна деталізація сприяє швидшому виявленню аномалій чи виникненню певних тенденцій в динаміці довкільних показників. Це дозволяє оперативно реагувати на потенційні проблеми та вдосконалювати стратегії управління довкільними ресурсами.

Візуалізація даних може розкрити складні взаємозв'язки між різними параметрами, які можуть залишитися непоміченими у текстовій формі. Вона дозволяє виявляти кореляції та причинно-наслідкові зв'язки, що сприяє глибокому розумінню довкільних процесів.

Крім того, візуалізація створює можливість ефективного взаємодії з даними для широкого кола аудиторії. Навіть ті, хто не має глибоких знань у сфері науки чи статистики, можуть легко зрозуміти важливі показники та тренди, що полегшує широке поширення інформації та прийняття рішень.

Загалом, візуальна деталізація даних вносить вагомий внесок у розвиток наукових досліджень, прийняття управлінських рішень та формування ефективних стратегій збереження довкілля. Вона робить дані більш доступними, зрозумілими та дієвими для різних зацікавлених сторін, що сприяє сталому розвитку та екологічній освіті.

2.1.1 Фактори впливу візуальної деталізації результатів моніторингу

В сучасному світі, де дані стають все більш важливим ресурсом для прийняття рішень та розвитку суспільства, візуальна деталізація даних грає вирішальну роль в зрозумінні та аналізі інформації. Це особливо актуально в контексті моніторингу довкілля, де величезний обсяг даних потребує ефективних інструментів візуалізації для виявлення тенденцій, аномалій та шляхів оптимізації ресурсів.

Важливо враховувати, як різноманітні чинники впливають на візуальну деталізацію даних, оскільки вони визначають успішність інтерпретації та використання інформації. В цьому контексті розглядатимемо п'ять ключових аспектів, що визначають, як візуалізація може сприяти розумінню та аналізу результатів моніторингу довкілля.

Вплив на візуальну деталізацію даних визначається декількома факторами:

- **Обсяг та тип даних:** Кількість та характер інформації може визначати оптимальний спосіб її візуалізації. Деякі дані краще представляти графіками чи діаграмами, тоді як інші можуть вимагати карт чи хмар тегів.
- **Аудиторія:** Хто буде користувачем візуальних даних? Врахування особливостей аудиторії, їхніх знань та очікувань важливо для створення ефективних візуалізацій.
- **Мета візуалізації:** Яка інформація має бути передана через візуальний елемент? Якщо це аналіз трендів, порівняння значень або виявлення аномалій, це впливатиме на вибір типу графіку чи діаграми.
- **Дизайн та естетика:** Чіткість та привабливість візуальної деталізації важливі для залучення уваги та зрозумілості. Вибір кольорів, шрифтів і загального дизайну може впливати на сприйняття інформації.
- **Технічні обмеження:** Обсяг оброблюваної інформації, швидкість завантаження та можливості відображення на конкретних пристроях чи платформах також враховуються при виборі методів візуалізації.

Ці чинники взаємодіють, формуючи оптимальний підхід до візуальної деталізації даних для конкретного випадку.

2.1.2 Візуальна деталізації результатів моніторингу за допомогою SDF-графіки

Використання SDF-графіки (Signed Distance Field) у візуальній деталізації даних відкриває нові можливості для створення зображень та анімацій з високою якістю та деталізацією. Цей метод вигідний для представлення об'єктів через їхні контури та форми, особливо в 2D та 3D графіці. Однак, реалізація SDF-графіки вимагає уважного врахування ряду чинників для досягнення оптимального результату.

Вплив на візуальну деталізацію за допомогою SDF-графіки визначається кількома ключовими факторами, які включають обсяг та тип даних, аудиторію користувачів, мету візуалізації, дизайн та естетику, та технічні обмеження. Однак, при роботі з SDF-графікою, також слід враховувати розмір та роздільну здатність текстур, специфікацію контурів, підтримку анімації, обсяг та складність геометрії, і можливості інтеграції з іншими технологіями.

Ці аспекти взаємодіють, формуючи оптимальний підхід до візуальної деталізації даних за допомогою SDF-графіки, що дозволяє досягти високої якості та ефективності в процесі візуалізації.

Візуальна деталізація за допомогою SDF-графіки (Signed Distance Field) враховує специфічні аспекти, які можуть впливати на якість та ефективність візуалізації:

- **Розмір та роздільна здатність текстур:** Висока роздільна здатність SDF-графіки дозволяє забезпечити деталізовані та високоякісні текстури для візуалізації об'єктів. Розмір текстур впливає на якість зображення, адже більші текстури можуть вимагати більше ресурсів.

- **Специфікація контурів:** SDF-графіка дозволяє представити контури об'єктів з високою точністю. Визначення точних контурів може покращити зовнішній вигляд та читабельність графічних елементів.
- **Підтримка анімації:** Якщо вимагається анімація об'єктів, то потрібно враховувати вплив частоти оновлення та обчислень на швидкодію анімації при використанні SDF-графіки.
- **Обсяг та складність геометрії:** Складність форм та кількість геометричних об'єктів можуть впливати на продуктивність та швидкість візуалізації. Велика кількість об'єктів чи деталей може збільшити обчислювальне навантаження.
- **Інтеграція з іншими технологіями:** Врахування можливостей інтеграції SDF-графіки з іншими технологіями, такими як освітлення чи тінювання, є ключовим для створення комплексних та реалістичних сцен.

Загально кажучи, вибір параметрів SDF-графіки та їх взаємодія з конкретними характеристиками сцени визначають якість та ефективність візуальної деталізації.

2.2 Розробка та деталізація SDF-графіки для прикладів візуалізації

Весь процес розробки SDF-графіки, використання фрагментних шейдерів та створення візуалізацій результатів моніторингу довкілля буде здійснюватися за допомогою програмного забезпечення Unity. Unity - це інтегроване середовище розробки, яке надає широкий набір інструментів для створення ігор, симуляцій та інтерактивних додатків.

Основною складовою розробки SDF-графіки на Unity є використання фрагментних шейдерів. Ці шейдери використовуються для обчислення значень в кожному пікселі екрану, де кожен піксель може представляти відстань до найближчого об'єкта на сцені, а також інші характеристики, такі як колір, освітлення тощо.

- **Інтерфейс та Редагування в Unity:** Програмне забезпечення Unity має інтуїтивний інтерфейс, який дозволяє розробникам легко створювати сцени, редагувати об'єкти та взаємодіяти з компонентами графіки. Розробка SDF-графіки відбудуватиметься у цьому зручному середовищі.
- **Мова Програмування та Сценарії на C#:** Unity використовує мову програмування C# для написання сценаріїв та логіки. Розробка SDF-графіки буде базуватися на написанні програмних кодів на C#, де реалізація алгоритмів SDF та фрагментних шейдерів буде виконуватися.
- **Шейдери в Shader Lab:** Для реалізації фрагментних шейдерів, Unity використовує мову Shader Lab. Розробники можуть створювати та оптимізувати шейдери, визначаючи, як кожен піксель сцени буде відображений, освітлення та колір, забезпечуючи точні та ефективні візуалізації.
- **Взаємодія та Перегляд Результатів:** Unity також дозволяє в реальному часі переглядати та тестувати результати розробки. Розробники можуть взаємодіяти зі сценою, переглядати анімації та робити зміни на льоту, щоб досягти бажаного ефекту.

Програмне забезпечення Unity в цьому випадку виступає як основна платформа для розробки SDF-графіки, а його інструменти та можливості роблять процес розробки ефективним та доступним для розробників.

2.2.1 Створення фрагментного шейдеру для відображення SDF-графіки

Фрагментний шейдер (fragment shader), також відомий як піксельний шейдер, є одним із типів шейдерів у програмуванні графіки, використовуваним для обробки окремих пікселів об'єкта або сцени. Цей тип шейдера використовується для визначення кольору та інших характеристик кожного окремого пікселя, який відображається на екрані.

Основним завданням фрагментного шейдера є визначення кольору пікселя на екрані на основі різних факторів, таких як текстур, освітлення, ефекти та інші

параметри, які можуть бути використані для створення візуально привабливого зображення.

Основна структура фрагментного шейдера включає код, що визначає кінцевий колір пікселя, а також може включати різноманітні обчислення та ефекти для поліпшення візуальної якості сцени.

У цьому проєкті ми збираємось розробити шейдер, який використовує SDF-графіку для відображення результатів моніторингу.

Сама ідея полягає в тому, щоб використовувати SDF які містять відстані від кожного пікселя до найближчого контуру об'єкта на сцені.

Крім того, планується дослідження можливостей розширення SDF-графіки для створення цікавих візуальних ефектів, таких як м'яккі тіні, ефекти освітлення та інші відтінки, які можуть покращити реалізм відображення сцени.

Для початку процесу реймарчингу необхідно згенерувати нормалізовані вектори, які визначають напрямок променя від камери для кожного окремого пікселя на екрані. Це є ключовим етапом у визначенні того, які об'єкти чи частини сцени відобразатимуться на кожному пікселі.

Зазвичай цей процес включає в себе обчислення нормалізованого вектора променя для кожного пікселя, який виходить від точки спостереження (зазвичай розташованої в камері) і направлено до певної точки на сцені або об'єкта. Цей вектор представляє напрямок "погляду" камери для кожного пікселя на екрані.

Отримавши нормалізований вектор променя для кожного пікселя, можна використовувати його для подальших обчислень.

Цей підхід дозволяє ефективно моделювати те, як світло та об'єкти взаємодіють із сценою на основі кута, під яким камера бачить кожен піксель. В результаті, реймарчинг може створити реалістичні та здорово виглядаючі зображення з врахуванням всіх аспектів освітлення та матеріалів на сцені.

Нормалізований вектор променя можна ефективно відобразити у вигляді кольору, що дозволяє використовувати кольори для візуалізації напряму поверхні об'єкта в просторі.

Один з популярних способів візуалізації нормалізованого вектора променя - це представлення його компонентів як кольорів RGB. Кожна компонента (частка) вектора відображається відповідною складовою кольору: червоний (R), зелений (G), та синій (B). Цей підхід дозволяє спростити сприйняття напрямку вектора та зробити його візуально доступним.

Наприклад, якщо вектор має напрям вздовж осі X, то відповідний колір може бути представлений як $(1, 0, 0)$, що відповідає червоному кольору. Аналогічно, для вектора вздовж осі Y відображається $(0, 1, 0)$ (зелений колір), а для вектора вздовж осі Z - $(0, 0, 1)$ (синій колір) (Рисунок 2.1).

Цей метод використання кольорів для представлення нормалізованих векторів може бути корисним у візуальному аналізі напрямку освітлення, поверхневих нормалей в графіці, або при розробці алгоритмів, пов'язаних із розрахунками у тривимірному просторі.



Рисунок 2.1 Відображення нормалізованих векторів у вигляді кольорів

Після створення нормалізованих векторів, які представляють напрямок променя від камери для кожного пікселя, починається написання алгоритму реймарчингу. Цей процес включає проходження променя через сцену, аналіз взаємодії з об'єктами та визначення кольору кожного пікселя на екрані відповідно до інформації, зібраної в результаті променевого трасування.

Цей алгоритм представляє функцію `raymarch`, яка використовує техніку `ray marching` для відтворення графічного зображення на основі трасування променів в тривимірному просторі. Основна ідея полягає в тому, щоб прослідкувати промінь, який виходить з точки спостереження (`ro`) в напрямку (`rd`) і взаємодіє з об'єктами у сцені.

Код реймарчеру:

```
float4 raymarch(float3 ro, float3 rd)
{
    float4 pixelColor = float4(0, 0, 0, 0);
    const int maxstep = 250;
    const float drawdist = 40;
    float t = 0;
    [loop]
    for (int i = 0; i < maxstep; ++i)
    {
        float3 p = ro + rd * t;
        float4 objectData = GetDistance(p);
        if (objectData < 0.001)
        {
            pixelColor = float4(1, 1, 1, 1);
            break;
        }
        else if (t > drawdist) { return pixelColor; }
        t += objectData;
    }
    return pixelColor;
}
```

Основні етапи реймарчингу включають:

- **Ініціалізація ret:** Створення змінної *ret*, яка представляє кінцевий колір пікселя. Початкове значення встановлено на чорний колір (0, 0, 0, 0).
- **Цикл реймарчингу:** За допомогою циклу для реймарчингу променя, який триває максимум 250 ітерацій. В кожній ітерації обчислюється точка *p* на промені (розташована на відстані *t* від точки спостереження).
- **Отримання даних об'єкта:** Викликається функція *GetDistance(p)*, яка повертає дані про відстань від точки *p*.
- **Перевірка зіткнення з об'єктом:** Перевіряється, чи відстань до об'єкта менша за певне значення (0.001). Якщо так, це означає зіткнення з об'єктом, і виконуються подальші обчислення.
- **Вихід з циклу:** Якщо точка вийшла за максимально встановлену відстань для відображення (*drawdist*), функція повертає поточне значення *ret*.
- **Ітерація та оновлення t:** Якщо зіткнення не виявлено, відстань *t* до точки на промені оновлюється на відстань до найближчого об'єкта.
- **Повернення значення ret:** Після виконання циклу функція повертає кінцеве значення *ret*, яке представляє колір пікселя, обчислений за допомогою реймарчингу.

Візуальне зображення роботи цього алгоритму було представлено на (Рисунку 1.6).

Після додавання на сцену сфери за формулою (1.2) результатом використання функції *raymarch* буде відображення сцени з урахуванням нового об'єкта. Сфера, додана на сцену, буде взаємодіяти з променем та впливати на кінцевий колір пікселя (Рисунок 2.2).

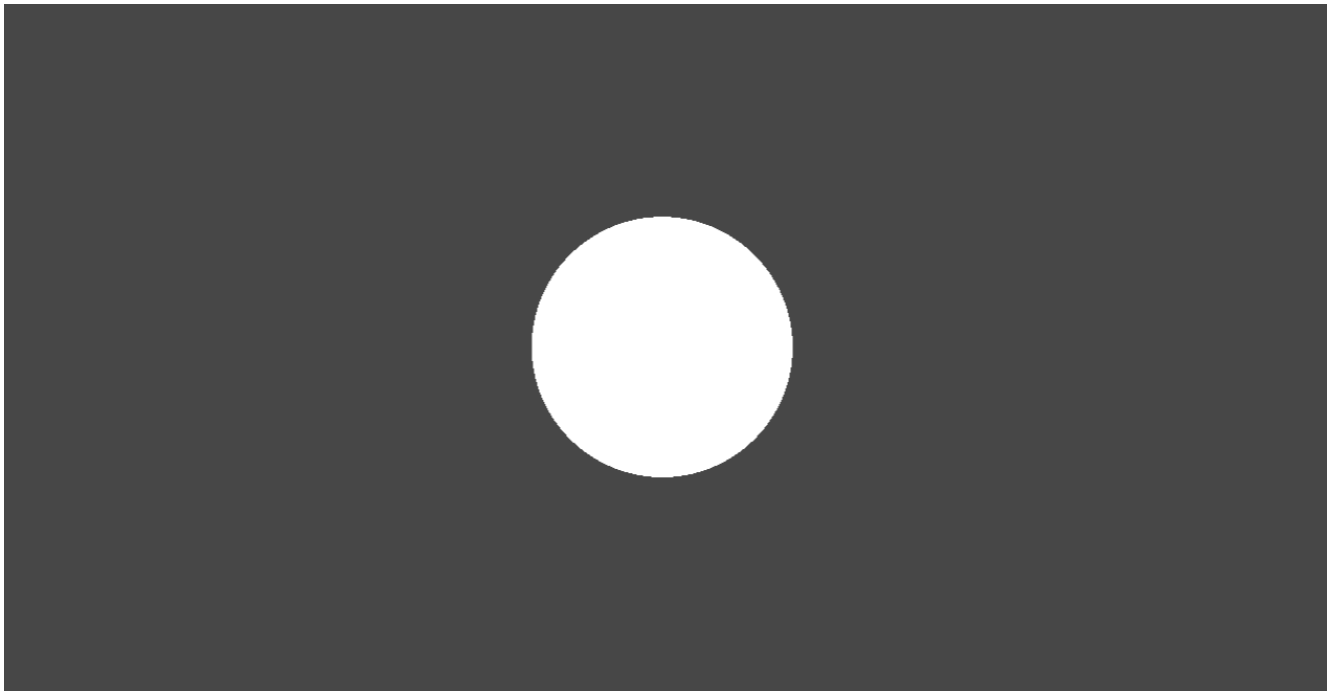


Рисунок 2.2 Відображення сфери

На рисунку 2.2 стає очевидним, що розпізнавання сфери у сцені ускладнюється відсутністю чіткого освітлення і нестачею визначеності у текстурі чи матеріалі сфери. Для покращення візуального сприйняття об'єкта та виокремлення його форми, ми можемо додати обчислення нормалей.

Цей алгоритм представляє функцію `calcNormal`, яка використовується для обчислення нормалі для точки на поверхні об'єкта. Нормалі є важливими для розрахунку освітлення та тіней на поверхні, що допомагає покращити візуальну якість зображення.

Код розрахунку нормалей:

```
float3 calcNormal(in float3 p)
{
    float d0 = GetDistance(p);
    const float2 epsilon = float2(.001, 0.0);
    float3 d1 = float3(
        GetDistance(p - epsilon.xyy),
        GetDistance(p - epsilon.yxy),
        GetDistance(p - epsilon.yyx));
    return normalize(d0 - d1);
}
```

Основні етапи цієї функції включають:

- **Отримання відстані до поверхні:** Викликається функція `GetDistance(p)` для отримання відстані від точки p до поверхні об'єкта. Значення зберігається в $d0$.
- **Визначення зміщень для обчислення нормалі:** Визначається `epsilon`, що представляє собою дуже малі значення. Створюється вектор $d1$, який містить відстані від точки p до поверхні в різних напрямках, використовуючи зміщення на `epsilon` вздовж координатних осей.
- **Обчислення нормалі:** Використовуючи отримані значення, обчислюється нормалі для точки p . Різниця між відстанями $d0$ та $d1$ для кожного напрямку визначає компоненти нормалі. Результат нормалізується, щоб забезпечити одиничну довжину.

Результат роботи цієї функції можна побачити на Рисунку 2.3.

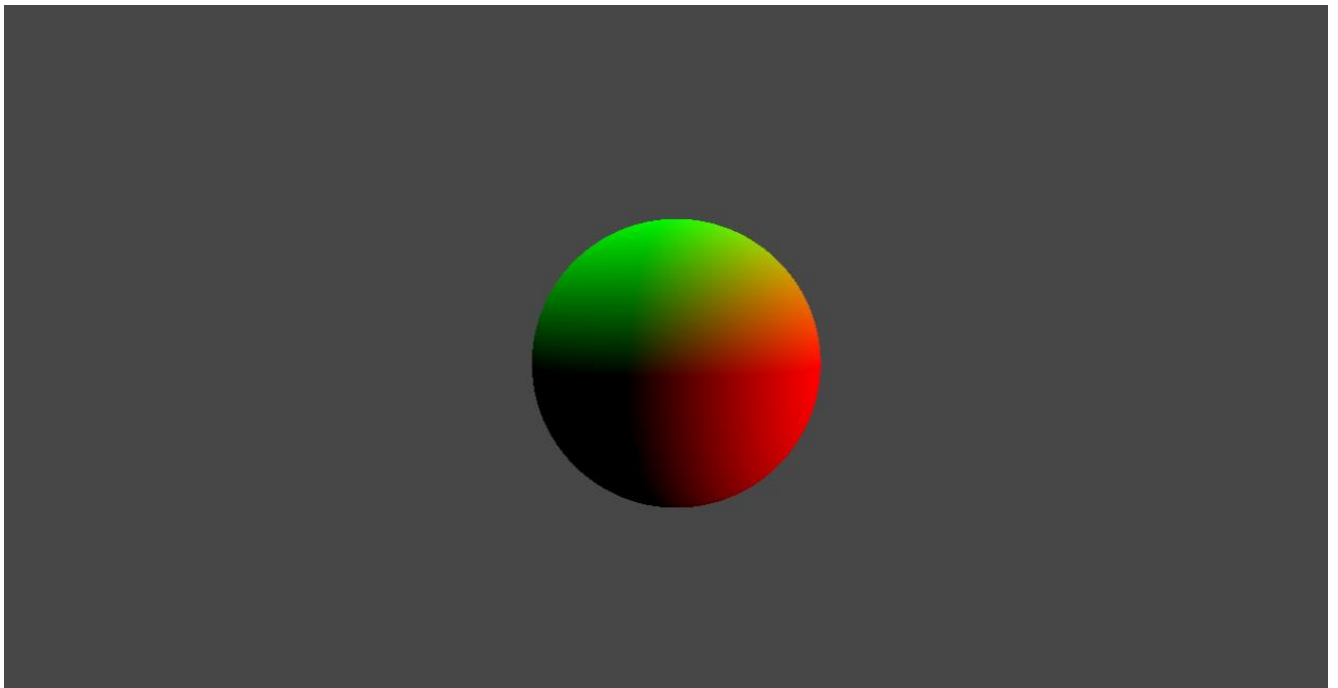


Рисунок 2.3 Відображення нормалей сфери за допомогою кольорів

Після визначення нормалей для точок на поверхні об'єкта ми можемо додати дифузійне освітлення для покращення візуальної якості сцени. Дифузійне освітлення враховує напрямок, під яким падає світло на поверхню, і визначає, наскільки ця поверхня освітлюється.

Зазвичай для розрахунку дифузійного освітлення використовують формулу Ламберта, яка виглядає наступним чином:

$$I_{diffuse} = \max(0, L * N) \quad (2.1)$$

Де L - вектор напрямку світла, N - нормаль до поверхні в даній точці. Результат дифузійного освітлення ($I_{diffuse}$) обмежується від 0 до 1.

За допомогою обчислених нормалей у функції `calcNormal` та вектора напрямку світла можна застосувати формулу Ламберта для кожного пікселя на поверхні об'єкта. Це дозволить створити м'які та реалістичні тіні, залежні від кута падіння світла на поверхню об'єкта (Рисунок 2.4).

Такий підхід додає візуальну глибину та реалістичність до сцени, дозволяючи кольорам та текстурам виглядати більш природньо на освітленій поверхні.

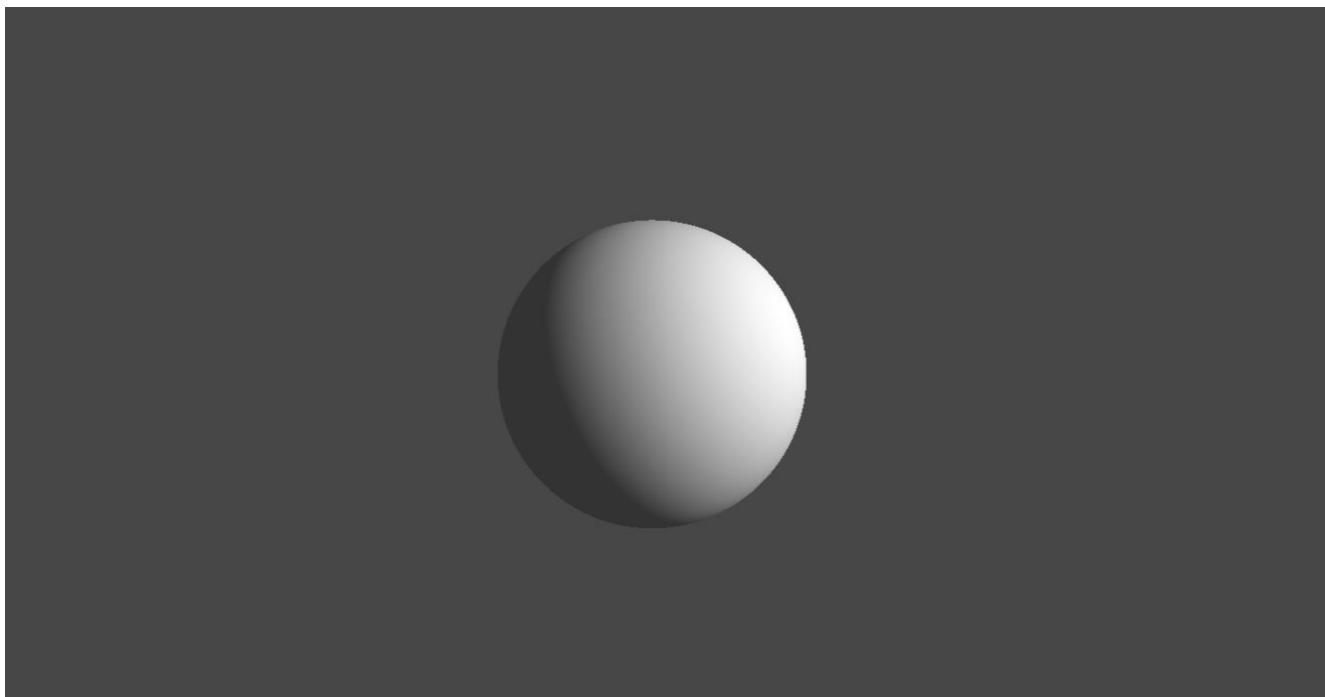


Рисунок 2.4 Дифузійне освітлення сфери

Після впровадження дифузійного освітлення та врахування нормалей для об'єктів на сцені можливо розширити сцену, додаючи ще декілька фігур. Це включає в себе розміщення та конфігурацію нових геометричних об'єктів, таких як сфери, прямокутники, чи інші форми (Рисунок 2.5).

Приклад коду сфери:

```
float SphereDistance(float3 p, float3 center, float r)
{
    return distance(p, centre) - r;
}
```

Приклад коду куба:

```
float CubeDistance(float3 p, float3 center, float3 s)
{
    float3 o = abs(p.xyz - centre) - s;
    float ud = length(max(o, 0));
    float n = max(max(min(o.x, 0), min(o.y, 0)), min(o.z, 0));
    return ud + n;
}
```

Приклад коду торуса:

```
float TorusDistance(float3 p, float3 centre, float r1, float r2)
{
    float2 q = float2(length((p.xyz - centre).xz) - r1, p.xyz.y - centre.y);
    return length(q) - r2;
}
```

Кожен новий об'єкт може мати свою власну текстуру, матеріал, та параметри освітлення. Застосовуючи ті самі принципи реймарчингу та обчислення нормалей для нових об'єктів, можливо створити складні та інтересні візуальні композиції.

Здатність додавати декілька об'єктів на сцену дозволяє створювати більш складні та захоплюючі графічні сцени. Для кожного об'єкта можна налаштовувати його взаємодію зі світлом, тінями, та іншими елементами сцени.

Це відкриває можливості для творчості та дозволяє створити різноманітні та цікаві графічні сцени, які відображають різні об'єкти та їх взаємодію у тривимірному просторі.

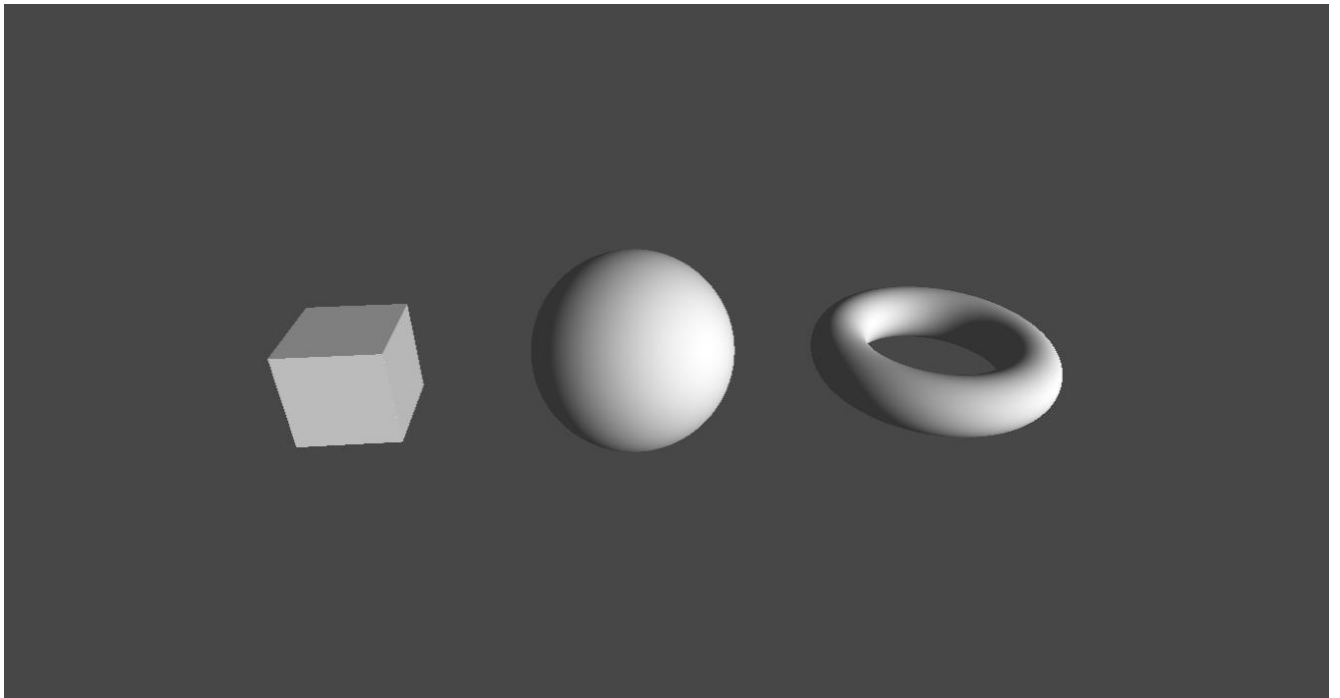


Рисунок 2.5 3-вимірні примітиви

Після розширення сцени декількома об'єктами було додано SDF HeightMap для відображення рельєфу сцени за допомогою карти висот. За допомогою SDF HeightMap можливо створювати рельєфні поверхні та відтворювати складні ландшафти або геометрію.

У цьому алгоритмі визначається функція `sdfHeightMap`, яка використовується для обчислення відстані до певної точки у тривимірному просторі.

Код функції `sdfHeightMap`:

```
float sdfHeightMap(float3 p, float2 scale)
{
    float3 b = float3(scale.x / 2.0f, 0.1, scale.y / 4.0f);
    float3 q = abs(p) - b;
    float g = sin(atan2(1, 2));
    float2 uv = getUv(p, scale);
    float2 texCoord = (uv / scale - float2(0.5, 0.5));
    float h = tex2D(_HeightMapTex, texCoord);
    h *= 0.03;
    q.y -= h;
    q.y *= g;
    q.x = max(0, q.x);
}
```

```

        q.y = max(0.0, q.y);
        q.z = max(0.0, q.z);
        float dist = length(q);
        return dist;
    }
float2 getUv(float3 p, float2 scale)
    {
        float3 b = float3(scale.x / 2.0f, 0.1, scale.y / 4.0f);
        float2 uv = float2(min(p.x, b.x), min(p.z, b.z));
        float aspectRatio = _HeightMapTex_TexelSize.y / _HeightMapTex_TexelSize.x;
        uv.y *= aspectRatio;
        return uv;
    }

```

Давайте розглянемо основні етапи цієї функції:

- **Ініціалізація параметрів:** Визначаються параметри **b** (розміри блоку), **q** (вектор відстаней від точки до блоку) та **uv** (координати текстури).
- **Корекція аспектного відношення:** Коригується аспектне відношення **uv** координат відносно розмірів текстури **_HeightMapTex**.
- **Отримання висоти з HeightMap:** Обчислюється **texCoord**, щоб визначити координати текстурного пікселя в області блоку. Потім висота (**h**) отримується з текстури **_HeightMap** та масштабується.
- **Коригування відстані від точки до блоку:** Відстань **q.y** коригується відповідно до значення **g** (синус арктангенса відношення 1 до 3). Це допомагає створити плавні переходи та змінюваність у висоті.
- **Виправлення від'ємних значень:** **q.x**, **q.y** та **q.z** виправляються так, щоб вони були не менше 0.
- **Обчислення відстані:** Обчислюється відстань **dist** як довжина вектора **q**.
- **Повернення значення відстані:** Функція повертає обчислену відстань **dist**. Це значення використовується в реймарчингу для обчислення, як далеко точка знаходиться від поверхні.

Карта висот (або **HeightMap**) представляє собою двовимірне графічне зображення, де кожен піксель кодує висоту відповідної точки на поверхні

тривимірного об'єкта чи ландшафту. Зазвичай, чорний колір відповідає найнижчій точці (мінімальній висоті), а білий — найвищій точці (максимальній висоті).

Використання карт висот є поширеним способом відтворення рельєфу та ландшафту у комп'ютерній графіці. Ці карти дозволяють створювати деталізовані та реалістичні віртуальні ландшафти.

Для прикладу ми використовуватимемо карту висот Закарпаття (Рисунок 2.6). Ця конкретна карта висот містить інформацію про рельєф регіону Закарпаття в Україні. Використовуючи дану карту висот, ми зможемо відобразити гірські хребти, долини та інші географічні особливості цього регіону у вигляді тривимірної графічної сцени.

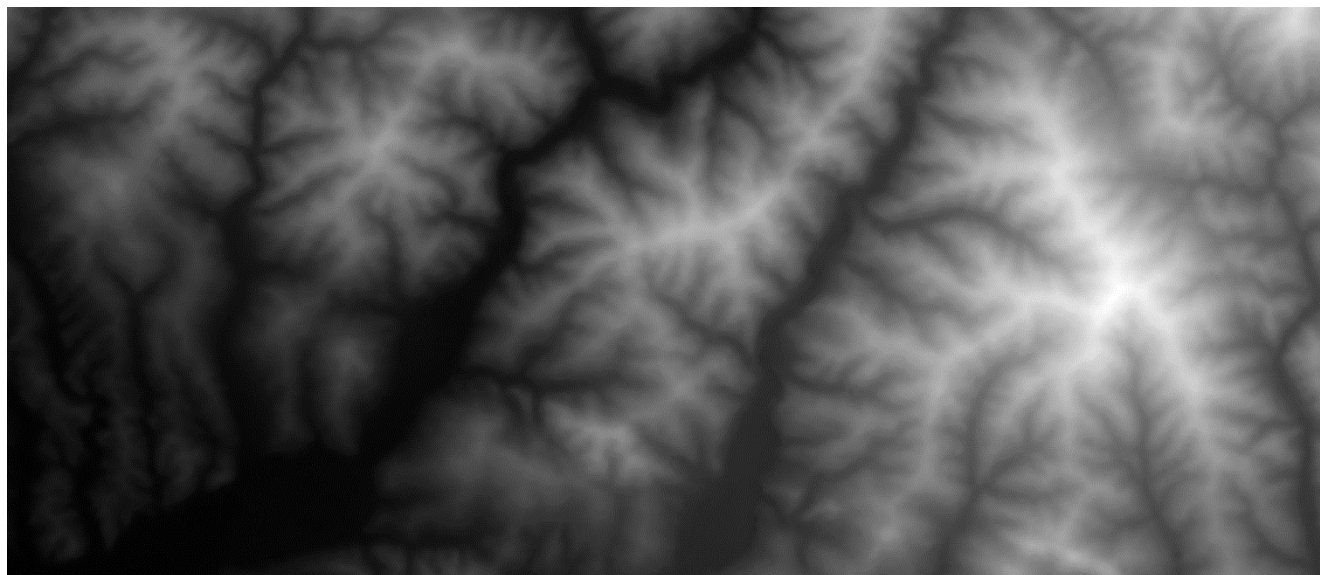


Рисунок 2.6 Карта висот Закарпаття

Беручи карту висот Закарпаття для прикладу роботи `sdHeightMap`, ми отримуємо можливість відобразити рельєф цього регіону у тривимірному просторі. Карта висот Закарпаття містить інформацію про висоту різних частин ландшафту, включаючи гори, долини та інші географічні особливості.

Використовуючи функцію `sdHeightMap` з цією картою висот, можна створити вражаюче візуальне представлення рельєфу Закарпаття в тривимірному графічному середовищі (Рисунок 2.7). Значення відстані, отримані з `sdHeightMap`, визначатимуть висоту точок на сцені відносно вказаного рельєфу.

Розміщення об'єктів на сцені, освітлення та інші властивості можна налаштувати відповідно до характеристик рельєфу, що дозволяє створювати віртуальні репліки реального ландшафту.

Такий підхід дозволяє використовувати реальні географічні дані для створення, які відображають природні форми та контури рельєфу реальних ландшафтів.

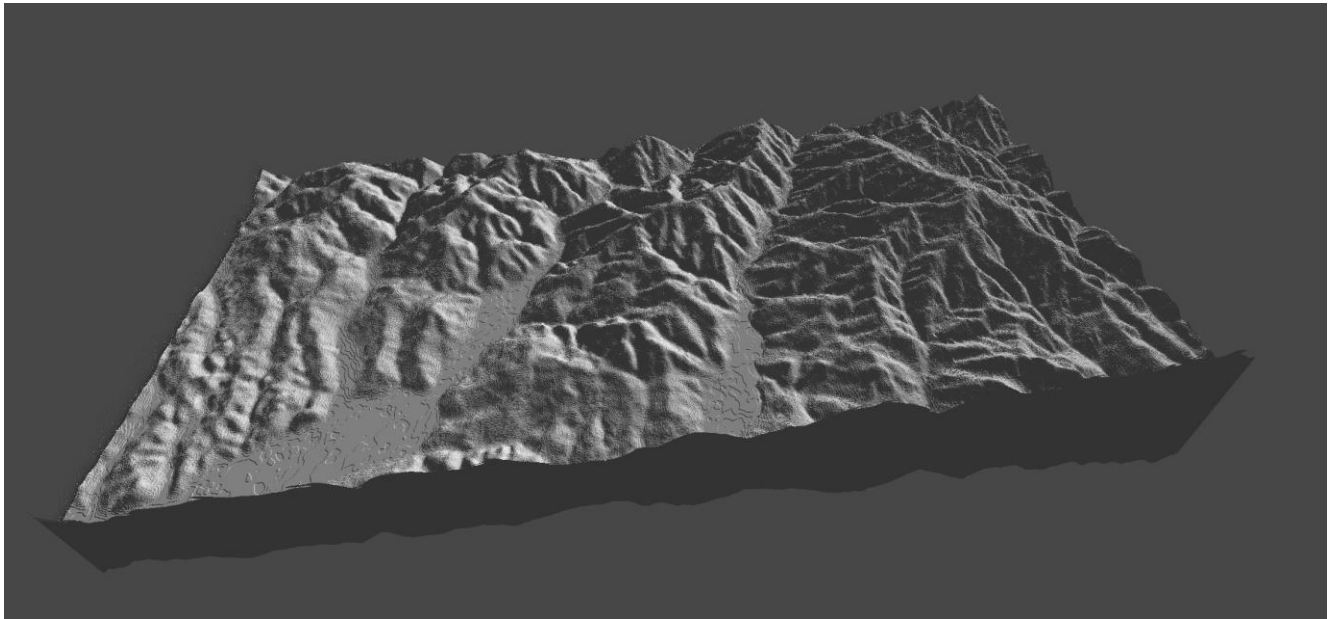


Рисунок 2.7 Створений рельєф за допомогою карти висот

Після використання карт висот та відображення рельєфу Закарпаття на сцені, можна врахувати додавання тіней для поліпшення візуального вигляду. Тіні додають глибину та реалізм до тривимірної сцени, створюючи враження взаємодії об'єктів та джерела світла (Рисунок 2.8).

Код функції для створення тіней:

```
float shadow(float3 p)
{
    float3 lightDir = -_LightDir;
    float kd = 1;
    int step = 0;
    [loop]
    for (float t = 0.1; step < 30 && kd > 0.001;)
    {
        float d = GetDistance(p + t * lightDir);
        if (d < 0.001) { kd = 0; }
```

```

    else { kd = min(kd, 16 * d / t); }
    t += d;
    step++;
}
return kd;
}

```

Основні етапи цього алгоритму можна описати так:

- **Ініціалізація параметрів:** Визначення напрямку світла (`lightDir`), коефіцієнта падіння світла (`kd`), та початкового значення змінної для кількості ітерацій (`step`).
- **Обчислення тінів:** Використання циклу для проведення тіньового тесту для кожного пікселя на поверхні. В цьому тесті вимірюється відстань вздовж напрямку світла (`lightDir`) від точки на поверхні до перешкоди.
- **Оцінка видимості:** Якщо відстань до перешкоди менше певного порогового значення (0.001), це вказує на те, що точка перебуває у тіні. В іншому випадку, визначається коефіцієнт падіння світла (`kd`), який визначає ступінь затемнення в залежності від відстані та порогового значення.
- **Оновлення параметрів:** Переход до наступної ітерації з оновленням змінної `t` та обмеженням на кількість ітерацій.
- **Повернення коефіцієнта тіней:** Повертається оцінка коефіцієнта падіння світла (`kd`), який використовується для модифікації освітлення в реймарчингу.

Цей алгоритм дає можливість враховувати тіні в залежності від розташування джерела світла та перешкод на шляху світла до поверхні. Така техніка додає реалізм до візуального представлення сцени та покращує сприйняття глибини.

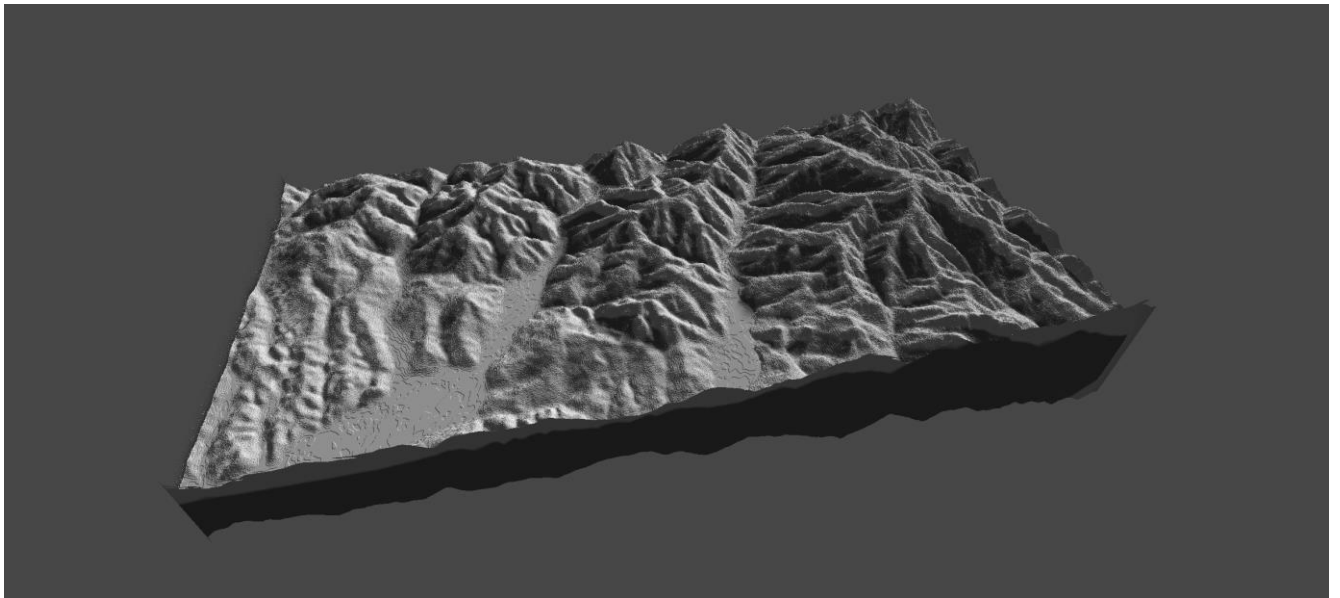


Рисунок 2.8 Додавання тіней до сцени

Тепер, коли у нас вже є рельєф Закарпаття, тіні та інші ефекти, ми можемо додати зображення (Рисунок 2.9) на поверхню цього рельєфу, надаючи сцені більш природний та деталізований вигляд (Рисунок 2.10).

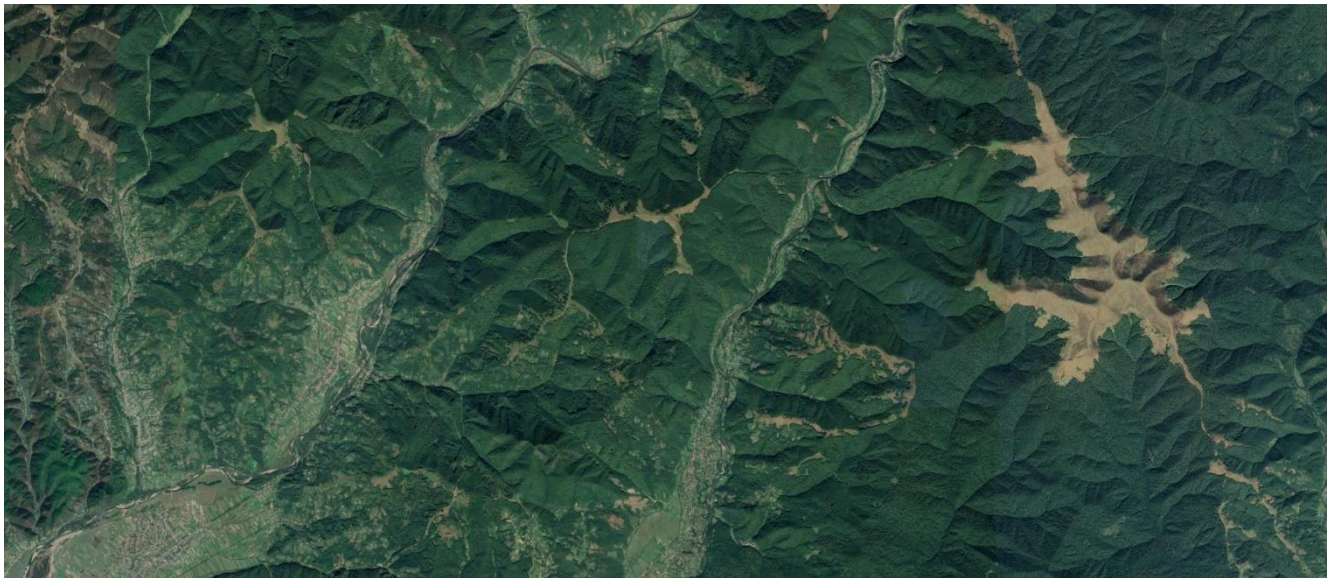


Рисунок 2.9 Зображення Закарпаття

У цьому алгоритмі використовується функція `getHeightMapColor`, яка дозволяє отримати колір для кожної точки рельєфу на основі зображення (текстури), яке використовується як картографія висот.

Код отримання кольору поверхні рельєфу:

```
float3 getHeightMapColor(float3 p, float2 scale)
{
    float2 uv = getUv(p, scale);
    float2 texCoord = (uv / scale - float2(0.5, 0.5));
    return float3(tex2D(_ReliefTex, texCoord).rgb);
}
```

Основні етапи цього алгоритму:

- **Ініціалізація параметрів:** Визначається розмір блоку (b), а також обчислюються координати текстурного пікселя (texCoord) для використання у подальших операціях.
- **Корекція аспектного відношення:** Коригується аспектне відношення uv координат відносно розмірів текстури.
- **Обчислення текстурних координат:** Використовуються обчислені uv координати та масштабу, щоб отримати фінальні текстурні координати для використання у функції tex2D.
- **Отримання кольору з текстури:** Використовується функція tex2D для отримання кольору з текстури (_ReliefTex) за вказаними текстурними координатами.
- **Повернення результату:** Повертається колір зображення для використання у подальших візуальних операціях.

Цей алгоритм дозволяє використовувати зображення в якості текстури для рельєфу, що може значно підвищити деталізацію та реалістичність віртуального середовища.

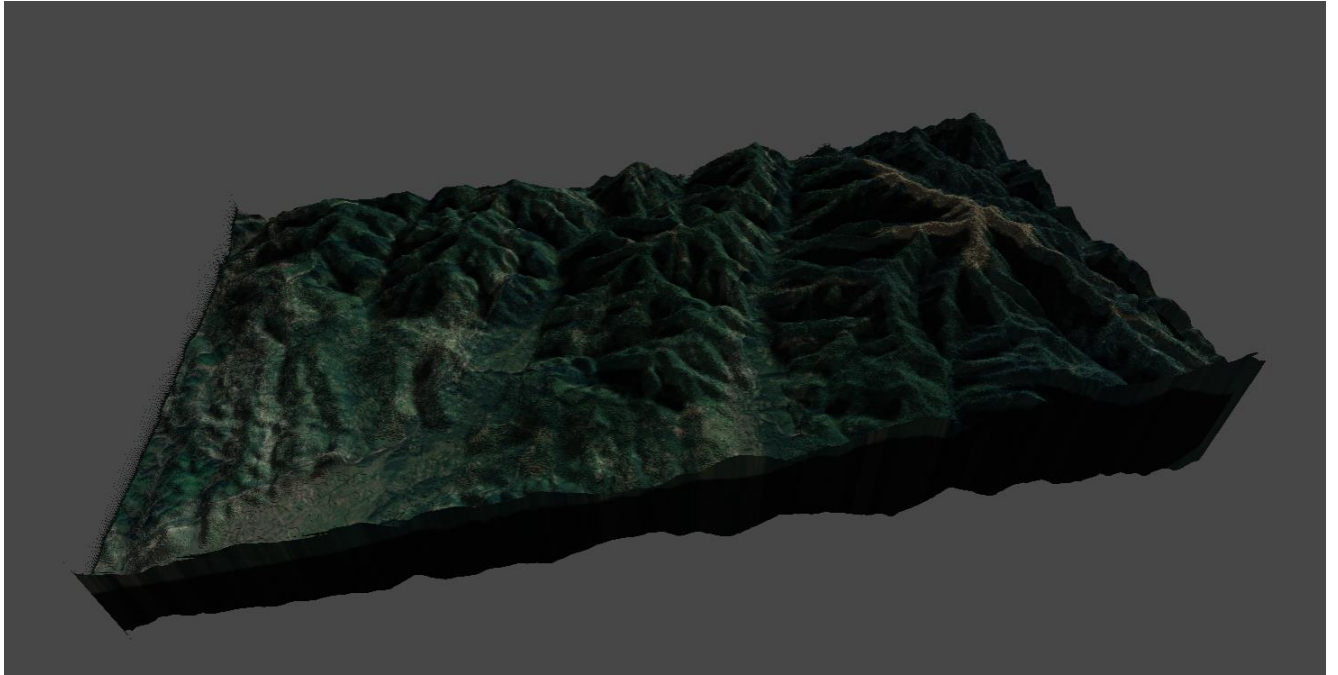


Рисунок 2.10 Відображення зображення Закарпаття на його рельєфі

Тепер, коли вже було успішно додано основні елементи, такі як рельєф, тіні, текстури та інші важливі ефекти, можемо розглянути можливість створити інтерактивний засіб для користувача.

2.2.2 Створення прикладу візуалізації результатів моніторингу атмосферного повітря за допомогою SDF-графіки

Для початку ми використаємо карту висот Києва (Рисунок 2.11), яка надасть нам інформацію про рельєф та висотні особливості міста. Ця карта висот дозволить нам створити тривимірний рельєф, відображаючи характерні зміни висот на поверхні.

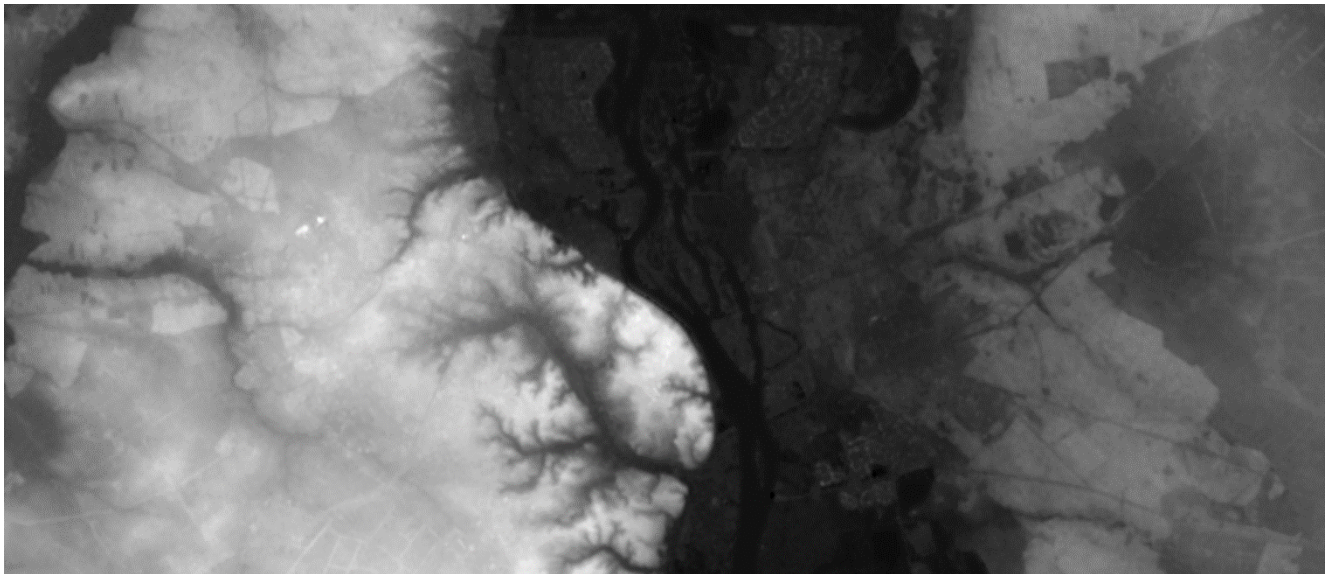


Рисунок 2.11 Карта висот Києва

Зображення Києва (Рисунок 2.12) буде використано як текстура для деталізації поверхні рельєфу. Ця текстура додасть деталі міста, такі як дороги та інші особливості.

Комбінація цієї карти висот та текстури дозволить нам створити візуальний образ Києва у віртуальному середовищі. Після цього ми зможемо розглянути додавання інтерактивних елементів та інших деталей для збагачення віртуального досвіду користувача.

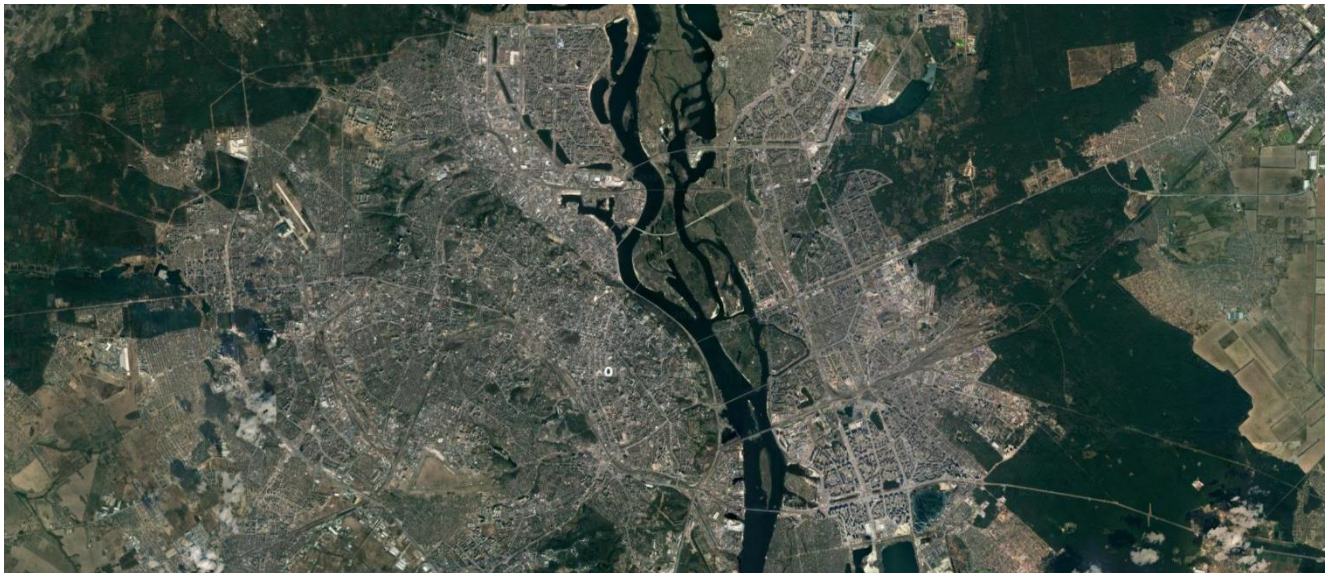


Рисунок 2.12 Зображення Києва

Використовуючи описані вище методи та етапи, ми отримаємо результат віртуального відображення Києва (Рисунок 2.13). Рельєф, створений за

допомогою карт висот, передасть рельєфні особливості міста, чи інші висотні контури.



Рисунок 2.13 Зображення Києва та його рельєфа

Для відображення місць моніторингу забруднення атмосферного повітря ми будемо використовувати сфери. Сферичні об'єкти нададуть нам можливість виділити та візуалізувати ці місця на віртуальній карті міста.

Використання сфер дозволить нам легко ідентифікувати місця моніторингу забруднення, а їх розташування на поверхні міста буде інтуїтивно зрозумілим для користувача.

Такий підхід сприяє наглядності та зрозумілості відображення даних про якість повітря у віртуальному місті (Рисунок 2.14).

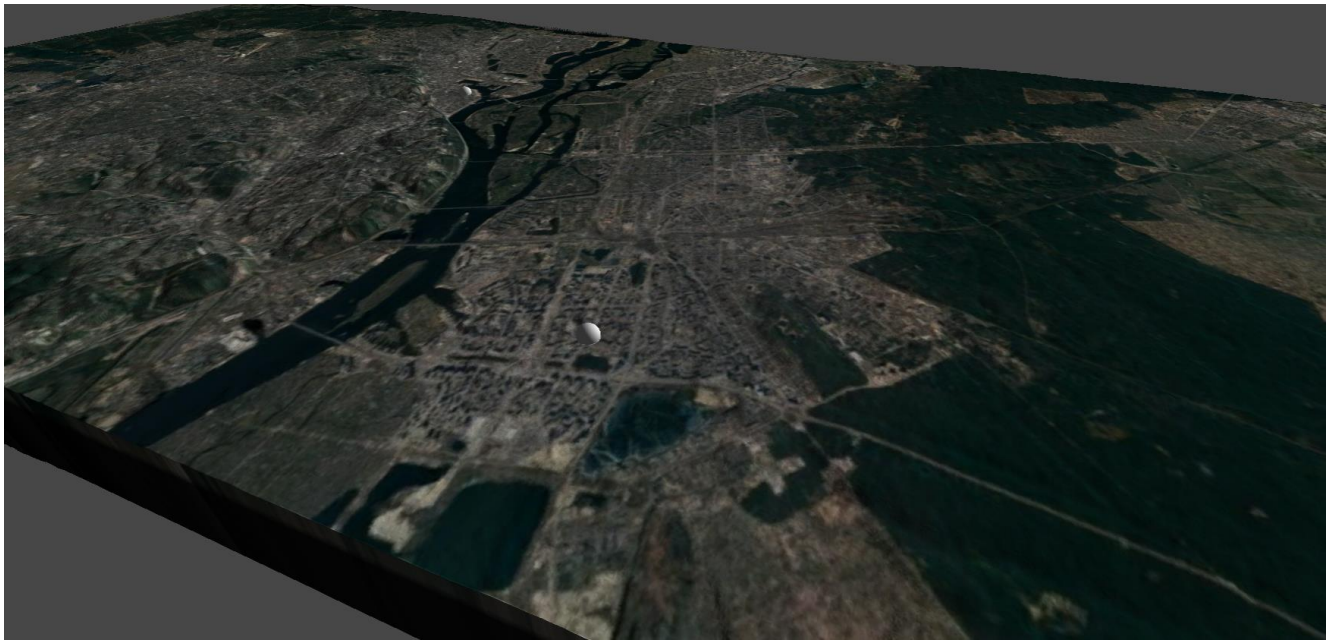


Рисунок 2.14 Зображення місця моніторингу за допомогою сфери

Далі ми плануємо додаємо інтерфейс користувача, який дозволить вибрати конкретний забрудник атмосфери, за яким користувач може спостерігати зміни рівня забруднення. Цей інтерфейс надасть користувачеві можливість вибору та вивчення конкретних даних про якість повітря (Рисунок 2.15).

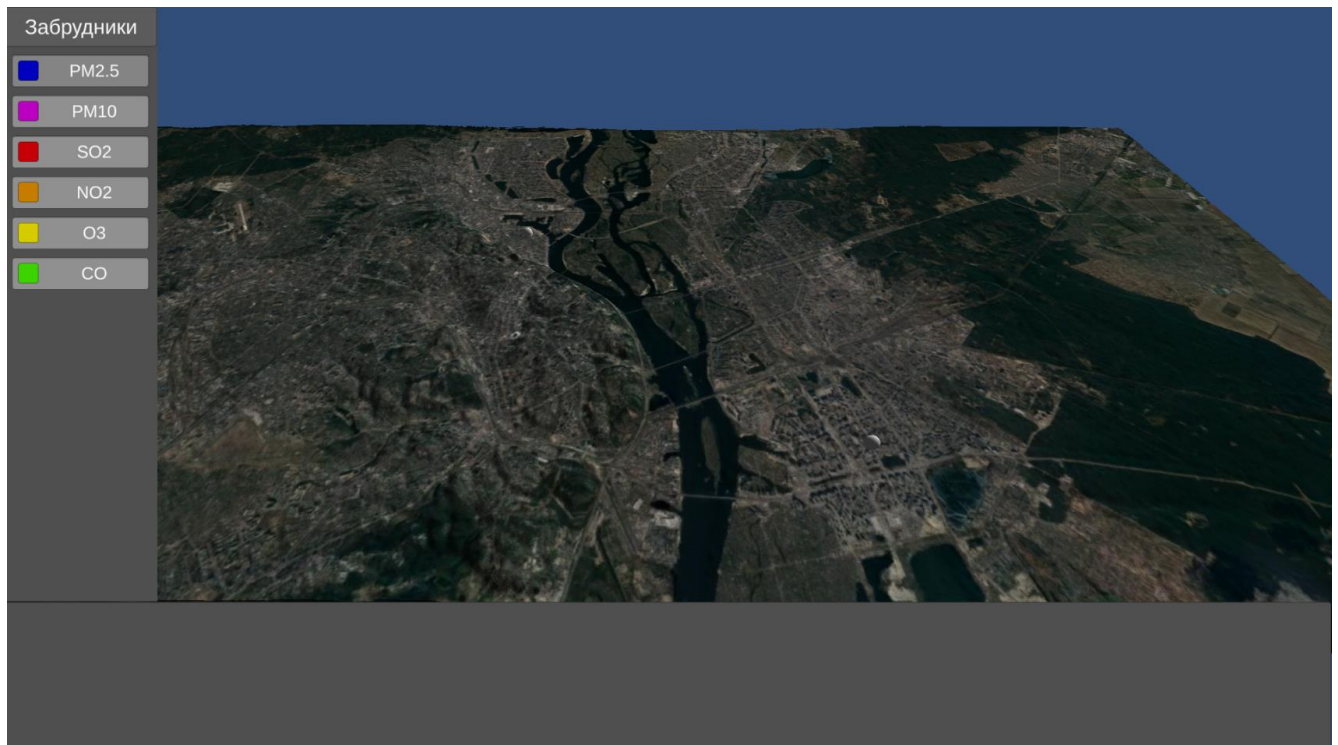


Рисунок 2.15 Інтерфейс та забрудники

При виборі конкретного забрудника в інтерфейсі він буде позначений зеленим кольором, щоб виділити його серед інших. Всі сфери, які відповідають за місця моніторингу становитимуться тим кольором, який позначений біля назви обраного забрудника (Рисунок 2.16).

Це візуальне позначення спростить користувачеві спостереження та взаємодію з обраним забрудником та місцями моніторингу. Зелений колір та відповідна позначка на сферах вказуватимуть користувачеві на те, які саме області повітряного простору його обраного забрудника наразі піддаються огляду.

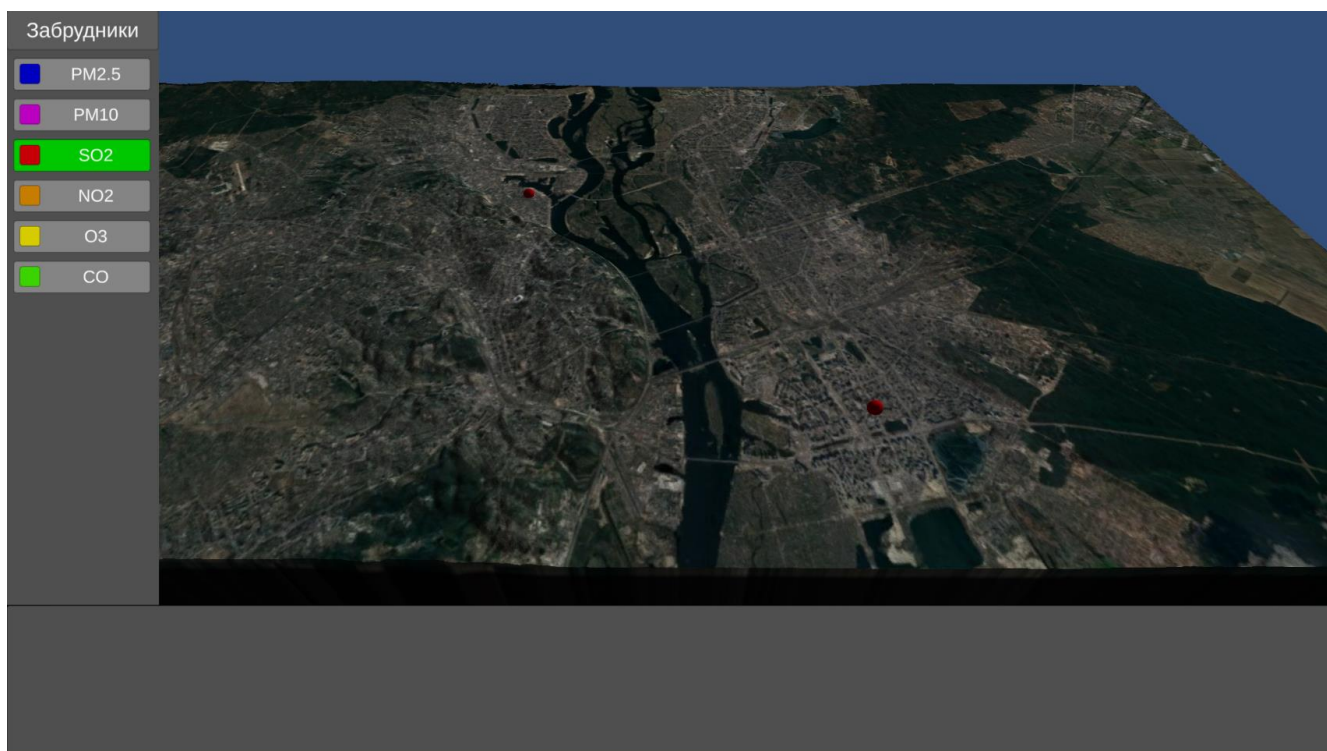


Рисунок 2.16 Зміна кольору сфери залежно від вибраного забрудника

Далі було додано в інтерфейс "Налаштування", де користувач може налаштувати відображення "Назву вулиці" та "Значення забрудника". В налаштуваннях користувач може визначити, яка інформація буде відображатися над сферами (Рисунок 2.17).

Можливі параметри налаштувань включають:

- **Назва вулиці:** Вмикання або вимикання відображення назв вулиць, щоб користувач міг зорієнтуватися у розташуванні місць моніторингу на мапі міста.

- **Значення забрудника:** Вмикання або вимикання відображення значень рівня забруднення над сферами, щоб користувач мав можливість отримати інформацію про стан атмосферного повітря безпосередньо на карті.

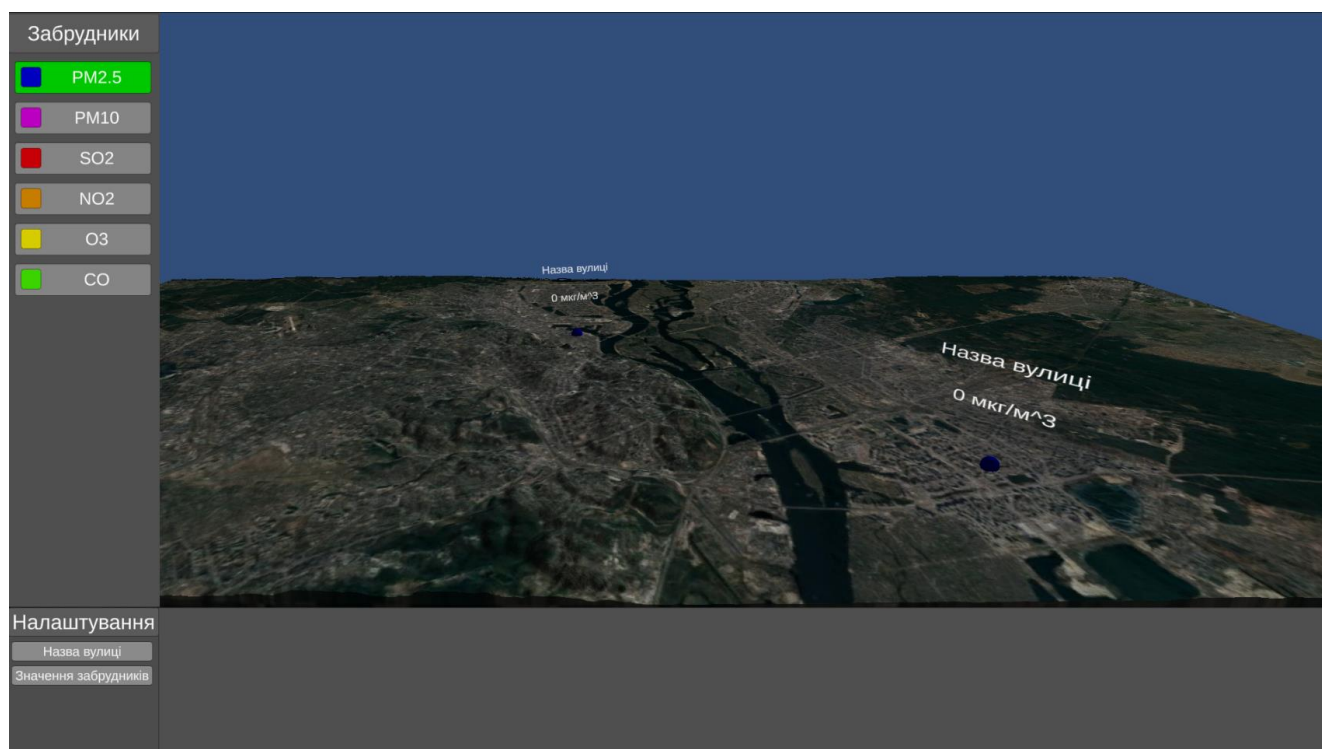


Рисунок 2.17 «Налаштування» для відображення інформації про місце моніторингу

Далі було додано до інтерфейсу "Часову шкалу", яка відповідала за встановлення проміжку часу для перегляду даних. У цій частині інтерфейсу користувач може вказати, за який період часу він бажає отримати інформацію про рівень забруднення повітря.

Окремо в інтерфейсі додано слайдер для зміни дати переглядаємої інформації. Користувач може вручну встановлювати дату або пересувати слайдер, щоб переглядати дані за різні дні чи періоди. Це дозволяє здійснювати динамічний аналіз та спостереження за змінами рівня забруднення в різний час (Рисунок 2.18).

Ці додаткові можливості інтерфейсу "Часова шкала" дозволяють користувачеві більш гнучко налаштовувати часові рамки та взаємодіяти з динамічними даними щодо забруднення повітря у вибраних областях міста.

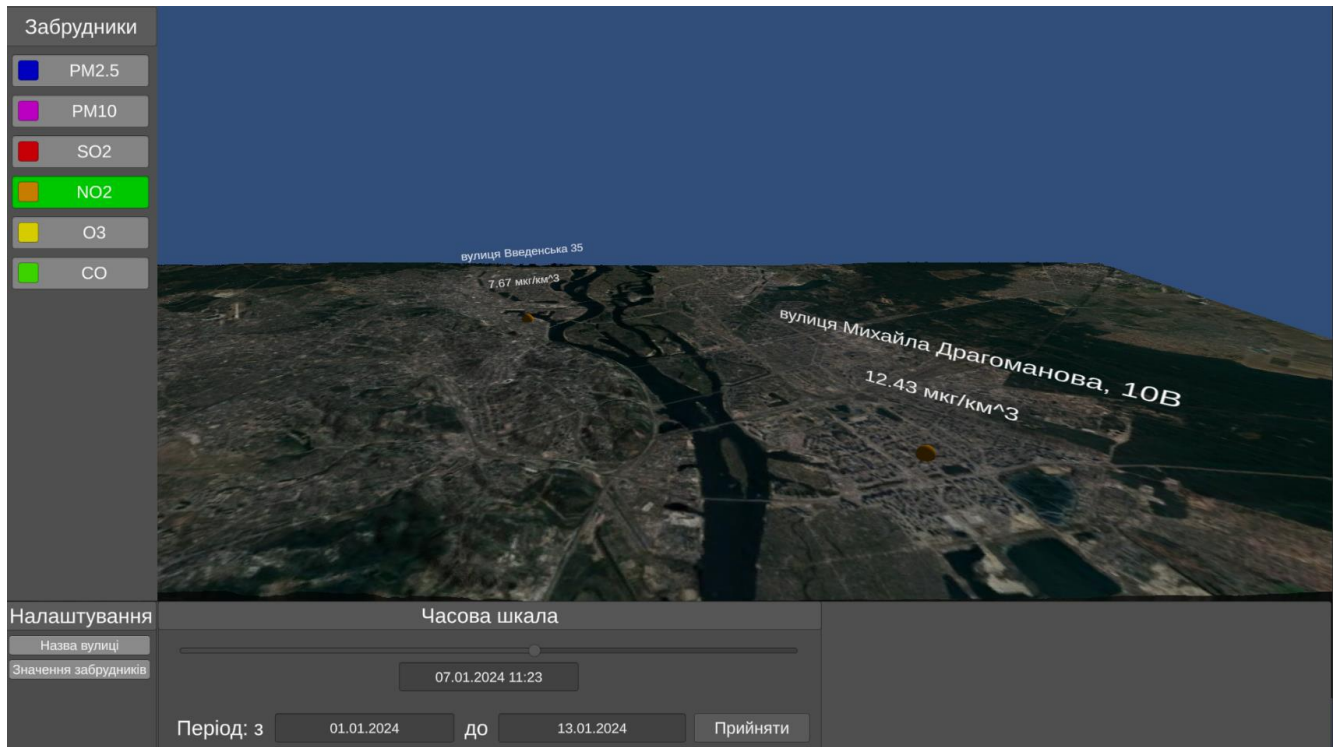


Рисунок 2.18 Відображення «Часова Шкала» в інтерфейсі

Завершуючи ми успішно розробили віртуальний інтерфейс для моніторингу рівня забруднення атмосфери у місті, зокрема для місць моніторингу, які відображені сферами на карті. Наша система дозволяє користувачеві не лише отримувати дані про якість повітря, а й зручно взаємодіяти та налаштовувати відображення інформації.

Інтерфейс включає в себе інтерактивне відображення забрудників, а також можливість налаштовувати відображення назв вулиць та значень забрудників. Крім того, за допомогою "Часової шкали" користувач може визначити період часу для аналізу та відслідковування динаміки змін.

З впровадженням цих функцій та налаштувань, наш віртуальний моніторинг забруднення повітря стає потужним інструментом для обліку та аналізу якості повітря у місті. Такий інтерфейс не лише надає інформацію, а й робить її доступною та зрозумілою для широкого кола користувачів, сприяючи збереженню та покращенню якості навколишнього середовища.

РОЗДІЛ 3. ОЦІНКА ЕФЕКТИВНОСТІ ЗАСТОСУВАННЯ МЕТОДІВ SDF-ГРАФІКИ ТА ВИЗНАЧЕННЯ МОЖЛИВОСТЕЙ І ШЛЯХІВ ЇХ ОПТИМІЗАЦІЇ ДЛЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ СПРИЙНЯТТЯ ТА АНАЛІЗУ РЕЗУЛЬТАТІВ

В сучасному світі, де екологічні проблеми стають надзвичайно актуальними, застосування новітніх технологій в моніторингу довкілля є критично важливим завданням. Серед інноваційних підходів вирізняється використання методів SDF-графіки для візуалізації та аналізу даних про забруднення повітря та інших аспектів навколишнього середовища. Ці методи дозволяють створити реалістичні та інтуїтивно зрозумілі зображення, що сприяє підвищенню ефективності сприйняття інформації користувачем.

Однак, для того щоб SDF-графіка стала невід'ємною складовою екологічного моніторингу, необхідно не лише визнати її потенціал, а й провести глибоку оцінку її ефективності. У цьому контексті, вивчення можливостей оптимізації стає ключовим завданням. Забезпечення швидкодії, точності та плавності відображення є важливими аспектами для забезпечення якісного аналізу та сприйняття результатів моніторингу.

Таким чином, даний розділ розкриває застосування SDF-графіки в моніторингу довкілля та визначає основні аспекти, які підлягають оцінці. Зосереджуючись на можливостях оптимізації, ми прагнемо визначити шляхи підвищення продуктивності та забезпечення максимально ефективного використання SDF-графіки для інформаційного аналізу та взаємодії з результатами моніторингу довкілля.

2.1 Оцінка ефективності SDF-графіки

В цьому розділі ми докладно розглянемо ефективність застосування методів SDF-графіки. Зосереджуючись на важливих аспектах візуалізації та аналізу даних про довкілля, ми визначимо та оцінимо ключові характеристики SDF-графіки, які впливають на її ефективність у сфері моніторингу.

Першим кроком у нашому дослідженні буде вивчення візуальної чіткості та зрозумілості SDF-графіки. Ми розглянемо, наскільки точно та зрозуміло вона передає складні дані про забруднення довкілля, а також які аспекти впливають на сприйняття інформації користувачем.

Далі ми звернемо увагу на динамічність та адаптивність SDF-графіки. Оцінюючи її здатність адаптуватися до змін у середовищі, ми визначимо, наскільки ефективно вона відображає динаміку забруднення повітря та інших параметрів в реальному часі.

Третім аспектом для розгляду буде взаємодія з користувачем. Ми вивчимо, наскільки ефективно користувач може взаємодіяти з SDF-графікою, регулюючи параметри та отримуючи додаткову інформацію для більш глибокого розуміння даних.

3.1.1 Візуальна чіткість та зрозумілість

Вивчаючи «Рисунок 2.17», де був представлений приклад рельєфу Києва, разом із використанням сфер для відображення місць забруднення, можна відзначити, що SDF-графіка дійсно забезпечує чітке та зрозуміле відображення для користувача. Графічне представлення рельєфу видається деталізованим та реалістичним, а місця забруднення відзначаються відповідним чітким візуальним сигналом.

Використання сфер для позначення точок забруднення надає ефективний засіб візуалізації, що дозволяє користувачеві легко розпізнати та взаємодіяти з конкретними областями інтересу. Такий підхід підсилює зрозумілість та легкість

сприйняття інформації про рівень забруднення, забезпечуючи користувачеві чітке відображення стану довкілля.

Інформація про ступінь забруднення та точне місце знаходження передається у вигляді тексту, розташованого над кожною сферою. Це забезпечує користувачеві додаткову контекстну інформацію, яка дозволяє не лише визначити місце забруднення, а й отримати конкретні числові показники ступеня забруднення.

Це підвищує рівень зрозумілості та деталізації інформації, надаючи користувачеві можливість отримувати конкретні дані про забруднення просто з візуальної представленості. Зазначимо, що такий підхід також відкриває можливості легкої зміни типів відображення цієї інформації, що дозволяє користувачам адаптувати візуалізацію до своїх власних потреб та вподобань.

Отже, можна зробити висновки, що SDF-графіка, представлена в даному контексті, відповідає вимогам чіткості та зрозумілості візуального відображення рівня забруднення, що робить її ефективним інструментом для користувачів моніторингу довкілля.

3.1.2 Динамічність та адаптивність

При аналізі можливостей адаптації SDF-графіки до динамічних змін у рівні забруднення та передачі часових динамік, слід звернутися до «Рисунку 2.17». На цьому рисунку демонструється часова шкала, яка реалізована за допомогою слайдера. Змінюючи положення слайдера, користувач може взаємодіяти з графікою та відображати стан забруднення на певний момент часу.

Такий підхід дає можливість ефективно адаптувати візуалізацію до динамічних змін у рівні забруднення, відображаючи їхню еволюцію в часі. Користувач може спостерігати за змінами, використовуючи інтерактивний інструмент у вигляді слайдера, що спрощує взаємодію та дозволяє аналізувати часові динаміки забруднення довкілля.

Отже, SDF-графіка, представлена на «Рисунку 2.17», виявляється ефективною в передачі часових змін у рівні забруднення, надаючи користувачам зручний та інтуїтивний інструмент для спостереження за динамікою стану навколишнього середовища.

3.1.3 Взаємодія з користувачем

Аспект взаємодії з користувачем у візуалізації SDF-графіки дійсно вражає своєю різноманітністю та інтерактивністю, як продемонстровано на «Рисунку 2.17». У цьому прикладі користувач має можливість здійснювати різноманітні взаємодії за допомогою інтерфейсу.

Однією з ключових можливостей є вибір забрудника, за яким користувач спостерігатиме за даними моніторингу. Це реалізується інтерактивним вибором із списку або іншим зручним способом, що надає користувачеві можливість фокусуватися на конкретних аспектах довкілля.

Додатково, інтерфейс дозволяє користувачеві налаштовувати відображення даних про місця моніторингу, встановлювати часові рамки для аналізу інформації в залежності від потреб, масштабувати сцену для детальнішого огляду, а також обертати камеру для отримання різних перспектив.

Отже, взаємодія з користувачем у візуалізації SDF-графіки дозволяє не лише спостерігати за даними моніторингу, але й активно адаптувати відображення до власних потреб та перевіряти різні аспекти стану довкілля, надаючи користувачеві широкий спектр інтерактивних можливостей.

2.2 Оптимізація SDF-графіки

Оптимізація SDF-графіки визначається як важливий етап в розвитку та застосуванні даного методу в графічних застосунках. Структура та реалізація SDF-графіки можуть вимагати значних обчислювальних ресурсів, особливо при великих об'ємах даних чи складних сценах. У цьому підрозділі ми розглянемо різні аспекти оптимізації SDF-графіки та шляхи підвищення її ефективності.

На фоні стрімкого розвитку графічних технологій та зростання складності візуалізаційних завдань, важливим є забезпечення оптимальної продуктивності SDF-графіки. В цьому контексті, аналіз різних підходів до оптимізації є критичним для створення ефективних та швидких графічних додатків, зокрема тих, що використовують SDF-графіку.

Задачі оптимізації SDF-графіки охоплюють різні аспекти, включаючи швидкість обчислень, використання пам'яті, апаратне прискорення та алгоритмічні вдосконалення. Для розуміння та управління цими аспектами важливо вивчати та розробляти ефективні стратегії оптимізації, спрямовані на забезпечення максимальної продуктивності графічних застосунків, що використовують SDF-графіку.

3.2.1 Алгоритмічні оптимізації

Оптимізація алгоритмів відображення SDF-графіки можливо далі оптимізувати за допомогою різних методів для поліпшення продуктивності та швидкодії цих алгоритмів. Нижче розглянуті деякі з популярних методів оптимізації:

Методи реймарчингу:

- **Adaptive Step Size:** Адаптивне налаштування розміру кроку реймарчингу в залежності від градієнта або властивостей сцени дозволяє більш ефективно просуватися по променю, скорочуючи кількість кроків там, де це можливо.
- **Early Ray Termination:** Передчасне припинення реймарчингу у випадках, коли можна визначити, що промінь не зіткнеться з об'єктом, сприяє зменшенню обчислювальних витрат.

Метод конусного реймарчингу (Cone Marching):

- **Adaptive Cone Tracing:** Конусний реймарчинг - це метод, який використовує конусні промені замість точкових променів для ефективного визначення перетинів з поверхнею об'єктів. Використовуючи конуси замість

точок, можливо скоротити кількість кроків реймарчингу та отримати адаптивність до форми об'єктів.

- **Cone Angle Optimization:** Оптимізація кута конусу дозволяє більш ефективно визначати напрямок променю та зменшує кількість кроків, необхідних для зіткнення з поверхнею.

Оптимізації обчислень:

- **GPU Parallelization:** Використання паралельних обчислень на графічному процесорі (GPU) для виконання обчислень SDF-графіки дозволяє використовувати потужності сучасних відеокарт для прискорення роботи алгоритмів.
- **Approximation Techniques:** Заміна обчислювально складних функцій більш простими апроксимаціями може зменшити витрати на обчислення, при цьому забезпечуючи достатню точність для візуального відображення.

Інші методи оптимізації:

- **Bounding Volume Hierarchy (BVH):** Впорядкування об'єктів у дереві обмежувачів об'ємів дозволяє ефективно прискорити пошук перетину променю з об'єктами.

Ці методи можуть бути комбіновані для досягнення максимального ефекту оптимізації SDF-графіки в графічних застосунках.

3.2.2 Оптимізація для мобільних платформ

Оптимізація SDF-графіки для мобільних платформ важлива з урахуванням обмежених ресурсів таких пристроїв. Нижче наведено кілька конкретних технік оптимізації для покращення продуктивності на мобільних платформах:

Апаратне прискорення:

- **Vulkan API:** Використання сучасних API, таких як Vulkan, які дозволяють ефективно використовувати обчислювальні ресурси пристроїв та забезпечують кращий контроль над апаратними можливостями.

- **Mobile GPU Features:** Використання специфічних функцій, доступних на мобільних GPU, таких як tile-based rendering, щоб максимально ефективно використовувати ресурси графічного процесора.

Низькорівневе програмування:

- **Shader Optimizations:** Використання оптимізацій на рівні шейдерів, таких як усунення невикористовуваних обчислень та об'єднання інструкцій для зменшення витрат на виконання.
- **Custom GPU Shaders:** Виробництво спеціалізованих шейдерів для конкретних завдань, враховуючи обмежені можливості мобільних GPU.

Оптимізація пам'яті:

- **Texture Compression:** Використання стиснення текстур для зменшення обсягу пам'яті, використовуваної для зображення деталей.
- **Memory Management:** Ефективне управління пам'яттю, зменшення використання пам'яті та уникнення зайвих алокацій.

Також важливим аспектом оптимізації для мобільних платформ є зменшення роздільної здатності рендерингу, оскільки велика роздільна здатність може призвести до зайвих витрат обчислювальних ресурсів.

Ці техніки дозволяють оптимізувати SDF-графіку для мобільних платформ, забезпечуючи ефективну роботу на пристроях з обмеженими ресурсами.

3.2.3 Апаратне прискорення

Апаратне прискорення - це використання спеціалізованого обладнання для виконання конкретних обчислювальних або графічних завдань замість загального програмного коду на центральному процесорі (CPU). Це дозволяє виконати деякі операції значно швидше, оскільки вони виконуються на спеціальному обладнанні, призначеному саме для цього типу завдань.

В контексті графіки та обчислень, апаратне прискорення часто використовується для обробки графічних операцій, наприклад, у графічних

процесорах (GPU). GPU спеціально розроблені для виконання багатьох паралельних обчислень, які зазвичай зустрічаються у графічних застосунках. Це може включати в себе рендеринг графіки, обчислення фізики, обробку текстур, а також інші графічні завдання.

Апаратне прискорення може також використовуватися у сфері штучного інтелекту, обробки великих обсягів даних, криптографії та інших областях, де велика кількість однотипних операцій може бути ефективно оброблена за допомогою спеціалізованого обладнання.

У цій роботі при розробці SDF-графіки було використано апаратне прискорення, зокрема, використовуючи спеціалізовані функціональності графічних процесорів (GPU). GPU є потужним інструментом для виконання паралельних обчислень, і він ідеально підходить для завдань, пов'язаних з графікою та обробкою великого обсягу даних.

Використання апаратного прискорення у вигляді GPU дозволило оптимізувати обчислювальні завдання, пов'язані з SDF-графікою, забезпечуючи високу продуктивність та швидкість відображення графічних об'єктів. Це дозволило реалізувати складні алгоритми реймарчингу та обчислення SDF з високою швидкістю, що сприяло взаємодії з користувачем у режимі реального часу та забезпечило плавність відображення сцени.

2.3 Інтерактивність та користувацький досвід

В сучасному світі розробка графічних додатків надає особливий акцент на інтерактивність та користувацький досвід. Зростаюча потреба в застосуванні відмінних візуальних рішень для ефективного взаємодії користувача визначає важливість розробки інтерфейсів, які не лише передають інформацію, але й взаємодіють з користувачем інтуїтивно та ефективно.

В даному підрозділі розглядається роль SDF-графіки в створенні інтерактивних додатків та підвищенні користувацького досвіду. Аналізується взаємодія користувача з графічним інтерфейсом, його зручність та ефективність у

сприйнятті інформації. Звертається увага на використання SDF-графіки для досягнення високої якості візуальних ефектів, що сприяють покращенню користувацького враження.

Підкреслюється значення інтерактивності у забезпеченні більш глибокого занурення користувача у віртуальні середовища та можливості реагування на його взаємодію. Особлива увага приділяється підходам до створення інтерактивних елементів та їх впливу на користувацький досвід в контексті відображення SDF-графіки.

3.3.1 Можливості взаємодії

Взаємодія з SDF-графікою визначається багатьма функціональними можливостями, спрямованими на полегшення користування та зрозумілість відображених даних. У представленому прикладі з другого розділу відзначається велика кількість інструментів для взаємодії з візуалізацією. Користувач може здійснювати вибір забрудника, налаштовувати відображення інформації, взаємодіяти з часовою шкалою, а також вільно масштабувати зображення рельєфу та переміщати камеру на сцені.

Ці можливості забезпечують не лише велику гнучкість для адаптації до конкретних завдань, але й роблять процес взаємодії із зображенням зручним та інтуїтивно зрозумілим. Комбінація вибору, налаштувань та інтерактивних опцій робить користування SDF-графікою високофункціональним та придатним для різних сценаріїв використання.

3.3.2 Інтерфейс користувача

Інтерфейс користувача в SDF-графіці був спроектований з урахуванням максимальної зручності та ефективності для взаємодії з візуалізацією атмосферного повітря. Представлений в другому розділі інтерфейс має мінімальний, але функціональний набір інструментів, спрямованих на аналіз та спостереження за рівнем забруднення.

"Забрудники" дозволяють користувачам обирати конкретного забрудника, за яким вони бажають провести моніторинг. "Налаштування" використовуються для відображення інформації про точки моніторингу та їх параметри. "Часова шкала" надає можливість змінювати дані в реальному часі, а також переглядати їх за певний період.

Мінімалізм та можливість розширення функціоналу інтерфейсу дозволяють користувачам зосередитися на важливих аспектах моніторингу довкілля, забезпечуючи при цьому зручність і ефективність взаємодії. Цей підхід дозволяє швидко реагувати на потреби користувачів та легко адаптувати інтерфейс до конкретних завдань аналізу та спостереження.

2.4 Адаптація до потреб користувача

У даному підрозділі ми зосередимося на оцінці адаптації SDF-графіки до різних потреб користувачів, зокрема у сфері моніторингу довкілля. Вивчимо, як цей графічний інтерфейс може відповідати різним категоріям користувачів, враховуючи їхні унікальні вимоги та завдання.

На першому етапі розглядатимемо різні профілі користувачів, серед яких екологи, дослідники, громадськість та владні органи. Визначимо, як SDF-графіка може відповідати специфіці їхніх індивідуальних потреб, забезпечуючи зручний та ефективний інтерфейс для взаємодії з даними моніторингу довкілля.

Другий аспект стосується обробки багатовимірних даних. Проаналізуємо, як SDF-графіка може враховувати різноманітні аспекти моніторингу, такі як рівні різних забрудників, вплив метеорологічних умов та інші фактори, що впливають на стан довкілля.

Третій напрямок стосується адаптованої інтерфейсної взаємодії. Розробимо інтерфейс, який легко адаптується до потреб користувачів різних вікових категорій та рівнів технічної підготовки. Забезпечимо його доступність та зручність використання для широкого спектру аудиторії, незалежно від їхнього досвіду та особистих вимог.

3.4.1 Різні профілі користувачів

SDF-графіка визначається своєю гнучкістю та спроможністю задовольняти різні потреби користувачів різних профілів. Враховуючи конкретний приклад, який був представлений, можна визначити, що ця техніка відображення корисна для громадськості, забезпечуючи доступність та зрозумілість інформації про забруднення довкілля.

Однак, розширивши функціонал, наприклад, додавши відображення даних у вигляді графіків чи інших аналітичних інструментів, SDF-графіка може бути ефективним інструментом для досліджень та наукових досліджень, задовольняючи потреби дослідників та екологів. Таким чином, можна визначити, що SDF-графіка може успішно адаптуватися до різних профілів користувачів, забезпечуючи високу ефективність та функціональність.

3.4.2 Багатовимірні дані

SDF-графіка, використовуючи свою гнучкість, може ефективно враховувати різноманітні аспекти моніторингу довкілля. Наприклад, в приведеному прикладі відображення точок моніторингу на рельєфі дозволяє аналізувати рівні забруднення в конкретних місцях. За допомогою текстового відображення інформації про місце, SDF-графіка дозволяє враховувати додаткові параметри.

Такий підхід до відображення багатовимірних даних дозволяє зручно взаємодіяти з різними аспектами моніторингу, роблячи інформацію доступною та зрозумілою для користувача. Сполучаючи графічне та текстове відображення, SDF-графіка стає потужним інструментом для вивчення різних вимірів стану довкілля та їх взаємодії.

2.5 Можливості розширення та інтеграції

У контексті сучасних технологічних рішень ключовим аспектом розробки графічних додатків є можливості їх розширення та інтеграції з іншими системами.

У цьому підрозділі розглядаються різноманітні можливості розширення та інтеграції SDF-графіки, які спрямовані на покращення функціональності та розширення сфери її застосування.

В першу чергу розглядається питання розширення функціоналу SDF-графіки. Оцінюється можливість додавання нових функцій або розширення існуючих, щоб забезпечити користувачам більше інструментів та можливостей у використанні графічного інтерфейсу.

Другий аспект стосується інтеграції SDF-графіки з геоданими. Розглядається можливість плідної співпраці з картографічними даними та іншими джерелами інформації, такими як дані моніторингу забруднення, для забезпечення користувачам повної та зрозумілої інформації.

3.5.1 Розширення функціоналу

Розширення функціоналу SDF-графіки відкриває широкі можливості для включення нових функцій або удосконалення існуючих, що може значно поліпшити користувацький досвід. Наприклад, в рамках приведеного прикладу можна легко модифікувати відображення інформації про місце забруднення, використовуючи різні методи або стилі представлення.

Додавання графіків до візуалізації може відкрити нові способи аналізу та сприйняття даних. Застосування звичайних графіків разом із SDF-графікою може створити інтегровану інтерактивну платформу для більш детального вивчення та розуміння моніторингу довкілля.

Такі розширення можуть підвищити корисність та адаптивність SDF-графіки до різних потреб користувачів, роблячи її більш ефективною та зручною для використання.

3.5.2 Інтеграція з геоданими

Інтеграція з геоданими в SDF-графіці виявляється важливим аспектом для забезпечення повноцінної та зрозумілої інформації користувачам. Наш підхід до цього питання демонструється в другому розділі, де розглядається можливість додавання рельєфу та відображення точок моніторингу на цьому рельєфі.

У конкретному випадку, SDF-графіка дозволяє відобразити геодані, такі як картографічні дані чи дані моніторингу забруднення, на тривимірному рельєфі. Це дозволяє користувачам з легкістю визначати географічне положення точок моніторингу та аналізувати зв'язок між географічним розташуванням та рівнем забруднення.

Такий підхід сприяє більш глибокому розумінню результатів моніторингу та забезпечує користувачам контекстуальну картину стану довкілля. Інтеграція з геоданими розширює можливості SDF-графіки, забезпечуючи користувачам більш повне та збалансоване відображення даних.

Використовуючи картографічні дані та дані моніторингу забруднення, SDF-графіка стає потужним інструментом для аналізу впливу різних факторів на стан довкілля. Інтеграція з геоданими відкриває нові можливості для розширення функціоналу та забезпечення користувачів комплексною та деталізованою інформацією.

ВИСНОВКИ

В результаті проведених досліджень було обґрунтовано ефективність використання методів SDF-графіки для представлення тривимірних об'єктів візуалізації еколого-економічного моніторингу та запропоновано відповідний інструментарій.

Декілька описових прикладів, які можуть бути особливо ефективними з використанням SDF-графіки, включають об'ємні хмари точок для кращого аналізу геометричних та топологічних властивостей об'єктів, об'ємні теплові карти для інформативного відображення розподілу значень у тривимірному просторі, та об'ємні анімації з часовою "глибиною" для передачі динаміки змін в часі. Крім того, SDF-графіка може бути успішно використана для візуалізації об'ємних моделей рельєфу та ґрунту, водних ресурсів та атмосферних явищ.

Застосування SDF-графіки має наступні переваги в порівнянні з традиційними методами:

- Інтерактивність. Використання методів SDF-графіки дозволяє користувачам взаємодіяти з візуалізацією у реальному часі, надаючи можливість маніпулювати об'єктами та отримувати деталізовану інформацію за запитом.
- Деталізація відображення об'єктів. Дозволяє створювати більш докладні та реалістичні об'єкти, що поліпшує сприйняття та аналіз отриманих даних.

Запропонований інструментарій надає можливість використовувати SDF-графіку для візуалізації даних моніторингу. Це дозволяє ефективно відображати об'єкти з високою деталізацією та точністю, сприяючи кращому розумінню та аналізу даних моніторингу.

Отриманні результати дають нам підстави для твердження про доцільність включення методів SDF-графіки в загальний інструментарій візуалізації еколого-економічного моніторингу.

БІБЛІОГРАФІЧНИЙ СПИСОК

1. Моніторинг довкілля [Електронний ресурс]. - Режим доступу: https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%BD%D1%96%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3_%D0%B4%D0%BE%D0%B2%D0%BA%D1%96%D0%BB%D0%BB%D1%8F
2. Моніторинг довкілля: підручник / [Боголюбов В.М., Клименко М.О., Мокін В. Б. та ін.]; за ред. проф. В.М. Боголюбова. Вид. 2-ге, переробл. і доповн. – Київ: НУБіПУ, 2018. – 435 с.
3. Ray Marching and Signed Distance Functions [Електронний ресурс]. - Режим доступу: <https://jamie-wong.com/2016/07/15/ray-marching-signed-distance-functions/>
4. Ray Marching написана Michael Walczyk [Електронний ресурс]. - Режим доступу: <https://www.tylerbovenzi.com/RayMarch/>
5. Історія екологічного моніторингу [Електронний ресурс]. - Режим доступу: <https://inl.gov/content/uploads/2023/07/Summary-of-Historical-Environmental-Monitoring.pdf>
6. Коваленко Ю. Л. Моніторинг довкілля : конспект лекцій для студентів 2 і 3 курсів денної та 3 курсу заочної форм навчання за спеціальностями 183 – Технології захисту навколишнього середовища та 101 – Екологія / Ю. Л. Коваленко ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2020. – 144 с.
7. Поняття екологічного моніторингу [Електронний ресурс]. - Режим доступу: <https://osvita.ua/vnz/reports/ecology/18980/>

ДОДАТОК А

Код реймарчеру:

```
float4 raymarch(float3 ro, float3 rd)
{
    float4 pixelColor = float4(0, 0, 0, 0);

    const int maxstep = 250;
    const float drawdist = 40;
    float t = 0;
    [loop]
    for (int i = 0; i < maxstep; ++i)
    {
        float3 p = ro + rd * t;
        float4 objectData = GetColorAndDistance(p);

        if (objectData.w < 0.001)
        {
            float3 n = calcNormal(p);
            float diffuce = max(dot(-_LightDir.xyz, n), 0);
            diffuce = lerp(0.2, 1.0, diffuce);
            float shadows = lerp(0.2, 1.0, shadow(p));
            pixelColor = float4(1, 1, 1, 1);
            pixelColor *= float4(diffuce.xxx, 1);
            pixelColor *= float4(objectData.xyz, 1);
            pixelColor *= float4(shadows.rrr, 1);
            break;
        }
        else if (t > drawdist) { return pixelColor; }

        t += objectData.w;
    }

    return pixelColor;
}
```

Код розрахунку нормалі в даній точці:

```
float3 calcNormal(in float3 p)
{
    float d0 = GetColorAndDistance(p).w;
    const float2 epsilon = float2(.001, 0.0);
    float3 d1 = float3(
        GetColorAndDistance(p - epsilon.xyy).w,
        GetColorAndDistance(p - epsilon.yxy).w,
        GetColorAndDistance(p - epsilon.yyx).w);
    return normalize(d0 - d1);
}
```

Код розрахунку тіней:

```
float shadow(float3 p)
{
    float3 lightDir = -_LightDir;
    float kd = 1;
    int step = 0;
    [loop]
    for (float t = 0.1; step < 30 && kd > 0.001;)
    {
        float d = GetColorAndDistance(p + t * lightDir).w;
        if (d < 0.001) { kd = 0; }
        else { kd = min(kd, 16 * d / t); }
        t += d;
        step++;
    }
    return kd;
}
```

Код отримання дистанції до об'єкту:

```
float4 GetShapeDistance(Shape shape, float3 p)
{
    float3 Point = Rotate(p, shape.rotation);
    switch (shape.shapeType)
    {
        case 0:
            return CubeDistance(Point, shape.position, shape.scale);
            break;
        case 1:
```

```

        return SphereDistance(Point, shape.position, shape.radii.x);
        break;
    case 2:
        return TorusDistance(Point, shape.position, shape.radii.x,
shape.radii.y);
        break;
    case 5:
        return sdHeightMap(Point, shape.radii);
        break;
    default:
        return 40;
        break;
    }
}

```

Код формування дистанції до об'єкту та його кольору :

```

float4 GetColorAndDistance(float3 p)
{
    float shapeDst, outDst = 40;
    float3 outColor = 1;
    float4 output = float4(outColor, outDst);
    Shape s;
    [loop]
    for (int i = 0; i < _numShapes; i++)
    {
        s = _shapes[i];
        shapeDst = GetShapeDistance(s, p);
        switch (s.blendType)
        {
            case 0: // Default raymarching render
                if (shapeDst < outDst)
                {
                    output = s.shapeType == 5 ? float4(getHeightMapColor(p,
s.radii), shapeDst) : float4(s.color.rgb, shapeDst);
                }
                break;
            case 1: // Blend
                output = s.shapeType == 5 ? Blend(outDst, shapeDst, outColor,
getHeightMapColor(p, s.radii), s.blendStrength)

```

```

        : Blend(outDst, shapeDst, outColor, s.color.rgb,
s.blendStrength);
        break;
    case 2: // Cut
        if (-shapeDst > outDst)
            output = s.shapeType == 5 ? float4(getHeightMapColor(p,
s.radii), -shapeDst) : float4(s.color.rgb, -shapeDst);
            break;
    case 3: // Mask
        if (shapeDst > outDst)
            output = s.shapeType == 5 ? float4(getHeightMapColor(p,
s.radii), shapeDst) : float4(s.color.rgb, shapeDst);
            break;
        default:
            break;
    }
    outColor = output.rgb;
    outDst = output.w;
}
return output;
}

```

Код отримання дистанцій куба:

```

float CubeDistance(float3 p, float3 centre, float3 s)
{
    float3 o = abs(p.xyz - centre) - s;
    float ud = length(max(o, 0));
    float n = max(max(min(o.x, 0), min(o.y, 0)), min(o.z, 0));
    return ud + n;
}

```

Код отримання дистанцій сфери:

```

float SphereDistance(float3 p, float3 centre, float r)
{
    return distance(p, centre) - r;
}

```

Код отримання дистанцій торуca:

```
float TorusDistance(float3 p, float3 centre, float r1, float r2)
{
    float2 q = float2(length((p.xyz - centre).xz) - r1, p.xyz.y - centre.y);
    return length(q) - r2;
}
```

Код для отримання uv координат:

```
float2 getUv(float3 p, float2 scale)
{
    float3 b = float3(scale.x / 2.0f, 0.1, scale.y / 4.0f);
    float2 uv = float2(min(p.x, b.x), min(p.z, b.z));
    float aspectRatio = _HeightMapTex_TexelSize.y /
_HeightMapTex_TexelSize.x;
    uv.y *= aspectRatio;
    return uv;
}
```

Код для отримання кольору для карти висот:

```
float3 getHeightMapColor(float3 p, float2 scale)
{
    float2 uv = getUv(p, scale);
    float2 texCoord = (uv / scale - float2(0.5, 0.5));
    return float3(tex2D(_ReliefTex, texCoord).rgb);
}
```

Код для отримання дистанції до карти висот:

```
float sdHeightMap(float3 p, float2 scale)
{
    float3 b = float3(scale.x / 2.0f, 0.1, scale.y / 4.0f);
    float3 q = abs(p) - b;
    float g = sin(atan2(1, 2));
    float2 uv = getUv(p, scale);
    float2 texCoord = (uv / scale - float2(0.5, 0.5));
    float h = tex2D(_HeightMapTex, texCoord);
    h *= 0.03;
    q.y -= h;
    q.y *= g;
    q.x = max(0, q.x);
    q.y = max(0.0, q.y);
    q.z = max(0.0, q.z);
    float dist = length(q);
    return dist;
}
```