

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Інформаційна система дослідження географічних
проекцій на поверхні Землі в двовимірному та тривимірному
просторах»

на здобуття освітнього ступеня магістра
зі спеціальності 121 Інженерія програмного забезпечення
освітньо-професійної програми «Інженерія програмного забезпечення»

*Кваліфікаційна робота містить результати власних досліджень. Використання
ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело*

_____ Іван ЗАЙЦЕВ
(підпис)

Виконав: здобувач вищої освіти групи ПДМ-62
_____ Іван ЗАЙЦЕВ

Керівник: _____ Віктор ГРЕБЕНЮК
доктор філософії (PhD)

Рецензент: _____

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Магістр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

_____ Ірина ЗАМРІЙ

«_____» _____ 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Зайцеву Івану Сергійовичу

1. Тема кваліфікаційної роботи: «Інформаційна система дослідження географічних проєкцій на поверхні Землі в двовимірному та тривимірному просторах»

керівник кваліфікаційної роботи Віктор ГРЕБЕНЮК, доктор філософії (PhD),

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «15» жовтня 2024 р. № 320.

2. Строк подання кваліфікаційної роботи «17» грудня 2024 р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, алгоритми та структури даних для роботи з географічними проєкціями.

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

1. Дослідження алгоритмів та структур даних для роботи з географічними проєкціями

2. Аналіз географічних проекцій на поверхні Землі в двовимірному та тривимірному просторах.
3. Розробка інформаційної системи дослідження географічних проекцій на поверхні Землі в двовимірному та тривимірному просторах.
5. Перелік ілюстративного матеріалу: *презентація*
1. Мета, об'єкт та предмет дослідження.
 2. Актуальність роботи.
 3. Алгоритм пошуку оптимальних проекцій.
 4. Математична модель створення гібридної проекції.
 5. Програмні засоби реалізації.
 6. Практичний результат.
 7. Порівняння ефективності.
6. Дата видачі завдання «16» жовтня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	16.10-23.10.2024	
2	Вивчення матеріалів для аналізу географічних проекцій на поверхні Землі	24.10-30.10.2024	
3	Дослідження алгоритмів та структур даних для роботи з географічними проекціями	31.10-04.11.2024	
4	Порівняння точності алгоритмів та структур даних для роботи з географічними проекціями	05.11-09.11.2024	
5	Застосування двовимірних та трьовимірних прокцій для дослідження та аналізу географічних даних	10.11-16.11.2024	
6	Розробка інформаційної системи для дослідження географічних проекцій	17.11-22.11.2024	
7	Оформлення роботи: вступ, висновки, реферат	23.11-25.11.2024	
8	Розробка демонстраційних матеріалів	26.11-27.11.2024	
9	Попередній захист роботи	27.11-06.12.2024	

Здобувач вищої освіти

(підпис)

Іван ЗАЙЦЕВ

Керівник

кваліфікаційної роботи

(підпис)

Віктор ГРЕБЕНЮК

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 82 стор., 1 табл., 16 рис., 30 джерел.

Мета роботи – покращення точності дослідження географічних проекцій за рахунок відображення географічних проекцій на поверхні Землі в двовимірному та тривимірному просторах.

Об'єкт дослідження – процес дослідження географічних проекцій на поверхні Землі в двовимірному та тривимірному просторах.

Предмет дослідження – засоби та методи дослідження географічних проекцій у двовимірному та тривимірному просторах.

У роботі використано різноманітні методи, такі як аналіз існуючих алгоритмів та методів, що застосовуються для роботи з географічними проекціями.

Проведено аналіз сучасних методів обробки географічних проекцій, включаючи Polygon buffer та Polygon distance, з урахуванням їх переваг і недоліків, таких як: недостатня адаптивність для роботи з різними географічними проекціями, недостатня ефективність при обробці географічних даних в різних проекціях.

Розроблено та оптимізовано:

1. Алгоритм динамічного пошуку оптимальних проекцій, який дозволяє автоматично визначати найбільш відповідну проекцію залежно від координат заданих полігонів, що підвищує точність і ефективність обчислень.
2. Алгоритм створення гібридних оптимальних проекцій, який забезпечує більш точне відображення географічних об'єктів, поєднуючи характеристики декількох проекцій.
3. Інформаційну систему для дослідження географічних проекцій на поверхні Землі в двовимірному та трьовимірному просторах.

Проведено експерименти для перевірки ефективності запропонованих алгоритмів, що підтвердили їх переваги над існуючими методами в умовах роботи з географічними проекціями.

КЛЮЧОВІ СЛОВА: ГЕОГРАФІЧНІ ПРОЕКЦІЇ, ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ, ДВОВИМІРНИЙ ПРОСТІР, ТРИВИМІРНИЙ ПРОСТІР, ТОЧНІСТЬ, ЕФЕКТИВНІСТЬ, ГЕО-ПРОСТОРОВІ АЛГОРИТМИ ТА СТРУКТУРИ ДАНИХ.

ABSTRACT

Text part of the master's qualification work: 82 pages, 1 table, 16 pictures, 30 sources.

The purpose of the work – is to improve the accuracy of geographic projection research by mapping geographic projections onto the Earth's surface in two-dimensional and three-dimensional spaces.

Object of research – the process of studying geographical projections on the Earth's surface in two-dimensional and three-dimensional spaces.

Subject of research – tools and methods for studying geographic projections in two-dimensional and three-dimensional spaces.

The study employs various methods, such as the analysis of existing algorithms and techniques used for working with geographic projections.

An analysis of modern methods for processing geographic projections was conducted, including Polygon buffer and Polygon distance, taking into account their advantages and disadvantages, such as insufficient adaptability for working with different geographic projections and inefficiency in processing geographic data in various projections.

The following were developed and optimized:

1. A dynamic algorithm for searching optimal projections, which allows for automatic determination of the most suitable projection depending on the coordinates of the given polygons, thereby improving the accuracy and efficiency of calculations.

2. An algorithm for creating hybrid optimal projections, which provides more accurate representation of geographic objects by combining the characteristics of multiple projections.

3. An information system for studying geographic projections on the Earth's surface in two-dimensional and three-dimensional spaces.

Experiments were conducted to verify the effectiveness of the proposed algorithms, confirming their advantages over existing methods in the context of working with geographic projections.

KEYWORDS: GEOGRAPHICAL PROJECTIONS, INFORMATION TECHNOLOGIES, TWO-DIMENSIONAL SPACE, THREE-DIMENSIONAL SPACE, ACCURACY, EFFICIENCY, GEO-SPATIAL ALGORITHMS AND DATA STRUCTURES.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	11
ВСТУП.....	12
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	14
1.1 Географічні проекції	14
1.2 Системи координат та стандарти	19
1.3 Структури даних для роботи з географічними проекціями	21
1.4 Огляд актуальних наукових робіт в галузі географічних проекцій	25
2 АНАЛІЗ МЕТОДІВ ТА АЛГОРИТМІВ ДЛЯ РОБОТИ З ГЕОГРАФІЧНИМИ ПРОЕКЦІЯМИ	32
2.1 Алгоритми та методи для роботи з географічними проекціями.....	32
2.2 Алгоритми перетворення координат.....	36
2.3 Оцінка деформацій у географічних проекціях	40
2.4 Автоматизація обробки географічних проекцій.....	43
2.5 Формати та інструменти для географічних проекцій	45
3 РОЗРОБКА АЛГОРИТМІВ ТА ІНФОРМАЦІЙНОЇ СИСТЕМИ	48
3.1 Алгоритм пошуку оптимальних проекцій	48
3.2 Математична модель створення гібридної географічної проекції	50
3.3 Інформаційна система дослідження проекцій	54
3.4 Порівняння точності оптимальних та гібридних проекцій.....	57
3.5 Програмні засоби реалізації	60
3.6 Розгортання інформаційної системи.....	62
ВИСНОВКИ	65
ПЕРЕЛІК ПОСИЛАНЬ	67
ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ.....	70
ДОДАТОК Б. ЛІСТИНГИ ПРОГРАМНИХ МОДУЛІВ	76

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ГІС - Географічні інформаційні системи

EPSG - Ідентифікатори для геодезичних систем координат (European Petroleum Survey Group)

WGS - Глобальна геодезична система координат (World Geodetic System)

UTM - Система проекцій (Universal Transverse Mercator)

PROJ - Формат моделі проекції

WKT - формат для представлення геометричних об'єктів (Well-Known Text)

ВСТУП

У сучасному світі картографія займає ключове місце серед інструментів для дослідження, аналізу і планування у різних галузях, таких як географія, геодезія, екологія, інженерія та містобудування. Одним із основних завдань картографії є перенесення поверхні Землі на площину за допомогою географічних проекцій. Це дозволяє відображати географічні об'єкти і явища у зрозумілому вигляді для подальшого аналізу та прийняття рішень. Проте жодна проекція не може точно передати всі характеристики сфероїдної поверхні Землі через її геометричну складність. Це обумовлює появу різних типів спотворень, які можуть торкатися форм, площ, відстаней або кутів. Вивчення цих спотворень та оптимізація методів роботи з ними є важливими завданнями для фахівців, що працюють у сфері геоінформаційних технологій.

Із розвитком інформаційних технологій виникає потреба у створенні програмного забезпечення, яке дозволяє ефективно досліджувати і візуалізувати географічні проекції. Це стосується як традиційного двовимірного простору, так і тривимірного моделювання, що стає все більш популярним для аналізу складних природних процесів та планування територій. Розробка таких технологій сприятиме: покращенню точності картографічних даних, розширенню можливостей аналізу спотворень різних проекцій, створенню нових інструментів для підтримки прийняття рішень у науковій та прикладній діяльності.

Мета роботи – покращення точності дослідження географічних проекцій за рахунок відображення географічних проекцій на поверхні Землі в двовимірному та тривимірному просторах.

Об'єкт дослідження – процес дослідження географічних проекцій на поверхні Землі в двовимірному та тривимірному просторах.

Предмет дослідження – засоби та методи дослідження географічних проекцій у двовимірному та тривимірному просторах.

Для досягнення мети вирішувалися наступні завдання:

1. Проведено дослідження і аналіз існуючих наукових джерел, щоб отримати повне уявлення про різноманітні методи та технології, що використовуються для дослідження географічних проекцій.
2. Розроблено алгоритм динамічного пошуку оптимальних проекцій, який дозволяє автоматично визначати найбільш відповідну проекцію залежно від координат заданих полігонів, що підвищує точність і ефективність обчислень.
3. Розроблено алгоритм створення гібридних оптимальних проекцій, який забезпечує більш точне відображення географічних об'єктів, поєднуючи характеристики декількох проекцій.
4. Розроблено інформаційну систему для дослідження географічних проекцій на поверхні Землі в двовимірному та трьовимірному просторах.
5. Використано статистичні методи для оцінки та порівняння результатів різних алгоритмів. Визначено точність, швидкість та стабільність кожного алгоритму.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Географічні проекції

Географічні проекції – це математичні способи перетворення координат точок земної поверхні з еліпсоїдної форми на площину. Вони необхідні для створення карт, оскільки неможливо зобразити криву поверхню Землі на площині без певних спотворень. Основна задача, яку вирішують географічні проекції це передача географічних даних з мінімальними спотвореннями форми, площі, відстаней та напрямків [1].

Географічні проекції класифікуються за різними властивостями, які вони зберігають або спотворюють. Основні види проекцій:

1. Рівнокутні (конформні).

Рівнокутні проекції зберігають форми малих об'єктів, забезпечуючи, що малі кути на поверхні Землі відповідають таким самим кутам на карті. Це важливо для навігаційних цілей, оскільки кути на картах відповідають реальним напрямкам. Проте такі проекції спотворюють площу.

2. Рівновеликі (еквівалентні).

Рівновеликі проекції зберігають площі об'єктів, що означає, що відносна площа будь-яких двох об'єктів на карті відповідає їхній реальній площі на поверхні Землі. Це важливо для тематичних карт, таких як карти розподілу населення, оскільки вони точно передають пропорції площ. Проте ці проекції спотворюють форму об'єктів, особливо біля країв карти.

3. Рівнопроміжні (еквідистантні).

Рівнопроміжні проекції зберігають відстані від певної точки (часто від центру проекції) до всіх інших точок. Це зручно для карт, де важливо показати правильні відстані від центральної точки. Проте ці проекції спотворюють як форму, так і площу об'єктів, особливо на віддалених від центру ділянках карти.

4. Компромісні проекції.

Компромісні проєкції прагнуть зменшити всі типи спотворень, але не повністю усувають жодного з них. Вони забезпечують збалансоване відображення форми, площі, відстані та напрямку, що робить їх універсальними для різних типів карт.

Також розрізняють такі форми проєкцій:

1. Циліндричні.

Проекційна поверхня має вигляд циліндра, що торкається земної поверхні на екваторі або на певній широті. Ці проєкції мають особливість розтягування від полюсів, що призводить до спотворення площ. Найвідоміша циліндрична проєкція це проєкція Меркатора. На рисунку 1.1 зображено приклад циліндричної проєкції [2].

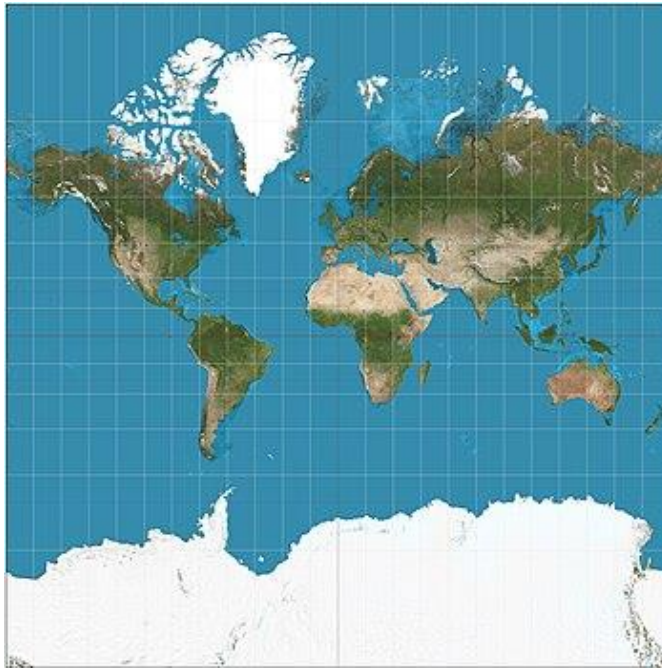


Рис. 1.1 Приклад циліндричної проєкції

2. Конічні.

Проекційна поверхня має форму конуса, який торкається земної поверхні або в певній точці, або вздовж лінії, що проходить через певні широти. Конічні проєкції добре підходять для відображення територій, розташованих на середніх широтах, наприклад, для карт Європи. На рисунку 1.2 зображено приклад конічної проєкції [2].



Рис. 1.2 Приклад конічної проекції

3. Азимутальні.

Проекційна поверхня є плоскою і торкається земної кулі в одній точці. Вони зазвичай використовуються для зображення полярних регіонів або для визначення напрямку між точками на земній поверхні. На рисунку 1.3 зображено приклад азимутальної проекції [2].

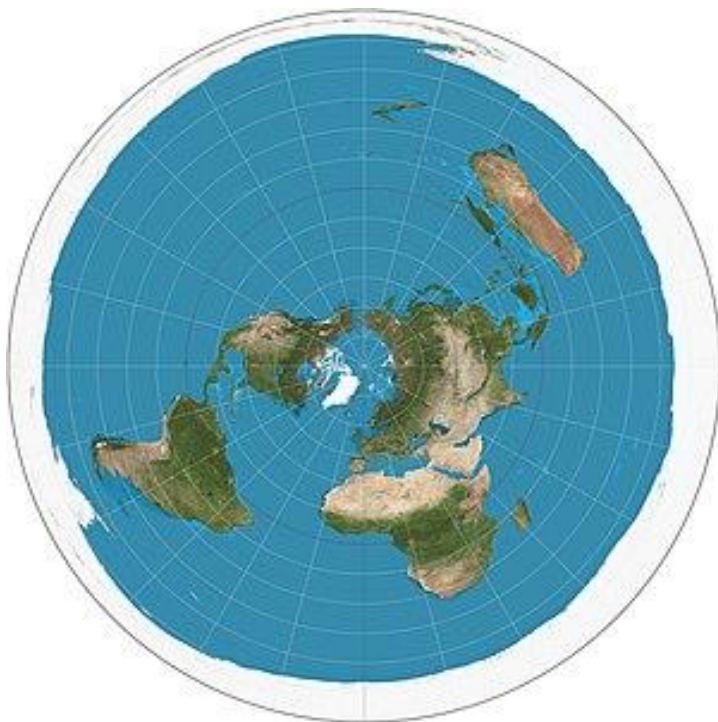


Рис. 1.3 Приклад азимутальної проекції

4. Переривчасті.

Передбачають розрив картографічної поверхні, тобто картографічне зображення розділяється на окремі частини або сегменти. Такі проекції використовуються, щоб уникнути значних спотворень на певних ділянках карти. На рисунку 1.4 зображено приклад переривчастої проекції.

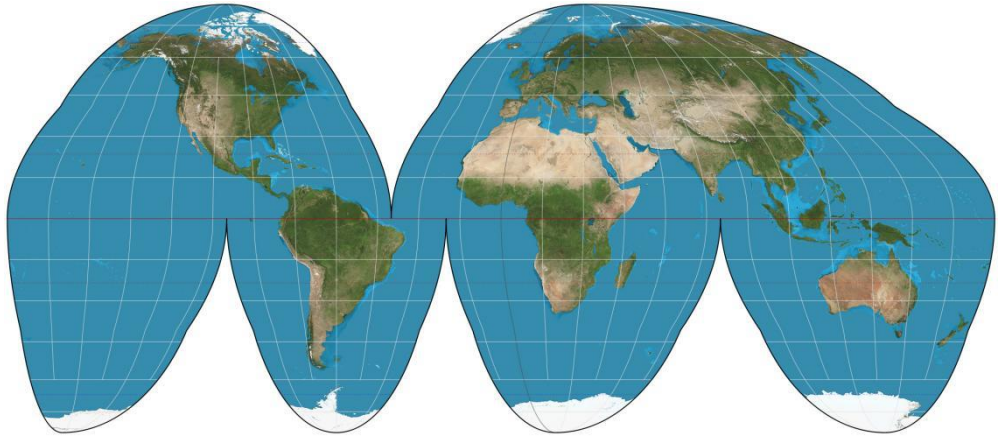


Рис. 1.4 Приклад переривчастої проекції

Географічні проекції використовуються для широкого спектра завдань, зокрема:

1. Перетворення сферичної поверхні Землі у площинну карту для візуалізації географічних об'єктів.
2. Відображення форми, розмірів та розташування об'єктів з урахуванням необхідної точності.
3. Розрахунку масштабів для створення карт із заданим ступенем деталізації.
4. Визначення площ географічних об'єктів із застосуванням відповідних проекцій для мінімізації спотворень.
5. Обчислення відстаней між точками на картах із різним типом проекції.
6. Розрахунку кутів між напрямками для точного орієнтування.
7. Створення тематичних карт для вивчення специфічних географічних, кліматичних або демографічних особливостей.

8. Планування транспортних маршрутів для морських, авіаційних і наземних перевезень.

9. Моделювання геопросторових даних для аналізу в геоінформаційних системах.

10. Наукових досліджень для вивчення процесів на Землі в залежності від їх географічного розташування.

Кожна проекція має свої переваги і недоліки, які враховуються залежно від конкретного завдання. Вибір проекції залежить від конкретних задач і того, які властивості карти є найбільш важливими для дослідження. Різні проекції підходять для різних масштабів і регіонів. Наприклад, для навігаційних досліджень важливі рівнокутні проекції, тоді як для демографічних досліджень краще використовувати рівновеликі проекції.

Вибір географічної проекції є критично важливим для правильного відображення географічних об'єктів на карті, так і для точності розрахунків пов'язаних з цими об'єктами.

Для того, щоб краще зрозуміти, як форма проекції з тривимірного простору переноситься на площину, можна розглянути три основні типи проекцій: азимутальні, конічні та циліндричні. Кожна з них має свою особливість у способі відображення земної кулі на площині. На Рисунку 1.5 зображено приклади цих проекцій.

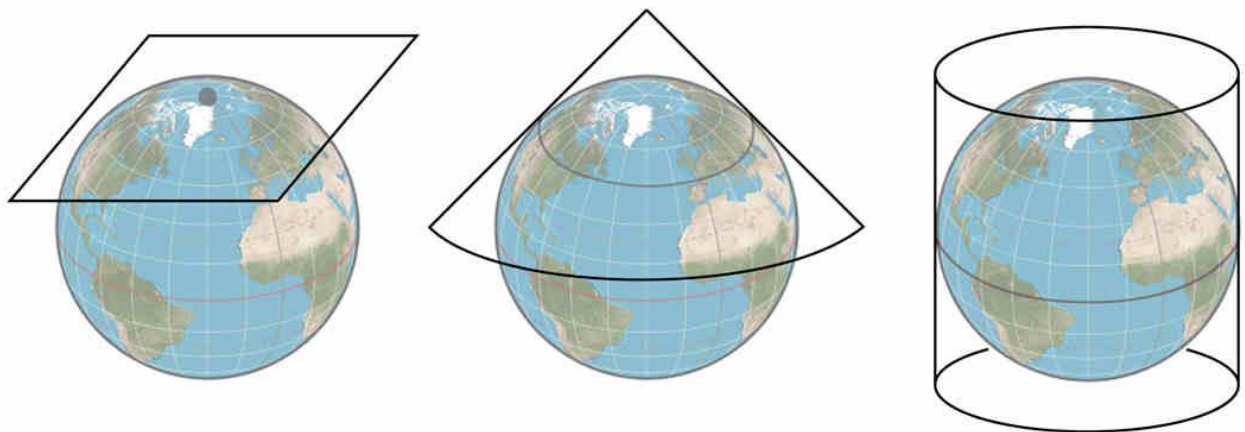


Рис. 1.5 Приклад проекцій

1.2 Системи координат та стандарти

WGS (World Geodetic System) — це глобальна геодезична система координат, яка визначає точні позиції на земній поверхні. Найбільш відома її версія — WGS 84, яка використовується в сучасних навігаційних системах, таких як GPS.

Основні характеристики WGS:

1. Система координат.

WGS використовує тривимірну декартову систему координат, де кожна точка визначається трьома координатами: широта (latitude), довгота (longitude) та висота (altitude).

Географічні координати (широта і довгота) визначаються на основі еліпсоїда, що наближає форму Землі.

2. Еліпсоїд.

WGS 84 базується на еліпсоїді з певними параметрами: великий піввісь ($a = 6378137$ м) і сплюснутість ($f = 1/298.257223563$).

Це дозволяє моделювати форму Землі як еліпсоїд, що краще відображає геометрію планети, ніж простий сферичний підхід.

3. Використання в навігації.

WGS 84 є стандартом для глобальних позиціонувальних систем, таких як GPS. Всі GPS приймачі використовують цю систему для визначення місця розташування.

Вона забезпечує точність у межах кількох метрів, хоча за допомогою диференційної корекції точність може бути значно вищою.

4. Глобальний охоплення.

WGS 84 охоплює всю планету, включаючи океани, і використовується для створення карт, географічних баз даних, а також для маркування точок на поверхні Землі в науці та інженерії.

WGS 84 є однією з найбільш поширених геодезичних систем, що дозволяє точне визначення координат для навігації, картографії, геоінформаційних систем і багато інших застосувань.

EPSG (European Petroleum Survey Group) — це система кодів для географічних та координатних систем. Кожен код EPSG ідентифікує конкретну географічну проекцію, систему координат або трансформацію між ними. EPSG-коди використовуються в географічних інформаційних системах для стандартизації та полегшення роботи з різними геопросторовими даними.

Основні категорії EPSG-кодів:

1. Географічні системи координат: визначають положення точок на поверхні Землі з використанням широти та довготи, наприклад:

EPSG:4326 — це географічна система координат на основі глобальної геодезичної моделі WGS84.

2. Проекції: це методи проекції тривимірної поверхні Землі на двовимірну площину, зазвичай для картографічних цілей. Наприклад:

EPSG:3857 — це поширена проекція для веб-карт, зокрема у Google Maps, яка використовується для відображення Землі на площині.

3. Довготи і широти: Визначають положення точок на основі географічних координат.

4. Трансформації: описують методи, як одні координати можна перетворити в інші. Вони використовуються для зміни систем координат, наприклад, між WGS 84 та координатними системами, які використовуються для картографії на локальних рівнях.

Система EPSG є відкритим стандартом і підтримується багатьма геопросторовими програмами [3].

1.3 Структури даних для роботи з географічними проекціями

Для проведення операцій із географічними проекціями використовуються спеціалізовані структури даних, які забезпечують ефективну обробку просторової інформації [4]. Їх застосування дозволяє:

1. Ефективно здійснювати пошук географічних об'єктів у базах даних, забезпечуючи швидкий доступ до інформації.
2. Виконувати трансформацію об'єктів між різними географічними проекціями, що є необхідним для роботи з даними у різних координатних системах.
3. Обчислювати відстані між об'єктами, враховуючи їх розташування на площині, з урахуванням кривизни земної поверхні.
4. Визначати кути між об'єктами для вирішення задач просторової орієнтації та аналізу.
5. Використовувати методи індексації для оптимізації доступу до просторових даних.
6. Виконувати геопросторовий аналіз, наприклад, визначення області покриття, буферизацію або розрахунок перетинів об'єктів.
7. Працювати з великими обсягами географічних даних, зберігаючи їх структуру та забезпечуючи швидку обробку.
8. Інтегрувати геодані з різних джерел, забезпечуючи їх узгодженість у загальній системі координат.
9. Підтримувати операції в реальному часі, що особливо важливо для навігаційних систем або моніторингу змін у геопросторовому середовищі.

Загалом, структури даних для роботи з географічними проекціями є основою для ефективного аналізу та обробки просторової інформації. Вони дозволяють зберігати та маніпулювати даними про об'єкти на поверхні Землі, забезпечуючи точність та оптимізацію розрахунків при трансформаціях між різними проекціями. Використання відповідних структур даних, таких як багатокутники, координатні системи та просторові індекси, є критичним для досягнення високої ефективності і точності в задачах географічного аналізу та візуалізації. Правильний вибір і

налаштування таких структур в контексті конкретних завдань дозволяє знизити витрати на обробку даних, покращити швидкодію системи і забезпечити точність результатів при роботі з великими обсягами просторової інформації.

У таблиці 1.1 підсумовано основні характеристики цих структур даних, щоб краще зрозуміти їх відмінності та вибрати кращий для конкретних завдань.

Таблиця 1.1

Порівняння характеристики структур даних геопросторових індексів

Структура	Переваги	Недоліки
Quadtree	Завдяки ітеративному поділу простору, пошук об'єктів у конкретній області може бути дуже швидким	Неефективна при великій щільності даних або коли об'єкти нерівномірно розподілені
K-d tree	Ефективне для пошуку в k-вимірному просторі, включаючи пошук за найближчим сусідом	Менш ефективне при динамічних операціях (вставка, видалення), що може потребувати перебудови дерева.

Вибір відповідної структури даних залежить від конкретних вимог. Quadtree підходить для двовимірних даних з великою кількістю порожніх областей, тоді як K-d tree ефективна для багатокимірних даних та операцій пошуку найближчих сусідів.

Quadtree (дерево квадрантів) — це структура даних, що використовується для ієрархічного поділу двовимірного простору на квадранти. Базується на рекурсивному розбитті простору на чотири рівні частини. Кожен вузол дерева відповідає квадратній області простору, яка може бути розбита на підобласті, якщо вона містить більше одного об'єкта чи задовольняє інші критерії деталізації.

Алгоритм побудови:

1. Ініціалізація: простір, який містить усі об'єкти, представляється великим прямокутником або квадратом.
2. Розбиття на квадранти: простір розбивається на чотири рівні квадранти.
3. Рекурсивний поділ: якщо квадрант містить більше одного об'єкта, він ділиться на чотири нові підквадранти. Цей процес триває, поки кожен квадрант не містить не більше одного об'єкта, або поки не буде досягнуто певного порогу глибини дерева [5].

На рисунку 1.6 зображено приклад структури даних дерева квадрантів.

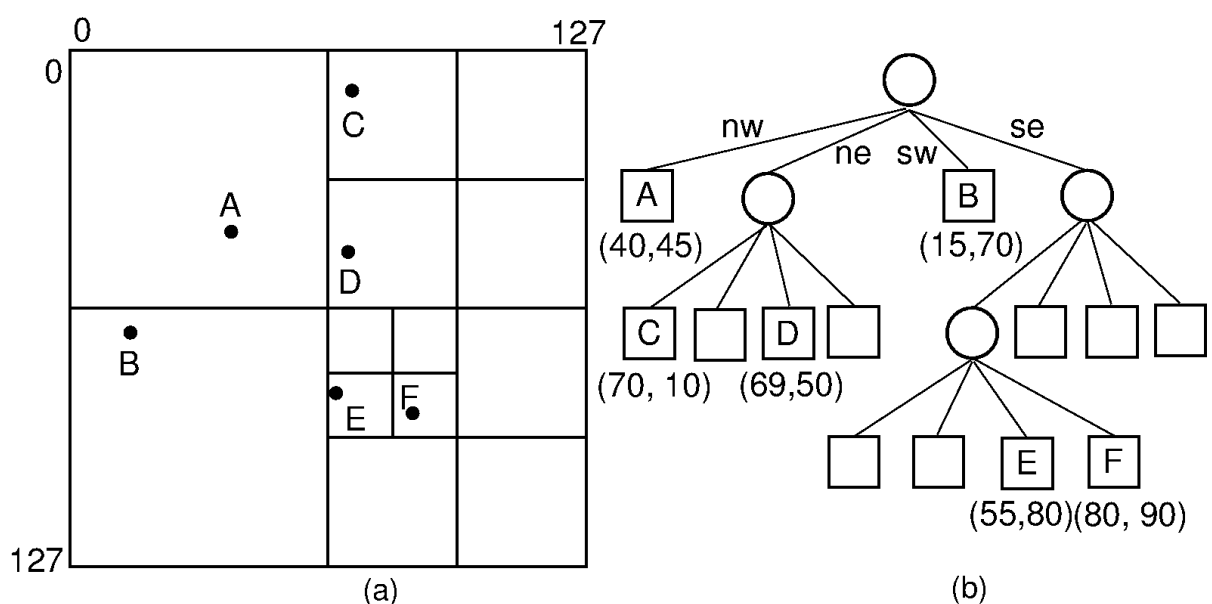


Рис. 1.6 Приклад дерева квадрантів

К-вимірне дерево (K-d tree) — це структура даних, яка використовується для зберігання точок у К-вимірному просторі. Базується на рекурсивному розбитті простору на два підпростори, які розділяються по одному з вимірів.

Алгоритм побудови:

1. Вибір осі для розбиття: починаючи з кореня дерева, обирається одна з осей.
2. Сортування точок: для поточного рівня дерева точки сортуються по вибраній осі.

3. Розбиття на підмножини: точка, що знаходиться в середині відсортованого списку по вибраній осі, стає коренем піддерева. Ліва підпіддерева містить точки, що мають менше значення по цій осі, а права більше.

4. Рекурсивний поділ: алгоритм рекурсивно будує ліве та праве піддерева, вибираючи для кожного рівня нову вісь для розбиття і повторюючи кроки 2-3.

На рисунку 1.7 зображено приклад структури даних К-вимірного дерева.

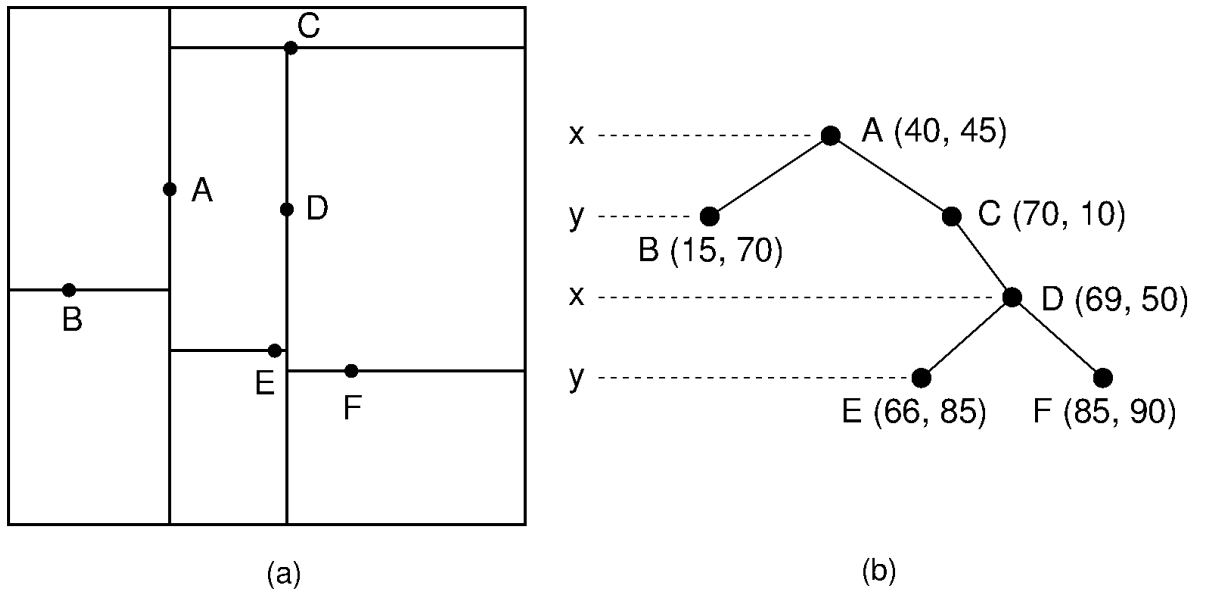


Рис. 1.7 Приклад К-вимірного дерева

Ці структури даних використовуються при роботі з геопросторовими індексами.

Геопросторові індекси використовуються для ефективного зберігання, пошуку та обробки просторових даних, таких як координати точок, лінії або полігони. Вони оптимізують виконання запитів, які стосуються географічних об'єктів, зокрема для таких завдань:

1. Пошук найближчих об'єктів: наприклад, знайти найближчі пункти обслуговування на карті. Геопросторові індекси дозволяють швидко знаходити об'єкти в заданій області або в межах певної відстані.

2. Перевірка перетину геометрій: визначення, чи перетинаються дві геометрії, наприклад, чи знаходиться точка всередині полігону чи лінія перетинає інший об'єкт.

3. Регіональні запити: наприклад, запити для виведення всіх об'єктів, які знаходяться в межах певної області.

4. Операції з буферами та відстанями: геопросторові індекси дозволяють ефективно виконувати запити, що стосуються відстані між точками чи обчислення буферних зон навколо об'єктів [5].

На рисунку 1.8 зображено приклад геопросторових індексів.

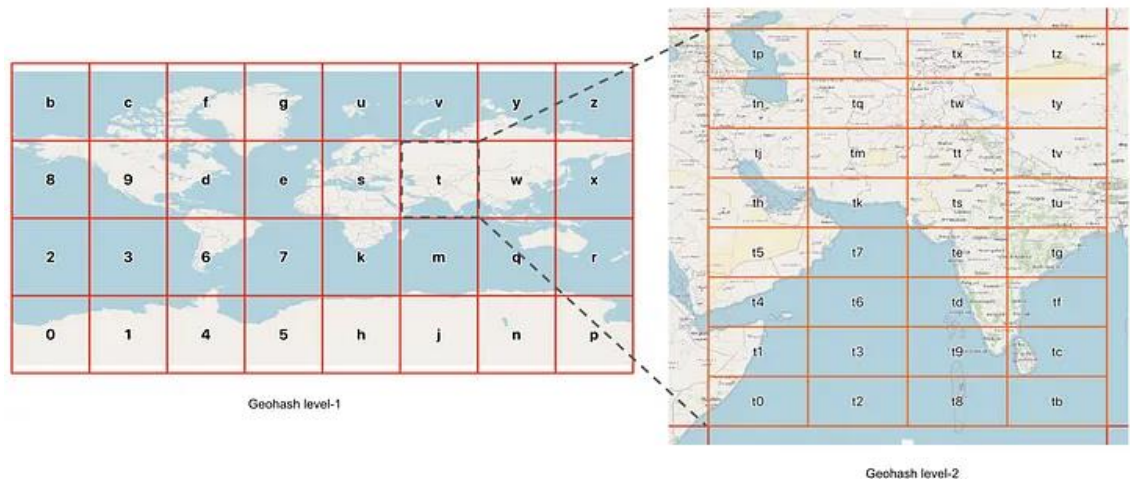


Рис. 1.8 Приклад геопросторових індексів

1.4 Огляд актуальних наукових робіт в галузі географічних проекцій

Наукова стаття «On the Definition of Standard Parallels in Map Projections», автором якої є М. Lapaîne, присвячена аналізу і визначенню стандартних паралелей у картографічних проекціях. Стандартні паралелі – це широти, де викривлення масштабу проекції є мінімальним або повністю відсутнім. Вони мають важливе значення при створенні картографічних проекцій, особливо циліндричних і конічних, які часто використовуються в географії, геодезії та картографії.

Основні положення статті:

1. Теоретичне обґрунтування: у статті розглядається математичне обґрунтування вибору стандартних паралелей у різних типах картографічних проекцій. Автор аналізує, як їх розташування впливає на точність та масштабну стабільність карти.

2. Типи проекцій:

- Конічні проекції: тут стандартні паралелі визначаються так, щоб мінімізувати масштабні викривлення в зоні між ними. У багатьох випадках вони встановлюються симетрично щодо середньої широти регіону;

- Циліндричні проекції: вибір стандартних паралелей також грає важливу роль, наприклад, у проекції Меркатора, де спотворення масштабу значно зростає з віддаленням від екватора;

3. Практичне застосування: визначення стандартних паралелей залежить від призначення карти та географічного регіону, який вона охоплює. Наприклад, для навігаційних цілей використовуються проекції з екватором як стандартною паралеллю, а для регіональних карт – дві паралелі, що обмежують регіон.

4. Математичні моделі: у статті представлені формули, що дозволяють точно розрахувати положення стандартних паралелей для різних типів проекцій.

5. Історичний контекст: М. Lapaine також аналізує історичний розвиток концепції стандартних паралелей, починаючи з класичних географічних карт до сучасних методів цифрового картографування.

Стаття підкреслює важливість стандартних паралелей як одного з ключових елементів оптимізації картографічних проекцій. Автор наголошує, що правильний вибір стандартних паралелей знижує викривлення масштабу та робить карту більш точною й придатною для конкретних завдань.

Ця публікація є цінним джерелом для картографів, геодезистів і спеціалістів з геоінформаційних систем, які займаються створенням і аналізом картографічних проекцій [6].

У книзі "Geographic Information Science and Systems, 4th Edition" автори Пол Лонглі, Майкл Гудчайлд, Девід Меджі, і Девід Рінделл детально розглядають

концепції географічних інформаційних систем та науки про просторові дані. Однією з ключових тем є картографічні проекції та їхній вплив на геопросторові дані.

Основні аспекти розділу про проекції:

1. Природа картографічних проекцій

Картографічні проекції є способом перетворення поверхні земної кулі (тривимірної сфери або еліпсоїда) у двовимірне представлення. Автори розглядають, як це перетворення викликає неминучі спотворення: форми, площі, відстаней або напрямків.

2. Класифікація проекцій.

У книзі пояснюються основні типи проекцій, включаючи:

- циліндричні (наприклад, Меркатора);
- конусні (наприклад, проекція Ламберта);
- азимутальні (наприклад, проекція Ламберта);

Висвітлено, як вибір проекції залежить від конкретної мети картографування.

3. Вплив спотворень.

Автори наголошують на важливості врахування спотворень, що виникають у процесі проекції, і підкреслюють необхідність вибору проекції, що найкраще відповідає завданням аналізу або візуалізації.

4. Геопросторові дані та проекції.

Розглядається вплив проекцій на обробку геопросторових даних у географічних інформаційних системах:

- як різні проекції впливають на точність обчислень (наприклад, площу, відстані чи буферизацію);
- чому важливо зберігати метадані про системи координат у геопросторових наборах даних;
- переваги використання уніфікованих стандартів, таких як EPSG-коди;

5. Рекомендації для практичного застосування.

У книзі наведено практичні поради щодо вибору та роботи з проекціями:

- використання геодезичних основ (еліпсоїдів, таких як WGS84);

- застосування локальних проекцій дрібномасштабного картографування;
- вибір глобальних проекцій для аналізу великих регіонів;

Значення для геопросторового аналізу:

Картографічні проекції є фундаментальним елементом ГІС. Неправильний вибір проекції може призвести до помилкових результатів, особливо у складних геометричних методах, таких як буферизація чи аналіз відстаней, які активно використовуються у географічному моделюванні [7].

Наукова робота «A Proposed Hybrid Spatial Data Structure Based on KD Tree and Quad Tree» автора Mahmood A. Mahmood пропонує новий гібридний підхід для роботи зі складними просторовими даними. Основна ідея полягає в об'єднанні характеристик KD-дерева та Quad-дерева, щоб оптимізувати пошук і зберігання просторових об'єктів. Ця методика забезпечує адаптивність до різних типів даних, таких як точки та багатокутники, і покращує продуктивність у порівнянні зі стандартними деревами.

Основні особливості роботи:

1. Комбінація структур: KD-дерева добре підходять для роботи з точками, тоді як Quad-дерева ефективні при роботі з багатокутниками. Гібридна структура використовує Quad-дерево для загальної організації простору, а KD-дерево — для обробки точкових даних у межах кожного вузла.

2. Оптимізація продуктивності: за експериментальними результатами, гібридна структура знижує час пошуку в порівнянні з традиційними деревами:

- для точок — до 56.6% у порівнянні з KD-деревом і до 45% з Quad-деревом;
- для багатокутників — до 71.4% у порівнянні з Quad-деревом;

3. Практичне застосування: така структура корисна для геоінформаційних систем, які працюють із великими наборами просторових даних із різною щільністю розподілу.

Незважаючи на переваги, метод має недоліки, зокрема у високій складності реалізації та значному часі на побудову гібридної структури. Однак він демонструє потенціал для застосування в реальних сценаріях, таких як управління даними у

ГІС-системах із низькими обчислювальними потужностями клієнтських пристроїв [8].

Публікація "Geospatial Data Management Research: Progress and Future Directions" дає вичерпний огляд поточного стану досліджень і тенденцій у галузі управління геопросторовими даними, висвітлюючи як основні досягнення, так і проблеми, що виникають на шляху інтеграції нових технологій у цю сферу.

Основні досягнення та тенденції:

2. Обробка великих даних (Big Data).

Геопросторові дані, особливо ті, що генеруються сенсорами, супутниками або мобільними пристроями, характеризуються великими обсягами та високою швидкістю надходження. Це створює виклик для традиційних методів обробки інформації. В результаті, нові методи обробки великих даних, такі як розподілені обчислення та хмарні технології, стали важливими для ефективного зберігання та аналізу геопросторової інформації. Ці технології дозволяють значно зменшити витрати часу і ресурсів, одночасно забезпечуючи високу точність і оперативність даних.

3. Інтеграція різних типів даних.

В сучасному управлінні геопросторовими даними важливо інтегрувати різноманітні джерела інформації, такі як радарні дані, супутникові знімки, мобільні дані, відкриті дані, тощо. Для цього розробляються методи, що дозволяють комбінувати різні формати даних і отримувати єдину картину, що забезпечує більш точні прогнози та рішення. Це особливо важливо для сфер, як-от моніторинг навколишнього середовища, планування міст, сільське господарство тощо.

4. Штучний інтелект і машинне навчання.

Використання штучного інтелекту (ШІ) та машинного навчання в аналізі геопросторових даних дозволяє автоматизувати багато аспектів обробки даних, зокрема класифікацію зображень, прогнозування та оптимізацію маршрутів. Це стало важливим етапом у розвитку геопросторових технологій, оскільки дозволяє швидше та точніше обробляти та інтерпретувати великі обсяги даних.

5. Динамічні та інтелектуальні картографічні системи.

Розвиток технологій дозволяє створювати інтелектуальні картографічні системи, які не лише відображають дані, а й дозволяють користувачам взаємодіяти з ними в реальному часі. Це включає в себе системи підтримки прийняття рішень, які аналізують геопросторові дані та пропонують різні варіанти дій залежно від ситуації.

Проблеми та виклики:

1. Масштабованість та ефективність.

Одним з основних викликів є забезпечення масштабованості обробки геопросторових даних. Традиційні бази даних не завжди здатні ефективно працювати з великими обсягами інформації, тому використовуються новітні підходи, такі як бази даних, що підтримують просторові індекси, та хмарні рішення для зберігання і обробки.

2. Точність і якість даних.

Геопросторові дані часто містять помилки або неповноту, що ускладнює їх використання для прийняття рішень. Одним з важливих напрямків є розробка методів, що дозволяють оцінити і покращити якість даних, а також перевіряти їх достовірність перед використанням у критичних додатках, таких як екологічний моніторинг чи планування інфраструктури.

3. Конфіденційність та безпека.

У сучасних системах управління геопросторовими даними зростає значення забезпечення конфіденційності та безпеки, особливо коли йдеться про чутливу інформацію, як-от дані про місце розташування осіб або важливі інфраструктурні об'єкти. Розробка нових методів захисту даних, таких як шифрування або анонімізація, є необхідною для забезпечення безпеки.

Публікація є важливим ресурсом для дослідників і практиків у галузі геопросторових даних, оскільки вона не лише підсумовує досягнення, але й вказує на ключові напрямки для подальших досліджень. Вона допомагає визначити найбільш актуальні виклики та можливості, що стоять перед цією сферою, і пропонує рекомендації щодо оптимізації обробки та використання геопросторових даних у різних галузях.

В ході проведеного огляду актуальних наукових робіт у галузі географічних проекцій, було виявлено значний розвиток теоретичних і практичних аспектів цієї теми. Зокрема, сучасні дослідження зосереджуються на оптимізації методів перетворення просторових даних, покращенні точності при відображенні земної поверхні в різних координатних системах, а також на застосуванні новітніх математичних і комп'ютерних моделей для вирішення проблеми трансформації проекцій. Зростає інтерес до гібридних та адаптивних проекцій, які враховують специфічні характеристики території чи масштабів дослідження. Зазначений розвиток сприяє точнішому аналізу географічних даних, а також розвитку технологій візуалізації та моделювання.

Таким чином, географічні проекції залишаються важливою складовою геопросторових досліджень, і подальші наукові розробки сприятимуть підвищенню ефективності картографії та геоінформаційних систем [9]

2 АНАЛІЗ МЕТОДІВ ТА АЛГОРИТМІВ ДЛЯ РОБОТИ З ГЕОГРАФІЧНИМИ ПРОЕКЦІЯМИ

2.1 Алгоритми та методи для роботи з географічними проекціями

Основні методи для роботи з географічними проекціями:

1. Буфер геометрії (Polygon buffer) — для створення області навколо геометрії яка створюється на певній відстані від її меж. Використовується для аналізу просторових відстаней. Наприклад, якщо потрібно визначити зону впливу від дороги або річки.

2. Дистанція геометрії (Polygon distance) — для розрахунку відстані між геометріями. Використовується для аналізу вістаней між різними точками на геометрії. Наприклад, якщо потрібно розрахувати відстань між центром та початком різних населених пунктів.

3. Перетин геометрій (Intersection of geometries) — це операція, яка визначає спільну область між двома або більше геометріями. Вона дозволяє знайти частини геометрій, які мають спільні точки або області. Наприклад, якщо потрібно знайти зони, де перетинаються два транспортні потоки або зони впливу різних об'єктів, то використовується перетин. Ця операція корисна при аналізі взаємодії різних об'єктів на місцевості.

4. Об'єднання геометрій (Union of geometries) — дозволяє об'єднати дві або більше геометрії в одну. Це дає змогу створювати великі об'єкти із суміші кількох просторових елементів. Наприклад, можна об'єднати різні ділянки лісу чи парку, щоб створити загальний коридор для досліджень біорізноманіття або розрахунків територій для охорони навколишнього середовища.

5. Різниця геометрій (Difference of geometries) — це операція, яка знаходить частину геометрії, що залишається після виключення перетину з іншою геометрією. Наприклад, при аналізі земельних ділянок можна визначити, яку частину території

вже використовують під забудову, виключаючи ділянки, зайняті іншими об'єктами, такими як дороги або водні ресурси.

6. Буфер із множинними відстанями (Multi-ring buffer) — це розширення буферної операції, яке створює декілька зон навколо об'єкта на різних відстанях. Цей метод корисний, наприклад, для оцінки впливу шуму або забруднення на різних відстанях від джерела, таких як аеропорти або промислові об'єкти.

7. Визначення центроїда (Centroid calculation) — центроїд геометрії є її "центром тяжіння" або середньою точкою, яку можна використовувати для аналізу, наприклад, для розрахунку відстані до інших геометричних об'єктів або для порівняння розмірів територій. Ця точка не завжди перебуває всередині об'єкта, зокрема для складних форм.

8. Тривимірна геометрія (3D geometry) — це розширення стандартних двовимірних операцій на тривимірний простір. Це включає створення 3D-буферів, 3D-дистанцій та перетинів для геометрій, що мають висоту або глибину (наприклад, для аналізу об'єктів у гірських районах або підземних структурах).

9. Опукла оболонка (Convex Hull) — це операція, яка знаходить найменшу опуклу геометрію, що повністю охоплює задану множину точок або геометричних об'єктів. Ідея полягає в тому, щоб побудувати багатокутник, вершини якого є точками з множини, а сам багатокутник містить усі ці точки всередині або на його межах [10].

Проблеми з географічними координатами:

1. Нерівномірність відстаней:

Відстань між двома точками на земній поверхні в географічних координатах не є сталою. Це пов'язано з тим, що один градус довготи відповідає різній відстані в залежності від широти. На екваторі 1 градус довготи дорівнює приблизно 111 км, на полюсах відстань між градусами довготи наближається до нуля. Це призводить до того, що відстань у градусах може бути сильно спотворена, якщо не враховувати цю зміну;

2. Різні одиниці вимірювання:

У географічних системах координат (широта/довгота) одиницею виміру є градуси, що не є безпосередньо пов'язаними з лінійними одиницями вимірювання, такими як метри;

Якщо намагатися використовувати такі одиниці як метри для аналізу просторових відстаней, важливо правильно трансформувати координати в проекційну систему, де одиниці вимірювання лінійні;

Вирішення проблеми з географічними координатами:

Щоб уникнути спотворень, географічні координати (широта та довгота) часто перетворюються в проекційну систему координат, де одиниці виміру є лінійними і постійними на всій території. Одним з таких варіантів є використання проекцій, таких як:

- проекція UTM (Universal Transverse Mercator) одна з найбільш поширених проекцій, яка дає лінійні одиниці виміру в метрах;

- проекція Ламберта або Меркатора, які застосовуються в різних географічних регіонах;

Алгоритм дій для розрахунку з використанням методів для роботи з географічними проекціями:

1. Визначення системи координат геометрії.

Для правильного розрахунку потрібно щоб геометрія та вхідні дані мали однакову одиницю виміру.

2. Перетворення в одиниці виміру.

Якщо геометрія та вхідні дані мають різні одиниці виміру, щоб запобігти спотворенням і зберегти точність, слід перетворити географічні координати в метричні одиниці за допомогою відповідної проекції.

3. Застосування необхідного методу.

Після того як геометрія та вхідні дані мають однакові одиниці виміру можна застосовувати необхідний метод для розрахунку, такий як буфер геометрії, дистанції геометрії, чи інші.

4. Зворотнє перетворення.

Після всіх обчислень важливо повернути дані до оригінальної системи координат, якщо це необхідно, для забезпечення відповідності з початковими даними або для подальшої роботи в оригінальній системі координат.

Для досягнення максимальної точності розрахунків важливо уважно вибрати оптимальну проекцію залежно від розташування геометрії. Кожна проекція має свої особливості, і їх вибір може суттєво вплинути на результати.

1. Аналіз розташування геометрії.

Необхідно визначити географічне положення геометрії. Це допоможе зрозуміти, які проекції найкраще підходять для цієї конкретної області. Наприклад, для об'єктів, розташованих поблизу екватора, підходять одні проекції, тоді як для полярних регіонів зовсім інші.

2. Вибір відповідної проекції.

Існують різні типи проекцій. Кожна з них має свої переваги та недоліки залежно від завдання. Наприклад, проекція Меркатора добре підходить для навігаційних цілей, але може спотворювати відстані в полярних областях.

3. Перевірка точності проекції.

Після вибору проекції необхідно перевірити її точність для конкретної геометрії. Це можна зробити, порівнюючи результати розрахунків у різних проекціях і визначаючи, які з них дають найточніші результати.

4. Корекція результатів.

Якщо обрана проекція виявляється не оптимальною, можливо, доведеться скоригувати розрахунки або вибрати іншу проекцію. Це дозволить мінімізувати похибки та забезпечити точність результатів.

5. Застосування проекції в розрахунках.

Після вибору оптимальної проекції її можна застосувати до геометрії. Це дозволить провести точні розрахунки, наприклад, створення буферних зон або визначення відстаней між об'єктами.

Правильний вибір проекції є ключовим етапом, який впливає на точність та достовірність геометричних розрахунків. Це не просто технічний крок, а важлива частина всього аналітичного процесу.

2.2 Алгоритми перетворення координат

Алгоритми перетворення координат між різними географічними проекціями включає в себе перетворення географічних координат (широти та довготи) у координати на плоскій поверхні та навпаки. Під час перетворення можуть виникати спотворення, і задача полягає в їх мінімізації для забезпечення необхідної точності [10].

Географічні координати (широта та довгота) спершу використовуються для визначення положення на Землі у сферичній або еліпсоїдальній системі. Після цього координати перетворюються в планарні координати на площині за допомогою певної проекції. Існує кілька основних типів проекцій, і кожна з них має власний алгоритм перетворення.

Алгоритм перетворення з географічних координат у площинні координати:

1. Циліндричні.

Для циліндричних проекцій використовуються рівняння, що передбачають накладання глобуса на циліндр. Широта (ϕ) та довгота (λ) перетворюються в координати (x) і (y) на площині.

Формула перетворення (проекція Меркатора):

$$x = R * (\lambda - \lambda_0),$$

$$y = R * \ln(\tan(\frac{\pi}{4} + \frac{\phi}{2})), \quad (2.1)$$

де R - радіус Землі;

λ - довгота в радіанах;

ϕ - широта в радіанах;

Циліндричні проекції створюються шляхом проектування поверхні Землі на циліндр, який потім розгортають у площину.

- спотворення площі: значні в областях, що віддалені від екватора, особливо в полярних регіонах;

- спотворення форми: зазвичай збережені поблизу екватора, але стають значними ближче до полюсів;

- спотворення відстані: точність зберігається лише в напрямку уздовж меридіанів або паралелей залежно від проекції;

- спотворення напрямку: у багатьох циліндричних проекціях, таких як Меркатора, напрямки зберігаються;

2. Конічні.

Для конічних проекцій одна зі складних частин це визначення параметрів, таких як стандартні паралелі. У проекції Ламберта координати визначаються за допомогою спрощених рівнянь, зокрема, використовуючи стандартні паралелі (де проекція має найменші спотворення).

Формула перетворення (проекція Ламберта):

$$n = \frac{\ln(\cos(\phi_1) * \sec(\phi_2))}{\ln(\tan(\frac{\pi}{4} + \frac{\phi_2}{2}) * \cot(\frac{\pi}{4} + \frac{\phi_1}{2}))'}$$

$$F = \frac{\cos(\phi_1) * \tan(\frac{\pi}{4} + \frac{\phi_1}{2})^n}{n},$$

$$\rho = R * F * \cot(\frac{\pi}{4} + \frac{\phi_1}{2})^n,$$

$$\rho_0 = R * F * \cot(\frac{\pi}{4} + \frac{\phi_0}{2})^n,$$

$$x = \rho * \sin(n * (\lambda - \lambda_0)),$$

$$y = \rho_0 - \rho * \cos(n * (\lambda - \lambda_0)), \quad (2.2)$$

де R - радіус Землі;

λ - довгота в радіанах;

λ_0 - центральна довгота (центральный меридіан);

ϕ - широта в радіанах;

ϕ_0 - широта де у координата буде 0 на центральному меридіані;

ϕ_1, ϕ_2 - стандартні паралелі (де проекція має мінімальні спотворення).

n - коефіцієнт конічної проекції;

F - коефіцієнт нормалізації;

Конічні проекції створюються шляхом проектування поверхні Землі на конус, який дотикається до планети уздовж однієї або двох паралелей.

- спотворення площі: помірні, але збільшуються в міру віддалення від стандартних паралелей;

- спотворення форми: форми добре зберігаються в межах стандартних паралелей, але викривляються поза ними;

- спотворення відстані: відстані найбільш точні вздовж стандартних паралелей;

- спотворення напрямку: Залежно від конкретної проекції, напрямки можуть зберігатися уздовж певних ліній, але загалом викривляються;

3. Азимутальні.

У азимутальних проекціях точка проекції розташована в центрі Землі або на одному з полюсів. Усі прямі лінії, що виходять з центральної точки, зберігають напрямки (азимуту).

Формула перетворення (проекція Ламберта):

$$k = \frac{2}{1 + \sin(\phi_1) * \sin(\phi) + \cos(\phi_1) * \cos(\phi) * \cos(\lambda - \lambda_0)},$$

$$x = k * \cos(\phi) * \sin(\lambda - \lambda_0),$$

$$y = k * [\cos(\phi_1) * \sin(\phi) - \sin(\phi_1) * \cos(\phi) * \cos(\lambda - \lambda_0)], \quad (2.3)$$

де R - радіус Землі;

λ - довгота в радіанах;

λ_0 - центральна довгота (центральный меридіан);

ϕ - широта в радіанах;

ϕ_0 - широта де у координата буде 0 на центральному меридіані;

ϕ_1, ϕ_2 - стандартні паралелі (де проекція має мінімальні спотворення).

n - коефіцієнт конічної проекції;

Азимутальні проекції проектують Землю на площину, дотичну до точки на її поверхні.

- спотворення площі: зростають із віддаленням від центральної точки, інколи значно;

- спотворення форми: форми об'єктів поблизу центральної точки відображаються точно, але стають сильно спотвореними на периферії;

- спотворення відстані: у рівновіддалених азимутальних проекціях відстань зберігається від центральної точки, але не між іншими точками;

- спотворення напрямку: напрямки від центральної точки точні, але можуть бути спотворені в інших місцях;

Отже циліндричні проекції підходять для карт, орієнтованих на екваторіальні регіони, конічні проекції краще підходять для середніх широт, азимутальні проекції корисні для відображення полюсів або регіонів навколо конкретної точки;

Алгоритми перетворення координат між різними проекціями є важливою складовою частиною географічних та картографічних систем. Вибір правильного методу залежить від конкретних вимог до точності, площі оброблюваної території та особливостей використання проекції.

На рисунку 2.1 [11], зображено приклад перетворення координат з трьовимірної поверхні Землі на площину.

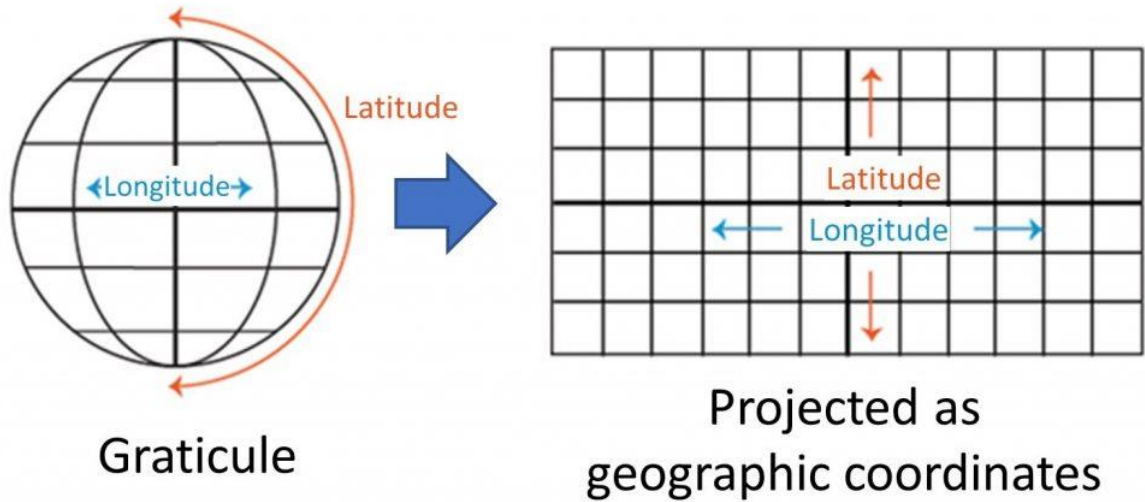


Рис. 2.1 Приклад перетворення координат

2.3 Оцінка деформацій у географічних проекціях

Географічні проекції завжди супроводжуються деформаціями, оскільки сферична (або еліпсоїдальна) поверхня Землі перетворюється на площину. Оцінка деформацій є ключовим етапом аналізу проекцій, адже вона дозволяє зрозуміти, наскільки точно проекція відображає реальні параметри.

При переході від поверхні Землі до площини виникають наступні типи деформацій:

- деформації кутів (конформність): вплив на форми об'єктів. У конформних проекціях кути між лініями зберігаються, але спотворюються довжини. Конформні проекції зберігають кути, але спотворюють відстані та площі. Це означає, що геометрія форм об'єктів на карті може залишатися точною, але їх масштаб може бути змінений. Наприклад, проекція Меркатора є конформною, що робить її корисною для навігації, оскільки вона зберігає напрямки і кути між об'єктами;

- деформації площ: зміна розмірів територій. Рівноплощинні проекції зберігають площі, але спотворюють форми. У рівноплощинних проекціях

зберігаються площі об'єктів, але форми можуть бути сильно спотворені. Такі проекції використовуються для тем, де важливі пропорції площ, наприклад, при представленнях статистичних даних, що потребують точності площ;

- деформації довжин: лінійні спотворення, які впливають на відстані між точками. Цей тип деформації змінює відстані між точками на карті. У циліндричних проекціях, таких як Меркатор, відстані на великих широтах збільшуються, в той час як на малих широтах спотворення є мінімальними. Проблема деформації довжин особливо важлива при вимірюваннях між точками на великих відстанях;

- комплексні деформації: комбінація перелічених спотворень, які одночасно впливають на форми, площі та відстані. Це спотворення, що виникає через зміну кутів між лініями на карті. Кутова деформація є важливою, коли потрібно зберегти орієнтацію ліній, як у картографії для навігації або побудови топографічних карт;

Для аналізу та оцінки деформацій використовуються різні метрики та алгоритми:

1. Індикатриса Тіссо.

Цей метод дає змогу побудувати еліпс, що представляє локальні деформації, з розмірами, що відповідають масштабам спотворень. Наприклад, якщо індикатриса є круглою, це означає, що спотворення на даній ділянці картки мінімальні (зберігаються пропорції). Якщо еліпс витягнутий, це вказує на більші деформації.

2. Деформаційні коефіцієнти.

Коефіцієнт лінійного розтягнення визначає зміну довжини між двома точками. Коефіцієнт площинної деформації визначає співвідношення площі на карті до реальної площі. Кутова деформація відображає зміну кута між перпендикулярними лініями.

4. Візуалізація деформацій.

Оцінка деформацій це важливий крок у при застосуванні географічних проекцій, який дозволяє забезпечити точність і надійність географічних даних.

На рисунку 2.2 [1012], зображено приклад деформацій на проекції Меркатора, можна побачити, як однакові за розміром фігури змінюють свої розміри залежно від їхнього місцезнаходження на карті. Це є характерним результатом для проекції Меркатора деформації площ, особливо біля полюсів.

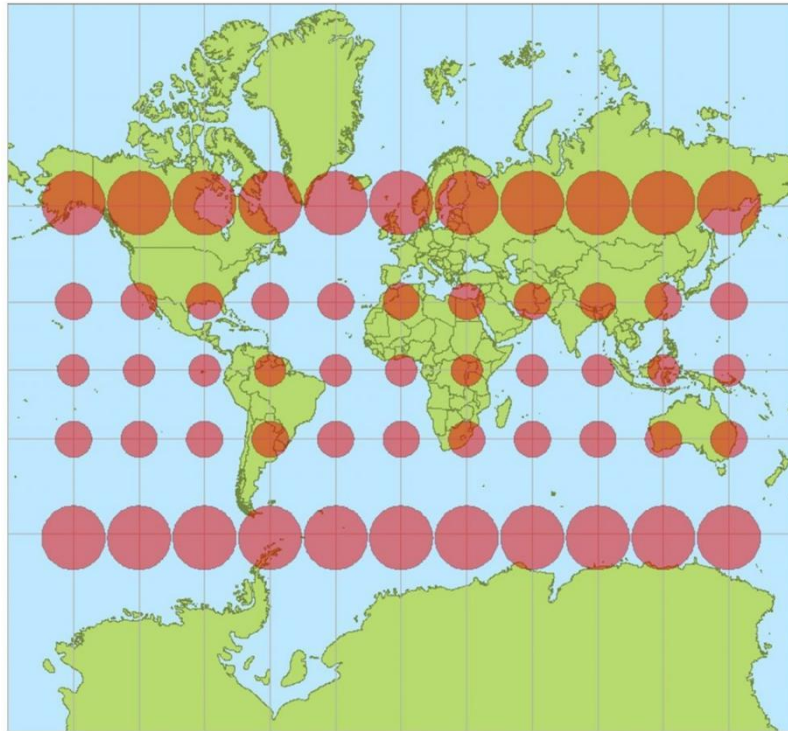


Рис. 2.2 Приклад деформації

Щоб зрозуміти, як виникають спотворення при перенесенні сферичної поверхні Землі на площину, важливо усвідомити природу таких перетворень.

Причина цього криється в геометричній несумісності сферичної поверхні з площиною. Неможливо розгорнути сферу без порушення однієї чи кількох властивостей: форми, площі, відстані або напрямків.

На малюнку 2.3 [1012], зображено сферу із нанесеною на неї сіткою меридіанів і паралелей, які рівномірно покривають поверхню. На сфері ці лінії зберігають правильні геометричні співвідношення, але при розгортанні в площину вони неминуче деформуються.

Ці спотворення виникають через різницю в масштабі, який неоднорідно змінюється залежно від розташування об'єктів на карті. В екваторіальних областях,

де проекція краще відповідає реальній сфері, спотворення мінімальні. У полярних регіонах або далеко від центру проекції деформації посилюються. Це може проявлятися у видовженні, сплюсненні або викривленні об'єктів.

На карті це спотворення можуть проявлятися у вигляді зсуву, розривання або стискання.

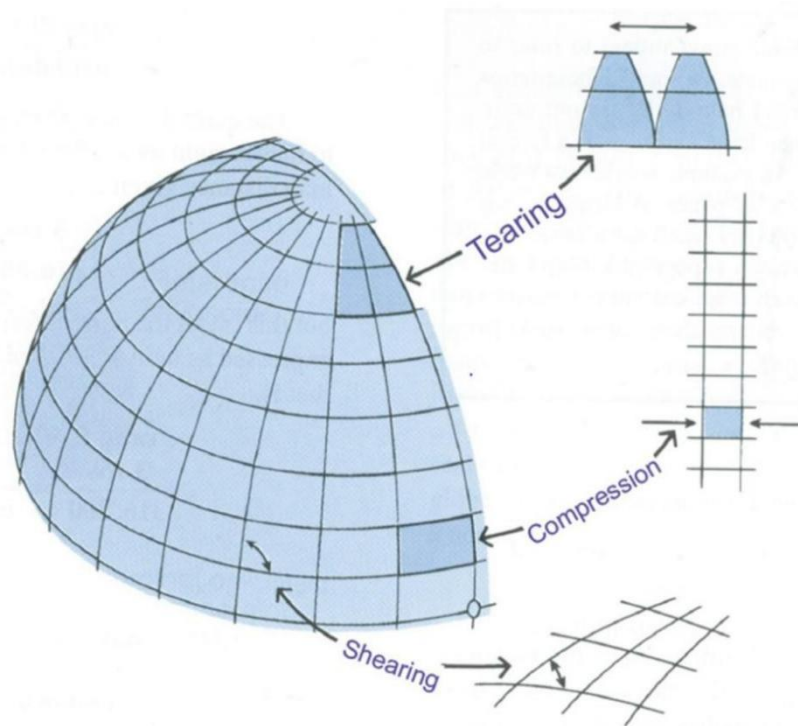


Рис. 2.3 Приклад деформації

2.4 Автоматизація обробки географічних проекцій

Обробка географічних проекцій та геометрії вимагає виконання складних математичних операцій, які можуть бути важкими для реалізації без використання спеціалізованих бібліотек. Ось основні проблеми, які виникають під час розв'язання таких завдань:

1. Перетворення географічних координат між проекціями.

Географічні координати (широта/довгота) використовуються у проекції EPSG:4326, але в деяких випадках може знадобитися перетворити ці координати в іншу проекцію, наприклад, UTM (EPSG:32633) або меркатор (EPSG:3857).

Перетворення між різними проекціями вимагає виконання математичних операцій, таких як трансформація матриць, врахування еліпсоїдних властивостей Землі та інші складні геодезичні розрахунки.

2. Обробка геометрії.

Геометричні операції (такі як знаходження перетину, об'єднання, різниці) з географічними об'єктами (точки, лінії, полігони) потребують точних обчислень для визначення просторових відносин між об'єктами. Наприклад, потрібно обчислити, чи перетинаються два полігони, чи точка знаходиться всередині полігона, чи точка знаходиться на певній відстані від лінії.

3. Обчислення відстані та буферів.

Для багатьох задач, таких як пошук найближчих об'єктів, необхідно обчислити відстань між географічними точками чи між об'єктами, а також створювати буфери навколо геометричних об'єктів. Врахування кривизни Землі та коректне обчислення відстані у різних проекціях може бути складним без правильних математичних інструментів.

4. Зберігання та інтеграція даних.

Географічні дані можуть зберігатися у різних форматах (наприклад, WKT, GeoJSON) і потребують інтеграції з іншими системами, такими як бази даних (наприклад, PostGIS). Без використання бібліотек складно працювати з такими форматами та обробляти дані на високому рівні.

GeoTools та JTS є двома популярними бібліотеками для обробки географічних даних, кожна з яких має свої особливості та сфери застосування.

1. GeoTools

GeoTools — це бібліотека з відкритим вихідним кодом, призначена для роботи з просторовими даними. Вона підтримує широкий спектр стандартів та форматів географічних даних, таких як WKT, GML, GeoJSON, shapefiles тощо.

Основні можливості:

- Обробка географічних даних: GeoTools підтримує безліч географічних операцій, таких як перетворення проекцій, обчислення буферів, знаходження перетину та обчислення відстані між об'єктами;

- Підтримка стандартів: GeoTools реалізує стандарти OGC (Open Geospatial Consortium), включаючи WMS, WFS, WCS, GML, і KML;
- Підтримка різних проекцій: GeoTools дозволяє працювати з різними географічними проекціями та здійснювати трансформації між ними;
- Інтеграція з базами даних: GeoTools можна використовувати для роботи з просторовими базами даних, такими як PostGIS, а також для імпорту/експорту даних;

JTS — це бібліотека для обробки просторової топології, яка надає потужні інструменти для роботи з геометрією, зокрема для маніпулювання точками, лініями, полігональними об'єктами, а також для виконання геометричних операцій.

Основні можливості:

- Геометричні операції: JTS дозволяє виконувати різноманітні операції з геометрією, наприклад, обчислення перетинів, об'єднань, різниць та буферів для багатьох типів геометрій (точки, лінії, полігони);
- Обробка просторових запитів: виконує просторові запити, такі як перевірка, чи містить геометрія іншу геометрію, чи перетинаються вони тощо;
- Аналіз топології: підтримка операцій для перевірки топологічних властивостей геометрій;

Отже, без спеціалізованих бібліотек, таких як GeoTools та JTS, автоматизація обробки географічних проекцій та геометрії є складним завданням, що вимагає значних зусиль у розробці алгоритмів та розуміння географічних обчислень. Використання цих бібліотек значно спрощує роботу, надаючи готові інструменти для вирішення більшості завдань, пов'язаних з географічними проекціями, геометрією та просторовими запитам.

2.5 Формати та інструменти для географічних проекцій

Географічні проекції і координатні системи використовуються для точного відображення географічних даних на плоских поверхнях, а також для забезпечення правильного поєднання різних географічних даних. Формати та стандарти

зберігання географічних проекцій, такі як WKT, KML, GeoJSON, і PROJ, мають важливе значення для забезпечення сумісності між різними географічними системами та програмами.

1. WKT — це текстовий формат для зберігання геометричних об'єктів та географічних проекцій. В основному використовується для представлення геометрій, таких як точки, лінії, полігони та їх варіанти, а також для визначення координатних систем. WKT є частиною стандарту OGC Simple Feature Specification, розробленого Open Geospatial Consortium.

WKT містить координати геометричних об'єктів, а також їх тип (наприклад, ``POINT``, ``LINESTRING``, ``POLYGON``).

У WKT також можна зберігати опис проекцій або географічних координатних систем через використання параметрів Spatial Reference System Identifier.

2. KML — це XML-формат, який був розроблений для зберігання та обміну геопросторовими даними, і в основному використовується для відображення даних на картографічних платформах, таких як Google Earth або Google Maps.

KML використовує XML для структурованого опису географічних об'єктів, таких як точки, лінії, полігони, а також метадані для визначення стилів, символів та шляхи.

KML підтримує лише стандартну географічну проекцію (WGS84), але можна вказати альтернативні координатні системи через властивість ``SRS``.

3. GeoJSON — це формат на основі JSON для зберігання географічних даних. GeoJSON є простим і легким форматом для зберігання точок, ліній, полігонів і колекцій геометрій. Формат активно використовується в веб-геоінформаційних системах.

GeoJSON визначає географічні об'єкти через координати та атрибути у форматі JSON. Він підтримує такі геометрії, як ``Point``, ``LineString``, ``Polygon``, а також колекції об'єктів (наприклад, ``FeatureCollection``).

GeoJSON використовує координати у стандартній проекції WGS84, але може включати метадані для підтримки інших систем.

4. PROJ — це стандарт для перетворення координат між різними географічними проекціями. Вона дозволяє перетворювати координати з однієї проекції в іншу.

PROJ використовується для опису математичних моделей для перетворення координат між різними системами.

У PROJ визначаються параметри проекції, такі як центральний меридіан, географічний масштаб, стандартні паралелі тощо.

Кожен з цих форматів має свої переваги та недоліки в залежності від потреб користувача. WKT і GeoJSON найбільше використовуються для зберігання географічних об'єктів у текстовому або JSON форматах, в той час як KML чудово підходить для візуалізації на картографічних платформах. PROJ, у свою чергу, є потужним інструментом для перетворення координат між різними проекціями і датурами.

У результаті огляду актуальних форматів та інструментів для географічних проекцій було виявлено широкий спектр рішень, які забезпечують ефективну роботу з географічними даними. Кожен з інструментів має свої переваги, залежно від специфіки задач, таких як точність відображення, швидкість обробки та здатність працювати з великими обсягами даних. Огляд показав, що для успішної реалізації географічних проекцій важливо вибрати оптимальні формати даних, а також інструменти, які забезпечують інтеграцію з різними типами систем та дозволяють виконувати складні геометричні операції з мінімальними витратами часу та ресурсів.

З огляду на швидкий розвиток технологій, майбутні інструменти для роботи з проекціями, ймовірно, будуть ще більш гнучкими та інтегрованими в сучасні інформаційні системи.

3 РОЗРОБКА АЛГОРИТМІВ ТА ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Алгоритм пошуку оптимальних проекцій

У сфері геоінформаційних систем правильний вибір картографічної проекції є критично важливим для забезпечення точності аналізу, візуалізації даних та просторових розрахунків. Зазвичай, проекції використовуються для відображення реальних географічних об'єктів на площині. Проте різні проекції мають свої специфічні властивості, які впливають на точність передачі відстаней, площ, кутів та форми.

Коли працюють із множиною полігонів у різних системах координат, виникають труднощі у визначенні оптимальної проекції для аналізу та візуалізації.

Основні проблеми:

1. Сумісність: полігони можуть використовувати різні одиниці виміру (наприклад, метри чи градуси), що робить обчислення некоректними.
2. Спотворення: обрані проекції можуть призводити до значних викривлень геометричних даних, особливо якщо полігони охоплюють великі території.
3. Автоматизація: процес вибору відповідної проекції зазвичай виконується вручну, що є трудомістким та схильним до помилок.

Алгоритм пошуку оптимальних проекцій вирішує цю проблему шляхом автоматичного аналізу та вибору проекцій, які найкраще відповідають геометричним характеристикам вхідних полігонів. Він враховує одиниці виміру, геометричні буфери та перевіряє відповідність даних до властивостей доступних проекцій. Такий підхід мінімізує ризик спотворень, забезпечуючи точність аналізу та спрощуючи роботу користувача.

На рисунку 3.1 зображено Алгоритм пошуку оптимальних проекцій та поетапні кроки виконання. Цей алгоритм забезпечує систематичний підхід до вибору географічних проекцій, що дозволяє мінімізувати похибки у відображенні просторових даних.

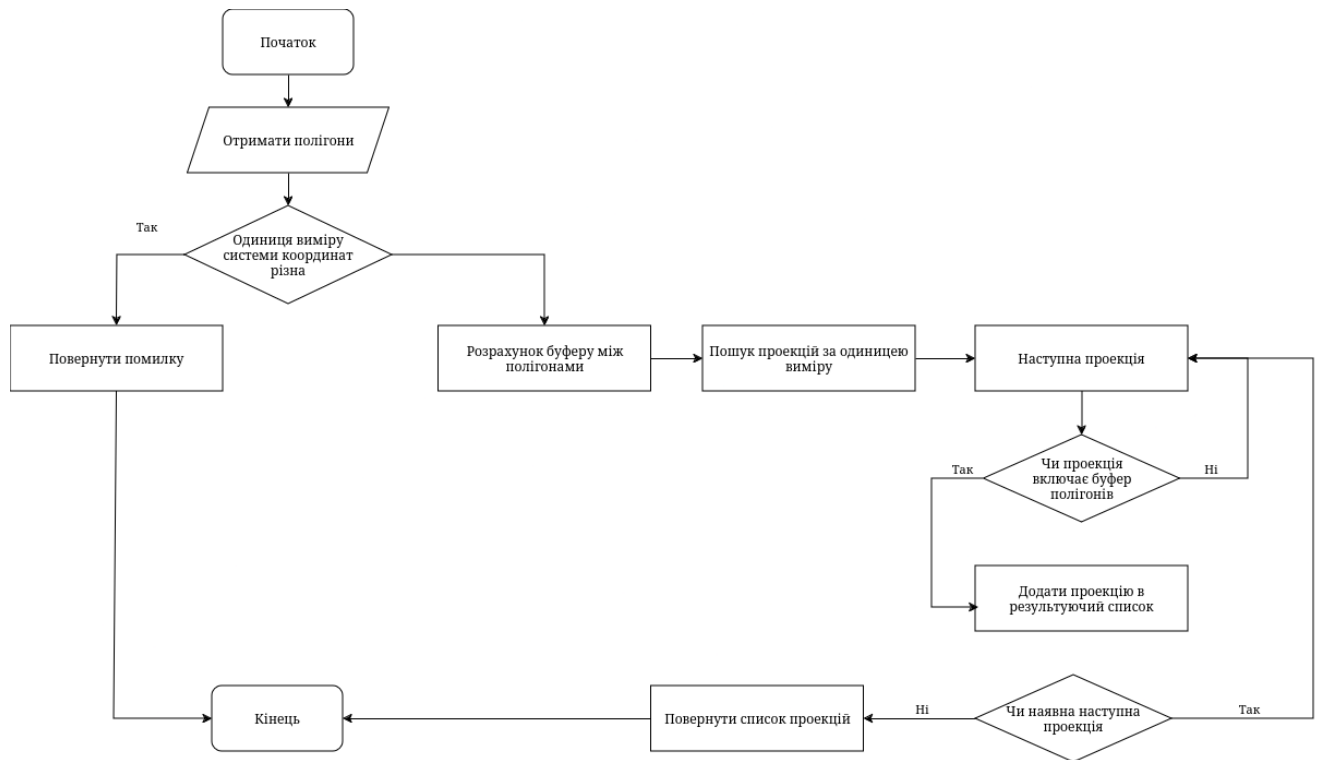


Рис. 3.1 Алгоритм пошуку оптимальних проекцій

Алгоритм пошуку оптимальних проекцій включає:

1. Отримання полігонів: на вхід алгоритм приймає набір геометричних полігонів. Ці полігони можуть представляти об'єкти чи області на карті.

2. Перевірка одиниці виміру: для кожного полігона визначається одиниця виміру (наприклад, метри, градуси, фути тощо). Якщо полігони мають різні одиниці виміру, алгоритм повертає помилку, оскільки некоректно працювати з полігонами, що мають різні системи координат.

3. Пошук проекцій за одиницею виміру: на основі визначеної одиниці виміру формується список доступних проекцій. Наприклад: Для метрів — EPSG:3857 та інші, для градусів — EPSG:4326, та інші.

4. Розрахунок буфера між полігонами: для кожного полігона створюється буфер певного розміру. Це робиться для того щоб об'єднати полігони, що забезпечить можливість підібрати оптимальну проекцію яка буде включати об'єднаний полігон.

5. Перевірка проєкцій: перевіряється, чи всі полігони і буфери коректно вписуються в межі цієї проєкції. Тобто чи не виникають спотворення, некоректні обрізання або виходи за межі системи координат.

6. Формування результуючого списку: якщо проєкція успішно враховує всі полігони і їх буфери, вона додається в результуючий список оптимальних проєкцій.

7. Повернення списку проєкцій: результатом алгоритму є список проєкцій, які підходять для обраного набору полігонів і задачі.

Переваги оптимальної проєкції:

1. Зменшення спотворень: за допомогою оптимальної проєкції для конкретної території можна значно зменшити спотворення, зокрема при вимірюванні відстаней, площ чи кутів між полігонами.

2. Точність обчислень: оптимальна проєкція дозволяє точно вимірювати відстань між географічними об'єктами, що є критично важливим для географічного аналізу та моделювання.

3. Покращення результатів у великих масштабах: для великих територій, де спотворення традиційних проєкцій стають значними, оптимальна проєкція дозволяє забезпечити високу точність вимірювань.

3.2 Математична модель створення гібридної географічної проєкції

Традиційні проєкції підходять для багатьох задач, пошук оптимальних проєкцій може значно ефективніше підібрати найоптимальнішу проєкцію для заданих полігонів. Однак проблема полягає в тому, що полігони можуть бути розташовані на різних ділянках земної поверхні, і жодна стандартна проєкція не може забезпечити мінімальні спотворення одночасно для всіх регіонів.

Гібридні географічні проєкції є вирішуть проблему роботи з географічними даними, розташованими в різних частинах світу, особливо якщо класичні проєкції не забезпечують достатньої точності. Гібридні проєкції створюються шляхом поєднання характеристик кількох базових проєкцій із врахуванням специфіки

задачі. Наприклад, це може бути рівномірний розподіл спотворень для різних полігонів або оптимізація відображення геометрії в межах певної області.

Особливості гібридних проекцій:

1. Гнучкість: параметри проекції можуть бути адаптовані для зменшення спотворень в заданих областях.
2. Локалізованість: дозволяє враховувати специфіку розташування полігонів (наприклад, близькість до полюсів або екватора).
3. Математична оптимізація: створення проекції базується на оптимізаційній моделі, яка мінімізує спотворення для конкретних задач.

Для створення гібридної проекції було розроблено математичну модель, яка дозволяє обчислювати параметри проекції, включно з координатами меридіанів і паралелей, для заданого набору точок.

Основні кроки моделі:

1. Збір вхідних даних: визначаються координати полігонів або точок, для яких необхідно мінімізувати спотворення.
3. Обчислення меридіанів: за допомогою параметричних функцій розраховуються координати меридіанів і паралелей для проекції.
4. Формат PROJ: результати моделі описуються у форматі PROJ, що дозволяє використовувати їх у інформаційних системах.

Формула (3.1) - розрахунок параметрів для моделі проекції.

$$zone = \min\left(\left\lfloor \frac{longitude + 180}{6} \right\rfloor + 1, 60\right),$$

$$centralMeridian = -183 + zone * 6,$$

$$falseNorthing = \begin{cases} 0, & latitude \geq 0 \\ 10000000, & latitude < 0 \end{cases} \quad (3.1)$$

де longitude - координата довготи;

latitude - координата широти;

6 - один мередіан;

60 - всього мередіанів;

zone - розрахований мередіан;

centralMeridian - розрахований центральний мередіан;

falseNorthing - розраховане зміщення координатної сітки;

Модель проекції в форматі PROJ:

```
PROJCS["WGS 84 / Auto UTM",
  GEOGCS["WGS 84",
    DATUM["WGS_1984",
      SPHEROID["WGS_1984", 6378137, 298.257223563]],
    PRIMEM["Greenwich", 0],
    UNIT["Decimal_Degree", 0.0174532925199433]],
  PROJECTION["Transverse_Mercator"],
  PARAMETER["Latitude_of_Origin", 0],
  PARAMETER["Central_Meridian", ${centralMeridian}],
  PARAMETER["False_Easting", 500000],
  PARAMETER["False_Northing", ${falseNorthing}],
  PARAMETER["Scale_Factor", 0.9996],
  UNIT["Meter", 1]]
```

Модель проекції в форматі PROJ використовується для перетворення географічних координат (широта, довгота) в координати на площині карти (x, y) та навпаки. Це основа для роботи з картами в геоінформаційних системах.

Принцип роботи моделі проекції:

1. Геодезична основа.

Модель проекції базується на використанні геодезичних даних, які визначають форму Землі. Формат PROJ дозволяє враховувати референц-еліпсоїд і дати (наприклад, WGS84), які задають форму та розмір Землі для обчислень.

2. Перетворення географічних координат.

Географічні координати (широта та довгота), які описують положення точок на поверхні Землі, перетворюються у площинну систему координат. Цей процес враховує вибрану картографічну проекцію, яка визначає спосіб проектування тривимірної поверхні на двовимірну площину.

3. Викривлення у проекціях.

Будь-яка проекція неминує викликає викривлення, що може впливати на форму, площу, відстань або кути між об'єктами. PROJ дозволяє налаштовувати параметри проекції для зменшення цих викривлень залежно від цілей використання карти. Наприклад, одні проекції зберігають кути (конформні), інші площу (еквівалентні).

4. Проекційні параметри.

Для кожної проекції задаються параметри, які визначають її поведінку. Це може включати центральний меридіан, початкову широту, масштабні коефіцієнти, зміщення осей тощо. Ці параметри задають прив'язку проекції до реального світу.

5. Система координат.

Важливою частиною роботи з проекціями є узгодження систем координат. Формат PROJ дозволяє описувати системи координат за допомогою текстових специфікацій, забезпечуючи сумісність між різними джерелами даних.

Процес накладання проекції:

1. Ідентифікація системи координат.

Кожен набір географічних даних має власну систему координат. Для сумісного накладання шарів на карту всі вони повинні бути або у спільній системі координат, або трансформовані у неї.

2. Перетворення через проекцію.

Використовуючи модель PROJ, географічні координати трансформуються у площинні для відображення на карті. Це включає обчислення нових координат точок відповідно до вибраної проекції та її параметрів.

3. Рендеринг карти.

Після перетворення дані накладаються на карту, забезпечуючи їхнє правильне просторове розташування. Це дозволяє точно аналізувати взаємне розташування об'єктів, відстані, межі тощо.

3.3 Інформаційна система дослідження проекцій

Для того щоб продемонструвати роботу географічних проекцій, була розроблена інформаційна система дослідження географічних проекцій на поверхні Землі в двовимірному та тривимірному просторах. Ця система дозволяє користувачам вивчати, аналізувати та порівнювати різні типи географічних проекцій, що використовуються для відображення кривизни Землі на площині. Застосування географічних проекцій є важливим для картографії, навігації та аналізу просторових даних, оскільки Земля є сферичною поверхнею, і для її точного представлення в двовимірному вигляді необхідно здійснювати певні математичні перетворення.

Інформаційна система надає можливість обрати різні типи проекцій, такі як конформні проекції, рівномірно розподілені, а також багато інших, що використовуються для специфічних задач. Система дозволяє користувачам аналізувати точність цих проекцій, їх властивості, а також вплив на спотворення форми, площі, кута чи відстані. Зокрема, можна оцінити, як кожна проекція впливає на спотворення геометричних характеристик на різних частинах картографічного зображення.

В рамках тривимірного простору система демонструє, як географічні проекції взаємодіють з тривимірними моделями Землі. Важливим аспектом є інтерактивні функції, що дозволяють змінювати параметри проекцій і спостерігати за тим, як змінюється відображення різних географічних об'єктів. Користувачі можуть порівнювати проекції в реальному часі, що дозволяє краще зрозуміти, як обрана проекція може впливати на точність даних, а також на можливість їх

використання в конкретних застосуваннях, таких як супутникові знімки, геопросторовий аналіз або навігаційні системи.

Для більш глибокого розуміння основних принципів географічних проекцій система містить навчальні матеріали, що охоплюють теоретичні аспекти, такі як властивості різних типів проекцій. Крім того, система підтримує візуалізацію проекцій у вигляді карт, що дозволяє проводити порівняння між різними методами відображення Землі, розглядаючи їх переваги та обмеження залежно від конкретних задач.

На рисунках 3.2 та 3.3 зображено приклади системи, яка демонструє роботу географічних проекцій у двовимірному та тривимірному просторах.

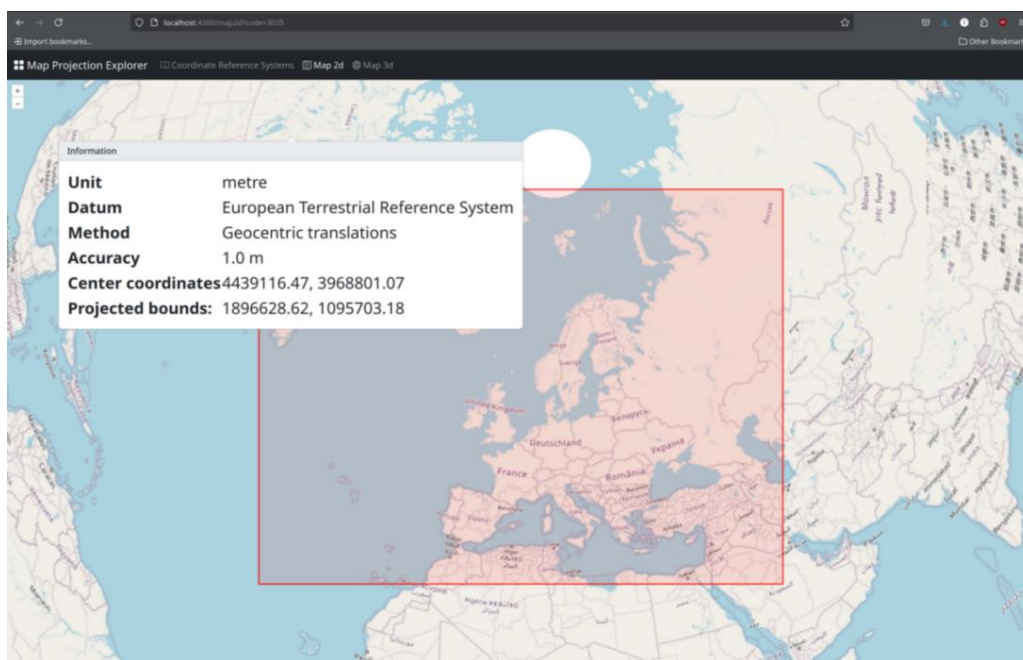


Рис. 3.2 Приклад проекції в двовимірному просторі

Рисунок 3.1 ілюструє результат застосування двовимірних географічних проекцій на плоску поверхню Землі. На ньому можна побачити різні типи проекцій, де кожна з них має свої властивості та рівень спотворення при трансформації географічних координат. Кожна з проекцій має різні ділянки на карті, де спотворення форм, площ або відстаней є мінімальними або максимально вираженими, що допомагає користувачам вибрати оптимальну проекцію залежно від конкретних завдань.

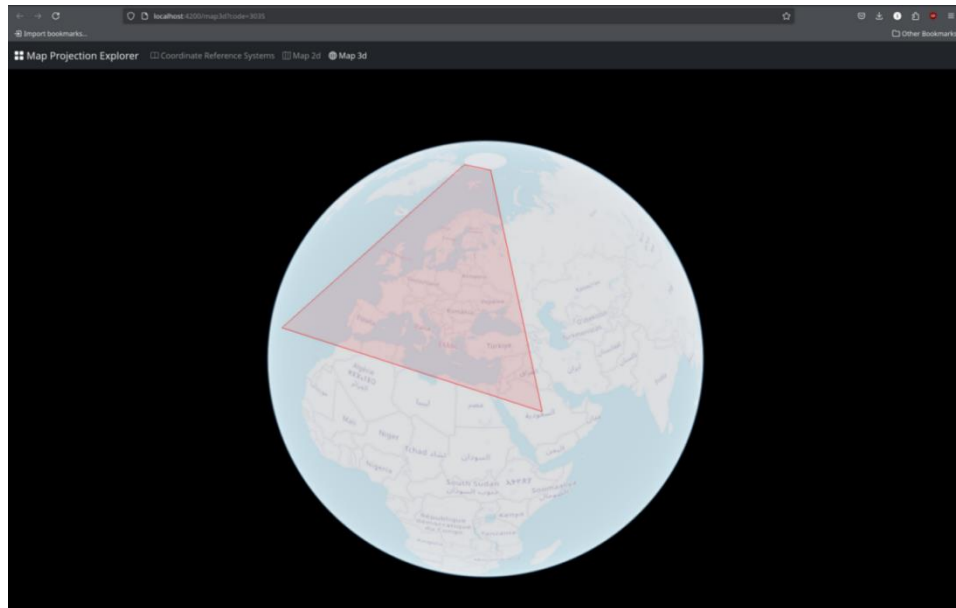


Рис. 3.3 Приклад проєкцій в трьовимірному просторі

Рисунок 3.2 демонструє тривимірний варіант відображення географічних проєкцій. Тут показано, як географічні проєкції застосовуються до моделі Землі, створеної за допомогою 3D-графіки. Користувач може побачити, як різні проєкції впливають на форму Землі, її континенти та океани в тривимірному просторі. Це дозволяє глибше зрозуміти, як кожна проєкція коригує відображення Землі на площині та як вона змінює сприйняття простору в контексті тривимірної моделі.

 A screenshot of the 'Map Projection Explorer' application showing a search results page for 'europe'. The page displays a table of map projections with columns for Name, Code, and Action. The table lists various projections such as ETRS89-extended / LCC Europe, WGS 84 / North Pole LAEA Europe, and others. Each row has a 'Load more' button next to it.

Name	Code	Action
ETRS89-extended / LCC Europe	3034	Load more
ETRS89-extended / LAEA Europe	3035	Load more
WGS 84 / North Pole LAEA Europe	3575	Load more
PTRA08 / LCC Europe	5632	Load more
PTRA08 / LAEA Europe	5633	Load more
REGCAN95 / LCC Europe	5634	Load more
REGCAN95 / LAEA Europe	5635	Load more
TUREF / LAEA Europe	5636	Load more
TUREF / LCC Europe	5637	Load more
ISN2004 / LAEA Europe	5638	Load more
ISN2004 / LCC Europe	5639	Load more
ISN2016 / LAEA Europe	9039	Load more
ISN2016 / LCC Europe	9040	Load more
WGS 84 / GLANCE Europe	10596	Load more
WGS 84 / Equi7 Europe	27704	Load more

Рис. 3.4 Приклад списку проєкцій

На рисунку 3.4 представлено приклад інтерфейсу, що демонструє список доступних географічних проекцій. Цей інтерфейс має кілька функцій, що забезпечують зручність роботи з великою кількістю проекцій.

1. Фільтрація по назві або коду: користувач може здійснювати фільтрацію списку проекцій, вводячи текст у відповідні поля для пошуку за назвою або за кодом проекції. Це дозволяє швидко знаходити конкретні проекції серед великої кількості доступних варіантів.

2. Посторінкове завантаження списку проекцій: для зручності навігації і зменшення навантаження на систему, список проекцій завантажується посторінково. Це дозволяє користувачеві не завантажувати одразу всю інформацію, що може бути корисним при роботі з великою кількістю проекцій.

3. Вибір декількох проекцій для дослідження: інтерфейс підтримує можливість вибору кількох проекцій одночасно. Користувач може вибрати кілька проекцій для подальшого аналізу чи дослідження, що забезпечує гнучкість у роботі з даними.

Завдяки таким функціям, інтерфейс дозволяє ефективно працювати з великими обсягами географічних даних і зручніше проводити аналіз та дослідження різних проекцій.

3.4 Порівняння точності оптимальних та гібридних проекцій

Результати точності геометричних операцій з використанням методів Polygon Buffer та Polygon Distance

1. Polygon Buffer (Буферизація полігонів).

Метод буферизації полігонів створює новий полігон, що представляє область на певній відстані від вихідного полігону. У випадку використання географічних проекцій, точність результатів може змінюватися залежно від обраної проекції. Розглянемо три варіанти проекцій: звичайну, оптимальну (за самотійно розробленим алгоритмом), та гібридну (також самотійно розроблену).

Звичайна проекція.

У випадку звичайної проекції, точність операції буферизації часто знижується через спотворення, особливо в районах, які далеко від екватора. Відстані на полюсах можуть бути значно спотворені, що призводить до помилок у визначенні точної форми буфера. Особливо це відчувається при великих відстанях або коли потрібно створити великі буфери навколо полігонів.

Оптимальна проекція.

Оптимальна проекція, алгоритм якої був розроблений самостійно, дозволяє зменшити спотворення, адаптуючи проекцію до конкретних географічних умов. У порівнянні зі звичайною проекцією, вона дає більш точні результати при виконанні буферизації, оскільки враховує особливості території, для якої проводиться операція. Для таких полігонів, які займають значну частину території, оптимальна проекція дозволяє зменшити похибки, зберігаючи точність форми буфера на більшій частині простору.

Гібридна проекція.

Гібридна проекція, що була розроблена самостійно, комбінує різні типи проекцій в залежності від частини світу, де проводяться операції. У районах з мінімальними спотвореннями використовується одна частина проекції, тоді як в інших – застосовується інша, що дозволяє досягти максимальної точності. Гібридна проекція значно покращує результати буферизації порівняно з іншими методами, оскільки враховує локальні особливості кожної території, і, як правило, дає найменші похибки в різних частинах світу.

2. Polygon Distance (Відстань між полігонами).

Метод розрахунку відстані між двома полігонами визначає мінімальну відстань між їхніми обрисами. Як і у випадку з буферизацією, точність цього методу значно залежить від вибору проекції.

Звичайна проекція.

Використання звичайної проекції при обчисленні відстані між полігонами може призвести до значних похибок, особливо якщо полігони знаходяться далеко один від одного або розташовані в полярних регіонах. Спотворення геометрії при

великій відстані можуть викривити реальну відстань між полігонами, що призведе до менш точних результатів.

Оптимальна проекція.

При використанні оптимальної проекції точність обчислення відстані між полігонами значно покращується. Завдяки адаптованому методу проекції для конкретної території, спотворення зменшуються, і результат наближається до реальної відстані між полігонами. Особливо помітна перевага оптимальної проекції в середніх та великих масштабах.

Гібридна проекція.

У випадку гібридної проекції, точність результатів при обчисленні відстані між полігонами є найвищою. Оскільки для кожної частини території використовуються найбільш підходящі проекції, спотворення відстаней зводяться до мінімуму. Це дозволяє отримати найбільш точні результати в глобальному контексті, зокрема для великих територій або при необхідності точних вимірів між полігонами, що знаходяться на різних континентах.

Підсумки результатів точності:

1. Звичайна проекція: відзначається значними спотвореннями при великих відстанях, особливо на полюсах, що призводить до менш точних результатів при геометричних операціях.
2. Оптимальна проекція: покращує точність в межах конкретної території, зменшуючи спотворення при буферизації та розрахунку відстаней, але залишається менш універсальною в порівнянні з гібридною.
3. Гібридна проекція: забезпечує найкращі результати завдяки використанню адаптованих методів для різних частин світу, даючи високу точність при будь-яких геометричних операціях.

Ці результати показують важливість вибору правильної проекції для досягнення точних результатів у геометричних операціях, таких як буферизація та визначення відстаней між полігонами.

Рисунок 3.4 демонструє результати точності геометричних операцій, виконаних на різних типах даних та за різних умов. У ньому представлено

порівняння результатів виконання операцій, таких як обчислення буферних зон, визначення відстаней між геометричними об'єктами, що враховують кривизну земної поверхні.

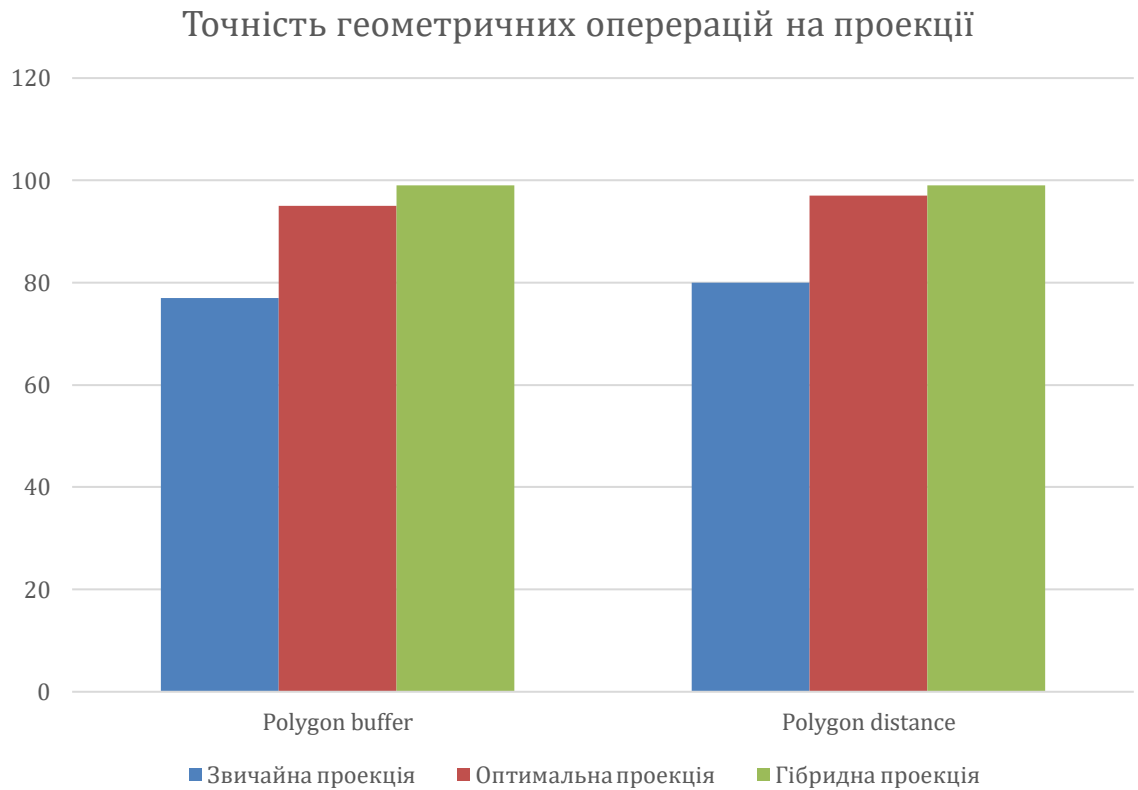


Рис. 3.4 Результати точності геометричних операцій

3.5 Програмні засоби реалізації

Для розробки інформаційної системи для дослідження географічних проекцій на поверхні Землі в двовимірному та тривимірному просторах, було застосовано сучасний набір технологій, який забезпечує високу продуктивність, масштабованість та зручність використання.

Git використовується для управління версіями та організації командної роботи. Ця система дозволяє ефективно координувати зміни в коді, забезпечувати прозорість розробки та підтримувати історію внесених правок. Завдяки можливості створення гілок та можливості інтеграції з CI/CD, Git допомагає автоматизувати перевірки, тестування та розгортання системи.

У фронтенді основною мовою є TypeScript. Цей інструмент, завдяки строгій типізації, дозволяє зменшити кількість помилок під час розробки, а також полегшує масштабування та рефакторинг коду. Його інтеграція з Angular робить розробку багатофункціонального інтерфейсу зручною та ефективною.

Angular як фреймворк забезпечує модульну структуру додатка, дозволяючи створювати динамічні компоненти та інтерактивний інтерфейс користувача. Завдяки підтримці RxJS обробка асинхронних даних, таких як запити до API чи потокові дані, виконується плавно й оптимально.

OpenLayers відповідає за роботу з картографічними даними. Ця бібліотека дозволяє відображати географічні карти, завантажувати шари даних у різних форматах, а також виконувати просторові операції, такі як побудова буферів, аналіз перетинів та обчислення геометричних характеристик об'єктів. OpenLayers також забезпечує інтеграцію з тривимірними даними, що дозволяє відображати моделі Землі в двовимірному та тривимірному просторах і порівнювати спотворення різних географічних проекцій.

На бекенді використовується мова програмування Go, яка вирізняється своєю високою продуктивністю та простотою у використанні. Завдяки вбудованій підтримці багатопотоковості мова забезпечує швидке виконання задач, таких як обробка запитів до серверу чи виконання складної бізнес-логіки. Go також ідеально підходить для побудови високонавантажених API та обробки великих обсягів географічних даних.

У роботі з базою даних застосовуються спеціалізовані бібліотеки для взаємодії з PostgreSQL, що забезпечує ефективність роботи з геопросторовими даними.

У ролі бази даних використовується PostgreSQL з розширенням PostGIS, яке додає потужні можливості для роботи з геопросторовими даними. PostgreSQL зберігає геометричні та географічні об'єкти, такі як точки, лінії та багатокутники, дозволяючи виконувати складні запити. Наприклад, обчислення відстаней між об'єктами, визначення їх перетинів, створення буферів навколо геометрій чи аналіз топологічних відносин. Постійно використовуються просторові індекси типу, що

значно підвищує швидкість виконання географічних запитів навіть на великих обсягах даних.

Для управління схемою бази даних використовується Liquibase. Цей інструмент дозволяє автоматизувати процес міграцій, забезпечуючи синхронізацію структури бази даних між різними середовищами розробки, тестування та продакшену. Liquibase також надає можливість документувати зміни в базі, що полегшує підтримку системи.

Ця система забезпечує зручність для користувачів завдяки сучасному інтерфейсу, який дозволяє інтерактивно досліджувати властивості географічних проекцій. Користувачі можуть не лише переглядати карти, а й аналізувати просторові дані, порівнювати спотворення проекцій, виконувати геометричні операції та навіть взаємодіяти з тривимірною моделлю Землі.

3.6 Розгортання інформаційної системи

Для розгортання інформаційної системи для дослідження географічних проекцій на поверхні Землі в двовимірному та тривимірному просторах, необхідно налаштувати інфраструктуру на базі Proxmox, встановити Kubernetes для керування контейнерами, підняти базу даних PostgreSQL для зберігання інформації про проекції, а також запустити контейнери для бекенду та фронтенду.

Proxmox — це віртуалізаційна платформа з відкритим вихідним кодом, яка дозволяє встановлювати та управляти віртуальними машинами та контейнерами на фізичних серверах. Proxmox базується на гіпервізорі KVM, що дозволяє забезпечити ефективну віртуалізацію різних типів ресурсів, таких як обчислювальна потужність, пам'ять та мережеві ресурси.

Proxmox має простий та зрозумілий інтерфейс користувача, що дозволяє швидко налаштувати та управляти віртуальними машинами та контейнерами. Крім того, Proxmox підтримує клонування та резервне копіювання віртуальних машин та контейнерів, що забезпечує зручне управління та зменшує час на відновлення в разі непередбачуваних ситуацій.

Proxmox також підтримує віддалене управління та моніторинг за допомогою зручного API та різноманітних інтеграцій з іншими інструментами та платформами. Загалом, Proxmox є потужним та зручним інструментом для віртуалізації та управління інфраструктурою, який дозволяє ефективно використовувати ресурси.

Для того щоб розробити універсальний спосіб запуску системи було вибрано Docker для контейнеризації та Kubernetes для оркестрування.

Docker — це платформа для створення, розпакування та запуску контейнерів. Контейнери є ізольованими віртуальними середовищами, які містять застосунок та всі залежності, необхідні для його роботи. Docker забезпечує стандартизацію процесу розгортання додатків, що дозволяє ефективно використовувати ресурси та забезпечує більш просте та швидке розгортання додатків.

Kubernetes — це система управління контейнерами, яка дозволяє автоматизувати розгортання, масштабування та керування додатками, які запуснені в контейнерах. Kubernetes забезпечує автоматичне масштабування додатків, балансування навантаження та автоматичне відновлення після відмови. Використання Kubernetes дозволяє забезпечити високу доступність та надійність додатків.

Kubernetes розгортався за допомогою Kubespray, що є потужним інструментом для автоматизації процесу розгортання кластерів Kubernetes на різних інфраструктурних середовищах, таких як фізичні сервери, віртуальні машини або хмарні платформи. Kubespray є відкритим проєктом, який використовує Ansible для автоматизації налаштування та розгортання Kubernetes, що дозволяє швидко і ефективно впроваджувати кластер. Kubespray підтримує різні операційні систем. Він також надає можливість налаштування таких функцій, як високодоступні кластери, оновлення, забезпечення безпеки, а також інтеграцію з інструментами для моніторингу та логування.

Разом Docker та Kubernetes дозволяють легко та ефективно розгорнути, масштабувати та керувати додатками в хмарних середовищах. Використання цих технологій дозволяє забезпечити високу продуктивність та ефективне використання ресурсів сервера. Завдяки можливості автоматичного

масштабування та управління життєвим циклом контейнерів, ці технології дозволяють забезпечувати оптимальну роботу навіть при значних навантаженнях.

Загалом, ця інфраструктура забезпечує надійне і ефективне середовище для дослідження і аналізу географічних проекцій, яке дозволяє глибше зрозуміти, як різні проекції впливають на точність і спотворення даних.

Використання сучасних технологій на серверній і клієнтській частинах дає змогу забезпечити високу продуктивність і масштабованість системи, а також інтерактивність при роботі з географічними даними.

Ця інфраструктура створена для забезпечення надійного і високопродуктивного середовища, яке сприяє глибокому дослідженню географічних проекцій та їх впливу на точність і спотворення даних. Використання сучасних технологій, таких як Angular, OpenLayers, PostgreSQL і Go, дозволяє побудувати ефективну та масштабовану систему, яка може адаптуватися до зростання обсягів даних і складності завдань. Задіяні інструменти підтримують інтерактивність і візуалізацію даних у реальному часі, що є особливо важливим для аналізу географічних моделей.

Користувач отримує змогу працювати з різноманітними проекціями, порівнювати їх, аналізувати вплив на відображення поверхні Землі в двовимірному та тривимірному просторах. Система також сприяє розширенню знань у сфері картографії, надаючи науковцям і розробникам інструменти для дослідження різних аспектів картографічних методів. Завдяки інтеграції технологій для керування базами даних та обробки складних запитів забезпечується швидкий доступ до великих обсягів географічної інформації, що робить систему корисною не лише для теоретичних досліджень, але й для практичного застосування.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи проведено детальний аналіз предметної галузі та існуючих аналогічних систем, методів та алгоритмів, що застосовуються для роботи з географічними проекціями. Це дало змогу визначити основні підходи до дослідження географічних проекцій, виявити їх переваги й недоліки, а також сформулювати ключові вимоги до розробленої системи.

Розроблена інформаційна система для дослідження географічних проекцій на поверхні Землі в двовимірному та тривимірному просторах. Система включає інструменти для візуалізації, аналізу та порівняння різних видів проекцій, що дозволяє оцінювати їх відповідність до визначених географічних і математичних вимог.

Розроблено алгоритм пошуку оптимальних проекцій. Цей алгоритм базується на комплексному аналізі параметрів, таких як спотворення площі, кутів, відстаней, що дає змогу автоматизувати вибір найбільш ефективних проекцій для конкретних завдань.

Розроблено математичну модель створення гібридної географічної проекції. Ця модель дозволяє поєднати переваги кількох стандартних проекцій для досягнення більш рівномірного розподілу спотворень у заданих регіонах.

У рамках роботи було проведено порівняння точності між звичайними, оптимальними та гібридними проекціями. Результати аналізу показали, що використання гібридних проекцій забезпечує рівномірніший розподіл спотворень у критичних зонах, особливо в умовах великих територій або складних геометричних конфігурацій. При цьому оптимальні проекції виявилися ефективними для задач з локальним масштабом, де необхідна мінімізація одного типу спотворень.

Таке порівняння підтвердило доцільність використання запропонованої гібридної моделі для підвищення точності візуалізації та аналізу просторових даних.

Результати дослідження апробовано та опубліковано у наступних тезах доповіді на конференціях:

1. Зайцев І.С., Жебка В.В. Розробка інформаційної технології для дослідження географічних проекцій на поверхні Землі в двовимірному та тривимірному просторах (типи проекції) // Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу – Київ: ДУІКТ, 2024. С. 182-184.

2. Зайцев І.С., Жебка В.В. Розробка інформаційної технології для дослідження географічних проекцій на поверхні Землі в двовимірному та тривимірному просторах (алгоритми проекцій) // Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу – Київ: ДУІКТ, 2024. С. 186-189.

ПЕРЕЛІК ПОСИЛАНЬ

1. Map projections. [Електронний ресурс]. – Режим доступу: <https://www.axismaps.com/guide/map-projections> (дата звернення: 20.11.2024)
2. Map Projections Classification. [Електронний ресурс]. – Режим доступу: <https://www.mdpi.com/2673-7086/2/2/19> (дата звернення: 20.11.2024)
3. Fritz Kessler. Working with Map Projections: A Guide to their Selection. CRC Press, 2019. 299 с.
4. Dr Michael J. de Smith. Geospatial Analysis 7th Edition, 2024. 592 с.
5. Scale and Projections. [Електронний ресурс]. – Режим доступу: <https://open.lib.umn.edu/mapping/chapter/3-scale-and-projections/> (дата звернення: 20.11.2024)
6. Geospatial Data Structures. [Електронний ресурс]. – Режим доступу: <https://opensourcegisdata.com/geospatial-data-structures-advantages-and-disadvantages.html> (дата звернення: 20.11.2024)
7. On the Definition of Standard Parallels in Map Projections. [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/376284736_On_the_Definition_of_Standard_Parallels_in_Map_Projections (дата звернення: 20.11.2024)
8. Paul A. Longley. Geographic Information Science and Systems, 4th Edition. Wiley, 2015. 496 с.
9. A Proposed Hybrid Spatial Data Structure based on KD Tree and Quad Tree. [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/339433736_A_Proposed_Hybrid_Spatial_Data_Structure_based_on_KD_Tree_and_Quad_Tree (дата звернення: 20.11.2024)
10. Geospatial Data Management Research: Progress and Future Directions. [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/339047709_Geospatial_Data_Management_Research_Progress_and_Future_Directions (дата звернення: 20.11.2024)

11. John N. Hatzopoulos. Geospatial Computational Methods: Algorithms of Computational Methods for Geomatics. Universal Publishers, 2024, 556 с.
12. Map projections. [Электронный ресурс]. – Режим доступа: <https://www.axismaps.com/guide/map-projections> (дата звернения: 20.11.2024)
13. Map Projections Classification. [Электронный ресурс]. – Режим доступа: <https://www.mdpi.com/2673-7086/2/2/19> (дата звернения: 20.11.2024)
14. Fritz Kessler. Working with Map Projections: A Guide to their Selection. CRC Press, 2019. 299 с.
15. Dr Michael J. de Smith. Geospatial Analysis 7th Edition, 2024. 592 с.
16. Scale and Projections. [Электронный ресурс]. – Режим доступа: <https://open.lib.umn.edu/mapping/chapter/3-scale-and-projections/> (дата звернения: 20.11.2024)
17. Geospatial Data Structures. [Электронный ресурс]. – Режим доступа: <https://opensourcegisdata.com/geospatial-data-structures-advantages-and-disadvantages.html> (дата звернения: 20.11.2024)
18. On the Definition of Standard Parallels in Map Projections. [Электронный ресурс]. – Режим доступа: https://www.researchgate.net/publication/376284736_On_the_Definition_of_Standard_Parallels_in_Map_Projections (дата звернения: 20.11.2024)
19. Paul A. Longley. Geographic Information Science and Systems, 4th Edition. Wiley, 2015. 496 с.
20. A Proposed Hybrid Spatial Data Structure based on KD Tree and Quad Tree. [Электронный ресурс]. – Режим доступа: https://www.researchgate.net/publication/339433736_A_Proposed_Hybrid_Spatial_Data_Structure_based_on_KD_Tree_and_Quad_Tree (дата звернения: 20.11.2024)
21. Geospatial Data Management Research: Progress and Future Directions. [Электронный ресурс]. – Режим доступа: https://www.researchgate.net/publication/339047709_Geospatial_Data_Management_Research_Progress_and_Future_Directions (дата звернения: 20.11.2024)

22. John N. Hatzopoulos. Geospatial Computational Methods: Algorithms of Computational Methods for Geomatics. Universal Publishers, 2024, 556 с.
23. Map projections. [Электронный ресурс]. – Режим доступа: <https://www.axismaps.com/guide/map-projections> (дата звернения: 20.11.2024)
24. Map Projections Classification. [Электронный ресурс]. – Режим доступа: <https://www.mdpi.com/2673-7086/2/2/19> (дата звернения: 20.11.2024)
25. Fritz Kessler. Working with Map Projections: A Guide to their Selection. CRC Press, 2019. 299 с.
26. Dr Michael J. de Smith. Geospatial Analysis 7th Edition, 2024. 592 с.
27. Scale and Projections. [Электронный ресурс]. – Режим доступа: <https://open.lib.umn.edu/mapping/chapter/3-scale-and-projections/> (дата звернения: 20.11.2024)
28. Geospatial Data Structures. [Электронный ресурс]. – Режим доступа: <https://opensourcegisdata.com/geospatial-data-structures-advantages-and-disadvantages.html> (дата звернения: 20.11.2024)
29. On the Definition of Standard Parallels in Map Projections. [Электронный ресурс]. – Режим доступа: https://www.researchgate.net/publication/376284736_On_the_Definition_of_Standard_Parallels_in_Map_Projections (дата звернения: 20.11.2024)
30. Paul A. Longley. Geographic Information Science and Systems, 4th Edition. Wiley, 2015. 496 с.

ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ



КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Магістерська робота

«ІНФОРМАЦІЙНА СИСТЕМА ДОСЛІДЖЕННЯ ГЕОГРАФІЧНИХ ПРОЕКЦІЙ НА ПОВЕРХНІ ЗЕМЛІ В ДВОВИМІРНОМУ ТА ТРИВИМІРНОМУ ПРОСТОРАХ»

Виконав: студент групи ПДМ-62 Зайцев Іван Сергійович

Керівник: доктор філософії, доцент кафедри ІІЗ Гребенюк Віктор
Вікторович

Київ - 2025

МЕТА, ОБ'ЄКТА ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: покращення точності дослідження географічних проекцій за рахунок відображення географічних проекцій на поверхні Землі в двовимірному та тривимірному просторах.

Об'єкт дослідження: процес дослідження географічних проекцій на поверхні Землі в двовимірному та тривимірному просторах.

Предмет дослідження: засоби та методи дослідження географічних проекцій у двовимірному та тривимірному просторах.

АКТУАЛЬНІСТЬ РОБОТИ

Недоліки наявних алгоритмів:

1. Недостатня адаптивність алгоритмів для роботи з географічними проекціями.
2. Недостатня ефективність алгоритмів для роботи з географічними проекціями.

Недоліки наявних методів:

1. Polygon buffer - невідповідність масштабів та розмірів якщо полігони знаходяться в різних проекціях.
2. Polygon distance - неточність розрахунку якщо полігони знаходяться в різних проекціях.

Вирішення:

1. Алгоритм динамічного пошуку оптимальних проекцій в залежності від координат заданих полігонів.
2. Алгоритм створення гібридних оптимальних проекцій.

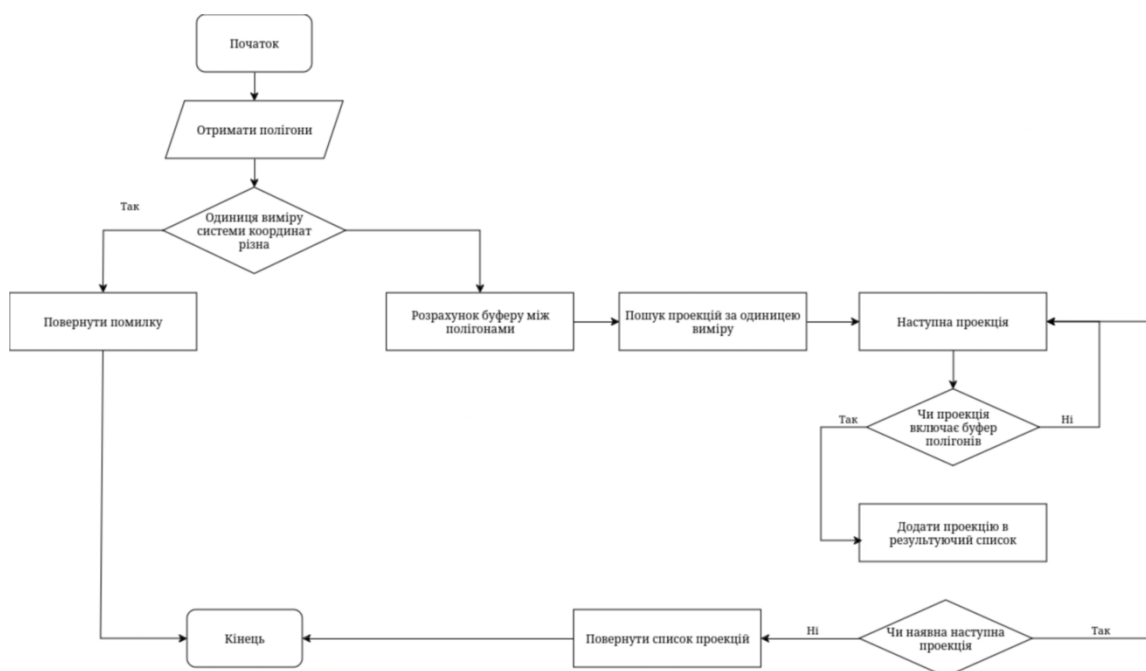
3

АКТУАЛЬНІСТЬ РОБОТИ

	EPSG	Розроблена система
Алгоритм пошуку оптимальних проекцій	Мануальний вибір проекцій	Пошук проекцій на основі координат полігонів
Алгоритм створення гібридних проекцій	Обмежений набір проекцій	Створення гібридних використовуючи математичну модель
Алгоритми розрахунку проекцій	Не використовуються	Використання геопросторових структур даних

4

АЛГОРИТМ ПОШУКУ ОПИТМАЛЬНИХ ПРОЕКЦІЙ



5

МАТЕМАТИЧНА МОДЕЛЬ СТВОРЕННЯ ГІБРИДНОЇ ПРОЕКЦІЇ

longitude - координата довготи
latitude - координата широти

Один сегмент UTM зони 6°
Всього UTM зон 60 (360 ÷ 6 = 60)

$$zone = \min\left(\left\lfloor \frac{longitude + 180}{6} \right\rfloor + 1, 60\right)$$

$$centralMeridian = -183 + zone * 6$$

$$falseNorthing = \begin{cases} 0, & latitude \geq 0 \\ 10000000, & latitude < 0 \end{cases}$$

Модель проекції в форматі PROJ

```

PROJCS["WGS 84 / Auto UTM",
  GEOGCS["WGS 84",
    DATUM["WGS_1984",
      SPHEROID["WGS_1984", 6378137, 298.257223563]],
    PRIMEM["Greenwich", 0],
    UNIT["Decimal_Degree", 0.0174532925199433]],
  PROJECTION["Transverse_Mercator"],
  PARAMETER["Latitude_of_Origin", 0],
  PARAMETER["Central_Meridian", ${centralMeridian}],
  PARAMETER["False_Easting", 500000],
  PARAMETER["False_Northing", ${falseNorthing}],
  PARAMETER["Scale_Factor", 0.9996],
  UNIT["Meter", 1]]
  
```

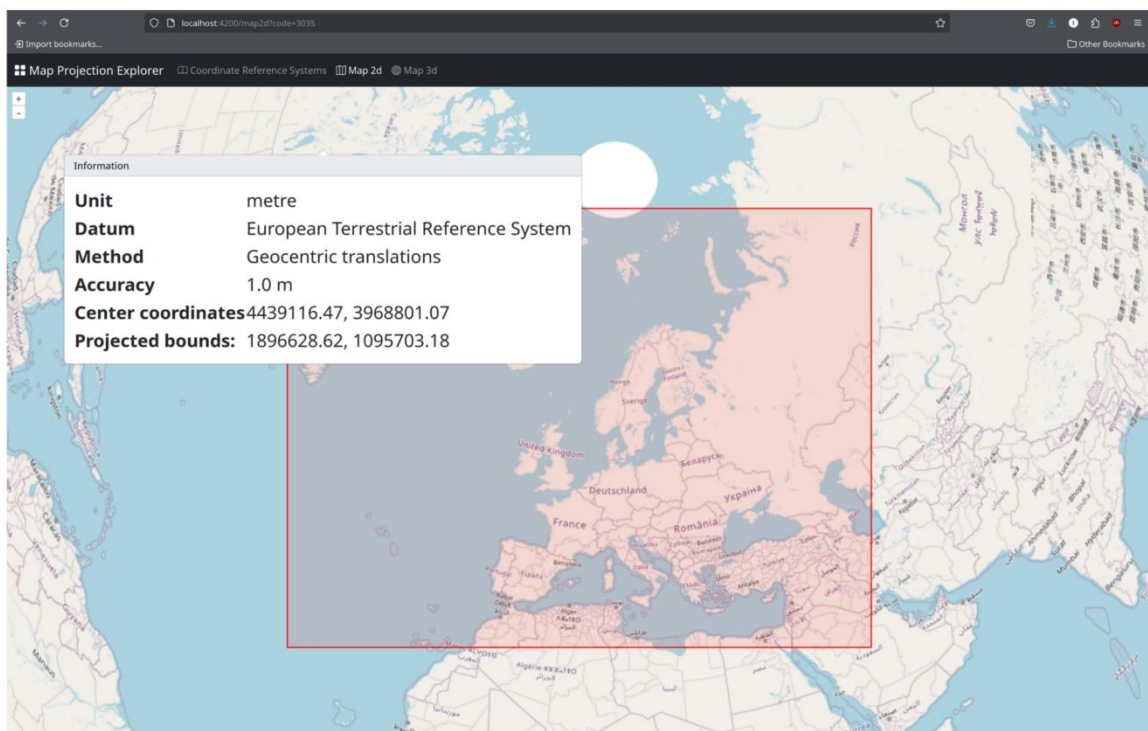
6

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



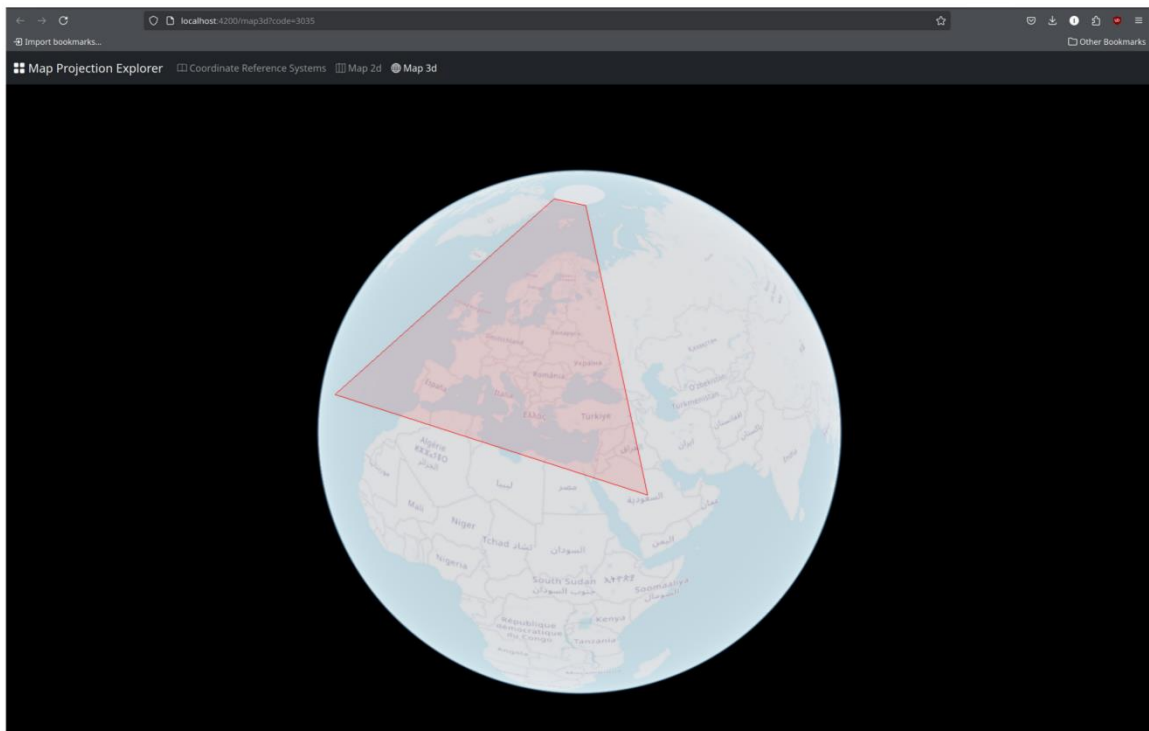
7

ПРАКТИЧНИЙ РЕЗУЛЬТАТ



8

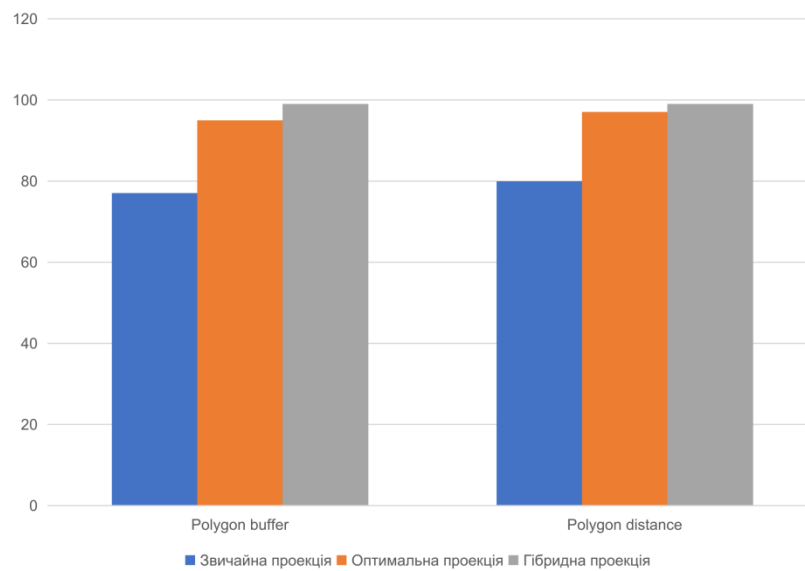
ПРАКТИЧНИЙ РЕЗУЛЬТАТ



9

ПОРІВНЯННЯ ЕФЕКТИВНОСТІ

Точність геометричних операцій на проекції



10

ВИСНОВКИ

1. Проведено детальний аналіз предметної галузі та існуючих аналогічних систем, методів та алгоритмів.
2. Розроблена інформаційна система для дослідження географічних проекцій на поверхні Землі в двовимірному та тривимірному просторах.
3. Розроблено алгоритм пошуку оптимальних проекцій.
4. Розроблено математичну модель створення гібридної географічної проекції.
5. Проведено порівняння точності між звичайними, оптимальними та гібридними проекціями.

11

ПУБЛІКАЦІЇ ТА АПРОБАЦІЯ РОБОТИ

Тези доповідей:

1. Зайцев І. С. Розробка інформаційної технології для дослідження географічних проекцій на поверхні Землі в двовимірному та тривимірному просторах (типи проекції) // Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу – Київ: ДУІКТ, 2024. С. 182-184.
2. Зайцев І. С. Розробка інформаційної технології для дослідження географічних проекцій на поверхні Землі в двовимірному та тривимірному просторах (алгоритми проекцій) // Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу – Київ: ДУІКТ, 2024. С. 186-189.

12

ДОДАТОК Б. ЛІСТИНГИ ПРОГРАМНИХ МОДУЛІВ

```

backend/Makefile
migrate:
    docker run --net=host --rm \
        --volume
    ${CURDIR}/config/sql/liquibase/changelog
    liquibase/liquibase:latest update \
        --changelog-file=changelog/changelog.xml --
    classpath=/liquibase/changelog --
    url=${$LIQUIBASE_DATABASE_URI

backend/Dockerfile
FROM golang:1.22.4-alpine3.20

WORKDIR /app

COPY . /app

RUN go build -o /application cmd/main.go

EXPOSE 8080

CMD ["/application"]

backend/config.yaml
server:
    port: 8080

backend/cmd/main.go
package main

import (
    _ "github.com/lib/pq"
    "github.com/rs/cors"
    "github.com/spf13/viper"
    "log"
    "map-projection-explorer-
backend/internal/repository"
    http_server "map-projection-explorer-
backend/internal/server/http"
    "map-projection-explorer-
backend/internal/service"
    "net/http"
)

var (
    epsgRepository
    repository.EpsgExtentRepository
    srsRepository repository.SrsRepository
    crsService service.CrsService
)

func main() {
    setupConfig()
    setupRepositories()
    setupServices()
    setupHttpServer()

    viper.SetConfigName("config")
    viper.SetConfigType("yaml")
    viper.AddConfigPath(".")
    viper.SetConfigFile("")
    viper.ReadInConfig()
    if err != nil {
        log.Fatalf("Error reading config file, %s", err)
    }

    viper.BindEnv("database.uri",
        "DATABASE_URI")

    func setupRepositories() {
        db, err :=
        repository.NewDatabase(viper.GetString("database.u
ri"))
        if err != nil {
            log.Fatalf("Failed to connect to database: %s",
err)
        }

        epsgRepository =
        repository.NewEpsgExtentRepository(db)
        srsRepository = repository.NewSrsRepository(db)
    }

    func setupServices() {
        crsService =
        service.NewCrsService(epsgRepository,
srsRepository)
    }

    func setupHttpServer() {
        mux := http.NewServeMux()
        handler := cors.Default().Handler(mux)

        server := http_server.NewServer(crsService)
        http_server.RegisterServerHandlers(mux, server)

        log.Println("Starting http server")
    }
}

```

```

    err :=
http.ListenAndServe(":"+viper.GetString("server.port
"), handler)
    if err != nil {
        log.Fatalf("Failed to start http server, %s", err)
    }
}

```

backend/config/sql/changelog.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<databaseChangeLog

```

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"

```

```

xmlns="http://www.liquibase.org/xml/ns/dbchangel
og"

```

```

xsi:schemaLocation="http://www.liquibase.org/xml/n
s/dbchangelog

```

```

http://www.liquibase.org/xml/ns/dbchangelog/dbchan
gelog-4.1.xsd">

```

```

    <include file="1.0.0__create_tables.sql"/>
    <include file="1.1.0__insert_data.sql"/>
    <include file="1.2.0__alter_keys.sql"/>

```

```

</databaseChangeLog>

```

backend/internal/domain/dto/crs.go

```

package dto

```

```

type CrsRecordDefinitionType string

```

```

var (
    CrsRecordDefinitionTypeProj4
    CrsRecordDefinitionType = "proj4"
)

```

```

type Page[T any] struct {
    Content    []*T `json:"content,omitempty"`
    NextPageCursor *int
`json:"nextPageCursor,omitempty"`
}

```

```

type CrsRecord struct {
    Name string
`json:"name,omitempty"`
    Code int `json:"code,omitempty"`
    Definition *CrsRecordDefinition
`json:"definition,omitempty"`
    Extent *CrsRecordExtent
`json:"extent,omitempty"`

```

```

    ExtentCenter *CrsRecordExtentCenter
`json:"extentCenter,omitempty"`
}

```

```

type CrsRecordDefinition struct {
    Type CrsRecordDefinitionType
`json:"type,omitempty"`
    Value string `json:"value,omitempty"`
}

```

```

type CrsRecordExtent struct {
    WestLongitude *float64
`json:"westLongitude,omitempty"`
    SouthLatitude *float64
`json:"southLatitude,omitempty"`
    EastLongitude *float64
`json:"eastLongitude,omitempty"`
    NorthLatitude *float64
`json:"northLatitude,omitempty"`
}

```

```

type CrsRecordExtentCenter struct {
    Longitude *float64 `json:"longitude,omitempty"`
    Latitude *float64 `json:"latitude,omitempty"`
}

```

backend/internal/domain/dto/error.go

```

package dto

```

```

var (
    ErrorCodeNotFound ErrorCode =
    ErrorCode{Name: "NOT_FOUND", Value: 404}
    ErrorCodeInvalidRequest ErrorCode =
    ErrorCode{Name: "INVALID_REQUEST", Value:
    400}
    ErrorCodeInternalServerError ErrorCode =
    ErrorCode{Name: "INTERNAL_SERVER_ERROR",
    Value: 500}
)

```

```

type ErrorCode struct {
    Name string
    Value int
}

```

```

type ServiceError struct {
    Code ErrorCode `json:"code,omitempty"`
    Message string `json:"message,omitempty"`
}

```

```

func (e ServiceError) Error() string {
    return e.Message
}

```

backend/internal/domain/model/model.go

```
package model
```

```
type EpsgExtentRecord struct {
    Name      string
    Code      int
    BboxSouthBoundLat float64
    BboxWestBoundLon float64
    BboxNorthBoundLat float64
    BboxEastBoundLon float64
}
```

```
type SrsRecord struct {
    Proj4text string
}
```

```
backend/internal/repository/api.go
```

```
package repository
```

```
import (
    "database/sql"
    "map-projection-explorer-
backend/internal/domain/model"
)

func NewDatabase(uri string) (*sql.DB, error) {
    db, err := sql.Open("postgres", uri)
    if err != nil {
        return nil, err
    }

    err = db.Ping()
    if err != nil {
        return nil, err
    }
    return db, nil
}
```

```
type EpsgExtentRepository interface {
    FindAllAfterCode(search string, afterCode *int,
size int) ([]*model.EpsgExtentRecord, error)
    FindByCode(code int)
(*model.EpsgExtentRecord, error)
}
```

```
type SrsRepository interface {
    FindByCode(code int) (*model.SrsRecord, error)
}
```

```
backend/internal/repository/epsg_repository.go
```

```
package repository
```

```
import (
    "database/sql"
    "map-projection-explorer-
backend/internal/domain/model"
```

```
)
```

```
type epsgExtentRepository struct {
    db *sql.DB
}

func NewEpsgExtentRepository(db *sql.DB)
EpsgExtentRepository {
    return &epsgExtentRepository{db: db}
}
```

```
func (e *epsgExtentRepository) FindAllAfterCode(
    search string, afterCode *int, size int)
([]*model.EpsgExtentRecord, error) {
```

```
    query := `
        SELECT
            ec.coord_ref_sys_name,
            ec.coord_ref_sys_code
        FROM epsg_coordinatereferencesystem ec
        WHERE
            (LOWER(ec.coord_ref_sys_name) LIKE
LOWER(CONCAT('%%', $1::varchar, '%%')) OR
            LOWER(ec.coord_ref_sys_code::varchar)
LIKE LOWER(CONCAT('%%', $1::varchar, '%%')))
        AND
            ($2::integer IS NULL OR
ec.coord_ref_sys_code > $2)
        ORDER BY ec.coord_ref_sys_code LIMIT $3`
```

```
    rows, err := e.db.Query(query, search, afterCode,
size)
    if err != nil {
        return nil, err
    }
    defer rows.Close()
```

```
    var records []*model.EpsgExtentRecord

    for rows.Next() {
        record := &model.EpsgExtentRecord{}

        err := rows.Scan(&record.Name,
&record.Code)
        if err != nil {
            return nil, err
        }
    }
```

```
    records = append(records, record)
}
return records, nil
}
```

```
func (e *epsgExtentRepository) FindByCode(code int)
(*model.EpsgExtentRecord, error) {
    query := `
        SELECT
```

```

        ec.coord_ref_sys_name,
        ec.coord_ref_sys_code,
        ee.bbox_south_bound_lat,
        ee.bbox_west_bound_lon,
        ee.bbox_north_bound_lat,
        ee.bbox_east_bound_lon
    FROM epsg_coordinatereferencesystem ec
    LEFT JOIN epsg_usage eu ON
ec.coord_ref_sys_code = eu.object_code
    LEFT JOIN epsg_extent ee ON eu.extent_code
= ee.extent_code
    WHERE ec.coord_ref_sys_code = $1`

```

```
row := e.db.QueryRow(query, code)
```

```
record := model.EpsgExtentRecord{}
```

```

err := row.Scan(&record.Name, &record.Code,
    &record.BboxSouthBoundLat,
    &record.BboxWestBoundLon,
    &record.BboxNorthBoundLat,
    &record.BboxEastBoundLon)

```

```

    if err != nil {
        return nil, err
    }
    return &record, nil
}

```

```
backend/internal/repository/srs_repository.go
```

```
package repository
```

```

import (
    "database/sql"
    "map-projection-explorer-
backend/internal/domain/model"
)

```

```

type srsRepository struct {
    db *sql.DB
}

```

```

func NewSrsRepository(db *sql.DB) SrsRepository {
    return &srsRepository{db: db}
}

```

```

func (s *srsRepository) FindByCode(code int)
(*model.SrsRecord, error) {
    query := `
        SELECT
            proj4text
        FROM spatial_ref_sys
        WHERE srid = $1`

```

```
row := s.db.QueryRow(query, code)
```

```
record := model.SrsRecord{}
```

```

err := row.Scan(&record.Proj4text)
if err != nil {
    return nil, err
}
return &record, nil
}

```

```
backend/internal/server/http/api.go
```

```
package http_server
```

```
import "net/http"
```

```

type CrsHttpServer interface {
    FindCoordinateReferenceSystem(w
http.ResponseWriter, r *http.Request)
    FindAllCoordinateReferenceSystems(w
http.ResponseWriter, r *http.Request)
}

```

```

func RegisterServerHandlers(mux *http.ServeMux,
server CrsHttpServer) {
    mux.HandleFunc("GET      /api/v1/coordinate-
reference-systems/{code}",
server.FindCoordinateReferenceSystem)
    mux.HandleFunc("GET      /api/v1/coordinate-
reference-systems",
server.FindAllCoordinateReferenceSystems)
}

```

```
backend/internal/server/http/crs_server.go
```

```
package http_server
```

```

import (
    "encoding/json"
    "log"
    "map-projection-explorer-
backend/internal/domain/dto"
    "map-projection-explorer-
backend/internal/service"
    "net/http"
    "strconv"
)

```

```

type crsServer struct {
    crsService service.CrsService
}

```

```

func NewServer(crsService service.CrsService)
CrsHttpServer {
    return &crsServer{crsService: crsService}
}

```

```

func FindCoordinateReferenceSystem(w http.ResponseWriter, r *http.Request) {
    code, err := strconv.Atoi(r.PathValue("code"))
    if err != nil {
        serviceError := &dto.ServiceError{Code:
        dto.ErrorInvalidRequest, Message: "Request path
        value 'code' invalid"}
        writeResponse(w, serviceError,
        http.StatusBadRequest)
        return
    }

    crsRecord, serviceErr :=
    c.crsService.FindCoordinateReferenceSystem(code)
    if serviceErr != nil {
        writeResponse(w, serviceErr,
        serviceErr.Code.Value)
        return
    }

    writeResponse(w, crsRecord, http.StatusOK)
}

func FindAllCoordinateReferenceSystems(w http.ResponseWriter, r *http.Request) {
    query := r.URL.Query()

    search := query.Get("search")

    pageCursor, err :=
    extractPageCursor(query.Get("pageCursor"))
    if err != nil {
        serviceError := &dto.ServiceError{Code:
        dto.ErrorInvalidRequest, Message: "Request
        parameter 'pageCursor' invalid"}
        writeResponse(w, serviceError,
        http.StatusBadRequest)
        return
    }

    pageSize, err :=
    strconv.Atoi(query.Get("pageSize"))
    if err != nil {
        serviceError := &dto.ServiceError{Code:
        dto.ErrorInvalidRequest, Message: "Request
        parameter 'pageSize' invalid"}
        writeResponse(w, serviceError,
        http.StatusBadRequest)
        return
    }

    crsRecords, serviceErr :=
    c.crsService.FindAllCoordinateReferenceSystems(se
    arch, pageCursor, pageSize)
    if serviceErr != nil {
        writeResponse(w, serviceErr,
        serviceErr.Code.Value)
        return
    }

    writeResponse(w, crsRecords, http.StatusOK)
}

func extractPageCursor(pageCursor string) (*int, error)
{
    if pageCursor == "" {
        return nil, nil
    }
    val, err := strconv.Atoi(pageCursor)
    if err != nil {
        return nil, err
    }
    return &val, nil
}

func writeResponse(w http.ResponseWriter, response
any, status int) {
    data, err := json.Marshal(response)
    if err != nil {
        http.Error(w, "Error serializing response",
        http.StatusInternalServerError)
        return
    }

    w.Header().Add("Content-Type",
    "application/json")
    w.WriteHeader(status)
    _, err = w.Write(data)
    if err != nil {
        log.Println("Error writing response:", err)
    }
}

backend/internal/service/api.go

package service

import (
    "map-projection-explorer-
    backend/internal/domain/dto"
)

type CrsService interface {
    FindCoordinateReferenceSystem(code int)
    (*dto.CrsRecord, *dto.ServiceError)
    FindAllCoordinateReferenceSystems(search
    string, pageCursor *int, pageSize int)
    (*dto.Page[dto.CrsRecord], *dto.ServiceError)
}

backend/internal/service/crs_service.go

```

```
package service
```

```
import (
    "database/sql"
    "errors"
    "log"
    "map-projection-explorer-
backend/internal/domain/dto"
    "map-projection-explorer-
backend/internal/domain/model"
    "map-projection-explorer-
backend/internal/repository"
    "math"
)
```

```
type crsService struct {
    epsgRepository
    repository.EpsgExtentRepository
    srsRepository repository.SrsRepository
}
```

```
func NewCrsService(epsgRepository
repository.EpsgExtentRepository, srsRepository
repository.SrsRepository) CrsService {
    return &crsService{epsgRepository:
epsgRepository, srsRepository: srsRepository}
}
```

```
func (c *crsService) FindCoordinateReferenceSystem(code int)
(*dto.CrsRecord, *dto.ServiceError) {
    epsgRecord, err :=
c.epsgRepository.FindByCode(code)
    if errors.Is(err, sql.ErrNoRows) {
        return nil, &dto.ServiceError{Code:
dto.ErrorCodeNotFound, Message: "Extent with such
code not found"}
    }
    if err != nil {
        log.Printf("Failed to find epsg record by
code: %s", err)
        return nil, &dto.ServiceError{Code:
dto.ErrorCodeInternalServerError, Message: "Unknown
error"}
    }
}
```

```
    srsRecord, err :=
c.srsRepository.FindByCode(code)
    if errors.Is(err, sql.ErrNoRows) {
        return nil, &dto.ServiceError{Code:
dto.ErrorCodeNotFound, Message: "SRS with such
code not found"}
    }
    if err != nil {
        log.Printf("Failed to find srs record by
code: %s", err)
    }
}
```

```
        return nil, &dto.ServiceError{Code:
dto.ErrorCodeInternalServerError, Message: "Unknown
error"}
    }
}
```

```
    crsRecord :=
convertToCrsRecordFull(epsgRecord, srsRecord)
    return crsRecord, nil
}
```

```
func (c *crsService) FindAllCoordinateReferenceSystems(
search string, pageCursor *int, pageSize int)
(*dto.Page[dto.CrsRecord], *dto.ServiceError) {
```

```
    if pageSize <= 0 || pageSize == math.MaxInt {
        return nil, &dto.ServiceError{Code:
dto.ErrorInvalidRequest, Message: "Invalid page size"}
    }
```

```
    epsgRecords, err :=
c.epsgRepository.FindAllAfterCode(search,
pageCursor, pageSize+1)
    if err != nil {
        log.Printf("Failed to find epsg records by
code: %s", err)
        return nil, &dto.ServiceError{Code:
dto.ErrorCodeInternalServerError, Message: "Unknown
error"}
    }
```

```
    var crsRecords []*dto.CrsRecord
    for _, epsgRecord := range epsgRecords {
        crsRecords = append(crsRecords,
convertToCrsRecord(epsgRecord))
    }
```

```
    if len(epsgRecords) > pageSize {
        crsRecords = crsRecords[:pageSize]
    }
    page := dto.Page[dto.CrsRecord]{
        Content: crsRecords,
        NextPageCursor:
findNextPageCursor(crsRecords, len(epsgRecords),
pageSize),
    }
    return &page, nil
}
```

```
func findNextPageCursor(records []*dto.CrsRecord,
recordsLen, pageSize int) *int {
    if len(records) == 0 || recordsLen <= pageSize {
        return nil
    }
    return &records[len(records)-1].Code
}
```

```

func          convertToCrsRecord(epsgRecord
*model.EpsgExtentRecord) *dto.CrsRecord {
    crsRecord := dto.CrsRecord{
        Name: epsgRecord.Name,
        Code: epsgRecord.Code,
    }
    return &crsRecord
}

```

```

func          convertToCrsRecordFull(epsgRecord
*model.EpsgExtentRecord,          srsRecord
*model.SrsRecord) *dto.CrsRecord {
    crsRecord := dto.CrsRecord{
        Name: epsgRecord.Name,
        Code: epsgRecord.Code,
        Definition: &dto.CrsRecordDefinition{
            Type:
dto.CrsRecordDefinitionTypeProj4,
            Value: srsRecord.Proj4text,
        },
        Extent: &dto.CrsRecordExtent{
            WestLongitude:
&epsgRecord.BboxWestBoundLon,
            SouthLatitude:
&epsgRecord.BboxSouthBoundLat,
            EastLongitude:
&epsgRecord.BboxEastBoundLon,
            NorthLatitude:
&epsgRecord.BboxNorthBoundLat,
        },
        ExtentCenter:
calculateExtentCenter(epsgRecord),
    }
    return &crsRecord
}

```

```

func          calculateExtentCenter(epsgRecord
*model.EpsgExtentRecord)
*dto.CrsRecordExtentCenter {
    n := epsgRecord.BboxNorthBoundLat
    s := epsgRecord.BboxSouthBoundLat
    w := epsgRecord.BboxWestBoundLon
    e := epsgRecord.BboxEastBoundLon

    lon := w + 180 + (360-(w+180)+e+180)/2.0
    lat := (n-s)/2.0 + s

    return &dto.CrsRecordExtentCenter{
        Longitude: &lon,
        Latitude: &lat,
    }
}

```