

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ НАЗЕМНИХ СПОРУД І АЕРОДРОМІВ
Кафедра аерокосмічної геодезії та землеустрою

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

_____ Юрій ВЕЛИКОДСЬКИЙ

«___» _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА (ПОЯСНЮВАЛЬНА ЗАПИСКА)

ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ «БАКАЛАВР»

Тема: «Використання даних OpenStreetMap для 3D-моделювання міст»

Виконавець: Потапенко Анна Сергіївна, студентка групи ГС-412Б

Керівник: Великодський Юрій Іванович, к.ф.-м.н., ст.досл. _____

Нормоконтролер: Іщенко Наталія Федорівна, PhD, доцент _____

КИЇВ 2024

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет наземних споруд і аеродромів

Кафедра аерокосмічної геодезії та землеустрою

Спеціальність 193 «Геодезія та землеустрій»

Освітньо-професійна програма «Геоінформаційні системи і технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____Юрій ВЕЛИКОДСЬКИЙ

«__» _____ 2024р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи

Потапенко Анні Сергіївні

1. Тема кваліфікаційної роботи: «Використання даних OpenStreetMap для 3D-моделювання міст», затверджена наказом ректора від 22.04.2024 року № 601/ст.
2. Термін виконання роботи: з 20 травня 2024 р. по 16 червня 2024 р.
3. Вихідні дані роботи: дані OpenStreetMap.
4. Зміст пояснювальної записки: Огляд теоретичних аспектів 3D-моделювання та структури даних OpenStreetMap. Опрацювання даних, отриманих з OpenStreetMap та їх картографування.
5. Перелік обов'язкового ілюстративного матеріалу: таблиці, рисунки.
6. Календарний план-графік:

№ п/п	Завдання	Термін виконання	Підпис керівника
1.	Отримання завдання	20.05.2024	
2.	Формування змісту роботи	20.05.2024- 21.05.2024	
3.	Опрацювання літературних джерел за тематикою роботи	21.05.2024- 22.05.2024	

4.	Робота над першим розділом роботи	23.05.2024- 26.05.2024	
5.	Робота з тематикою другого розділу роботи	27.05.2024- 31.05.2024	
6.	Робота над практичною частиною роботи	31.05.2024- 07.06.2024	
7.	Формування висновків. Оформлення пояснювальної записки	08.06.2024- 10.06.2024	
8.	Підготовка до захисту. Захист роботи	11.06.2024- 16.06.2024	

Дата видачі завдання: «20» травня 2024 р.

Керівник кваліфікаційної роботи: _____ Великодський Ю.І.

Завдання прийняв до виконання: _____ Потапенко А.С.

РЕФЕРАТ

Загальний обсяг пояснювальної записки до дипломної роботи «Використання даних OpenStreetMap для 3D-моделювання міст» складає 63 сторінки та містить 15 рисунків, 3 таблиці, 20 використаних джерел.

Об'єктом дослідження кваліфікаційної роботи є дані OpenStreetMap.

Мета роботи: дослідити можливість використання даних OpenStreetMap для 3D-моделювання міського середовища та проаналізувати наповненість цих даних у розрізі регіонів України.

Методи дослідження: аналіз літературних джерел та наукових публікацій по темі дослідження, статистичний та картографічний.

Результати бакалаврської роботи можуть бути використані для подальших та більш детальних досліджень.

OPENSTREETMAP, 3D-МОДЕЛЮВАННЯ, OVERPASS TURBO, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, BUILDING INFORMATION MODELING, ДЖЕРЕЛО ДАНИХ, ГЕОГРАФІЧНІ ОБ'ЄКТИ.

ЗМІСТ

ВСТУП	6
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	8
РОЗДІЛ 1. 3D-МОДЕЛЮВАННЯ МІСЬКОГО СЕРЕДОВИЩА	9
1.1. Основні поняття про 3D-моделювання	9
1.2. Методи створення 3D-моделей міського середовища	12
1.3. Опис BIM-технологій.....	18
РОЗДІЛ 2. OPENSTREETMAP ЯК ДЖЕРЕЛО ДАНИХ.....	25
2.1. Огляд OpenStreetMap.....	25
2.2. Структура та характеристика даних у OpenStreetMap.....	29
2.3. 3D-візуалізатори OpenStreetMap	37
РОЗДІЛ 3. АНАЛІЗ МОЖЛИВОСТІ ВИКОРИСТАННЯ ДАНИХ	
OPENSTREETMAP ДЛЯ 3D-МОДЕЛЮВАННЯ МІСТ.....	44
3.1. Вебсервіс Overpass Turbo.....	44
3.2. Аналіз даних, отриманих за допомогою вебсервісу Overpass Turbo....	51
ВИСНОВКИ.....	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	56
ДОДАТКИ	59

ВСТУП

Актуальність теми. З невідпинним розвитком сучасних технологій у світі підвищується і потреба у знаходженні швидких, точних та ефективних рішень, які будуть спрощувати отримання 3D-моделі міста з метою створення архітектурного проекту перед початком будівництва, або ж для управління містом та планування розвитку його інфраструктури та просторової організації.

Дані OSM є багатим і всеосяжним джерелом просторової інформації про міста, включаючи дороги, будівлі, землекористування, об'єкти благоустрою тощо. Ці детальні дані дозволяють точно відображати міське середовище у 3D-моделях, що сприяє кращій візуалізації, аналізу та прийняттю рішень у різних сферах, таких як містобудування, архітектура, транспорт та управління надзвичайними ситуаціями. Використання даних OSM для 3D-моделювання сприяє співпраці та залученню громади. Оскільки OSM є відкритою платформою, яка дозволяє будь-кому надавати геопросторові дані та отримувати до них доступ, це сприяє створенню середовища співпраці, в якому окремі особи, громади та організації можуть працювати разом над створенням та вдосконаленням 3D-моделей міст. Ці колективні зусилля призводять до розробки більш повних та актуальних моделей міст, що відображають динамічну природу міст у часі.

Крім того, інтеграція даних OSM у робочі процеси 3D-моделювання підвищує доступність інструментів і технологій моделювання міст. У порівнянні з власницькими або комерційними наборами даних, OSM пропонує економічно ефективну та гнучку альтернативу для отримання просторових даних, роблячи їх доступними для ширшого кола користувачів, включаючи дослідників, студентів та малих розробників. Така демократизація даних дає можливість окремим особам та організаціям створювати 3D-моделі міст для різних цілей, таких як містобудівне моделювання, ігри, віртуальний туризм та освіта. Використання даних OSM для 3D-моделювання підтримує інтероперабельність та сумісність з іншими наборами геопросторових даних та платформами. Дані OSM можна легко інтегрувати з існуючим програмним забезпеченням ГІС, інструментами

3D-моделювання та середовищами віртуальної реальності, що дозволяє безперешкодно обмінюватися даними та співпрацювати між різними платформами. Така інтеперабельність підвищує універсальність і корисність даних OSM для додатків 3D-моделювання, дозволяючи користувачам використовувати широкий спектр інструментів і ресурсів для створення захоплюючих та інтерактивних моделей міст.

Загалом, застосування даних OpenStreetMap є дуже актуальною та впливовою у сфері містобудування, ГІС та розробки віртуальних середовищ. Використовуючи можливості даних OSM, практики та дослідники можуть створювати точні, спільні та доступні 3D-моделі міст, що сприятиме кращому розумінню, плануванню та управлінню міськими просторами в цифрову епоху.

Метою даної роботи є дослідження того, чи можливо використовувати дані OpenStreetMap для створення 3D-моделей міського середовища та оцінка широкості цих даних у різних регіонах України.

Для досягнення поставленої мети в кваліфікаційній роботі необхідно вирішити наступні **завдання**:

- отримати дані про наявність в базі даних OpenStreetMap тегів, які є основою для створення елементарної 3D-моделі міста;
- побудувати карти розподілу за регіонами України відсотку наявності відповідних тегів;
- здійснити аналіз отриманих статистичних даних;
- визначити, для моделювання яких регіонів України можуть використовуватись дані OpenStreetMap.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ГІС — Геоінформаційні системи

OSM — OpenStreetMap

САПР — Система автоматизованого проектування і розрахунку

BIM — Building Information Modeling — Інформаційна модель будівлі

GPS — Global Positioning System — Система глобального позиціонування

РОЗДІЛ 1. 3D-МОДЕЛЮВАННЯ МІСЬКОГО СЕРЕДОВИЩА

1.1. Основні поняття про 3D-моделювання

Вперше про термін «3D-моделювання» стало відомо у 1960 році. Його винахідниками були графічний дизайнер Вільям Феттер та Верн Хадсон. А в 1963 з'явилося перше програмне забезпечення для 3D-візуалізації під назвою Sketchpad. Дана програма стала основою для стрімкого розвитку комп'ютерного 3D-моделювання і запустила безповоротно його вдосконалення [12].

3D-моделювання — це цифровий процес, який передбачає створення тривимірних зображень об'єктів або середовищ за допомогою спеціалізованого комп'ютерного програмного забезпечення. Він також передбачає ретельну побудову віртуальної геометрії для точного відтворення форми, розмірів і просторових взаємовідносин задуманого об'єкта.

XYZ координати в 3D-моделюванні служать основою для точного розташування точок в тривимірному просторі. Заснована на декартовій системі координат, яку започаткував Рене Декарт у 17 столітті, ця система використовує три числові значення - X, Y та Z - для окреслення положення будь-якої точки. X-координата позначає горизонтальне положення, Y-координата означає вертикальне положення, а Z-координата вимірює глибину або відстань від спостерігача. Ці координати незамінні в спектрі дисциплін, включаючи комп'ютерну графіку, інженерію, архітектуру тощо, де точність розміщення об'єктів у 3D-середовищах має першорядне значення. Завдяки ретельному маніпулюванню координатами XYZ з'являється можливість розробляти та маневрувати об'єктами, щоб забезпечити не тільки їх просторову точність, але й реалістичну взаємодію у віртуальних сферах. Візуалізація координат XYZ часто представляється через графічне зображення осей: вісь X проходить горизонтально, вісь Y піднімається вертикально, а вісь Z розширюється перпендикулярно до обох. Цей візуальний фреймворк допомагає користувачам будувати складні 3D-структури з ретельною точністю. В цілому, координати XYZ

є основоположним елементом у сфері 3D-моделювання, надаючи стандартизовану структуру для розмежування положення, орієнтації та масштабу об'єктів [12].

Тривимірна модель являє собою фізичне тіло, яке використовує серію точок у 3D просторі, які з'єднуються між собою типами геометричних об'єктів, такими як трикутники, лінії та полігони. 3D-моделі з набором даних (точок та іншої інформації) можна створити вручну, алгоритмічно (процедурне моделювання) або сканувати. Їхні поверхні можна додатково ідентифікувати за допомогою відображення текстури.

Полігони є базовими геометричними компонентами тривимірної моделі. Тобто, фактично, вони є будівельними блоками для створення об'єктів різної складності у 3D-моделюванні. Кожен полігон створений хоча б з трьох вершин та ребер, які з'єднуючись, утворюють замкнену форму. Площина, що утворюється в результаті цього, називається гранню. Зазвичай вона є трикутним багатокутником, але є також чотиристоронні (квадри) та грані з кількома вершинами (n-гони). З'єднуючись між собою, полігони утворюють полігональну сітку, по суті, це і є 3D-моделлю [1].

Залежно від кількості полігонів, із яких складається тривимірна модель, можна виділити високополігональні та низькополігональні 3D-моделі. Високополігональна модель містить в собі велику кількість полігонів для детального та складного представлення об'єкта. Вона створюється шляхом додавання полігонів до поверхні об'єкта, який моделюється. Завдяки цьому з'являється можливість відобразити значно більшу кількість деталей, наприклад, складні текстури на об'єктах або ж дрібні деталі поверхні на архітектурних спорудах. Не зважаючи на виняткову деталізацію, високополігональна модель має недолік: вона вимагає вагомих обчислювальних ресурсів, що значно уповільнює роботу. Низькополігональна модель характеризується мінімальною кількістю полігонів, це дає змогу оптимізувати продуктивність. Крім зменшення кількості полігонів вона часто проходить процес під назвою оптимізація сітки. У даний процес входить вдосконалення сітчастої структури з метою збереження

загальної форми та зовнішнього вигляду тривимірної моделі. Таким чином модель не зазнає значних спотворень та є ефективною для візуалізації та взаємодії в реальному часі [1].

Текстурування необхідне для того, щоб додати візуальних ефектів до моделі, тобто для її деталізації. Дифузна, дзеркальна, нормальна та карта зміщення є базовими текстурами, які зазвичай використовуються у 3D-моделюванні. Дифузна карта є найпростішою, вона додає текстуру та колір моделі, а також імітує її матеріал. Функцією дзеркальної карти є контроль кількості світла, що відбивається від поверхні моделі. Це стимулює блиск або відбиття матеріалу. Карта нормалей імітує нерівності та складки моделі. Зазвичай вона використовується для додавання деталей без збільшення кількості полігонів. Завдяки карті зміщення на модель додається висота та глибина моделі для того, щоб деталізувати та зробити вигляд більш реалістичним.

Використовуючи наявні дані, програма для 3D-візуалізації створює фотореалістичне зображення об'єкта. Без візуалізації об'єкт залишається надто технічним для загального розуміння, оскільки необробленій 3D-моделі бракує реалістичності.

3D-візуалізація полягає у перетворенні складних цифрових наборів даних у візуально сприйнятливі зображення об'єктів. Це означає перетворення технічних складнощів у візуально привабливі репрезентації. .

Візуалізація зазвичай вважається найбільш складним і трудомістким етапом 3D-виробництва. Перш ніж досягти реалістичної візуалізації, фахівці з САПР повинні ретельно відтворити елементи реального світу, що сприймаються людським оком. Це передбачає маніпулювання цифровими даними за допомогою математичного представлення поверхонь і точок для побудови реалістичного 3D-середовища на екрані комп'ютера.

Тривалість процесу 3D-візуалізації значно варіюється залежно від складності об'єктів і сцен. Це пов'язано з тим, що програмне забезпечення ретельно фіксує кожен піксель зображення, що може зайняти багато часу.

1.2. Методи створення 3D-моделей міського середовища

За своєю суттю 3D-моделювання міського середовища передбачає цифрове відтворення міст, селищ та інших населених пунктів у тривимірному просторі. Цей процес охоплює фіксацію фізичних і просторових атрибутів міських ландшафтів, включаючи будівлі, вулиці, інфраструктуру та природні елементи, з високим ступенем точності та деталізації. Суть 3D-моделювання в цьому контексті полягає в його здатності забезпечити всебічне та всеосяжне розуміння міських просторів, сприяючи прийняттю обґрунтованих рішень, міському дизайну та просторовому аналізу.

Основними методами створення 3D-моделей міського середовища є фотограмметрія, лазерне сканування, процедурне моделювання.

Фотограмметрія — це наука та технологія отримання надійної тривимірної геометричної та фізичної інформації про об'єкти та навколишнє середовище з фотографічних зображень. На практиці фотограмметрія дозволяє проводити 3D вимірювання геометричної інформації об'єктів (наприклад, положення, орієнтації, форми та розміри) з фотографій [7]. Основними методами отримання 3D-інформації із зображень за допомогою фотограмметрії є орієнтація зображення, налаштування пучка і зіставлення зображень.

Орієнтація зображення передбачає відновлення даних позиції та орієнтації оптичних променів із зібраних зображень. Воно складається з двох етапів: внутрішнього орієнтування і зовнішнього орієнтування [7]. Внутрішнє орієнтування перетворює піксельні координати на координати простору зображення, як правило, використовуючи контрольні позначки. Зовнішнє орієнтування перетворює координати простору зображення в координати простору тривимірного об'єкта за допомогою рівнянь колінеарності. Параметри зовнішнього орієнтування, включаючи положення камери та кути повороту, є вирішальними для 3D-картографії. Вони можуть бути виміряні датчиками або отримані за допомогою просторової резекції, відносної орієнтації та абсолютної орієнтації. Відносна орієнтація встановлює внутрішні зв'язки між

зображеннями, тоді як абсолютна орієнтація масштабує, обертає та переводить 3D-модель у фактичну систему координат. Обидва методи можна застосовувати до окремих зображень або блоків зображень [8].

Коригування пучка є альтернативою методу відносної та абсолютної орієнтації. Воно використовує рівняння колінеарності для визначення оптичних променів, починаючи з точок зображення, проходячи через центр перспективи камери та досягаючи 3D-точок. Зіставляючи зв'язувальні точки на стереопарах або кількох зображеннях, пучок оптичних променів з'єднує зображення та зв'язує простір зображення з простором об'єкта. Коригування пучка спрямоване на покращення параметрів орієнтації зображення для забезпечення точного перетину оптичних променів у 3D-точках. Він працює за принципом найменших квадратів і включає чотири типи рівнянь спостереження.

Коригування пучка одночасно обчислює невідомі параметри та координати 3D-об'єктів зв'язувальних точок, використовуючи рівняння спостереження, контрольні точки та зв'язувальні точки, зіставлені на зображеннях. Він поєднує процеси резекції простору та перетину. Різні ваги призначаються спостереженням на основі їх точності, що дозволяє контролювати їхні внески. Спостереження з більшою точністю отримують вищі ваги, тоді як спостереження з більшою невизначеністю отримують менші ваги. Воно виправляє систематичні помилки та пропонує статистичну інформацію, включаючи залишки для оцінки параметрів [8].

Зіставлення зображень — це важливий процес визначення відповідних точок на кількох зображеннях із покриттям, що перекривається, що представляють ту саму точку об'єкта. Зазвичай воно покладається на пошук подібності в рівнях локальних фрагментів зображення або зіставлення фрагмента зображення з шаблоном. Зіставлення зображень може виконуватися попіксельно, відоме як щільне зіставлення, або шляхом зіставлення окремих точок або елементів шаблону, що називається збігом ознак.

Існують різні методи зіставлення зображень, наприклад нормалізоване взаємне кореляційне зіставлення, яке оцінює подібність між фрагментами

зображення на основі їхнього рівня взаємної кореляції рівня сірого. Зіставлення точок ознак, прикладом якого є масштабоінваріантне ознакове перетворення, виявляє точки ознак, інваріантні до змін масштабу, і зіставляє їх на основі градієнтів місцевих областей. Однак масштабоінваріантне ознакове перетворення не є достатньо результативним. Напівглобальне зіставлення поєднує глобальні та локальні підходи для попиксельного зіставлення, хоча його глобальна оптимізація може призвести до надмірного згладжування 3D-реконструкцій.

Ієрархічне зіставлення зображень передбачає зіставлення ознак, за яким слідує щільне зіставлення з використанням триангуляцій для збігу характеристик і країв. Трикутники динамічно адаптуються під час зіставлення, включаючи щойно зіставлені точки та ребра. Цей метод розставляє пріоритети зіставлення відмінних ознак, сприяючи надійним функціям і щільним обмеженням відповідності.

Традиційна фотограмметрія стикається з обмеженнями при 3D картографуванні та моделюванні міських територій через те, що вона зосереджена на зображеннях, знятих близько до надиру камери, встановленими на літальних апаратах. Зіставлення зображень у міських умовах є особливо складним завданням, що вимагає значних людських зусиль, особливо в густонаселених мегаполісах з високими будівлями. Однак, останніми роками вдосконалення апаратного та програмного забезпечення для збору даних та обробки зображень трансформувало фотограмметрію. Найсучасніші системи косокутної фотограмметрії тепер знімають повітряні косі зображення в міських районах з високою надлишковістю, гарантуючи, що кожна наземна точка буде видима на кількох зображеннях. Таке вдосконалення значно покращує автоматичне зіставлення зображень у міських умовах і надає детальну інформацію про фасади будівель.

Методи моделювання на основі лазерного сканування мають високу точність, надійність і не піддаються впливу змін у навколишньому середовищі. Системи МЛС виконують зйомку за допомогою наземних транспортних засобів.

Мобільне лазерне сканування (МЛС) - це передова технологія, призначена для отримання тривимірних даних про навколишні об'єкти під час руху скануючої системи [10].

Системи МЛС виконують зйомку за допомогою наземних транспортних засобів. Навігаційна система МЛС містить глобальну навігаційну супутникову систему (GNSS) та інерційний вимірювальний пристрій (ІВП), забезпечує траєкторію і положення транспортного засобу для генерації прив'язаних до місцевості 3D хмар точок.

GNSS надає дані про географічне положення та швидкість антени приймача GNSS, використовуючи сузір'я орбітальних супутників. Вимірювання положення обчислюється шляхом триангуляції супутникових сигналів у межах чіткого огляду антени приймача. Для визначення позиції має бути видно чотири супутники, і точність GNSS в ідеалі зростає, коли стає доступніше більше супутників. Однак існують деякі загальні джерела помилок, через які приймачі GNSS зазвичай мають точність позиціонування 1-2 м. Такі перешкоди, як будівлі чи дерева, можуть блокувати супутниковий сигнал, що призводить до ненадійної навігації. Інерційна навігаційна система обчислює відносне положення об'єкта з часом, використовуючи вимірювання обертання та прискорення від ІВП, який може вимірювати відносний рух у 3D-просторі. ІВП містить шість взаємодоповнюючих датчиків, які розташовані на трьох ортогональних осях. На кожній із трьох осей з'єднані акселерометр і гіроскоп, які вимірюють лінійне та обертальне прискорення відповідно. На основі вимірювань лінійного прискорення та прискорення обертання інерційна навігаційна система може обчислити положення та швидкість для всіх трьох осей.

Хмара точок створюється лазерними сканерами (лідарами), які можуть оцінити відстань до об'єкта, випромінюючи лазерне світло та вимірюючи час, потрібний світлу для повернення до датчика [10]. Обертовий лідар має огляд на 360°. З кожним обертом він може сканувати точки вздовж конуса, що походить від датчика, таким чином утворюючи одну круглу лінію сканування. Цей кут конуса змінюється на попередньо визначену величину після кожного повного

оберту з максимальним абсолютним кутом таким, що датчик не може сканувати область безпосередньо над або під ним. Перевагами лідару є забезпечення різних діапазонів вимірювання, мінімальна кутова роздільна здатність, низька похибка вимірювання, забезпечення як середньої, так і високої частоти дискредитації. Головним недоліком лідару є його висока вартість.

Завдяки лазерним сканерам навіть складні об'єкти можна оцифрувати з вражаючою точністю. Враховуючи різні технології оцифрування, все ще існує кілька обмежень, пов'язаних із умовами навколишнього середовища, оклюзією між об'єктами та можливостями датчиків, які обмежують повну ефективність глибини сцени, зафіксованої мобільним лазерним сканером. Неповні дані створять невизначеність для подальшої обробки. Щоб уникнути цього, необхідно мати відповідну повну версію даних. Для простого збору даних можна виконати повторне сканування для отримання нових даних. Однак інколи отримати повну версію 3D-даних шляхом повторного сканування може бути складно через оклюзію, спричинену об'єктами, або недоступність скануючого пристрою до зони спостереження, і тому потрібно доповнити дані вручну або автоматично [10]. Це створило область для доповнення відсутньої 3D-інформації даних МЛС або інших форм 3D-даних.

Параметричне моделювання — це процес моделювання, який дозволяє розробнику змінювати геометрію моделі, змінюючи її значення розмірів і оцінюючи сценарії проектування на основі інтеграції геометричних змінних і правил у ітераційному процесі проектування [13]. Генеративні, параметричні та алгоритмічні моделі є синонімами обчислювального дизайну, призначеного для використання «штучного інтелекту» та передової математики для створення складних форм віртуальної реальності. Параметричне моделювання здійснюється з використанням коду комп'ютерного програмування для визначення розмірів і геометрії моделі. В даний час обчислювальний і статистичний аналіз використовується в інженерії та містобудуванні; зокрема з розвитком методів мислення в процесах міського планування. Виникла необхідність вибору найкращих доступних технологій для їх інтеграції в

проектування та реалізацію проектів містобудування. Параметричне моделювання — це послідовний процес створення різних проектів або сценаріїв шляхом встановлення змінних архітектурних елементів. Визначення цих змінних має ідентифікувати взаємозв'язки між ними. Особливість параметричних моделей виникає внаслідок результатів, отриманих на основі математичних ймовірностей, заснованих на статистиці, оскільки це дозволяє користувачеві змінювати значення однієї змінної та виводити вплив на інші змінні, створюючи таким чином новий сценарій проектування на основі значень усіх змінних, тому немає потреби перемальовувати модель щоразу, коли вона потребує змін. Іншими словами, за допомогою параметричного моделювання дизайнер має змогу змінювати не лише окремі частини, а цілі форми моделі BIM [13].

Моделювання з використанням цих методів може сприяти етапам проектування великих проектів, а також вирішенню операційних проблем. Визначено багато переваг, включаючи централізоване керування даними, описову інформацію (просторові дані), покращення контролю витрат, дотримання графіка будівництва та взаємодію із зацікавленими сторонами з різних професій. Міську параметричну модель можна визначити як комп'ютер, заснований на міській моделі, який описує компоненти міської системи та перетворює кожен компонент на ряд змінних, які контролюють склад міських елементів у третьому вимірі, таких як будівлі, дороги, тротуари, стовпи освітлення, дерева, тіні, сидіння тощо. Параметричний підхід проектування забезпечує методологію для вираження вибраних елементів дизайну як змінних. Користувач може контролювати та змінювати початкові значення цих змінних, а також інтегрувати різні заздалегідь визначені правила та критерії, створюючи велику кількість альтернатив для однієї будівлі. Таким чином, можна відновити бачення міського контексту та створити нескінченну кількість сценаріїв [13]. З цієї причини інструменти параметричного проектування дозволяють залишати дизайн відкритим для модифікації з високим рівнем деталізації, щоб зрозуміти основні якості та елементи запропонованої конструкції.

Процес 3D-моделювання може бути детально розкритий на чотири етапи: імпорт 2D-географічних даних, визначення параметрів будівлі, розробка скрипту, застосування коду до міського плану.

На першому етапі дані про міську територію, такі як векторні або растрові файли, імпортуються у формі просторових шарів. Ці дані можуть включати інформацію про дороги, ділянки землі, водні об'єкти та інші важливі деталі міської інфраструктури. На другому етапі вводяться параметри, які визначають характеристики будівлі. Ці параметри можуть включати висоту будівлі, кількість поверхів, розміри вікон та дверей, форму даху, наявність зовнішніх елементів, таких як колони чи відвідні доріжки, та інші. На третьому етапі міський планувальник або розробник створює скрипт, який містить всі необхідні функції та підпрограми для перетворення інформації про будівлю в 3D-елементи. Цей скрипт може включати правила для розташування будівель, створення деталей фасаду, встановлення текстур та багато іншого. На четвертому етапі розроблений скрипт застосовується до всього 2D плану міста. Під час цього кроку можна використовувати текстури, форми прорізів або зображення, що описують зовнішню композицію будівлі. Кінцевим результатом є готова 3D-модель для всіх будівель міста, яка може бути детально налаштована та відредагована.

1.3. Опис BIM-технологій

Інформаційне моделювання будівель — це революційна технологія та процес, яка швидко змінила спосіб планування, проектування, будівництва та експлуатації будівель [2].

BIM – це не просто програмне забезпечення: це і процес, і програмне забезпечення. BIM означає не лише використання тривимірних інтелектуальних моделей, але й внесення значних змін у робочий процес і процеси реалізації проекту.

Ключова відмінність між технологією BIM і традиційним 3D-САПР полягає в їх підході до представлення будівель [2]. У той час як у звичайній 3D-

САПР для зображення будівлі використовуються окремі 3D-види, як-от плани, розрізи та фасади, BIM об'єднує всі аспекти будівлі в цілісну, взаємопов'язану модель. У традиційній САПР зміна одного вигляду вимагає перехресних посилань і оновлення всіх інших видів, процес, схильний до помилок і часто призводить до неповної документації. Крім того, дані в цих 3D-кресленнях зазвичай містять графічні об'єкти, такі як лінії, дуги та кола, яким бракує інтелектуальної контекстної семантики, яка є в моделях BIM. Навпаки, моделі BIM визначають об'єкти з точки зору будівельних елементів і систем, що охоплюють простори, стіни, балки та колони. Інформаційна модель будівлі накопичує повну інформацію, що стосується будівлі, охоплюючи як її фізичні атрибути, так і функціональні характеристики, а також деталі, пов'язані з усім її життєвим циклом, усі інкапсульовані в серії «розумних об'єктів».

BIM являє собою комплексний віртуальний процес, який об'єднує всі аспекти, дисципліни та системи об'єкта в єдину віртуальну модель. Це забезпечує безперебійну співпрацю між усіма зацікавленими сторонами, включаючи власників, архітекторів, інженерів, підрядників, субпідрядників і постачальників, забезпечуючи підвищену точність і ефективність порівняно з традиційними методами. Під час розробки моделі члени команди постійно вдосконалюють і коригують свої внески, щоб узгодити їх зі специфікаціями проекту та змінами дизайну, забезпечуючи максимальну точність до фізичного початку проекту. В основі BIM лежать два фундаментальних принципи: спілкування та співпраця. Успішне впровадження BIM залежить від раннього залучення всіх зацікавлених сторін проекту, що робить традиційні системи доставки проектів, такі як проектування-тендер-збудова, застарілими в проектах на основі BIM [4]. Нещодавно концепція Integrated Project Delivery з'явилася як природний супутник BIM, об'єднуючи ключовий досвід управління будівництвом, торгівлею, виробництвом, постачальником і виробником продукції разом із професіоналами з проектування та власником на початку процесу. Результатом цієї співпраці є дизайн, оптимізований для якості, естетики,

конструктивності, доступності, своєчасності та бездоганної інтеграції в управління життєвим циклом.

Використання BIM на етапі програмування проекту дає команді проекту можливість провести аналіз простору та отримати уявлення про тонкощі космічних стандартів і земельних норм. Інтеграція BIM з ГІС покращує планування проекту, допомагаючи у виборі місця та проведенні техніко-економічних і маркетингових досліджень. Переваги підходу «GIS-BIM» до аналізу включають [4]:

1. Сприяння оцінці потенційних об'єктів для забезпечення їх відповідності критеріям проекту з урахуванням технічних, фінансових та інших відповідних факторів.
2. Зменшення витрат, пов'язаних із потребами комунальних послуг та знесенням.
3. Зменшення ризику зіткнення з небезпечними матеріалами.

Удосконалення технології 3D-лазерного сканування революціонізувало можливість точно сканувати існуючі комунальні послуги та включати їх у моделі BIM. Крім того, 3D-лазерне сканування можна використовувати для точного сканування існуючих об'єктів, дозволяючи розробляти BIM-моделі в межах промислових допусків. Потім ці моделі можна використовувати для проектів реконструкції та адаптивного повторного використання існуючих будівель, демонструючи універсальність та ефективність інтеграції 3D-сканування з методологіями BIM.

Архітектори та інженери можуть скористатися перевагами залучення BIM на різних етапах розробки проекту, а саме: схематичного проектування, детального проектування і детального проектування конструкції.

Використання BIM на етапі підготовки до будівництва охоплює кілька ключових застосувань [4]:

1. Оцінка: підрядники можуть використовувати інформаційні моделі будівель для проведення точних кількісних обстежень і створення комплексних оцінок з високим ступенем точності.

2. Координація об'єкта: застосовуючи 3D- або 4D-моделі координації об'єкта, підрядники можуть розробити стратегію логістики об'єкта, розробити схеми руху та точно визначити потенційні небезпеки на об'єкті, сприяючи створенню більш прагматичного плану безпеки об'єкта.
3. Аналіз конструктивності: моделі BIM дозволяють команді проекту проводити ретельний аналіз конструктивності, дозволяючи ретельно планувати послідовність операцій на робочому місці.

На етапі будівництва BIM виконує кілька важливих функцій для команди проекту:

1. Відстеження прогресу проекту полегшено завдяки використанню 3D планів поетапності.
2. BIM допомагає під час зустрічей з координації торгівлі.
3. Інформацію про замовлення на зміну та список перфорацій можна бездоганно інтегрувати в моделі BIM.

Під час будівництва безперервне оновлення BIM-моделі є обов'язковим, щоб переконатися, що вона відображає найновішу інформацію. Ця оновлена модель стає безцінною для експлуатації та обслуговування будівлі менеджерами об'єктів після будівництва. Удосконалення технологій смартфонів і планшетів дозволило підрядникам і субпідрядникам використовувати моделі BIM на місці для вилучення інформації та координації.

Інформаційна модель будівлі містить повну інформацію про еволюцію об'єкта на етапах планування, проектування та будівництва. Менеджери об'єктів можуть використовувати цю велику кількість інформації, що сприяє підвищенню ефективності роботи та обслуговування об'єктів. Значна частина вартості життєвого циклу об'єкта — приблизно 85 % — припадає на період після будівництва [4].

Основою утиліти BIM для управління об'єктами є її здатність надавати вичерпну інформацію про будівлю, включаючи її приміщення, системи та компоненти. Основна мета — бездоганно інтегрувати ці дані в операції з

управління об'єктом. Наприклад, у BIM-моделі кляцання об'єкта надає доступ до відповідної інформації про системи та обладнання будівлі. Цей спрощений доступ до важливої інформації дозволяє працівникам технічного обслуговування ефективно отримувати доступ до життєво важливих даних, пов'язаних з різними системами всередині будівлі, тим самим спрощуючи та оптимізуючи завдання з обслуговування.

Попри численні переваги, які BIM може принести зацікавленим сторонам проекту, впровадження цієї технології супроводжується ризиками та перешкодами. BIM не є універсальним рішенням для кожного проекту або фірми. Ризики, пов'язані з використанням BIM, можна узагальнити у дві великі категорії: ризики, пов'язані з технологіями, та ризики, пов'язані з процесами, [6].

Основний ризик, пов'язаний із технологіями, пов'язаний із відсутністю стандартизованих протоколів BIM для інтеграції та керування моделями між міждисциплінарними командами. Досягнення повної інтеграції мультидисциплінарної інформації в рамках єдиної моделі BIM вимагає багатокористувацького доступу, що підкреслює важливість встановлення протоколів на етапі програмування проекту для забезпечення узгодженості інформаційного контексту та стилів форматування. Наразі відсутність стандартизованих протоколів спонукає кожен фірму прийняти власні стандарти, що потенційно може призвести до неузгодженості в моделі. Нездатність виявити ці невідповідності може призвести до неточностей і невідповідностей у моделі BIM, що підкреслює необхідність частих «аудитів моделі» командою проекту для пом'якшення таких проблем. Проблеми з функціональною сумісністю, хоча помітно зменшилися, продовжують становити значний ризик. Інтєроперабельність означає здатність обмінюватися даними між програмами для оптимізації автоматизації та уникнення повторного введення даних [5]. Незважаючи на те, що запровадження галузевих базових класів і XML-схем значною мірою вирішило проблеми сумісності, обидва підходи мають властиві обмеження. Вибираючи програмні додатки BIM, користувачі повинні провести ретельне дослідження сумісності.

Інший потенційний ризик виникає, коли члени команди проекту, окрім власника та архітектора/інженера, вносять дані, інтегровані в інформаційну модель будівлі. Проблеми з ліцензуванням можуть виникнути в сценаріях, коли постачальники обладнання та матеріалів надають проекти, пов'язані зі своїми продуктами, головному дизайнеру, щоб вплинути на специфікації обладнання. Хоча така практика може принести користь діловим стосункам, можуть виникнути проблеми з ліцензуванням, якщо проекти не були створені ліцензованим дизайнером у місці реалізації проекту [5].

Ризики, пов'язані з процесом, включають юридичні, договірні та організаційні проблеми. Найбільший ризик пов'язаний із визначенням права власності на BIM-дані та їх захистом за допомогою законів про авторське право та інших правових засобів. Наприклад, у той час як власник може розраховувати на те, що він стане власником дизайну, якщо він фінансуватиме його, члени команди, які надають конфіденційну інформацію, вимагають захисту своїх даних. Таким чином, право власності на дані передбачає індивідуальний підхід до кожного проекту, щоб запобігти стримуванням або перешкодам, які заважають учасникам повністю використовувати потенціал моделі. Щоб запобігти суперечкам щодо авторських прав, у контрактних документах мають бути визначені права власності та обов'язки.

Інша договірна проблема передбачає визначення того, хто контролює введення даних у модель і несе відповідальність за будь-які неточності. Взяття на себе відповідальності за оновлення та забезпечення точності даних BIM тягне за собою значний ризик. Переговори щодо відшкодування збитків і гарантій, а також вирішення питань про відмову від відповідальності є ключовими аспектами, які необхідно вирішити перед розгортанням технології BIM. Крім того, введення та перегляд даних BIM вимагає додаткового часу та витрат у процесі проектування та адміністрування проекту. Хоча ці витрати можуть бути компенсовані підвищенням ефективності, вони залишаються витратами, понесеними членом команди проекту. Таким чином, перед повним

використанням технології BIM необхідно визначити ризики, розподілити та покрити витрати на впровадження [5].

Інтегрована концепція BIM розмиває межі відповідальності, потенційно посилюючи ризик і відповідальність. Контракти про спільну інтегровану реалізацію проекту пропонують ефективний підхід до пом'якшення таких ризиків шляхом розподілу ризиків і вигоди від використання BIM між учасниками проекту.

РОЗДІЛ 2. OPENSTREETMAP ЯК ДЖЕРЕЛО ДАНИХ

2.1. Огляд OpenStreetMap

OpenStreetMap — це проект, спрямований на створення повної безкоштовної географічної бази даних світу з метою документування кожного географічного об'єкта в усьому світі [11]. Спочатку він був зосереджений на картографуванні вулиць, але потім розширив сферу застосування, включивши в нього різні елементи, такі як пішохідні доріжки, будівлі, водні шляхи та навіть окремі дерева. Окрім фізичних особливостей, проект охоплює адміністративні межі, деталі землекористування та абстрактні поняття, такі як автобусні маршрути.

Користувачі збирають дані, перетинаючи вулиці та стежки, використовуючи приймачі GPS для запису своїх переміщень. Потім ці дані використовуються для створення точок і ліній, які утворюють карти або допомагають у навігації. Незважаючи на те, що в першу чергу це відбувається за рахунок волонтерів, комерційні організації та державні органи також почали робити свій внесок.

Краудсорсинг, процес розподілу завдань через Інтернет і збирання результатів від груп людей, є центральним у методології OpenStreetMap. Він покладається як на волонтерів, так і на організації, використовуючи широку доступність ширококутового Інтернету.

Збір даних включає різні джерела, включаючи карти, на які не поширюється авторське право, бази даних загального користування та власні бази даних, надані компаніями. У проекті використовується система, схожа на Вікі, де картографи можуть додавати або редагувати об'єкти, забезпечуючи точність даних через повну історію редагування [11].

OpenStreetMap підтримує власне програмне забезпечення та модель даних, а не покладається на існуючі системи ГІС, що забезпечує гнучкість підходів до картографування. Проект також сприяє розробці програмного забезпечення,

результатом чого є безліч інструментів з відкритим кодом для створення, редагування та використання даних.

Усі дані в OpenStreetMap є у вільному доступі для використання за відкритою ліцензією, що дозволяє копіювати, змінювати та розповсюджувати. Це відрізняється від інших картографічних служб, які обмежують використання та зміну даних. OpenStreetMap охоплює широкий спектр географічних об'єктів за межами доріг. Співавтори мають свободу додавати будь-які географічні об'єкти, з якими вони стикаються, створюючи різноманітний набір даних, який включає телефонні будки, автобусні зупинки, парки, громадські туалети та місця поклоніння. Примітно, що пішохідні та велосипедні доріжки детально задокументовані в OpenStreetMap порівняно з іншими базами даних, і численні велосипедні карти та проекти маршрутів використовують ці дані.

OpenStreetMap часто оновлюється, що забезпечує постійний доступ до останніх даних. На відміну від власних баз даних, які можуть мати затримки у випуску оновлень для клієнтів, повна база даних OpenStreetMap, відома як файл Planet, випускається щотижня. Крім того, оновлення надаються щодня, щогодини і навіть щохвилини.

Одним з унікальних аспектів OpenStreetMap є політика відкритого редагування, що дозволяє користувачам виправляти помилки в даних і ділитися цими виправленнями зі спільнотою.

Для збору даних для OpenStreetMap існує декілька джерел: GPS треки, супутникові знімки, власноруч зроблені фотографії [11].

Збір даних для OpenStreetMap за допомогою GPS-треків складається з кількох кроків. Спочатку користувачі використовують пристрої GPS або смартфони з підтримкою GPS, щоб записувати свої переміщення під час подорожі дорогами, стежками чи іншими географічними об'єктами. Ці пристрої з регулярними інтервалами збирають дані про широту, довготу та іноді про висоту. Після запису користувачі завантажують свої GPS-треки на свої комп'ютери або безпосередньо на платформу OSM. Це можна зробити за

допомогою різних програмних засобів або програм, які дозволяють користувачам отримувати дані GPS зі своїх пристроїв і зберігати їх у сумісному форматі.

Далі завантажені GPS-треки необхідно узгодити з наявними картографічними даними в OSM. Використовуючи GPS-треки як орієнтир, користувачі відстежують шляхи доріг, пішохідних доріжок, велосипедних доріжок, річок та інших географічних об'єктів на карті. Вони створюють вузли (точки) уздовж GPS-треку та з'єднують їх, щоб утворити лінії, що представляють об'єкти.

Коли географічні об'єкти простежуються, картографи додають додаткові атрибути, щоб підвищити корисність і точність картографічних даних. Ці метадані містять назви доріг, типи покриття, обмеження швидкості та іншу відповідну інформацію. Після додавання об'єктів і атрибутів картографи переглядають і перевіряють точність своїх змін. Вони також можуть вносити додаткові коригування на основі супутникових зображень, місцевих знань або інших доступних джерел даних, щоб забезпечити правильність.

Як наслідок, користувачі зберігають свої зміни в базі даних OSM. Потім ці зміни стають видимими іншим користувачам і стають частиною спільної роботи зі створення та підтримки точної та актуальної карти світу. Збираючи дані за допомогою GPS-треків і додаючи їх до OSM, користувачі відіграють важливу роль у покращенні якості та охоплення картографічних даних, роблячи їх цінним ресурсом для навігації, досліджень та різноманітних інших програм.

Для отримання даних за допомогою супутникових зображень необхідно спочатку отримати доступ до них, або ж скористатись наявними знімками в OpenStreetMap. Далі виконується визначення областей картографування, які є застарілими, або ж тих, змін яких вимагає поставлена задача. Далі користувачі вносять дані про об'єкти, видимі на зображенні супутника і додають усі необхідні атрибути (такі як назви доріг, типи будівель, класифікація землекористування і т.д.).

Збір даних за допомогою особистих фотографій — це керований користувачем процес, який додає унікальну інформацію та деталі до бази даних

карти. Цей метод складається з кількох кроків, кожен з яких сприяє створенню повної та детальної карти.

Користувачі починають із фотографій різних географічних об'єктів за допомогою своїх смартфонів або камер. Ці фотографії можуть включати будівлі, дорожні знаки, орієнтири, природні об'єкти або будь-які інші об'єкти інтересу, які користувачі бачать поблизу.

Багато сучасних смартфонів і цифрових камер автоматично позначають фотографії GPS-координатами, надаючи точну інформацію про місцезнаходження для кожного зображення. Це геотегування гарантує точне розміщення фотографій на карті.

Коли користувачі роблять свої фотографії, вони завантажують їх на платформу OSM або інші картографічні інструменти, які підтримують інтеграцію фотографій. Ці платформи дозволяють користувачам додавати свої фотографії безпосередньо до бази даних карт.

Після завантаження своїх фотографій користувачі можуть прив'язати їх до географічної прив'язки, вирівнявши їх із наявними об'єктами карти в OSM. Це передбачає зіставлення місця, зображеного на фотографії, з відповідним місцем на карті. Геореференція забезпечує точне розміщення фотографій у просторовому контексті карти [11].

На додаток до геоприв'язки, користувачі коментують свої фотографії додатковими деталями або описами. Це може включати назву будівлі, тип рослинності, стан дороги або будь-яку іншу важливу інформацію, яка покращує багатство та зручність використання карти.

Після завершення географічної прив'язки та анотації користувачі зберігають свої фотографії в базі даних OSM. Ці внески стають частиною набору картографічних даних, надаючи цінну візуальну інформацію про географічні об'єкти.

2.2. Структура та характеристика даних у OpenStreetMap

Є три базових типи елементів в OpenStreetMap: точка, лінія, зв'язок (рис. 3.1). Всі об'єкти без виключень в OSM описуються цими типами даних і вже після цього наповнюються різноманітним комбінуванням тегів. Модель даних базується на ієрархічній структурі посилань, з цього можна зробити висновок, що будь-який наступний тип даних не нестиме в собі інформацію, яка міститься в попередніх типах, а буде створювати нову сутність, при цьому посилаючись на об'єкти попереднього типу. Не слід забувати, що абсолютно всі об'єкти без виключень мають в структурі даних OSM свій власний ідентифікатор, тобто ID, що є унікальним в даному типі об'єктів. Саме за цим ідентифікатором і відбувається посилання на, власне, об'єкт [14].

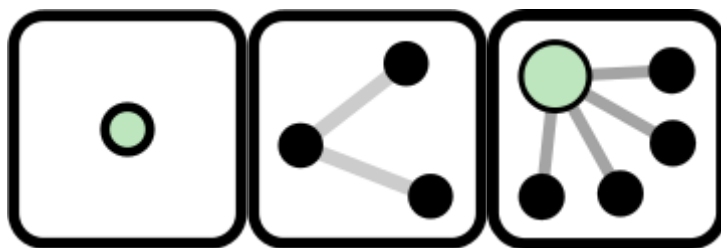


Рисунок 2.1. Базові типи елементів OSM

Точка (node) є мінімальним набором даних та базою в ієрархічній моделі, вона несе в собі інформацію про пару координат, а саме широту та довготу. Координати OSM використовуються в десятковій системі числення (base 10), оскільки їх набагато простіше обробляти, ніж хвилиний і секундний формати координат. Сама по собі точка - це або самостійний об'єкт, що описує якийсь точковий об'єкт (геометричний примітив), або, при цьому, взагалі має свої інформаційні складові. Точка — це самостійний об'єкт, який одночасно несе унікальну інформацію і може бути частиною іншого об'єкта.

Лінія (way) — складається з посилань на точкові об'єкти. Важливо, щоб вона містила хоча б одне посилання на вже існуючий об'єкт, що є точкою. Лінія з однієї точки суперечить поняттям базової геометрії і викликає незрозуміння при занадто крихких алгоритмах обробки даних, тому правильно задана лінія завжди

міститиме в собі як мінімум одне посилання на два існуючих об'єкта типу точка [14].

Порядок розташування точок на лінії важливий і характеризує послідовність розташування точок на лінії і напрямок самої лінії, тобто навіть якщо лінія замкнута. Точки можуть бути спільними для двох ліній, тобто входити в будь-яку кількість об'єктів ліній. У цьому випадку посилання на неї міститься в обох лініях. Таким чином, будується інтегральний граф об'єкта (в більшості випадків дорожній граф для розрахунку маршруту), який використовується для обчислення загальних елементів (точок) [14].

Останній тип даних — зв'язок (relation). Всі об'єкти, не рахуючи точки, вже є зв'язком, але лінії все одно виділено як окремий тип даних, тому що є більш поширеними та такими, що описують базові геометричні примітиви (полігони, полілінії та лінії). Зв'язок є універсальним типом даних і призначений для решти більш складних геометричних об'єктів, а також для таких геометричних об'єктів як списки, ієрархії взаємозв'язків та колекції.

Основна відмінність лінії від зв'язку, в цілому, полягає в тому, що лінія — це завжди набір точок, а зв'язок може бути набором як ліній, так і точок. Саме тому у зв'язках вказується не тільки ID, але і тип, до якого належить об'єкт.

Порядок перерахування членів враховується та має вагоме значення у зв'язку так само, як і в лінії при використанні даного зв'язку. Це пов'язано з тим, що зв'язок може бути не просто геометричною фігурою, а побудований, наприклад, для автомобіля маршрут. У такому разі у зв'язок вже буде входити не лише набір точок, а і ділянки доріг. Зв'язок також може бути частиною іншого зв'язку. Єдине що, зв'язок не може бути частиною самого себе, тобто мати посилання на самого себе [14].

Початковий об'єкт — це такий об'єкт, який входить до складу будь-якого іншого, але сам при цьому не містить в собі жодного об'єкта. По факту, це завжди точка. Кінцевим об'єктом вважається той, що найвищий по ієрархії і не входить до складу жодного з інших об'єктів. Це може бути і точка, і лінія, і зв'язок.

Теги являють собою пару ключ-значення. Вони використовуються для того, щоб описати кожен окремий об'єкт, який являється елементом даних. Набори тегів мають містити ключ лише один раз. Якщо не вказано ніякого конкретного тега — застосовується типове значення. Значення призначене для конкретизації об'єкта за його ключем. В більшості випадків значення являють собою довільний текст, але в деяких окремих випадках вони можуть використовуватись з переліку заздалегідь узгоджених [15].

Найбільш частою необхідною задачею для геопросторових даних є отримання графічного представлення об'єкта, що описані цими даними. Простіше кажучи, графічне представлення об'єкта — це візуалізація. Він зводиться до відмалювання геометричних примітивів, які утворюються з об'єктів трьох типів даних.

Геометричні примітиви формуються з точки, лінії, полілінії.

1. Точка: Представлена одним вузлом, точка відповідає просторовому розташуванню в географічних координатах, що перекладається на її положення на картографічній проекції. Координати (широта/довгота) переводяться в картографічні координати (x/y) з урахуванням проекції карти.
2. Лінія: Часто найкоротша відстань між двома точками, лінія відповідає об'єкту «лінія», що містить два вузлові об'єкти. Оскільки ми працюємо у плоскому просторі, відстань між будь-якими двома точками завжди є прямою лінією.
3. Полілінія: З'єднана послідовність відрізків, де кожен відрізок утворює лінію, з'єднану кінець в кінець. Вся послідовність становить єдиний об'єкт, зазвичай представлений об'єктом «лінія», що містить три або більше вузлів. Крім того, полілінія може бути об'єктом «зв'язок», що складається з послідовно включених об'єктів «лінії». Кожен наступний об'єкт починається там, де закінчується попередній.

Полігони, як підтип поліліній, представляють собою замкнені фігури, де остання точка збігається з першою. Вони можуть бути як об'єктами «шляху», що

містять декілька вузлів, так і об'єкти «relation», що включають послідовно з'єднані об'єкти «ways», які утворюють замкнутий контур. Останній вузол останнього включеного об'єкта «лінія» або «зв'язок» відповідає першому вузлу першого включеного об'єкта «лінія» [15].

Крім того, існують складні об'єкти, такі як багатокутники, які не можуть бути описані одним примітивом лінії. Вони будуються на основі об'єктів типу зв'язок, де `type=multipolygon`. На відміну від полігонів і поліліній, описаних як «лінія» або «зв'язок», складові об'єкти не можна спростити до одного об'єкта «лінія».

Таким чином, різноманітний спектр об'єктів в OSM, разом з їхньою ієрархічною організацією та унікальними ідентифікаторами, дозволяє комплексно представляти просторові дані, сприяючи багатству та універсальності платформи в картографуванні різних географічних об'єктів та структур.

Визначення того, що є «будівлею» в OpenStreetMap є комплексним, оскільки воно охоплює широкий спектр структур, окрім традиційних конструкцій. Хоча цей термін зазвичай стосується постійних структур, призначених для перебування людей, OpenStreetMap використовує більш інклюзивний підхід. На додаток до звичайних будівель, він включає такі об'єкти, як стаціонарні каравани, плаваючі будинки і, в деяких випадках, резервуари для зберігання під тегом `building`. Таке широке визначення відображає прагнення платформи відображати різноманітні просторові об'єкти, незалежно від традиційної класифікації.

При нанесенні будівель на мапу OpenStreetMap важливо створити єдиний контур для кожної будівлі, незалежно від того, чи це окремий об'єкт нерухомості, чи частина ряду терасових будинків. Для терасових будинків рекомендується створювати пов'язані контури, які мають спільні граничні вузли, щоб розрізнити кожну будівлю [16].

Ці контури можуть варіюватися від спрощених форм до детальних зображень, які точно відображають форму будівлі. Спочатку будівлі можуть бути

нанесені на карту простими груповими контурами, а пізніше уточнені більш детальними контурами або розділені на окремі об'єкти нерухомості.

Основний тег, який використовується для будівель - `building=yes`, що позначає будь-який тип будівлі. Крім того, конкретні типології можуть бути описані за допомогою таких значень, як `building=house`, `building=hut`, будівля=`garage` або `building=school`. Додаткові опції можна знайти у документації до тегу `building=*`, що дозволяє створювати нові значення, які відповідають існуючим конвенціям [16].

Важливо зазначити, що мітки будівель представляють саму фізичну структуру, а не її поточне використання чи мешканців. Ця гнучкість дозволяє створювати складні сценарії, такі як церква, розташована в багатоквартирному будинку (`building=apartments`), або апартаменти, розташовані в колишній церкві (`building=church`).

При додаванні адрес можна використовувати тег `addr=*`.

Входи до будівлі можуть бути позначені на вузлах вздовж контуру будівлі за допомогою тегів `entrance=*` і, можливо, `access=*`. Ці входи можуть бути пов'язані з описами доступу до пішохідних доріжок за допомогою тегів `highway=footway` або `highway=steps`. У деяких випадках входи можуть мати індивідуальні поштові адреси.

Методи тегування, в першу чергу, передбачають опис об'єму будівлі за допомогою двох типів площ [16]:

1. Контури будівлі: Ці контури представляють найбільш загальну область складної будівлі. Вони визначають загальну форму і площу будівлі, забезпечуючи базову основу для її об'єму і просторового планування.
2. Частини будівлі: На додаток до обрисів, частини будівлі використовуються для опису конкретних секцій будівлі, особливо тих, що мають різну висоту або інші атрибути. Ці частини пропонують більш дрібну деталізацію, що дозволяє точніше моделювати внутрішні та зовнішні особливості будівлі.

Контур будівлі охоплює загальну площу землі, яку займають усі частини будівлі, що зазвичай дорівнює площі її основи. Представлений у вигляді замкненого контуру або багатокутника з тегом `building=*`, він слугує фундаментальною формою будівлі.

Важливо, що атрибути будівлі, такі як адреса, назва, загальна висота та оператор, повинні бути позначені на самому контурі будівлі.

Крім того, контур будівлі підтримує сумісність з програмним забезпеченням для 2D-візуалізації, таким як Mapnik, та іншими споживачами даних, не орієнтованими на 3D-моделювання. Однак, якщо будівля містить будь-які області `building:part=*`, контур будівлі ігнорується для цілей 3D-візуалізації [16].

Різні секції будівлі, що відрізняються різними фізичними характеристиками, такими як висота або колір, зазвичай зображуються шляхом малювання областей у межах контуру будівлі та позначення їх тегом `building:part=*`. Хоча за замовчуванням `building:part=*` має значення `yes`, йому можна присвоїти будь-яке значення `building=*`.

Кожен сегмент контуру будівлі має бути заповнений областями `building:part=*`, кожна з яких має певну висоту та інші відповідні атрибути. Ці області можуть перекриватися або не перекриватися, в залежності від структури. Однак, в той час як перекриття 2D-об'єктів є звичайним і прийнятним, слід уникати перекриття 3D-об'єктів, особливо якщо вони мають спільні грані.

Теги атрибутів будівлі, які зазвичай застосовуються до областей, позначених тегом `building:part=*`, в основному використовуються для цілей 3D-візуалізації. Ці атрибути не враховуються 2D-візуалізаторами.

У випадках, коли будівля має складну структуру або має частини, що виходять за межі площі, можна використовувати зв'язок `type=building`, щоб організувати контур будівлі та її різні частини. Однак, якщо структура будівлі є простою, без частин, що виходять за межі площі, створення зв'язку `type=building` може бути непотрібним. У таких випадках, всі частини будівлі повинні бути розміщені в межах контуру будівлі.

Коли використовується зв'язок `type=building`, всі частини будівлі повинні бути перелічені як члени зв'язку з роллю «частина», тоді як сам контур будівлі повинен бути перелічений з роллю «контур» [17]. Частини будівлі можуть бути розташовані у будь-якій конфігурації відносно контуру будівлі у відношенні `type=building`. За відсутності зв'язку `type=building`, додатки повинні інтерпретувати всі частини будівлі в межах області контуру будівлі як невід'ємні компоненти цієї будівлі.

Теги, що використовуються для будівель та частин будівель, поділяються на три основні групи [17]:

1. Теги, призначені для позначення даху будівлі, мають префікс `roof`.
2. Теги, призначені для самої будівлі, за винятком даху (тобто фасадів), мають префікс `building`.
3. Теги, які стосуються як будівлі, так і її даху, не мають префікса.

Теги, які застосовуються як на фасадах будівель, так і на даху:

- `height=`: Вказує на відстань від найнижчої точки контакту з землею до верху даху будівлі, не враховуючи антени, шпилі та інше обладнання, встановлене на даху.
- `min_height=`: Вказує висоту під конструкцією будівлі. Важливо зазначити, що коли використовується `min_height`, висота все одно визначається як відстань від землі до верху конструкції. Наприклад, міст висотою 3 метри, де нижня частина мосту розташована на висоті 10 метрів над рівнем землі, матиме `min_height=10` і `height=13`.

Теги, що стосуються виключно фасадів будівлі під дахом:

- `building:levels=`: Вказує на кількість поверхів будівлі над рівнем землі, за винятком рівнів у даху. Цей тег допомагає точно текстурувати будівлю
- `building:min_level=`: Вказує всі пропущені рівні в частині будівлі, подібно до тегу `min_height` для загальної висоти.
- `building:material=`: Ідентифікує зовнішній матеріал, використаний для фасадів будівлі.

- `building:color=`: Визначає колір фасадів будівлі. Необхідно звернутись до параметра `color=*` для отримання можливих значень.

Важливо зазначити, що не існує спеціального тегу для висоти фасадів (`building:height`). Замість цього, це значення автоматично виводиться як загальна `height=*` мінус `roof:height=*`.

Теги для даху будівлі слугують для позначення форми даху: двосхилого, шатрового, плаского і т.д.

Теги `height`, `roof:height`, `building:levels` та `roof:levels` слугують різним цілям і мають проблеми сумісності між 2D та 3D представленнями у OpenStreetMap.

У 2D картографуванні, `building:levels` та `roof:levels` зазвичай позначають кількість поверхів або рівнів у будівлі та даху, відповідно.

У 3D-візуалізації, коли теги висот не вказані, теги `*:levels` транслюються в імітовані висоти. Зазвичай кожен поверх вважається висотою 3 метри при візуалізації.

Для кращої сумісності між 2D і 3D даними рекомендується додавати реальні висоти будівель. Це забезпечить узгодженість у представленні. Наприклад, замість десяткових значень, таких як `building:levels=1.5` та `roof:levels=0.7`, краще вказати висоту. Наприклад, `building:levels=1`, `roof:levels=0`, `height=6.6` метрів, `roof:height=2.1` метрів [17].

Фактична висота будівлі часто невідома, і для візуалізації зазвичай використовується значення за замовчуванням 3 метри на поверх. У таких випадках, як спортивні зали, де висота перевищує 3 метри за замовчуванням, вказівка фактичної висоти має вирішальне значення для точного відображення. Наприклад, `building:levels=1`, `height=6` метрів. Необхідно уникати використання фальшивих значень рівнів для імітації висот, які не відповідають реальній структурі будівлі. Використання десяткових значень може неточно відображати будівлі як у 2D, так і в 3D контекстах.

2.3. 3D-візуалізатори OpenStreetMap

Візуалізація карти передбачає перетворення необроблених геопросторових даних і тегів на візуальні зображення. Цей термін часто стосується створення растрових зображень або наборів растрових плиток, він також може включати виведення у векторних форматах. 3D-візуалізація можлива з використанням картографічних даних як вхідних. Можливість візуалізувати карти в різних стилях або підкреслювати специфічні особливості є переконливим аспектом відкритого доступу до геоданих.

Існує низка програм для візуалізації даних OpenStreetMap. Основною, та найкраще зрозумілою у використанні є F4 Map.

F4 Map [18], інтерфейс якого відображено на рис. 3.1, — це 3D-мапа, створена на основі даних OpenStreetMap за допомогою технології WebGL (API, створене для відображення 3D-графіки у браузері), що вимагає сумісного веб-браузера [19]. Ця карта пропонує 3D-види, які демонструють не лише будівлі OpenStreetMap, але й дерева та іншу просторову інформацію.



Рисунок 2.2. Інтерфейс F4 Map

Загальні теги, за допомогою яких обробляється більшість функцій F4 Map:

- colour=color description
- height=number in meters
- min_height=number in meters

Творці F4 Map надали великого значення візуалізації 3D-мапи. Затінення ґрунту досягається за допомогою тайлової графіки з використанням Mapnik

(інструментарій, призначенням якого є візуалізація карт), а також SVG (масштабованої векторної графіки) і текстур. Використання SVG та текстури покращує візуальне представлення ландшафту. Значна увага приділена деталям міста (зображено на рис 3.2) [19]:

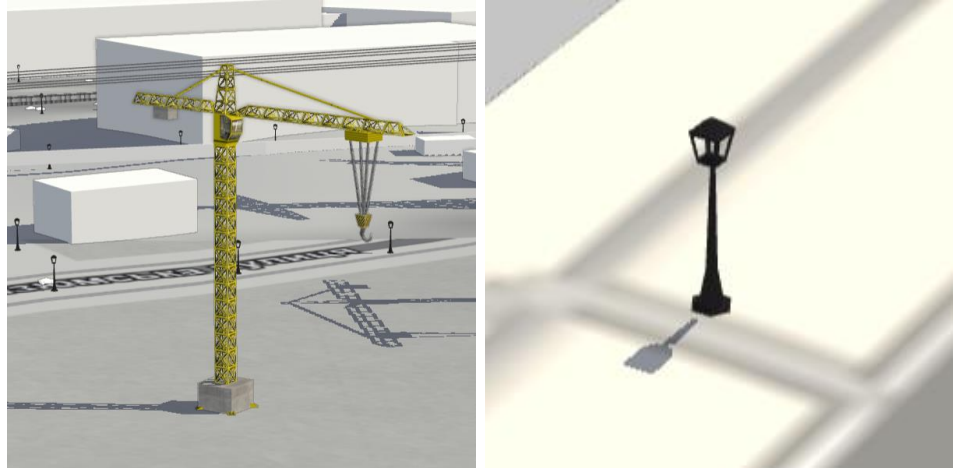


Рисунок 2.3. Приклад відображення деталей міста

1. Перешкоди та/або паркани: для `barrier=wall|hedge` генеруються широкі полігони, а для `barrier=fence` створюються текстуровані полігони.
2. Будівлі з `man_made=chimney` або `tower:type=cooling` мають ефект білого диму на даху, враховуючи поточний напрямок та швидкість вітру.
3. Окремі моделі вуличних ліхтарів створюються за допомогою точки `highway=street_lamp`, додаючи автентичності міському пейзажу.
4. Анімовані моделі вітрогенераторів розміщені на точках `power=generator` та `generator:source=wind`, зображуючи відновлювані джерела енергії.
5. Анімовані моделі виділяють точки `man_made=petroleum_well`, демонструючи промислову інфраструктуру.
6. Анімовані моделі будівельних кранів напіввипадковим чином розміщені на полігонах `landuse=construction polygons`, ілюструючи поточні проекти розвитку.
7. Лінії електропередач та другорядні лінії, представлені як `power=line|minor_line`, у стилі чорного каркасу зі спеціальними моделями точок опор ліній електропередач та опор. Тег `cables=(number)` працює на `power=line`, підвищуючи реалістичність.

8. Спеціальні моделі виконані багатокутниками і точками `man_made=lighthouse` з висотами, отриманими з тегів висоти або виведеними з тегів `seamark`.
9. Могили додаються напіввипадковим чином на полігонах `landuse=cemetery` або `amenity=grave_yard`.
10. Генератори анімованих частинок оживляють фонтани на полігонах і точках `amenity=fountain`, додаючи руху міському простору.
11. Анімовані моделі прапорів майорять на точках `golf=pin|hole` і `man_made=flagpole`, створюючи атмосферу міста.
12. Конкретні моделі човнів випадковим чином розміщуються на лініях з водним `waterway=river|canal` та `boat=yes|designated|permissive` або `route=ferry`, анімуючи водні об'єкти.
13. Спеціальні моделі повітряних шляхів додано на точки з `aerialway=pylon`, тоді як кабелі зображуються на лініях з `aerialway=cable_car|j-bar|t-bar|platter|drag_lift|gondola|chair_lift|mixed_lift`. Анімовані гондоли, додані як `aerialway=cable_car|gondola|gondola|chair_lift`, підвищують реалістичність.
14. Моделі скульптур розміщені на елементах з тегами `monument=statue`, `artwork_type|artwork:type=sculpture|statue`, з урахуванням тегів `height` та `min_height`.

Природні деталі ландшафту додано з використанням таких тегів (зображено на рис. 3.3) [19]:

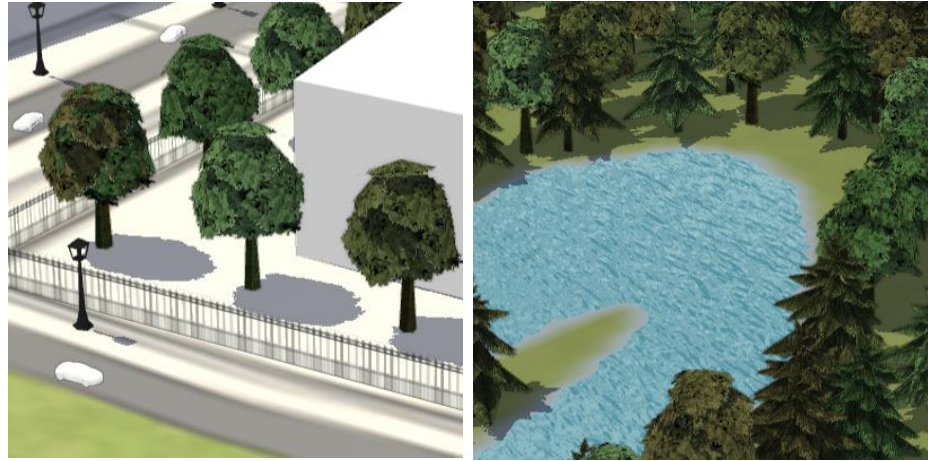


Рисунок 2.4. Приклад відображення природних деталей

1. Ефект води інтегрований у текстуру ґрунту, а напрямок хвиль при цьому є синхронізованим глобально по всьому світу.
2. Дерев, ліси та чагарники:
 - Дерев розміщуються на точках з міткою `natural=tree`, або напіввипадково вздовж ліній з міткою `natural=tree_row`. Вони також розкидані напіввипадковим чином в межах ділянок, позначених як `landuse=forest|orchard` або `natural=wood`.
 - На вибір листяних, хвойних та пальмових дерев впливає поточна широта, що забезпечує природну варіацію.
 - Кущі також розміщені напіввипадково в межах ділянок, позначених як `natural=heath|scrub`.
3. Полігони, позначені як `landuse=vineyard`, заповнюються вирівняними живоплотами випадково, що сприяє покращенню відображення сільськогосподарського ландшафту.
4. Моделі каменів розміщуються на точках, лініях і полігонах, позначених тегами `natural=stone` і `megalith_type=true`, це додає геологічних особливостей до навколишнього середовища.
5. Процес візуалізації снігу, хмар, дощу та інших погодних ефектів відбувається за допомогою API `worldweatheronline`. Дані про швидкість вітру використовуються для анімації хмар і синхронізації таких ефектів, як дим з димаря, що підвищує реалістичність.

Візуалізація будівель (зображено на рис 3.2) відбувається на основі простої специфікації 3D-будівель OpenStreetMap та реалізується відповідно до наступного переліку принципів [19]:



Рисунок 2.5. Приклад візуалізації будівлі

- Будь-яка `building:part`, яка не повністю міститься у будівлі, ігнорується. `building:part` успадковують всі теги, що пов'язані з будівлею, від так званої батьківської будівлі.
- Генерація висоти будівлі виконується напіввипадково з `way_id`, за умови, якщо не вказана конкретно за допомогою тегу `height` або `building:levels`. У якості стандарту в такому випадку використовується висота три метри на кожен поверх.
- Полігони частин будівлі, які мають перетин з полігоном контуру, видаляються, якщо тільки контур не позначено тегом `building:part=yes`.
- Контури з тегами `building:part=no` або `building:parts=vertical/horizontal` ігноруються у процесі візуалізації.

F4 Map підтримує прості форми дахів (зображено на рис. 3.3) (плаский, двосхилий, мансардний і так далі) з такими тегами: `roof:shape`, `building:roof:shape` та `building:shape`. Дахи будівель з тегами `man_made=chimney` або `tower:type=carendered` візуалізуються з використанням гіперболоїдної форми для точного відображення їхньої структури та зовнішнього вигляду. Якщо тег даху вказано `roof:orientation=along/across`, застосовується оптимально орієнтована обмежувальна рамка, щоб відповідно визначити орієнтацію хребта [19]. Для частин будівлі з тегами `building:part=steps` та `roof:shape=skillion`,

застосовується скілонна форму даху, щоб врахувати наявність сходинок. Якщо висота сходинок не вказана явно, за замовчуванням, використовується висота 16 см. Коли висоту даху не зазначено, вона буде згенерована і прийнята за 4 метри.

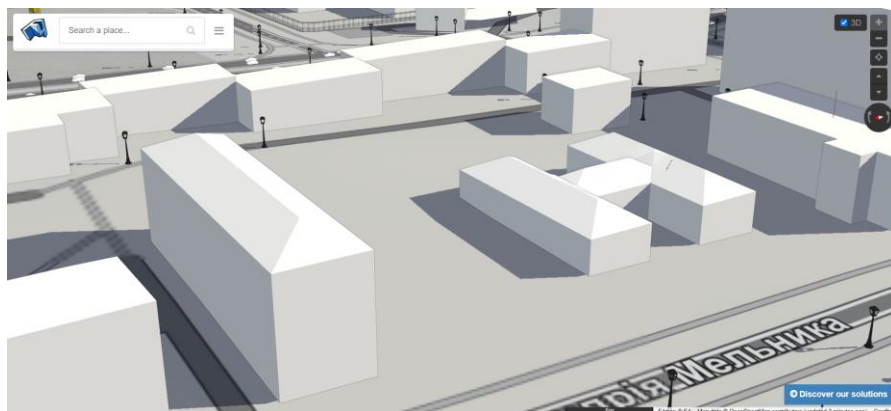


Рисунок 2.6. Візуалізація чотирисхилого даху

Висота стін розраховується як загальна висота будівлі (height) мінус висота даху (roof:height). Це забезпечує пропорційність стін загальній висоті будівлі та висоті даху. Якщо тег будівлі вказано як дах (building=roof) або wall=no walls, що вказує на те, що об'єкт є або лише дахом, або не має стін, то стіни не будуть створені як частина процесу візуалізації.

У F4 Map є недоліки, які можуть ускладнити, або унеможливити роботу. Так як візуалізатор покладається на відкриті джерела геопросторових даних, 3D-візуалізація може бути недостатньо точною, адже є області мапи без детальних даних. Оновлення даних OSM відображаються не одразу, що також може призвести до розбіжності даних [11]. Для візуалізації 3D-мапи в реальному часі необхідно мати апаратне забезпечення та швидкий інтернет, оскільки це досить ресурсозатратний процес. Відповідно, якщо пристрій буде недостатньо потужний, або ж нестабільне інтернет-з'єднання, буде сильно страждати плавна навігація. З даною проблемою стикнулась я особисто. Ще одним недоліком є обмеженість функцій. Наприклад, якщо мета — візуалізувати будинок з усіма його текстурами (такими як матеріал фасаду чи даху), зробити цього не вийде, так як F4 Map не в повній мірі підтримує текстури.

Попри всі недоліки, F4 Map все-таки можна використовувати для 3D-моделювання міст, оскільки він є зручним у використанні. Переглядати мапу

можна без плагінів і не лише на комп'ютері, а і на мобільних пристроях. Є можливість редагувати типи вмісту для відображення саме тих елементів, які необхідні. Візуалізатор з максимальною детальністю відображає геодані, включаючи дорожні мережі, визначні місця, громадський транспорт та ін.

РОЗДІЛ 3. АНАЛІЗ МОЖЛИВОСТІ ВИКОРИСТАННЯ ДАНИХ OPENSTREETMAP ДЛЯ 3D-МОДЕЛЮВАННЯ МІСТ

З метою оцінки можливості використання даних OpenStreetMap для 3D-моделювання у даній роботі було проведено статистичне дослідження наявності тегів, які описують 3D-характеристики будівель, у розрізі регіонів України. Для цього за допомогою вебсервіса Overpass Turbo [20] було пораховано кількість будівель у кожному регіоні, а також кількість будівель, що мають відповідні теги. На основі отриманих результатів було побудовано карти. Розглянемо усі етапи детально.

3.1. Вебсервіс Overpass Turbo

Overpass Turbo — це вебсервіс, розроблений спеціально для OpenStreetMap, який призначений для виконання будь-якого запиту Overpass API і виводить результат пошуку на інтерактивну мапу. Overpass API, в свою чергу, працює як онлайн база даних, тобто клієнт надсилає запит до API і отримує дані, що відповідають даному запиту.

Перед початком створення запитів до даного сервісу з метою збору статистичних даних необхідно визначити, які саме теги використовуються для побудови елементарної 3D-моделі будівлі у OpenStreetMap.

Першим важливим тегом є `building`. Він призначений для позначення території як будівлі і є відправною точкою для створення 3D-моделі. Без нього створення моделі унеможлиблюється.

Другим важливим тегом є `building:levels`. Він задає кількість поверхів будівлі, що дозволяє точно зазначити її вертикальний розмір. Це дає змогу переконатись, що висота 3D-моделі відповідає її аналогу на місцевості. Вертикальний розмір — це, по суті, базова інформація, необхідна для візуалізації.

Третій тег — `roof:shape`, який застосовується для опису форми даху будівлі. Його необхідність полягає в різноманітті та ідентифікованості профілю будівлі та у її відображенні з урахуванням форми та орієнтації.

Не менш важливим тегом є `building:material`, оскільки він є вирішальним у представленні матеріалу будівлі. З використанням даного тегу можна побудувати більш точну та візуально зрозумілішу 3D-модель будівлі.

Для того, щоб отримати дані про будівлі з відповідними тегами, необхідно створити запит до бази OpenStreetMap за допомогою Overpass Query Language.

Для зручності опрацювання даних було створено запит, який надаватиме ці дані одразу по всіх областях, а не по кожній окремо. Дані про АР Крим враховувати не будемо.

Важливо також зазначити, що будівлі у OpenStreetMap представлені такими типами даних як лінія та зв'язок [14]. За допомогою останнього відображено менше 1% будівель, тому у зборі та підрахунку даних ми його не враховуємо. Крім того, типом «зв'язок» часто вказується не просто будівля, а іноді якась її частина, що відрізняється за своїми характеристиками від основної (наприклад, має меншу/більшу кількість поверхів чи інший тип даху), це може додати певних неточностей до отриманих даних, що є ще одною причиною прийняти рішення не враховувати даний тип даних.

Створення запиту до OpenStreetMap складається з визначення області, для якої ми хочемо отримати дані (місто, регіон або країна); вибору тегу, за яким буде здійснюватися підрахунок.

Розглянемо здійснення запиту, який шукатиме загальну кількість будівель на прикладі міста Київ. Спочатку ми виконуємо запит, який знаходитиме область під назвою «Київ» у базі даних. Ось приклад: `area[name="Київ"]->.kyiv;`. Для зручності в подальшому використанні, області присвоєно назву «`kyiv`». Далі ми шукаємо всі будівлі в обраній області, використовуючи тип даних «лінії»: `way(area.kyiv)["building"]`. Після чого підраховуємо кількість знайдених об'єктів за допомогою команди «`out count`».

У результаті виконаної роботи із запитом було створено код:

```

[out:json];
(
  area[name="Київ"]->.kyiv;
  area[name="Київська область"]->.kyivska;
  area[name="Вінницька область"]->.vinnytska;
  area[name="Волинська область"]->.volynska;
  area[name="Дніпропетровська область"]->.dniproperovska;
  area[name="Донецька область"]->.donetska;
  area[name="Житомирська область"]->.zhytomyrska;
  area[name="Закарпатська область"]->.zakarpatska;
  area[name="Запорізька область"]->.zaporizhka;
  area[name="Івано-Франківська область"]->.ivano_frankivska;
  area[name="Кіровоградська область"]->.kirovohradska;
  area[name="Луганська область"]->.luhanska;
  area[name="Львівська область"]->.lvivska;
  area[name="Миколаївська область"]->.mykolaivska;
  area[name="Одеська область"]->.odeska;
  area[name="Полтавська область"]->.poltavska;
  area[name="Рівненська область"]->.rivnenska;
  area[name="Сумська область"]->.sumska;
  area[name="Тернопільська область"]->.ternopilka;
  area[name="Харківська область"]->.harkivska;
  area[name="Херсонська область"]->.hersonska;
  area[name="Хмельницька область"]->.khmelnytska;
  area[name="Черкаська область"]->.cherkaska;
  area[name="Чернівецька область"]->.chernivetska;
  area[name="Чернігівська область"]->.chernihivska;
);
(
  way(area.kyiv) ["building"];
);
out count;
(
  way(area.kyivska) ["building"];
);

```

```
out count;
(
    way(area.vinnytska) ["building"];
);
out count;
(
    way(area.volynska) ["building"];
);
out count;
(
    way(area.dnipropetrovska) ["building"];
);
out count;
(
    way(area.donetska) ["building"];
);
out count;
(
    way(area.zhytomyrska) ["building"];
);
out count;
(
    way(area.zakarpatska) ["building"];
);
out count;
(
    way(area.zaporizhka) ["building"];
);
out count;
(
    way(area.ivano_frankivska) ["building"];
);
out count;
(
    way(area.kirovohradska) ["building"];
);
```

```
);  
out count;  
(  
    way(area.luhanska) ["building"];  
);  
out count;  
(  
    way(area.lvivska) ["building"];  
);  
out count;  
(  
    way(area.mykolaiivska) ["building"];  
);  
out count;  
(  
    way(area.odeska) ["building"];  
);  
out count;  
(  
    way(area.poltavska) ["building"];  
);  
out count;  
(  
    way(area.rivnenska) ["building"];  
);  
out count;  
(  
    way(area.symska) ["building"];  
);  
out count;  
(  
    way(area.ternopilska) ["building"];  
);  
out count;  
(
```



```
    way(area.harkivska) ["building"];
);
out count;
(
    way(area.hersonska) ["building"];
);
out count;
(
    way(area.khmelnytska) ["building"];
);
out count;
(
    way(area.cherkaska) ["building"];
);
out count;
(
    way(area.chernivetska) ["building"];
);
out count;
(
    way(area.chernihivska) ["building"];
);
out count;
```

Після обробки мого запиту Overpass Turbo видає такий результат (його скорочено до відображення даних лише про перші 3 області):

```
{
  "type": "count",
  "id": 0,
  "tags": {
    "nodes": "0",
    "ways": "92240",
    "relations": "0",
    "areas": "0",
    "total": "92240"
  }
}
```

```

},
{
  "type": "count",
  "id": 0,
  "tags": {
    "nodes": "0",
    "ways": "436636",
    "relations": "0",
    "total": "436636"
  }
},
{
  "type": "count",
  "id": 0,
  "tags": {
    "nodes": "0",
    "ways": "203173",
    "relations": "0",
    "total": "203173"
  }
}

```

Після отримання даних про загальну кількість будівель у кожній з областей України приступаємо до визначення кількості будівель, для якої вказано дані про кількість поверхів, тип даху та матеріал фасаду. Для цього використовуватимемо запити, схожі до наведених вище. Визначення областей для дослідження залишається незмінним, оскільки ми продовжуємо аналізувати дані про Україну. Раніше ми використовували загальний тег, який відображає кількість усіх будинків, без будь-яких уточнюючих даних. Щоб додати ці уточнюючі дані, потрібно дописати для запита ще одну умову. Так, для того, щоб отримати дані про кількість будинків з указаною кількістю поверхів, необхідно до запиту додати тег `building:levels`. Тоді запит набуде такого вигляду: `way(area.kyiv)["building"]["building:levels"]`.

Аналогічним чином створюємо запити для визначення кількості будинків з указаним типом даху: `way(area.kyiv)["building"]["roof:shape"]`.

Останній крок — визначення кількості будівель, у яких вказано матеріал, з якого вони побудовані: `way(area.kyiv)["building"]["building:material"]`.

3.2. Аналіз даних, отриманих за допомогою вебсервісу Overpass Turbo

Завдяки вебсервісу Overpass Turbo [20] було отримано всі необхідні дані для того, щоб здійснити аналіз наповненості мапи України у OpenStreetMap станом на 20.05.2024 р.

Дані щодо кількості поверхових будинків, заданих тегом `building:levels`, у розрізі регіонів України представлено у таблиці 3.1 та у додатку А.

Таблиця 3.1

Кількість будинків з тегом `building:levels`

Область	Загальна кількість будинків	Будинки з указаною кількістю поверхів
Київ	92240	38055
Київська	436636	19164
Вінницька	203173	9229
Волинська	176938	25432
Дніпропетровська	464424	41969
Донецька	505884	184167
Житомирська	126956	25587
Закарпатська	310844	9081
Запорізька	156134	20219
Івано-Франківська	361322	7710
Кіровоградська	114890	5086
Луганська	331093	244896
Львівська	473823	39745
Миколаївська	141511	13044
Одеська	173876	22738
Полтавська	208168	33751
Рівненська	200439	30741
Сумська	125416	11553
Тернопільська	229691	7073
Харківська	507129	84549
Херсонська	68089	8118
Хмельницька	263862	37945
Черкаська	215051	10174
Чернівецька	270252	41347

Чернігівська	182596	6524
Автономна Республіка Крим	—	—

Проаналізувавши отриману таблицю, можна зазначити, що найбільше вказано поверхів у Луганській області — 73,97%, найменше у Закарпатській — 2,92%, середнє значення має місто Київ — 41,26%. Можна зробити висновок, що саме Луганська область є найбільш потенційно придатною для візуалізації її 3D-моделі. Для решти областей необхідно провести великий об'єм робіт з додавання тегів.

Дані щодо кількості будинків з указаним типом даху, що задані тегом `building:levels` представлено у таблиці 3.2, додатках Б і В.

Таблиця 3.2

Кількість будинків з тегом `roof:shape`

Область	Загальна кількість будинків	Будинки з указаним типом даху
Київ	92240	4288
Київська	436636	1064
Вінницька	203173	436
Волинська	176938	11260
Дніпропетровська	464424	8826
Донецька	505884	121500
Житомирська	126956	274
Закарпатська	310844	1968
Запорізька	156134	1490
Івано-Франківська	361322	828
Кіровоградська	114890	776
Луганська	331093	864
Львівська	473823	3144
Миколаївська	141511	385
Одеська	173876	898
Полтавська	208168	246
Рівненська	200439	1013
Сумська	125416	2125
Тернопільська	229691	547
Харківська	507129	983
Херсонська	68089	84
Хмельницька	263862	476
Черкаська	215051	1836
Чернівецька	270252	34618
Чернігівська	182596	118

Автономна Республіка Крим	—	—
------------------------------	---	---

Інформація щодо тегу roof:shape відрізняються від інформації про building:levels. У даному випадку областю з найбільшим показником вказаності тегу є Донецька — 24,02%, з найменшим — Чернігівська, там тег вказано лише для 0,06% будівель. Середній показник має Чернівецька область — 12,81%. За даним тегом жодна з областей не може повною мірою візуалізуватись, оскільки даних занадто мало.

Дані щодо кількості будинків з указаним матеріалом фасаду, що задані тегом building:material, представлено у таблиці 3.3, додатках Г і Д.

Таблиця 3.3

Кількість будинків з тегом building:material

Область	Загальна кількість будинків	Будинки з указаним матеріалом фасаду
Київ	92240	2805
Київська	436636	246
Вінницька	203173	146
Волинська	176938	44
Дніпропетровська	464424	896
Донецька	505884	9258
Житомирська	126956	693
Закарпатська	310844	21
Запорізька	156134	46
Івано-Франківська	361322	69
Кіровоградська	114890	28
Луганська	331093	26
Львівська	473823	257
Миколаївська	141511	137
Одеська	173876	74
Полтавська	208168	73
Рівненська	200439	777
Сумська	125416	70
Тернопільська	229691	49
Харківська	507129	64
Херсонська	68089	5
Хмельницька	263862	127
Черкаська	215051	489
Чернівецька	270252	15
Чернігівська	182596	56

Автономна Республіка Крим	—	—
------------------------------	---	---

Тег `building:material` вказаний для настільки малої кількості областей, що лише для міста Київ значення складає хоча б 3%. Для решти областей відсоток кількості будинків з указаним матеріалом фасаду коливається від 0,01% до 1,8%. Це вказує на те, що користувачі OpenStreetMap у дуже рідкісних випадках вказують для будівель тег `building:material`. Через це значно знижується точність та наочність візуалізації 3D-моделей будівель.

На основі результатів, отриманих за допомогою створення запитів до Overpass Turbo було створено відповідні тематичні карти. Для цього використано програмне забезпечення ArcGIS Pro.

Створення просторових шарів мапи виконувалося у форматі шейп-файлів, оскільки він є універсальним. Після чого вище вказані таблиці було підключено до таблиці атрибутів карти. Далі було проведено підготовку даних до візуалізації.

Процес створення статистичних картосхем складався зі змістовного наповнення та компоновання. Під час змістовного наповнення було обрано способи зображення і, власне, тематика карт. Відбулось налаштування шарів та стилів карти для передачі необхідної інформації.

Компоновання карти проводилось аби створити віртуальну картографічну модель, тобто кінцеве зображення карти. Першим кроком було обрання розміру та формату карти, далі було додано назву, легенду, масштаб та формат.

Метою побудови карт є наглядне відображення статистичних даних з таблиць для того, щоб їх було легше візуально сприймати, аналізувати та робити відповідні висновки.

Завдяки візуальному зображенню статистичних карт можна точно визначити, у якому регіоні України будівель із вказаним певним тегом більше, а в якому менше. Також можна зробити висновок про загальну ситуацію щодо будівель з вказаними тегами в розрізі регіонів України.

ВИСНОВКИ

У кваліфікаційній роботі шляхом опрацювання даних з OpenStreetMap було проведено аналіз можливості їх використання для 3D-моделювання міського середовища.

1. За допомогою запитів Overpass Turbo отримано дані про наявність в базі даних OpenStreetMap тегів, що описують поверховість будинків, форму їхнього даху та матеріал фасаду по регіонах України.
2. Побудовано карти розподілу за регіонами України відсотку наявності уточнювальних тегів у будівель, які необхідні для 3D-моделювання міст.
3. Найбільший відсоток вказаної кількості поверхів спостерігається у Луганській області (73,97%), а найменший — у Закарпатській (2,92%). Найбільший відсоток вказаного типу даху будівлі — 24,02% для Донецької області. Найбільший відсоток вказаного типу матеріалу фасаду — 3,04% для міста Києва.
4. Станом на сьогодні дані OpenStreetMap можуть використовуватись для 3D-моделювання міст в деяких регіонах України для задач, у яких достатньо інформації лише про кількість поверхів та контури будівель в плані.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Mark Keil J. Polygon decomposition. *Handbook of computational geometry*. 2000. P. 491–518. URL: <https://doi.org/10.1016/b978-044482537-7/50012-7> (дата звернення: 22.05.2024).
2. A. Borrmann, M König, C Koch, J Beetz. Building information modeling: Why? What? How?. *Building information modeling*. Cham, 2018. P. 1–24. URL: https://doi.org/10.1007/978-3-319-92862-3_1 (дата звернення: 24.05.2024).
3. Kensek K. M. Building information modeling. Taylor & Francis Group, London, 2014. 312p.
4. Hardin B., McCool D. BIM and construction management: proven tools, methods, and workflows. Wiley & Sons, Incorporated, John, 2015. 416 p.
5. Braun A, Tuttas S, Stilla U, Borrmann A. Building information modeling: technology foundations and industry practice. Springer, 2018. 609 p.
6. Mahamadu A.-M., Mahdjoubi L., Booth C. Challenges to bim-cloud integration: implication of security issues on secure collaboration. *2013 IEEE 5th international conference on cloud computing technology and science (cloudcom)*, Bristol, United Kingdom, 2–5 December 2013. 2013. URL: <https://doi.org/10.1109/cloudcom.2013.127> (дата звернення: 24.05.2024).
7. ASPRS – IMAGING AND GEOSPATIAL SOCIETY. URL: <https://www.asprs.org/a/society/about.html> (дата звернення 25.05.2024р.).
8. Lowe D. G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*. 2004. Vol. 60, no. 2. P. 91–110. URL: <https://doi.org/10.1023/b:visi.0000029664.99615.94> (дата звернення 25.05.2024р.).
9. History of 3D Rendering Explained: Where It All Began. URL: <https://visengine.com/3d-rendering-where-it-all-began/> (дата звернення 02.05.2024р.).

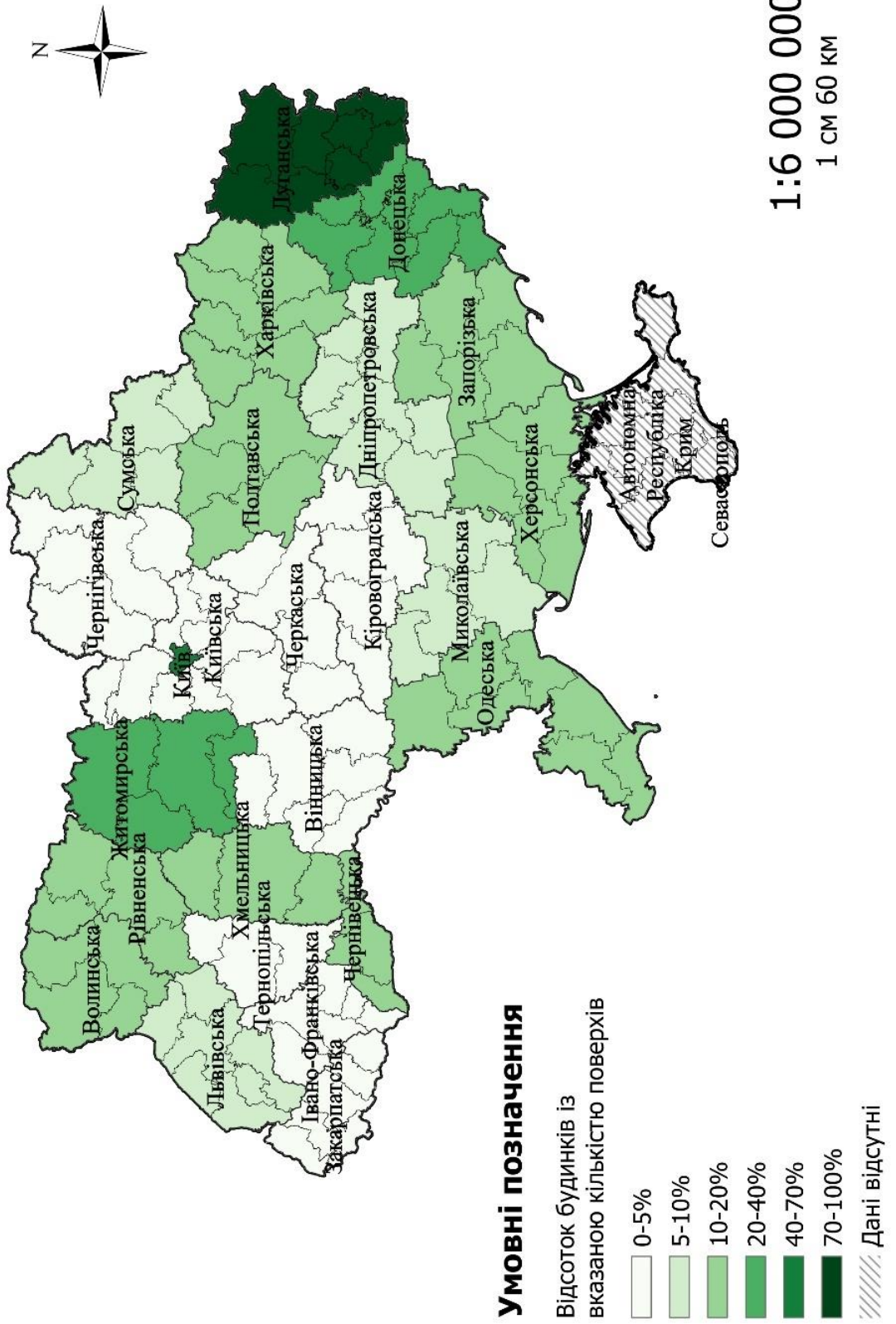
10. A. Kukko. Mobile Laser Scanning—System development, performance and applications. Finnish Geodetic Institute, Kirkkonummi, 2013. 247p.
11. Bennett J. OpenStreetMap: be your own cartographer. Birmingham, U.K. : Packt Pub., 2010. 234 p.
12. Gingold Y., Igarashi T., Zorin D. Structured annotations for 2D-to-3D modeling. *ACM transactions on graphics*. 2009. Vol. 28, no. 5. P. 1–9. URL: <https://doi.org/10.1145/1618452.1618494> (дата звернення: 24.05.2024).
13. Kalkan E., Okur F. Y., Altunışık A. C. Applications and usability of parametric modeling. *Journal of construction engineering, management & innovation*. 2018. Vol. 1, no. 3. URL: <https://doi.org/10.31462/jcemi.2018.03139146> (дата звернення: 24.05.2024).
14. P Mooney, M Minghini. A review of OpenStreetMap data. *Mapping and the citizen sensor*. 2017. P. 37–59. URL: <https://doi.org/10.5334/bbf.c> (дата звернення: 27.05.2024).
15. N Davidovic, P Mooney, L Stoimenov, M Minghini. Tagging in volunteered geographic information: an analysis of tagging practices for cities and urban regions in OpenStreetMap. *ISPRS international journal of geo-information*. 2016. Vol. 5, no. 12. P. 232. URL: <https://doi.org/10.3390/ijgi5120232> (дата звернення: 27.05.2024).
16. Hecht R., Kunze C., Hahmann S. Measuring completeness of building footprints in OpenStreetMap over space and time. *ISPRS international journal of geo-information*. 2013. Vol. 2, no. 4. P. 1066–1091. URL: <https://doi.org/10.3390/ijgi2041066> (дата звернення: 27.05.2024).
17. Simple 3D Buildings - OpenStreetMap Wiki. *OpenStreetMap Wiki*. URL: https://wiki.openstreetmap.org/wiki/Simple_3D_Buildings#Tagging_for_building_outlines_and_parts (дата звернення: 30.05.2024).
18. F4map Demo - Interactive 3D map. *F4map Demo - Interactive 3D map*. URL:

<https://demo.f4map.com/#lat=50.4474103&lon=30.3847060&zoom=17&camera.phi=50.42> (дата звернення: 30.05.2024).

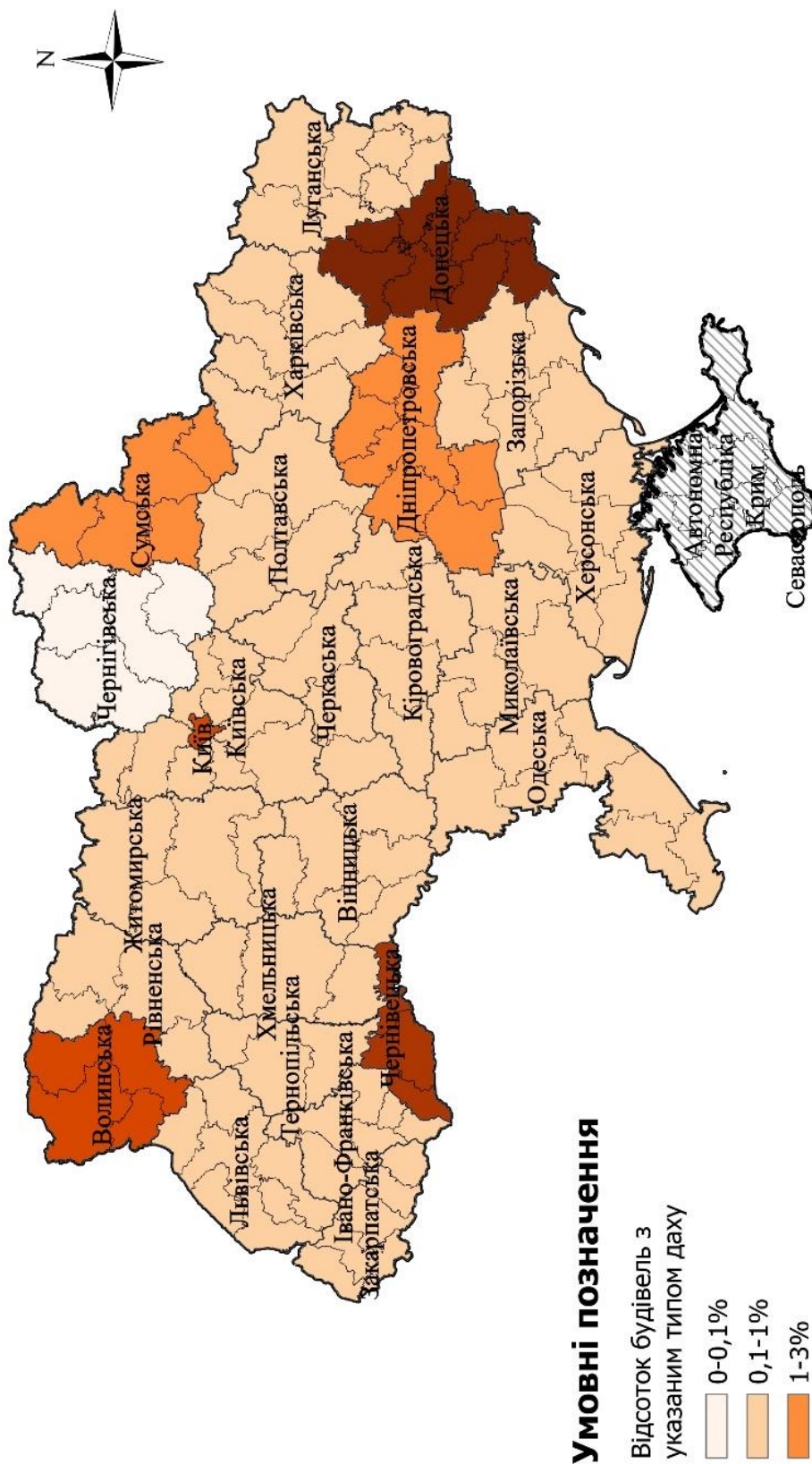
19. F4 Map - OpenStreetMap Wiki. *OpenStreetMap Wiki*. URL: https://wiki.openstreetmap.org/wiki/F4_Map (дата звернення: 30.05.2024).

20. Overpass turbo. *overpass turbo*. URL: <https://overpass-turbo.eu/> (дата звернення: 30.05.2024).

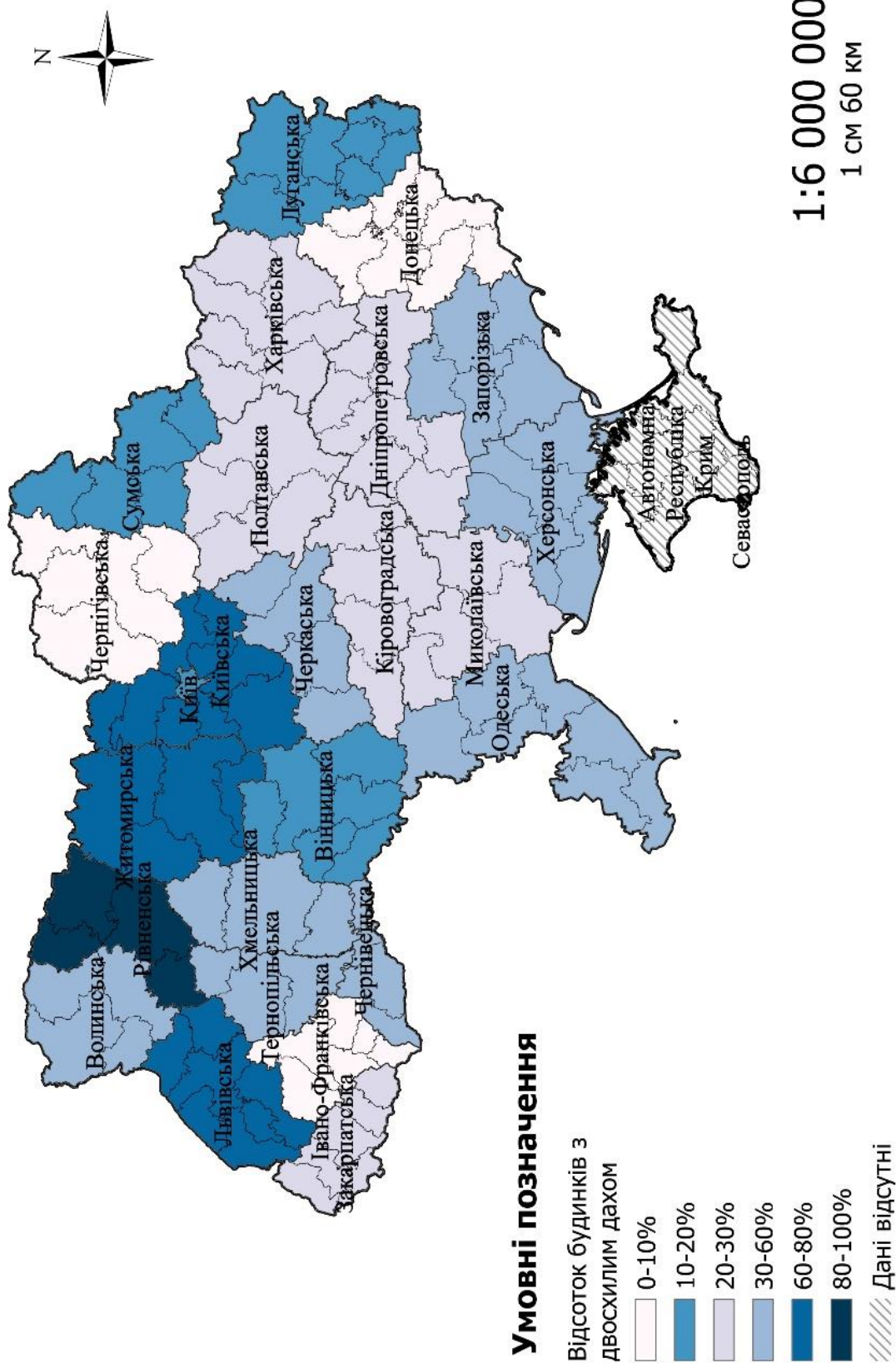
Будинки з вказаною кількістю поверхів у OpenStreetMap



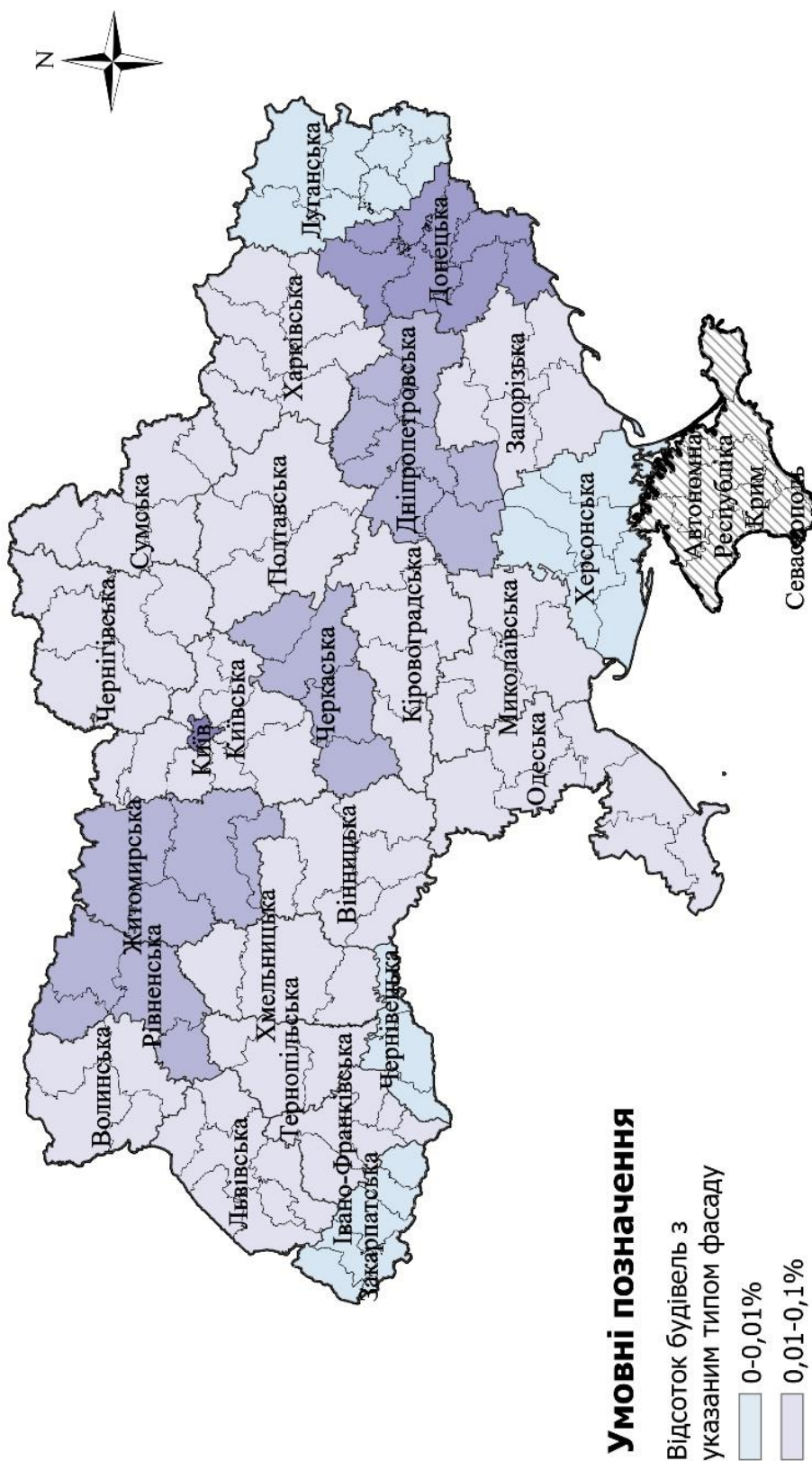
Будинки з указаним типом даху у OpenStreetMap



Будинки з двохилим типом даху у OpenStreetMap



Будинки з указаним типом фасаду у OpenStreetMap



Будинки з цегляним фасадом у OpenStreetMap

