

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет: Геоінформаційних систем та управління територіями

Кафедра: Геоінформатики і фотограмметрії

Освітній рівень: «магістр за ОПП»

Спеціальність: 193 «Геодезія та землеустрій»

Спеціалізація: Геоінформаційні системи і технології

**ЗАТВЕРДЖУЮ**

Декан факультету

доцент., к.т.н. Нестерненко О. В.

“\_3\_” грудня 2025 року

**ЗАВДАННЯ  
ДО ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ  
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА**

Демченко Марія Анатоліївна

(прізвище, ім'я, по батькові)

**1. Тема роботи:** Розробка методів автоматизованої обробки LiDAR-даних для геопросторового картографування

затверджена наказом ректора КНУБА № 1844/22/25 від 31.10.2025

**2. Керівник роботи** доц., к.т.н. Горковчук Юлія Вікторівна

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

**3. Строк подання студентом роботи до захисту:** 4.12.2025

**4. Зміст пояснювальної записки за розділами:**

Р. 1. Теоретичні основи та сучасні підходи до обробки LiDAR- даних.

Р. 2. Методичні засади топографічного картографування.

Р. 3. Розроблення методики автоматизованої обробки LiDAR-даних для топографічного картографування.

Р.4 Дослідна реалізація засобів автоматизації для обробки LiDAR-даних.

**5. Графічний матеріал за розділами**

Р. 1. 1.1 Принципу роботи LiDAR. Таблиця 1.1 Атрибути лідарного знімання. Таблиця 1.2 Порівняльна таблиця ПЗ для перегляду, аналізу та ручної обробки.

Р. 2 2.1 Рис. 2.1 Загальна технологічна схема створення цифрової карти

Р. 3. Рис. 3.1 Етапи цифрового картографування на основі LiDAR. Рис.3.2 Алгоритм плагіну.Рис.3.3 Приклади люків.Рис.3.4 Інтерфейс інструменту «Slip».Рис.3.5 Результат роботи функції.Рис.3.6 Функція яка прописується для переведення з CSV в JSON.Рис. 3.7 Завантаження JSON-бібліотеи шаблонів люків із каталогу плагіна.Рис.3.8 Функція зіставлення середнього кольору з шаблоном.Рис.3.9 Поле «class» та перевірку кольору.Рис. 3.10 Колірна сегментація по бібліотеці люків.Рис. 3.11 «Паспорт» плагіну «AutoVec».Рис. 3.12 Результат роботи плагіну «QGIS Plugin Builder».Рис.3.13 Початок роботи над плагіном в Qt Designer.Рис.3. 14 Клас діалогового вікна, який керує інтерфейсом та запускає векторизацію.Рис. 3.15 Заповнення полів вибору формату (comboBox\_2).Рис.3.16 Підключення обробників кнопок/подій.Рис. 3.17 Завантаження останнього використаного CRS.Рис. 3.18 Встановлення лімітів за замовчуванням.Рис.3.19 Відкриває діалог для вибору вхідної хмари точок (LAS/LAZ/XYZ).Рис.3.20 Відкриває діалог для вибору шляху та імені вихідного файлу.Рис.3.21 Визначає драйвер OGR на основі обраного формату.Рис. 3.22 Основна функція, яка збирає параметри та запускає алгоритм.Рис. 3.23 Фінальна версія користувацького інтерфейсу.Рис.3.24 Архітектура плагіну «AutoVec».Рис. 3.25 Функція PDAL-пайплайн на основі JSON-шаблону,що використовується для кольорокової фільтрації та попередньої.Рис.3.26 JSON-шаблон задає логіку фільтрації.

Р. 4. Рис.4.1 Автоматично створений векторний шар, після плагіну .Рис.4.2 Атрибутивна таблиця векторного шару після плагіну. Рис.4.3 Відображення векторного шару після плагіну. Рис.4.4 Користувацький інтерфейс плагіну. Рис.4.5 Поле «Cloud(LAS/LAZ/XYZ)» Кнопка «Choose cloud". Рис.4.6 Воксельна фільтрація і кластеризація точок для автоматичної векторизації. Рис.4.7 Вибір вихідного шару автоматичної векторизації.Рис. 4.8 Поле “Output file”.Рис.4.9 Поле вибору Ск.Рис.4.10 Кнопка «Run».

Рис.4.11 Запуск плагіну «AutoVec». Рис.4.12 Впливаюче вікно яке з'являється після завершення автоматичної векторизації.Рис.4.13 Інтерфейс плагіну.Рис.4.14 Атрибутивна таблиця векторного шару створеного після плагіну.Рис.4.15 Приклад автоматичної векторизації люків після плагіну.Рис.4.16 Приклад векторизації смітників.Рис.4.17 Приклад векторизації поодиноких дерев.Рис.4.18 Приклад векторизації висотних точок.Рис.4.19 Приклад векторизації дренажів.Рис.4.20 Приклад векторизації бордюрів.Рис.4.21 Приклад векторизації кущів.Рис.4.22 Приклад векторизації будинків.Рис.4.23 Приклад векторизації плитки.Рис.4.24 Приклад векторизації сходів.Рис.4.25 Топографічний план міста Ставангер масштабу 1:500.Рис.4.26 Ручна векторизація та автоматична.Рис.4.27 «Hatches\_classic» (ручна векторизація), «Hatches» (автоматична векторизація).Рис. 4.28 Налаштування для інструменту «Matrix distance».Рис.4.29 «InputID» - номер люка знайденого плагіном, «TargetId» - найближчий люк, «Distance» - Відстань між ними.Таблиця4.1Результати оцінки плагіну «AutoVec»

## 6. Календарний план виконання роботи:

Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту ( роботи )
Розділ 1 1.1. Сутність та принципи технології LiDAR.1.2 Огляд сучасних нструментів та програмних рішень для оброблення LiDAR-даних Висновки до розділу 1	13.10.2025
Розділ 2 2.1. Методика цифрового топографічного картографування 2.2 Використання LiDAR-даних у топографічному картографуванні Висновки до розділу 2	27.10.2025
Розділ 3. 3.1. Методика цифрового топографічного картографування на основі LiDAR-даних 3.2. Розроблення засобів автоматизації для обробки LiDAR-даних. Висновки до розділу 3.	10.11.2025
Розділ 4. 4.1 Програмна реалізація плагіну з автоматизації. 4.2Створення топографічного плану М1:500 на територію. 4.3 Оцінка якості. Висновки до розділу 4.	24.11.2025
Остаточне оформлення роботи	04.12.2025
Направлення роботи на рецензування, перевірку на плагіат	08.12.2025
Попередній захист роботи на кафедрі	

## 7. Консультанти розділів атестаційної випускної роботи

Розділ	Прізвище, ініціали та посада консультанта	Перевірів	
		дата	підпис
Розділ 1.	Горковчук Ю.В.		
Розділ 2.	Горковчук Ю.В.		
Розділ 3.	Горковчук Ю.В.		
Розділ 4.	Горковчук Ю.В.		

**8. Дата видачі завдання:** 13.10.2025

Зав. кафедри \_\_\_\_\_ Карпінський Ю.О.  
(підпис) (прізвище та ініціали)

Керівник \_\_\_\_\_ Горковчук Ю.В.  
(підпис) (прізвище та ініціали)

Студент \_\_\_\_\_ Демченко М.А.  
(підпис) (прізвище та ініціали)

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет геоінформаційних систем і управління територіями

Кафедра геоінформатики і фотограмметрії

**ПОЯСНЮВАЛЬНА ЗАПИСКА  
ДО АТЕСТАЦІЙНОЇ РОБОТИ  
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА**

на тему:

Розробка методів автоматизованої обробки LiDAR-даних для  
геопросторового картографування

Демченко Марія Анатоліївна

Київ – 2025 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет геоінформаційних систем і управління територіями

Кафедра геоінформатики і фотограмметрії

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

\_\_проф., д.т.н. Юрій КАРПІНСЬКИЙ

“3” грудня 2025 року

**ПОЯСНЮВАЛЬНА ЗАПИСКА  
ДО АТЕСТАЦІЙНОЇ РОБОТИ  
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА**

Розробка методів автоматизованої обробки LiDAR-даних для  
геопросторового картографування

Виконала студентка групи ГСТм-24  
193 «Геодезія та землеустрій»  
Геоінформаційні системи і технології  
Демченко М.А.

Керівник: Горковчук Ю.В., доц., к.т.н.

Київ-2025 р.

РЕЗЮМЕ (summary)		<b>Демченко Марія Анатлоліївна</b>	
до атестаційної випускної роботи студента:			
<i>ЗВО</i>	Київський національний університет будівництва і архітектури		
<i>Тема</i>	Розробка методів автоматизованої обробки LiDAR-даних для геопросторового картографування		
<i>Освітній ступінь</i>	Магістр за освітньо-науковою програмою навчання		
<i>Факультет</i>	Геоінформаційних систем та управління територіями		
<i>Кафедра</i>	Геоінформатики та фотограмметрії		
<i>Спеціальність</i>	193 Геодезія та землеустрій		
<i>Керівник</i>	Горковчук Юлія Вікторівна доц. к.т.н.		
<i>Обсяг роботи</i>	<i>пояснювальна записка</i>	<i>розділів</i>	<i>креслень формату А1</i>
	78	4	0
<i>Розділ 1</i>	У розділі розглянуто теоретичні основи технології LiDAR та принципи формування хмар точок як базового джерела просторових даних. Проаналізовано атрибутивну структуру LiDAR-даних, їх переваги та обмеження у топографічному картографуванні. Виконано огляд сучасного комерційного й відкритого програмного забезпечення для обробки хмар точок та визначено доцільність використання ГІС-платформ з відкритим кодом.		
<i>Розділ 2</i>	У розділі викладено методичні засади цифрового топографічного картографування з урахуванням чинних нормативних документів та вимог до точності великомасштабних планів. Проаналізовано класичні етапи створення цифрових топографічних карт і визначено особливості їх реалізації з використанням LiDAR-даних. Обґрунтовано переваги лазерного сканування порівняно з традиційними методами знімання.		
<i>Розділ 3</i>	У розділі розроблено удосконалену методику цифрового топографічного картографування на основі LiDAR-даних з високим рівнем автоматизації.		

	Запропоновано алгоритм обробки хмар точок, що включає класифікацію, та автоматичну векторизацію об'єктів. Описано архітектуру та логіку роботи QGIS-плагіна «AutoVec», розробленого для автоматизованої векторизації елементів топографічного плану.
<i>Розділ 4</i>	У розділі представлено практичну реалізацію розробленого плагіна автоматизації обробки LiDAR-даних у середовищі QGIS. Продемонстровано процес створення топографічного плану масштабу 1:500 для території м. Ставангер із використанням автоматизованих алгоритмів. Виконано оцінку якості отриманих результатів за показниками точності, повноти та відповідності нормативним вимогам.
<i>Висновки по роботі</i>	У роботі розроблено методику автоматизованої обробки LiDAR-даних для топографічного картографування, що забезпечує підвищення точності та ефективності створення великомасштабних карт. Запропоновано удосконалену технологічну схему цифрового картографування та реалізовано QGIS-плагін «AutoVec», який автоматизує процеси класифікації та векторизації об'єктів.  Практичну придатність методики підтверджено створенням топографічного плану масштабу 1:500 для території м. Ставангер. Результати оцінки якості засвідчили відповідність отриманих даних вимогам чинних нормативних документів, що підтверджує можливість застосування розробленого підходу у топографо-геодезичній практиці.
<p><b>Ключові слова:</b> LiDAR, лазерне сканування, цифрове картографування, топографічний план, автоматизація векторизації даних, плагін “AutoVec”</p> <p><b>Keywords:</b> LiDAR, laser scanning, digital mapping, topographic plan, data vectorization automation, AutoVec plugin</p>	

Керівник: \_\_\_\_\_/Горковчук Ю.В./

«11» грудня 2025

## ЗМІСТ

ВСТУП .....	11
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ТА СУЧАСНІ ПІДХОДИ ДО ОБРОБКИ LIDAR-ДАНИХ.....	13
1.1. Сутність та принципи технології LiDAR.....	15
1.2. Огляд сучасних інструментів та програмних рішень для оброблення хмар точок.....	19
Висновок до першого розділу .....	23
РОЗДІЛ 2. МЕТОДИЧНІ ЗАСАДИ ТОПОГРАФІЧНОГО КАРТОГРАФУВАННЯ.....	24
2.1. Методика цифрового топографічного картографування.....	25
2.2. Використання LiDAR-даних у топографічному картографуванні.....	30
Висновок до другого розділу:.....	32
РОЗДІЛ 3. РОЗРОБЛЕННЯ МЕТОДИКИ АВТОМАТИЗОВАНОЇ ОБРОБКИ LIDAR-ДАНИХ ДЛЯ ТОПОГРАФІЧНОГО КАРТОГРАФУВАННЯ.....	33
3.1. Методика цифрового топографічного картографування на основі LiDAR-даних.....	34
3.2. Розроблення засобів автоматизації для обробки LiDAR-даних.....	39
Висновки до третього розділу: .....	59
РОЗДІЛ 4. ТЕОРЕТИЧНІ ОСНОВИ ТА СУЧАСНІ ПІДХОДИ ДО ОБРОБКИ LIDAR-ДАНИХ.....	60
4.1. Програмна реалізація плагіну з автоматизації «AutoVec» .....	62
4.2. Створення топографічного плану М1:500 на територію м.Ставангер .....	66
4.3. Оцінка якості.....	72

Висновок до четвертого розділу.....	75
ВИСНОВОК.....	77
Список використаних джерел.....	78
ДОДАТКИ.....	81

## ВСТУП

Сучасні тенденції розвитку геоінформаційних технологій, дистанційного зондування Землі та автоматизації просторових обчислень зумовлюють зростання ролі лазерного сканування (LiDAR) у сфері топографічного та інженерного картографування. Технологія LiDAR (Light Detection and Ranging) забезпечує отримання високоточної тривимірної інформації про земну поверхню, об'єкти інфраструктури та природне довкілля, що робить її незамінним інструментом для створення цифрових моделей рельєфу, моніторингу територій, містобудівного планування та екологічного аналізу.

Актуальність теми зумовлена необхідністю підвищення ефективності та автоматизації процесів обробки LiDAR-даних у топографічному картографуванні. Незважаючи на широкий спектр наявних програмних рішень (TerraScan, ArcGIS Pro, QGIS, PDAL, LAStools), більшість із них потребують ручного втручання та значних витрат часу на класифікацію, векторизацію та побудову цифрових моделей рельєфу. Це створює потребу у розробленні методики автоматизованої обробки LiDAR-даних та відповідних програмних інструментів, які здатні скоротити тривалість виробничого циклу та забезпечити стандартизовану якість результатів.

Використання LiDAR-даних у топографічному картографуванні відкриває нові можливості для створення точних цифрових топографічних планів, побудови моделей рельєфу (DTM/DSM), 3D-векторизації природних та штучних об'єктів, а також інтеграції цих даних у геоінформаційні системи. Особливої уваги потребує розроблення плагінів та модулів автоматизації, які поєднують функціонал PDAL-пайплайнів, JSON-шаблонів і середовища QGIS.

Мета роботи – розробити методику та програмний засіб автоматизованої обробки LiDAR-даних для створення цифрових

топографічних карт і моделей рельєфу. Для роботи було обрано Створення топографічної карти.

Об'єкт дослідження – процес оброблення та використання LiDAR-даних у цифровому топографічному картографуванні.

Предмет дослідження – методика та програмні засоби автоматизованої обробки LiDAR-даних для топографічних карт.

**Розділ 1**  
**ТЕОРЕТИЧНІ ОСНОВИ ТА СУЧАСНІ ПІДХОДИ ДО ОБРОБКИ**  
**LIDAR-ДАНИХ**

					<b>ДИПЛОМНИЙ ПРОЕКТ</b>			
Змн.	Арк.	№ докум.	Підпис	Дата				
Виконала	Демченко МА.				Розробка методів автоматизованої обробки LiDAR-даних для геопросторового картографування	Літ.	Арк.	Аркушів
							13	10
Керівник	Горковчук Ю.В.				КНУБА, ГІСУТ, група ГСТМ-24			
Зав. каф.	Карпінський Ю.О.							

## 1.1. Сутність та принципи технології LiDAR

Лідар (транслітерація LIDAR англ. Light Identification, Detection and Ranging) — технологія отримання та обробки інформації про віддалені об'єкти за допомогою активних оптичних систем, що використовують явища відбиття світла і його розсіювання в прозорих і напівпрозорих середовищах [1]. Коли лазерний промінь натикається на об'єкт, наприклад, дерево чи будівлю, частина світла відбивається від нього в напрямку датчика. Засікаючи точний час повернення кожного лазерного імпульсу, система може розрахувати відстань до кожної відбитої точки. В основі цих розрахунків лежить метод визначення часу прольоту (ToF), за яким швидкість світла є величиною сталою.[2]

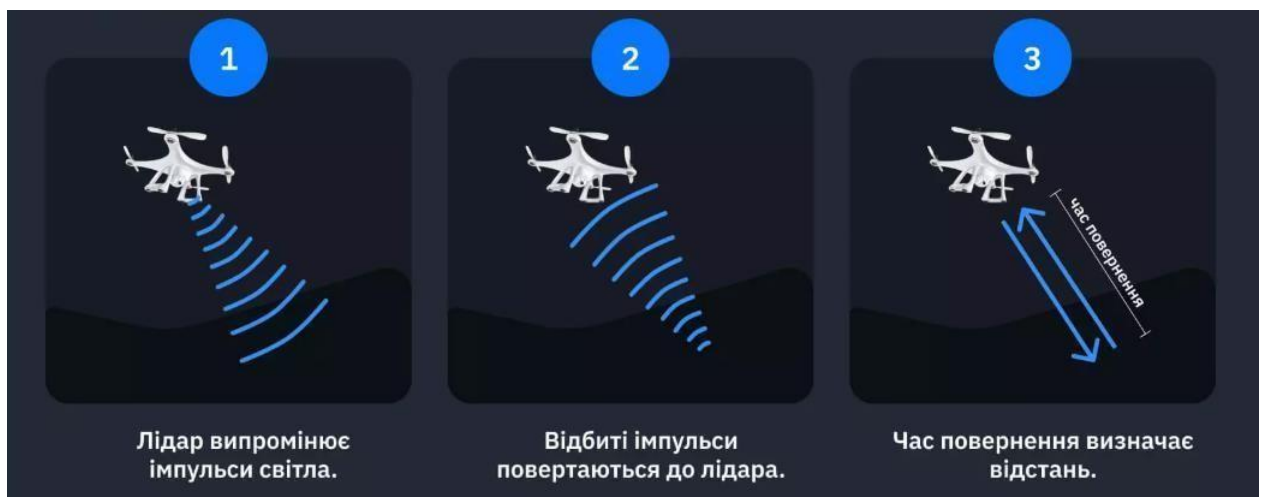


Рис. 1 Принцип роботи лідару [2]

Трьома основними компонентами системи виявлення та визначення дальності світла є лазер, сканер та детектор. Лазерне джерело генерує світло у вигляді переривчастих імпульсів, що дає змогу швидко просканувати область зондування в кількох напрямках. Після того як світло проходить шлях від системи до об'єкта і назад, воно реєструється сканером (зазвичай це дзеркало, що обертається або MEMS – мікроелектромеханічна система) і направляється на детектор для подальших розрахунків.[2]

LiDAR-сенсори випромінюють світло у вигляді лазерних імпульсів для вимірювання відстаней. Вони відомі своєю точністю та здатністю фіксувати детальні 3D-дані. Однак, як і будь-яка технологія, вони мають свої обмеження та специфічні атрибути, які слід враховувати:

**Пряма видимість:** LiDAR не може "бачити" крізь тверді предмети або густий дим. Важливо відповідно планувати свої місії сканування, забезпечуючи відсутність перешкод у критичних зонах.

**Відбиваючі поверхні:** Сильно відбиваючі поверхні, такі як скло або метал, можуть спричиняти багаторазові відбиття або "луну", що може призвести до неточностей у даних.

**Діапазон:** LiDAR-сенсори мають максимальний діапазон сканування, на який може впливати відбивна здатність сканованих поверхонь.

**Поле зору (FOV):** LiDAR-сенсори мають різні FOV, і важливо розуміти FOV конкретного сканера, необхідно враховувати щоб правильно розташовувати сканер щоб він міг захопити простір під правильним кутом.  
[4]

Цей процес повторюється мільйон разів за допомогою лідарних приладів і в результаті створюється складна карта досліджуваної області, відому як 3D-хмара точок.[5]

Хмара точок — це набір даних, що складається з мільйонів окремих точок у тривимірному просторі, кожна з яких має свої координати  $x$ ,  $y$ ,  $z$ . [6]

Для кожного записаного лазерного імпульсу зберігаються такі атрибути точки лідара: інтенсивність, кількість відбитих сигналів, кількість відбитих сигналів, значення класифікації точок, точки, що знаходяться на краю лінії польоту, значення RGB (червоний, зелений та синій), час GPS, кут сканування та напрямок сканування. У наступній таблиці описано атрибути, які можна надати кожній точці лідара.[6.1]

## Атрибути лідарного знімання [6.1]

Інтенсивність	Сила відбиття лазерного імпульсу, який генерував точку лідара.
Номер повернення	Випромінюваний лазерний імпульс може мати до п'яти відбиттів залежно від об'єктів, від яких він відбивається, та можливостей лазерного сканера. Перше відбиття — номер один, друге — номер два тощо
Кількість повернень	Загальна кількість повернень для заданого імпульсу. Наприклад, точка може бути поверненням два серед п'яти.
Класифікація точок	Кожна точка після пост-обробки може мати класифікацію, що визначає тип об'єкта: ґрунт, низька/висока рослинність, будівлі, вода, шум тощо.
Край лінії польоту	Прапорець, який позначає, чи знаходиться точка на краю смуги сканування (0 або 1).
RGB	Атрибути точки за каналами червоного, зеленого та синього, зазвичай отримана з одночасно зібраних зображень.
Час GPS	Мітка часу GPS, коли імпульс був випромінений із літального апарата, у секундах тижня GPS.
Кут сканування	Значення в градусах від -90 до +90. Вказує, де знаходився імпульс відносно літака під час сканування
Напрямок сканування	Напрямок руху скануючого дзеркала: 1 — позитивний, 0 — негативний. Додаткове значення вказує зміщення імпульсу відносно траєкторії польоту

У підсумку сукупність координат та атрибутивних даних, отриманих у процесі лазерного сканування, формує хмару точок — структурований набір просторових вимірів, у якому кожна точка має точне положення у тривимірному просторі й набір характеристик, що описують властивості поверхні відбиття. Хмара точок слугує основною формою представлення результатів LiDAR-знімання та є фундаментом для побудови цифрових моделей рельєфу, поверхні та інженерних об'єктів.[3] Перевагами LiDAR-сканування порівнянно з традиційними методами збору даних:

- Висока точність: LiDAR забезпечує точні вимірювання відстані, що дозволяє створювати детальні 3D-карти високої роздільної здатності.[24]
- Швидкість: LiDAR швидко збирає дані, що дозволяє використовувати їх у режимі реального часу, наприклад, для навігації автономних транспортних засобів.[24]

- Незалежність від освітлення: на відміну від камер, LiDAR не залежить від рівня освітлення та може ефективно працювати як вдень, так і вночі.[24]
- Проникна здатність: LiDAR може «бачити» крізь рослинні полози та атмосферні явища, що робить його корисним для створення точних моделей земної поверхні.[24]
- 3D-сканування: LiDAR дозволяє отримувати повні 3D-зображення об'єктів, необхідні для точного картографування та моделювання.[24]

Сумарно наведені властивості роблять LiDAR одним із найефективніших інструментів для топографічного картографування, оскільки технологія забезпечує високу точність відтворення рельєфу, значну швидкість охоплення територій та можливість роботи у складних або недоступних умовах. У результаті формується більш детальна, структурована й достовірна хмара точок, що створює основу для побудови цифрових моделей місцевості та подальшого аналізу. Разом із тим, попри очевидні переваги, використання LiDAR-технологій має низку обмежень і факторів, які впливають на якість та вартість кінцевого картографічного продукту.

Незважаючи на численні переваги, впровадження технології LiDAR у широку практику топографічного картографування стримується певними недоліками та технічними обмеженнями, до них входять:

- Обмежений радіус дії: Незважаючи на високу точність, радіус дії лідарів може бути обмеженим, особливо в умовах поганої видимості (туман, сильний дощ).[24]
- Чутливість до погоди: LiDAR може мати проблеми з роботою під час сильного дощу, снігу або пилу, що знижує якість даних.[24]
- Складність обробки даних: LiDAR генерує великі обсяги даних, для обробки та інтерпретації яких потрібні потужні обчислювальні ресурси.[24]

- Споживання енергії: Лідари можуть споживати більше енергії, ніж інші датчики, що може бути критично важливим для автономних систем з обмеженими ресурсами.[24]

Україна також використовує LiDAR системи для збирання даних та картографування. Технологія LiDAR сканування знайшла застосування у різних галузях, включаючи науку, транспорт, будівництво та архітектуру. Галузі використання LiDAR систем для топографічного картографування[19]

## 1.2. Огляд сучасних інструментів та програмних рішень для оброблення хмар точок

Для оброблення хмар точок використовується спеціалізоване програмне забезпечення (ПЗ), яке дозволяє імпортувати, візуалізувати, редагувати, класифікувати дані та створювати на їх основі 3D-моделі, цифрові моделі місцевості (ЦММ), ортофотоплани тощо. Основні програми, що застосовуються в цій галузі, включають як комерційні, так і відкриті рішення:

Комерційне програмне забезпечення:

Autodesk ReCap Pro. Це один із найпоширеніших інструментів для перетворення хмар точок на точні цифрові моделі об'єктів. ПЗ підтримує імпорт даних із лазерних сканерів, фотограмметрії та мобільних платформ, дозволяє очищувати хмару точок, виконувати сегментацію, реєстрацію та експорт у формати, сумісні з Revit, Civil 3D і іншими інженерними системами. ReCap Pro широко застосовується у BIM-проектах, коли потрібно отримати модель реального об'єкта з високою точністю.[7]

Hexagon 3DReshaper. Це універсальне програмне рішення для формування полігональних сіток (mesh) та інженерного аналізу. 3DReshaper працює з даними з наземних, мобільних та авіаційних сканерів, дозволяє будувати тендітні поверхні, виконувати контроль деформацій, генерувати об'ємні моделі та аналізувати геометрію у високій точності. Програма активно застосовується у машинобудуванні, топографії, гідротехнічних роботах та моделюванні об'єктів складної форми.[8]

CREDO 3D СКАН. Програмний продукт, розроблений для інженерної геодезії та проектування. CREDO 3D СКАН дозволяє виконувати автоматичну і напіваавтоматичну класифікацію хмар точок, будувати цифрові моделі рельєфу, генерувати поперечники, профілі, моделі дорожньої інфраструктури та інші інженерні об'єкти. ПЗ орієнтоване на оброблення

наземного та мобільного сканування у проектах дорожнього, промислового й цивільного будівництва.[9]

VisionLidar. Це програмне забезпечення з широким спектром інструментів для фільтрації, класифікації та аналізу хмар точок. VisionLidar підтримує роботу в режимі реального часу, що дозволяє інтерактивно оцінювати геометрію місцевості або споруд. Система використовується у міському моделюванні, кадастрових роботах та контролі інженерної інфраструктури.[10]

Pix4Dmapper / Pix4D Desktop. Популярне ПЗ для фотограмметрії, яке водночас підтримує оброблення щільних хмар точок, побудованих із знімків або комбінованих із LiDAR-даними. Pix4Dmapper дозволяє генерувати ортофотоплани, 3D-моделі, цифрові моделі поверхні та рельєфу, а також створювати сіткові моделі. ПЗ часто застосовується у сільському господарстві, будівництві та моніторингу територій.[11]

LixelStudio. Спеціалізована програма для роботи з ручними та портативними LiDAR-сканерами, включаючи мобільні сенсори. ПЗ дозволяє виконувати повний цикл опрацювання хмари точок: очищення, видалення шумів, сегментація, побудова поверхонь та експорт у формати, сумісні з інженерними платформами. LixelStudio особливо зручне для indoor-зйомок та моделювання інтер'єрів.[12]

Програмні забезпечення з відкритим кодом:

CloudCompare. Одна з найпотужніших безкоштовних платформ для візуалізації та редагування хмар точок. ПЗ підтримує фільтрацію шумів, сегментацію, класифікацію, порівняння моделей, реєстрацію (ICP), аналіз щільності та побудову профілів. CloudCompare активно використовується у наукових і технічних проектах для аналізу змін місцевості та інженерних конструкцій.[13]

GRASS GIS. Сучасна ГІС-платформа, що включає модулі для аналізу LiDAR-даних (v.in.lidar, r.in.lidar), побудови цифрових моделей рельєфу, класифікації даних і 3D-візуалізації. GRASS GIS використовується у екологічному моніторингу, гідрологічному моделюванні та аналізі природних ресурсів.[14]

Таблиця 1.2

Порівняльна таблиця ПЗ для перегляду, аналізу та ручної обробки

Назва ПЗ	Тип ліцензії	Основні функції	Переваги	Сфера застосування
Autodesk ReCap Pro	Комерційна (Autodesk)	Імпорт, реєстрація, фільтрація, сегментація, 3D- моделювання	Інтеграція з Revit, Civil 3D	ВІМ, геодезія, реконструкціях[7]
Hexagon 3DReshaper	Комерційна (Hexagon)	Mesh, аналіз поверхонь, порівняння моделей	Висока точність аналізу	Топографія, машинобудування[8]
CREDO 3D СКАН	Комерційна	Класифікація, ЦМР/ЦМП, профілі	Оптимізація для дорожніх проектів	Інженерна геодезія [9]
VisionLidar	Комерційна	Фільтрація, класифікація, аналіз у реальному часі	Швидка обробка хмари точок	Містобудування[10]
Pix4Dmappe r	Комерційна	Хмари точок, DSM/DTM, ортофото, 3D	Фотограмме трія	Будівництво, агromоніторинг[11]
LixelStudio	Комерційна	Сегментація, поверхні, LiDAR	Indoor оптимізація	Архітектура, інтер'єри[12]
CloudCompa re	Відкрита (GPL)	Редагування, сегментація, ICP,	Швидкість	Геодезія, контроль деформацій[13]

		профілі		
GRASS GIS	Відкрита (GPL)	Класифікація, DEM, LiDAR- аналітика	Інтеграція з ГІС	Кадастр, топографія[14]

QGIS дозволяє користувачам створювати карти з безліччю шарів, використовуючи різні картографічні проєкції. Карти можуть бути зібрані в різні формати і використовуватися з різною метою. У системі QGIS карти можуть складатися з растрових або векторних шарів. Типовими для такого роду програмного забезпечення, векторні дані зберігаються як точка, лінія, полігон. Підтримуються різні види растрових зображень. Програмне забезпечення може виконувати геоприв'язку зображень.[25]

#### Переваги QGIS:

- Відкритий код та безкоштовна ліцензія, що дозволяє широкому колу користувачів мати доступ до програми.
- Широкий набір функцій та інструментів для роботи з географічними даними.
- Підтримка великою спільнотою розробників, що забезпечує постійне оновлення та розширення функціональності програми.
- Можливість інтеграції з іншими ГІС та програмами через плагіни та розширення.
- Простий інтерфейс та простота використання.
- Можливість роботи з різноманітними геоданими та форматами файлів.[25]

## **Висновок до першого розділу:**

У межах розглянутих підпунктів було сформовано цілісне уявлення про технологію LiDAR та сучасні інструменти, які забезпечують повний цикл роботи з хмарами точок. Показано, що принцип дії лазерного сканування базується на точних вимірюваннях часу проходження імпульсу, а результатом цього процесу виступає хмара точок — просторово-атрибутивний набір даних, здатний відтворювати структуру місцевості з високою деталізацією. Виявлено, що внутрішня структура LiDAR-даних включає численні атрибути, такі як координати, інтенсивність, класифікаційні коди, кількість повернень та параметри сканування, що робить їх цінним джерелом для геоінформаційного аналізу.

Окремо підкреслено, що ефективне використання хмар точок потребує потужних програмних засобів. У сучасній практиці застосовуються як комерційні платформи (Autodesk ReCap Pro, Hexagon 3DReshaper, CREDO 3D СКАН, VisionLidar, Pix4Dmapper, LixelStudio), так і відкриті рішення (CloudCompare, GRASS GIS). Ці системи забезпечують імпорт, очищення, фільтрацію, сегментацію, класифікацію та моделювання на основі хмар точок, відкриваючи широкі можливості для створення цифрових моделей рельєфу, поверхонь, 3D-об'єктів та інженерних схем. Проаналізовані програмні засоби суттєво відрізняються за функціональністю, інструментами та сферою застосування, що дає змогу підібрати оптимальне рішення під конкретні завдання.

**Розділ 2**  
**МЕТОДИЧНІ ЗАСАДИ ТОПОГРАФІЧНОГО КАРТОГРАФУВАННЯ**

					<b>ДИПЛОМНИЙ ПРОЕКТ</b>			
Змн.	Арк.	№ докум.	Підпис	Дата				
Виконала	Демченко М.А.				Розробка методів автоматизованої обробки LiDAR-даних для геопросторового картографування	Літ.	Арк.	Аркушів
							24	18
Керівник	Горковчук Ю.В.					КНУБА, ГІСУТ, група ГСТМ-24		
Зав. каф.	Карпінський Ю.О.							

## 2.1 Методика цифрового топографічного картографування

Цифрове топографічне картографування стало логічним продовженням класичних картографічних практик у середовищі комп'ютерних технологій. Воно ґрунтується на створенні структурованої цифрової моделі місцевості, де просторові об'єкти подаються у вигляді метричних та атрибутивних даних. У цифрових картах зв'язок між просторовими об'єктами та семантикою реалізується через класифікаційну структуру — основу всього процесу побудови сучасної топографічної карти.[15]

Цифрова карта місцевості (ЦКМ) — цифрова модель місцевості, записана на магнітний носій у встановлених структурі і кодах стосовно до визначеної математичної основи, проекції і розграфленню, прийнятих для карт, і що відповідає встановленим для конкретного використання вимогам за точністю і змістом. Цифрові карти місцевості призначені для автоматизованого вивчення й оцінки місцевості, виконання розрахунків при плануванні і проектуванні інженерних споруджень і рішення інших спеціальних задач в автоматизованих системах [15].

Цифрові карти місцевості повинні задовольняти наступним вимогам:

- Для 1:5000: За основу береться аркуш карти масштабу 1:100 000, який ділиться на 25 квадратів, кожен з яких позначається арабською цифрою.
- 1:2000: Аркуш плану масштабу 1:5000 ділиться на чотири частини, які позначаються великими літерами українського алфавіту.
- 1:500: Аркуш плану масштабу 1:2000 ділиться на шістнадцять рівномірних частин. Кожна частина позначається арабською цифрою від 1 до 16. [28]

- створюватися в системі координат координат 2000 року (УСК-2000);[29]

- забезпечувати можливість машинного визначення даних про місце розташування об'єктів і їхніх характеристик; [15]

- включати цифрові значення кількісних, якісних характеристик і кодів об'єктів у Єдиній системі класифікації і кодування картографічної інформації; [15]

- мати класифікацію об'єктів і елементів місцевості, що відповідають класифікації, прийнятої для українських топографічних карт масштабів 1:25000 і 1:50000; [15]

- мати сумарну середню квадратичну помилку взаємного планового положення твердих контурів у межах номенклатурного аркуша ЦКМ не більш 50 м; [15]

- сумарну середню квадратичну помилку положення горизонталей по висоті не більш: [15]

5 м - для менш пересічних районів; [15]

10 м - для середньо пересічних районів;

20 м - для більш пересічних районів;

- додавати до масивів даних відповідних елементів змісту топографічної карти службово-довідкову інформацію (основні характеристики цифрової карти місцевості, наприклад: номенклатуру, грифа таємності, теоретичні розміри рамки аркуша топографічної карти, характеристики помилок цифрової інформації в плані і за висотою тощо); [15]

- мати структуру уявлення інформації ЦКМ, що забезпечує можливість внесення змін і доповнень без перекручування наявних даних і погіршення їх точних характеристик. [15]

Процес створення цифрових карт включає наступні основні етапи:

- 1) автоматизоване перетворення початкової картографічної інформації на цифрову форму;
- 2) символізація цифрової картографічної інформації та автоматизоване складання електронних карт;
- 3) розробка системи управління базами даних (СУБД), призначеної для роботи з електронними картками.[15]



Рис. 2.1 Загальна технологічна схема створення цифрової карти

Нормативні документи встановлюють вимоги до дозволеної дискретності даних, точності координат, якості векторизації, коректності топології та відповідності умовним знакам. У цифровому картографуванні важливу роль відіграють стандарти класифікації та кодифікації даних, зокрема державні класифікатори просторових об'єктів, які забезпечують уніфіковане подання інформації. [15]

До державних документів, що застосовуються у топографічному картографуванні, належать ДСТУ (державні стандарти України), ГСТУ

(галузеві стандарти), кодекси усталеної практики, міждержавні стандарти та технічні регламенти. Вони визначають вимоги до точності, структури просторових даних, класифікаторів та організації оброблення геопросторової інформації.

ДСТУ 7169:2010 «Якість геопросторових даних. Терміни та визначення» Регламентує вимоги до точності, повноти, узгодженості та логічної правильності геопросторових даних.

ДСТУ ISO 19157:2015 «Географічна інформація. Якість даних» Описує контроль якості цифрових карт: точність позиціонування, атрибутивна точність, повнота, узгодженість.

ДСТУ ISO 19115-1:2015 «Метадані географічної інформації» Регулює обов'язкову структуру метаданих цифрової карти.

Також для топографічного картографування обов'язкові відомчі українські нормативи «Умовні знаки для топографічних карт масштабів 1:500–1:5000», для точності картографічних карт «Інструкція з топографічного знімання у масштабах 1:500–1:5000 (ГКНТА-2.04-02-98)».

У діючій нормативній базі України встановлено чіткі вимоги до точності топографічних планів великого масштабу, зокрема 1:500. Такі плани відносяться до інженерних топографічних робіт, тому їх параметри регламентуються низкою документів: ДБН А.2.1-1-2014, Інструкцією з топографо-геодезичних робіт ГКНТА-2.04-02-98, а також Вимогами до точності топографічних планів масштабу 1:500–1:5000.

1. Планова точність. Для масштабу 1:500 середня квадратична похибка (СКП) планового положення точок місцевості не повинна перевищувати: 0,1мм у масштабі плану, що відповідає 5 см у натурі (0,05 м). Ця вимога застосовується до кутів будівель, осей інженерних споруд, жорстко

закріплених елементів інфраструктури. Норми встановлені у ГКНТА-2.04-02-98, табл. Точності великомасштабних планів.[16]

2. Висотна точність. СКП визначення висот для масштабу 1:500 становить: 0,05 м (5 см) для твердих, доступних точок рельєфу та об'єктів; 0,10 м (10 см) для м'яких та важкодоступних поверхонь (насипи, відкоси, ґрунтові ділянки).

Такі норми відповідають ДБН А.2.1-1-2014, а також вимогам Інструкції з топографо-геодезичних робіт, що регламентує порядок знімання рельєфу та висотної прив'язки об'єктів.[16],[17]

3. Точність відображення лінійних та контурних об'єктів. Для топографічного плану масштабу 1:500: СКП положення лінійних об'єктів — 0,1 мм у масштабі (5 см на місцевості) (дороги, паркани, комунікації).

Ці значення регламентовані вимогами до графічної точності Держгеокадастру та Укргеодезкартографії для великомасштабних планів.[16]

Вимоги до семантичної точності. Семантичні ознаки об'єктів (титри, позначення, атрибути) повинні відповідати: Класифікатору цифрової карти (КЦК). Умовним знакам масштабів 1:500–1:5000 (наказ Укргеодезкартографії №3 від 14.01.2000) .

Помилки у підписах та кодах не допускаються, оскільки вони безпосередньо впливають на подальше використання плану в проектуванні.[16] Вимоги до повноти відображення. Для масштабу 1:500 встановлено один із найвищих рівнів деталізації: Обов'язкове відображення всіх об'єктів шириною від 0,1–0,2 м; Нанесення всіх мікрооб'єктів (колодязів, люків, бордюрів, дрібних об'єктів благоустрою); Обов'язкове вказання матеріалів покриття, типів огорож, діаметрів труб, висот і відміток.

Повнота змісту підтверджується вимогами ДБН А.2.1-1-2014, розділ щодо інженерних вишукувань для будівництва. [17]

## 2.2 Використання LiDAR-даних у топографічному картографуванні

Традиційна технологія створення топографічних карт базується переважно на польових геодезичних вимірюваннях (тахеометрія) або фотограмметрії з використанням аерофотознімків [26]. Ці методи потребують значних часових витрат на польові роботи, особливо на ділянках зі складною рослинністю чи забудовою. Обробка даних також є трудомісткою, а отримання точної моделі рельєфу під кронами дерев було вкрай ускладненим [27]. Застосування технології лазерного сканування (LiDAR) принципово змінює цей процес. Основна відмінність полягає у методі збору даних. LiDAR використовує активний метод — випромінювання лазерних імпульсів та фіксацію часу їх повернення [27]. Це дозволяє отримувати мільйони точок з високою точністю за короткий проміжок часу.

Ключові відмінності внаслідок застосування LiDAR-даних:

**Швидкість збору даних:** Аерофотознімання або польова зйомка вимагають тривалого часу на планування та виконання робіт. LiDAR-зйомка дозволяє охопити великі площі значно швидше, що скорочує загальні терміни виконання картографічних робіт [27].

**Детальність та щільність:** Традиційні методи надають вибірккові точки або растрові зображення, з яких потім створюються ізолінії. LiDAR формує щільну хмару точок, що забезпечує надзвичайно детальну цифрову модель місцевості [27].

**Проникнення крізь рослинність:** Це, мабуть, найважливіша перевага. Частина лазерних імпульсів проникає крізь листя та гілки і досягає поверхні землі. В результаті цього, вперше стало можливим створювати точні цифрові моделі рельєфу (ЦМР) навіть у лісових масивах без необхідності проведення дороговартісних та довготривалих наземних зйомок [27].

Автоматизація обробки: Величезний обсяг даних, отриманий за допомогою LiDAR, обробляється за допомогою спеціалізованого програмного забезпечення. Це дозволяє автоматизувати класифікацію точок (земля, рослинність, будівлі, вода) та генерацію кінцевих картографічних продуктів, таких як ізолінії та матриці висот [27].

## **Висновок до другого розділу:**

У підсумку другого розділу було простежено, як сучасні цифрові технології перетворили класичне топографічне картографування на гнучкий, структурований і високотехнологічний процес. Цифровий підхід дав змогу переосмислити весь цикл отримання та опрацювання просторової інформації — від збору даних до формування готового картографічного продукту, придатного для подальших інженерних, проектних чи аналітичних задач.

Окреслено базові принципи та алгоритми цифрового картографування, підкреслено значення класифікаційної структури та метаданих як основи для коректного подання об'єктів і гарантії їхньої сумісності в різних ГІС-системах. Розглянуті етапи автоматизованої векторизації показали, що сучасні програмні комплекси дозволяють істотно прискорити отримання векторної моделі та підвищити її структурованість завдяки алгоритмам розпізнавання і топологічного контролю.

Особлива увага була приділена застосуванню LiDAR-технологій як одному з найпотужніших інструментів для топографічного картографування. Аналіз переваг показав, що LiDAR забезпечує найвищу доступну точність опису рельєфу, працює у складних природних умовах, формує надзвичайно деталізовану тривимірну хмару точок та значно скорочує час польових робіт. Разом із тим було підкреслено важливість урахування технічних і економічних обмежень цієї технології, зокрема чутливості до погодних умов, великих обсягів даних і потреби у високій обчислювальній потужності

**Розділ 3**  
**РОЗРОБЛЕННЯ МЕТОДИКИ АВТОМАТИЗОВАНОЇ ОБРОБКИ**  
**LIDAR-ДАНИХ ДЛЯ ТОПОГРАФІЧНОГО КАРТОГРАФУВАННЯ**

					<b>ДИПЛОМНИЙ ПРОЕКТ</b>					
<b>Змн.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>	Розробка методів автоматизованої обробки LiDAR-даних для геопросторового картографування			<b>Літ.</b>	<b>Арк.</b>	<b>Аркушів</b>
Виконала	Демченко М.А.									33
Керівник	Горковчук Ю.В.				КНУБА, ГІСУТ, група ГСТм-24					
Зав. каф.	Карпінський Ю.О.									

### **3.1. Методика цифрового топографічного картографування на основі LiDAR-даних**

Розвиток лазерного сканування дав можливість суттєво змінити традиційну методику створення цифрових топографічних карт. Якщо класична схема з пункту 2.1 орієнтувалася на ручну векторизацію та роботу з растровими підкладками, то LiDAR картографування ґрунтується на аналізі хмар точок, автоматизованій класифікації та побудові моделей рельєфу, що суттєво підвищує швидкість і точність виробничого циклу.

Удосконалена методика цифрового топографічного картографування з використанням LiDAR-даних спирається на класичний виробничий цикл (підготовка, цифрування, обробка, випуск), однак переносить значну частину процедур на роботу зі хмарою точок. Це підвищує точність, стабільність результатів та рівень автоматизації, що особливо важливо для масштабів 1:500.

Класичну схему «Загальна технологічна схема створення цифрової карти» в пункті 2.1 було покращено створену оновлену схему створення цифрової карти за допомогою лідару. Також було покращено другий пункт схеми за рахунок створення плагіну «AutoVec» для автоматичної векторизації люків.

Етапами вдосконаленої методики є:

-Перший етап підготовчий. Метою роботи було створення топографічного плану місцевості міста Ставангер, з точністю відповідно до чинних ДСТУ та ГКНТА (зокрема ГКНТА-2.04-02-98) на незабудованій території. Середня квадратична похибка (СКП) положення чітко контурованих об'єктів місцевості відносно найближчих пунктів опорної геодезичної мережі не повинна перевищувати 0,1 мм у масштабі плану. В результаті, для масштабу 1:500 це відповідає 5 см на місцевості. Гранична похибка (подвоєна СКП) не має перевищувати 0,2 мм, або 10 см на

місцевості. Формується перелік вимог до кінцевої моделі (точність об'єктів, щільність точок, формат даних).

Наступним кроком обирається метод зйомки (авіаційна, мобільна, наземна), проводиться оцінка щільності і точності хмари точок. Збір даних виконується паралельною реєстрацією GNSS/IMU.

-Другий етап розпочинається з попередньої обробки хмари точок. А саме з імпорту та конвертації. Завантаження даних у форматі LAS/LAZ, перевірка коректності геопросторової прив'язки, застосування поправок GNSS/IMU. Також необхідно видалити шуми аномальні значення та точки із надмірними відхиленнями. Після очищення та підготовки хмари виконується автоматизована класифікація точок яка включає в себе присвоєння класифікаційного коду (грунт, рослинність, будівлі, вода, технічний шум).

-Третім етапом є створення моделей рельєфу і поверхонь. Цифрова Модель Рельєфу (ЦМР / DTM). Побудова виконується виключно на основі точок, класифікованих як «грунт». ЦМР є базою для генералізації горизонталей, створення профілів та інженерного аналізу.

-Четвертий етап заключається у векторизації об'єктів та автоматичній векторизації люків. До автоматичної векторизації застосовуються алгоритми перетворення класифікованої хмари точок або похідних моделей (DTM/DSM) на векторні об'єкти. Генерація ізоліній створюється на основі ЦМР із заданим інтервалом перерізу рельєфу.

-П'ятий етап це нанесення об'єктів та атрибутизація

Сюди входить присвоєння сентиматики, векторні об'єкти кодуються відповідно до класифікатора картографічних об'єктів. Тут додаються числові й текстові атрибути, виконується ручне коригування дрібних елементів (лавки, парканчики, дроти), що характерно для масштабу 1:500.

## -Контроль якості та топологічна перевірка

Проводиться перевірка на: допустимі середньоквадратичні похибки згідно ДСТУ; відсутність топологічних помилок (розривів, дублювання, самоперетинів); відповідність умовним знакам та класифікаторам.

## -Формування фінального аркуша

Після контролю виконується: складання топографічного плану 1:500; імпорт шарів до ГІС-середовища; генерація метаданих; експорт у формат DXF, Geopackage, TIFF або PDF.

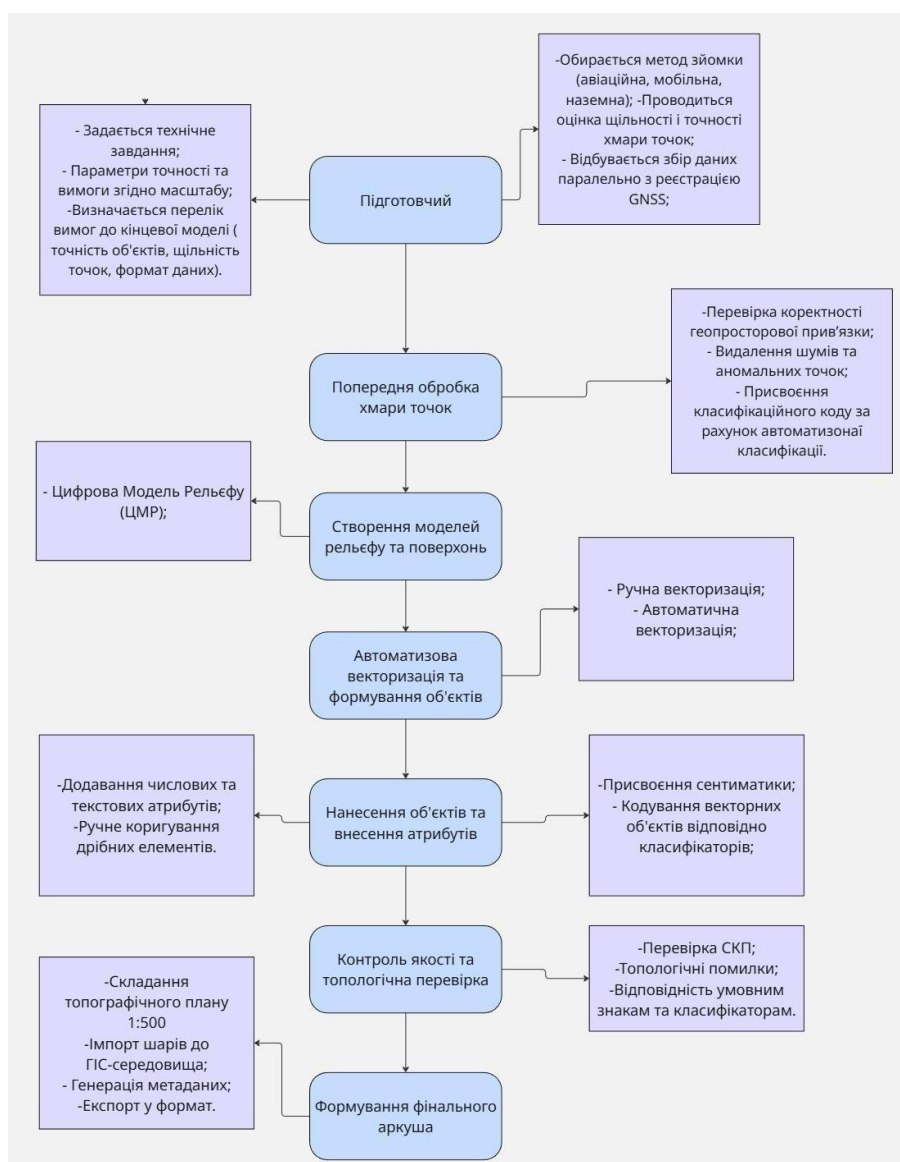


Рис. 3.1 Етапи цифрового картографування на основі LiDAR

Алгоритм роботи плагіну «AutoVec» для застосування його для етапу автоматичної векторизації. Він складається з основних трьох:

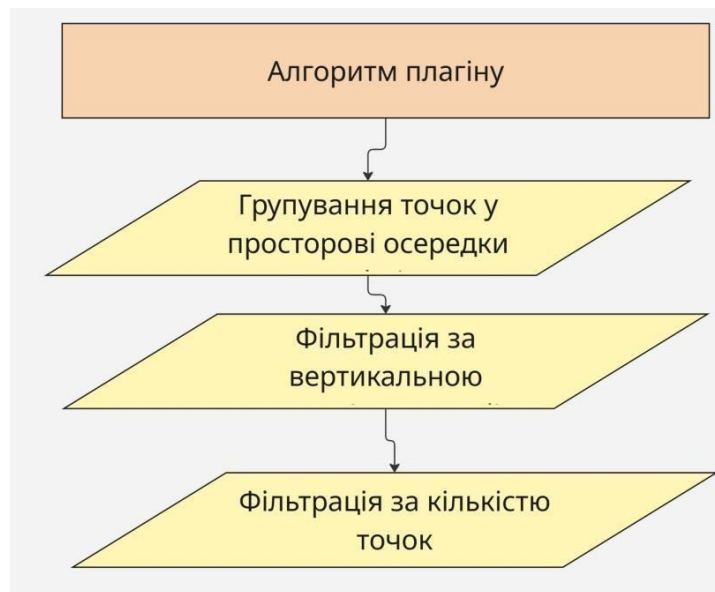


Рис.3.2 Алгоритм плагіну

-Групування точок у просторові осередки (voxel\_size);

Параметр voxel\_size ( в метрах наприклад 0,5). Це розмір сторони тобто векселя, який використовується для поділу простору. В алгоритмі це розмір осередку сітки (X x Y), який агрегує точка.Значення має бути приблизно рівним або трохи більшим за розмір решітки. Наприклад якщо решітка має розмір 0,5 на 0,5м необхідно встановити voxel\_size=0,5м.

-Фільтрація за вертикальною дисперсією (max\_dist);

Параметр max\_dist ( в метрах наприклад 0,05). Це максимальна різниця висот для всіх точок, які потрапили в один вексель. Для цього параметру значення має бути дуже малим, як правило це декілька сантиметрів.Оскільки одна решітка є плоским, горизонтальним об'єктом, її вертикальна дисперсія має бути мінімальною (0,01м-0,05м). Вертикальна дисперсія векселя (наприклад 0,5м) це ймовірно, схил, край бордюру, або частина дерева/стовпа.

-Фільтрація за кількістю точок;

Параметрами тут виступають `min_pts` (мінімальна кількість точок), `max_pts`(максимальна кількість точок). Це діапазон в якому має знаходитись кількість точок усередині одного векселя для того, щоб цей вексель вважався решіткою.

### 3.2. Розроблення засобів автоматизації для обробки LiDAR-даних

Автоматизація обробки хмар точок у геоінформаційних системах потребує продуманого вибору середовища розроблення, архітектури програмного продукту та інструментів інтеграції з високопродуктивними фреймворками обробки просторових даних. У межах цієї роботи основу автоматизації складає QGIS-плагін, створений для виконання класифікації, побудови цифрових моделей рельєфу та подальшої векторизації об'єктів на основі LiDAR-даних. Плагін поєднує інструменти PDAL, JSON-шаблони та засоби Python-розроблення, що дає можливість виконувати складні пайплайни обробки без необхідності використання зовнішніх програмних комплексів.

Основою для створення програмного рішення обрано QGIS як відкриту та розширювану платформу для ГІС-аналізу. Основним призначенням системи є обробка і аналіз просторових даних, підготовка різної картографічної продукції. Інтерфейс QGIS побудований на базі бібліотеки Qt. Designer має гнучку систему розширень, які можна створювати на мовах C++ і Python. Підтримуються різноманітні векторні і растрові формати з ESRI Shapefile і GeoTIFF включно.[25]

Основна реалізація виконується у Python 3.6, оскільки ця мова забезпечує просту інтеграцію з GDAL, PDAL, NumPy та іншими бібліотеками, необхідними для роботи з LiDAR-даними. Для створення графічного інтерфейсу використовується PyQt5 — стандартний фреймворк для плагінів QGIS 3.44.2. Формування структури плагіна здійснюється через офіційний інструмент Plugin Builder, який генерує каркас проєкту та файлову структуру.

Розроблення інструменту автоматизованої векторизації хмари точок у межах QGIS потребує чіткої організації внутрішньої структури плагіна та

визначення середовища, у якому виконується програмування. Основою виступає вбудований у QGIS механізм плагінів, що працює з використанням Python-інтерпретатора та API PyQGIS. Саме цей стек технологій забезпечує прямий доступ до геопросторових даних, систем візуалізації та інструментів оброблення. Крім того, екосистема QGIS підтримує інтеграцію бібліотек сторонніх розробників, що дає змогу розширити стандартний функціонал і створювати спеціалізовані алгоритми векторизації.

Створення JSON бібліотеки прикладів, розпочалось з додаванням хмари точок до QGIS, та створенням полігону навколо об'єктів, для подальшого вирізання.



Рис.3.3 Приклади люків

Потім для бібліотеки JSON необхідно вирізати ці фрагменти з хмари точок це можливо зробити за допомогою функції «Clip»

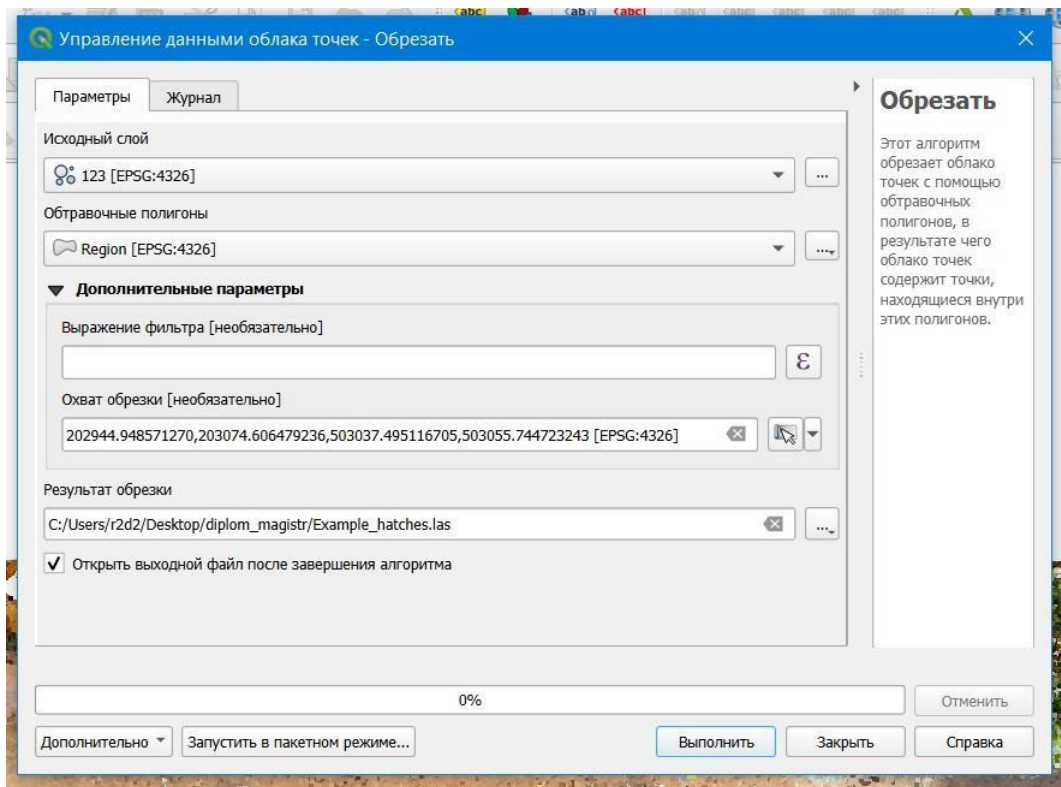


Рис.3.4 Интерфейс інструменту «Слір»

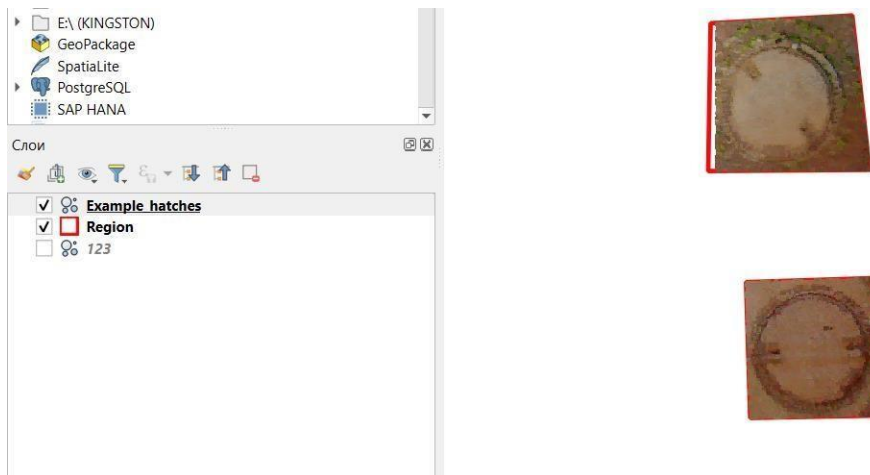


Рис.3.5 Результат роботи функції

Для інтеграції цієї хмари точок в бібліотеку, необхідно зберегти цю хмару точок в форматі CSV з необхідними параметрами це ( X, Y, Z, Red, Green, Blue, Intensity, ReturnNumber).

Буде створена таблиця з координатами та кольорами хмар точок

Наступним кроком додаємо цю таблицю в Qgis, необхідно відкрити таблицю атрибутів та створити нове поле з назвою «Json\_string» та текстовим типом та прописати функцію:

```
'{"x":'|'|"X"|'|','y":'|'|"Y"|'|  
'|"z":'|'|"Z"|'|  
'|"r":'|'|"Red"|'|','g":'|'|  
"Green"|'|','b":'|'|"Blue"|'|}''
```

Рис.3.6 Функція яка прописується для переведення з CSV в JSON

Потім необхідно експортувати цю таблицю в форматі GeoJSON з полів було обрано лише «json\_string».

Наступний структурний блок відповідає за логіку порівняння: під час аналізу всі точки вхідної хмари проходять процедуру класифікаційного співставлення з еталонними шаблонами кольорових ознак. На основі подібності формуються тимчасові кластери, які потім трансформуються у векторні геометрії — точкові або полігональні об'єкти. Перетворення реалізовано через PyQGIS-методи створення шарів, що дозволяє вивантажувати результати у форматах GeoPackage, Shapefile або залишати їх у середовищі проекту QGIS.

Ще один важливий компонент — модуль керування бібліотекою шаблонів. Він забезпечує додавання нових фрагментів хмар точок, їх статистичний аналіз та збереження у структурованому вигляді, що дозволяє розширювати можливості плагіна без переписування основного коду. Користувач має змогу завантажити власний зразок хмари, виділити потрібний фрагмент, а плагін автоматично сформує JSON-опис кольорової моделі об'єкта.

Графічний інтерфейс побудований на основі форми, створеної у Qt Designer та інтегрованої через PyQt5. Він містить мінімально необхідні елементи: панель завантаження хмари точок, список доступних шаблонів,

поле налаштувань параметрів подібності RGB та модуль керування векторним виходом. Така структура дозволяє користувачу працювати у кілька кліків, не взаємодіяти з консоллю та не налаштовувати складні параметри вручну, що значно пришвидшує процес векторизації.

Процес інтеграції бібліотеки в середовище плагіну:

```
def load_hatches_example(self):
    """
    Завантажує JSON-бібліотеку шаблонів люків із каталогу плагіна.
    Якщо файл відсутній або зламаний - повертає None.
    """
    plugin_dir = os.path.dirname(__file__)
    json_path = os.path.join(plugin_dir, "templates", "Hatches_example.json")

    if not os.path.exists(json_path):
        print(f"Попередження: бібліотека люків не знайдена: {json_path}")
        return None

    try:
        with open(json_path, "r", encoding="utf-8") as f:
            data = json.load(f)
        if "templates" in data and isinstance(data["templates"], list):
            return data["templates"]
        else:
            print("Попередження: у файлі бібліотеки немає ключа 'templates' або він некоректний")
            return None
    except Exception as e:
        print(f"Помилка читання JSON-бібліотеки люків: {e}")
        return None
```

Рис. 3.7 Завантаження JSON-бібліотеки шаблонів люків із каталогу плагіна

```
def match_hatches_template(self, r, g, b):
    """
    Повертає словник шаблону люка, якщо середній колір (r, g, b)
    потрапляє в діапазон хоча б одного шаблону бібліотеки.
    Якщо збігів немає або бібліотека відсутня - повертає None.
    """
    if not self.Hatches_example:
        return None

    for tpl in self.Hatches_example:
        try:
            if (tpl["r_min"] <= r <= tpl["r_max"] and
                tpl["g_min"] <= g <= tpl["g_max"] and
                tpl["b_min"] <= b <= tpl["b_max"]):
                return tpl
        except KeyError:
            # Якщо в шаблоні не вистачає якогось ключа - пропуск
            continue

    return None
```

Рис.3.8 Функція зіставлення середнього кольору з шаблоном

```
#поле класу об'єкта (тип люка) ---
fields.append(QgsField("class", QVariant.String))
```

Рис.3.9 Поле «class» та перевірку кольору

```

#колірна сегментація по бібліотеці люків
matched_tpl = None
if avg_r is not None and avg_g is not None and avg_b is not None:
    matched_tpl = self.match_Hatches_template(avg_r, avg_g, avg_b)
# Якщо плагін має знаходити тільки люки – пропускаємо всі воксели без збігу
if matched_tpl is None:
    continue
# Запис класу (назви шаблону) як атрибуту
feature.setAttribute("class", matched_tpl.get("name", "Hatches"))

```

Рис. 3.10 Колірна сегментація по бібліотеці люків

Концепція розроблюваного плагіна базується на ідеї повної автоматизації процесу векторизації хмари точок LiDAR за кольоровими характеристиками, без виконання цифрового моделювання рельєфу чи класифікації за стандартами ASPRS. Плагін орієнтований на роботу з уже очищеними або попередньо підготовленими хмарами точок, у яких атрибути RGB, інтенсивність або їх комбінації служать критеріями для виокремлення об'єктів, зокрема плагін орієнтований на створеній вручну бібліотеки з обраними конкретними об'єктами (люки).

У контексті проекту хмара точок сприймається не як основа для побудови цифрових моделей рельєфу, а як джерело колірних та просторових характеристик, достатніх для автоматичного відокремлення люків від навколишніх об'єктів. Для цього було передбачено використання двох взаємопов'язаних механізмів:

Колірна сегментація – аналіз RGB-компонентів кожної точки з метою виявлення кластерів, що відповідають характерним візуальним ознакам люків. Сегментація працює за принципом «пошук у діапазонах» із динамічними порогами, які задаються через інтерфейс плагіна.

Зіставлення з еталонними фрагментами – використання спеціальної бібліотеки шаблонів, що містить попередньо виділені фрагменти хмар точок люків, зібрані з реальних даних. Кожен шаблон відображає типову форму та

колір люка, що дозволяє алгоритму порівнювати локальні скупчення точок зі зразками і визначати ступінь відповідності.

У результаті люк розпізнається не як «клас об'єкта», а як комбінація колірних ознак і просторового патерну, що відповідає еталонному шаблону.

Архітектура плагіна складається з кількох логічних блоків, кожен із яких відповідає за окремий етап роботи. Вона вибудована за модульним принципом, щоб спростити оновлення функцій і масштабування.

### 1. Модуль управління даними

Виконує завантаження хмари точок у форматах LAS та LAZ. Передбачено зчитування координат X, Y, Z; оброблення RGB-компонентів; підготовку даних для подальших алгоритмів. Модуль працює на базі бібліотек laspy або PDAL Python bindings, що дає змогу працювати зі значними обсягами даних без втрати продуктивності.

### 2. Модуль колірної сегментації

Основний алгоритмічний блок, який виконує виділення точок, що можуть відповідати люкам. Відповідають функції: фільтрація за колірними діапазонами (RGB або HSV); первинне групування точок у колірні кластери; формування масиву підозрюваних зон. Сегментація налаштовується через інтерфейс плагіна — користувач задає допустимі інтервали або обирає один із заздалегідь підготовлених колірних профілів.

### 3. Бібліотека шаблонів хмар точок

Бібліотека шаблонів є ключовим елементом архітектури й забезпечує плагін можливістю розпізнавати люки за зразками реальних даних. Ця бібліотека має набір еталонних фрагментів хмари точок люків; опис їхніх структурних та візуальних властивостей; JSON-файли з параметрами пошуку (радіус, форма, насиченість кольору, щільність точок). Ця бібліотека виступає

«інтелектуальним ядром» плагіна, оскільки дозволяє алгоритму не лише аналізувати колір, а й впізнавати характерну геометрію люка.

#### 4. Модуль просторової кореляції

Порівнює локальні групи точок із шаблонами з бібліотеки. Функціонал: оцінювання схожості за принципом nearest-neighbour clustering; обчислення щільності та радіальної структури; визначення центру люка та орієнтовного радіусу; фільтрація помилкових збігів. Результатом роботи модуля є масив локальних зон, які з високою ймовірністю є люками.

#### 5. Модуль векторизації

Завершальний блок, який перетворює зони знайдених люків у векторні об'єкти. Типами вихідних даних є точкові шари (центроїди люків), полігони-кружки або квадрати (залежно від стилю оформлення топоплану). Модуль додатково формує атрибути: ID люка; площа / радіус; координати центру; індекс достовірності збігу. Готовий векторний шар автоматично додається до проекту QGIS.

#### 6. Модуль інтерфейсу користувача (GUI Module)

Графічний інтерфейс створено через Qt Designer, інтегрований у QGIS API, що забезпечує інтуїтивність та легке керування параметрами.

Інтерфейс містить:

- блок для вибору хмари точок;
- елементи налаштування колірних фільтрів;
- панель вибору шаблонів із бібліотеки;
- слайдери для параметрів радіуса, щільності та меж пошуку;
- кнопку запуску плагіну шару.

Qt Designer у цьому випадку виступає повноцінним модулем середовища розроблення, оскільки забезпечує створення та інтеграцію UI-компонентів безпосередньо у QGIS-плагін.

## 7. Модуль логування та експорту

Передбачає: логування параметрів запуску; фіксацію знайдених зон та можливість експорту векторних шарів у формати GeoPackage, Shapefile, GeoJSON. Графічний інтерфейс користувача (GUI) у плагіні AutoVec створюється як центральний компонент, що забезпечує інтуїтивну взаємодію з інструментами автоматичної векторизації хмари точок. Його побудова базується на принципах простоти, мінімальної когнітивної навантаженості та чіткої відповідності алгоритмічному ядру, орієнтованому на виявлення люків за кольором та порівнянням з еталонними зразками з бібліотеки. Концепція інтерфейсу спрямована на те, щоб користувач міг виконувати повний цикл обробки за декілька послідовних дій без доступу до програмного коду.

Інтерфейс побудований за модульним принципом: кожна група параметрів згрупована за функціональним призначенням — вхідні дані, бібліотека, параметри сегментації та параметри збереження результатів. Така структура дозволяє підтримати прозорість процесу, оскільки всі налаштування відображені у межах одного вікна й доступні без додаткових вкладок.

Робота над користувацьким інтерфейсом розпочалась з встановлення плагіну «QGIS Plugin Builder» та заповнення умовного «паспорту» плагіну, це необхідно для коректного встановлення плагіну, можливість показати його в менеджері плагінів, розуміння як назвати модуль, контроль сумісності версій, ідентифікації автора.

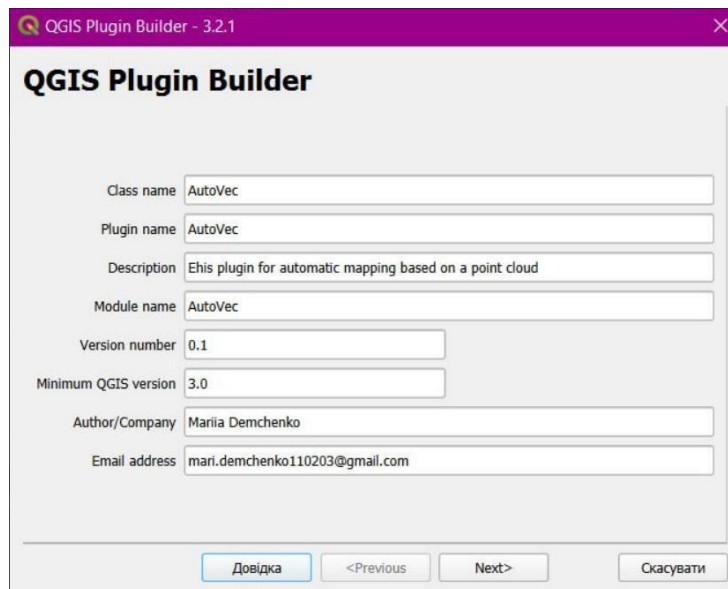


Рис. 3.11 «Паспорт» плагіну «AutoVec»

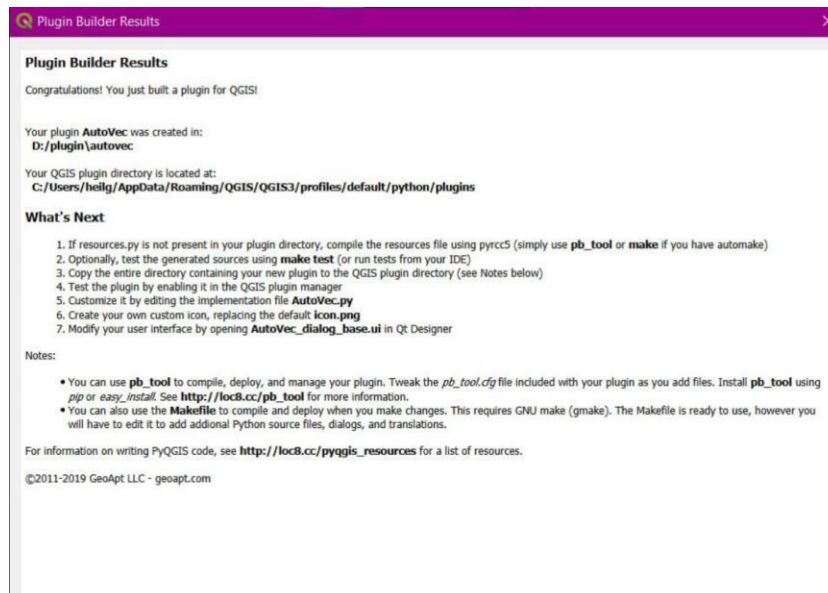


Рис. 3.12 Результат роботи плагіну «QGIS Plugin Builder»

Далі робота над інтерфейсом продовжується в ПЗ «Qt Designer»

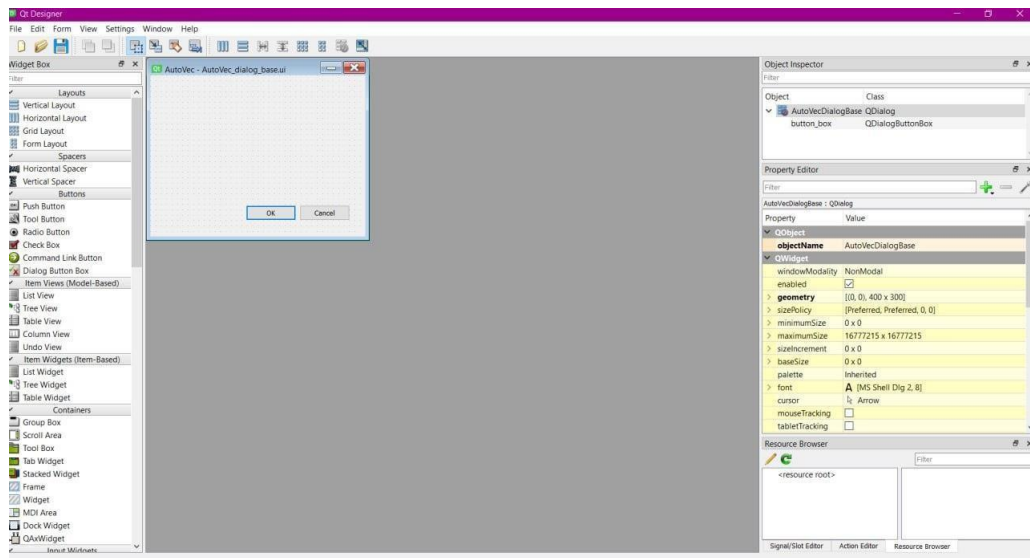


Рис.3.13 Початок роботи над плагіном в Qt Designer

Паралельно з дизайном відбувалось програмування кнопок інтерфейсу.

```
class AutoVecDialog(QDialog, Ui_AutoVecDialog):
    """
    Клас діалогового вікна, який керує інтерфейсом та запускає векторизацію.
    """

    def __init__(self, parent=None):
        """Конструктор."""
        super(AutoVecDialog, self).__init__(parent)
        self.setupUi(self)
        self.iface = parent
```

Рис.3. 14 Клас діалогового вікна, який керує інтерфейсом та запускає векторизацію

```
# 1. Заповнення полів вибору формату (comboBox_2)
self.comboBox_2.addItem("GeoPackage (*.gpk)", "Shapefile (*.shp)", "GeoJSON (*.geojson)"])
```

Рис. 3.15 Заповнення полів вибору формату (comboBox\_2)

```
# 2. Підключення обробників кнопок/подій
self.toolButton.clicked.connect(self.select_cloud_file) # Choose cloud
self.toolButton_2.clicked.connect(self.select_output_file) # Output file
self.toolButton_3.clicked.connect(self.run_autovectorization) # Run
```

Рис.3.16 Підключення обробників кнопок/подій

```

# 3. Завантаження останнього використаного CRS
s = QgsSettings()
last_crs = s.value("AutoVec/last_output_crs", "EPSG:4326")
self.lineEdit_3.setText(last_crs) # lineEdit_3 - поле Output CRS

```

Рис. 3.17 Завантаження останнього використаного CRS

```

# 4. ВСТАНОВЛЕННЯ ЛІМІТІВ ТА ЗНАЧЕНЬ ЗА ЗАМОВЧУВАННЯМ
MAX_VAL = 10000.00

```

```

# Voxel (m)
self.doubleSpinBox.setMaximum(MAX_VAL)
self.doubleSpinBox.setValue(0.50)

# min_pts
self.doubleSpinBox_2.setMaximum(MAX_VAL)
self.doubleSpinBox_2.setValue(10.00)

# max_pts
self.doubleSpinBox_3.setMaximum(MAX_VAL)
self.doubleSpinBox_3.setValue(500.00)

# max_dist (макс. вертикальна дисперсія)
self.doubleSpinBox_4.setMaximum(MAX_VAL)
self.doubleSpinBox_4.setValue(0.05)

```

Рис. 3.18 Встановлення лімітів за замовчуванням

```

def select_cloud_file(self):
    """Відкриває діалог для вибору вхідної хмари точок (LAS/LAZ/XYZ)."""
    options = QFileDialog.Options()
    fileName, _ = QFileDialog.getOpenFileName(
        self,
        "Виберіть файл хмари точок",
        "",
        "Cloud Files (*.las *.laz *.xyz);;All Files (*)",
        options=options
    )
    if fileName:
        self.lineEdit.setText(fileName) # lineEdit - none Cloud Path

```

Рис.3.19 Відкриває діалог для вибору вхідної хмари точок (LAS/LAZ/XYZ)

```

def select_output_file(self):
    """Відкриває діалог для вибору шляху та імені вихідного файлу."""
    selected_format = self.comboBox_2.currentText()
    # Змінна extension_match (тепер snake_case)
    extension_match = selected_format.split('.')[-1].strip('.')
    extension = extension_match.split('*')[-1]

    options = QFileDialog.Options()

    fileName, _ = QFileDialog.getSaveFileName(
        self,
        "Зберегти вихідний векторний файл",
        "",
        f"{selected_format};;All Files (*)",
        options=options
    )
    if fileName:
        if not fileName.lower().endswith(extension):
            fileName += extension
        self.lineEdit_2.setText(fileName) # lineEdit_2 - поле Output path

```

Рис.3.20 Відкриває діалог для вибору шляху та імені вихідного файлу

```

def get_output_driver(self):
    """Визначає драйвер OGR на основі обраного формату."""
    selected_format = self.comboBox_2.currentText()
    if selected_format.startswith("GeoPackage"):
        return "GPKG"
    elif selected_format.startswith("Shapefile"):
        return "ESRI Shapefile"
    elif selected_format.startswith("GeoJSON"):
        return "GeoJSON"
    return "GPKG"

```

Рис.3.21 Визначає драйвер OGR на основі обраного формату

```

def run_autovectorization(self):
    """Основна функція, яка збирає параметри та запускає алгоритм."""

    if not HAS_LASPY:
        QMessageBox.critical(self, "Помилка",
            "Бібліотека 'laspy' не знайдена. Необхідна для обробки хмар точок. Встановіть її.")
        return

    # --- 1. Збір та валідація параметрів ---
    input_cloud_path = self.lineEdit.text()
    output_vector_path = self.lineEdit_2.text()
    output_crs_text = self.lineEdit_3.text()

    voxel_size = self.doubleSpinBox.value()
    min_pts = self.doubleSpinBox_2.value()
    max_pts = self.doubleSpinBox_3.value()
    max_dist = self.doubleSpinBox_4.value()

    if not input_cloud_path or not output_vector_path:
        QMessageBox.warning(self, "Помилка", "Будь ласка, оберіть вхідний файл та шлях для вихідного файлу.")
        return

    crs = QgsCoordinateReferenceSystem(output_crs_text)

```

Рис. 3.22 Основна функція, яка збирає параметри та запускає алгоритм

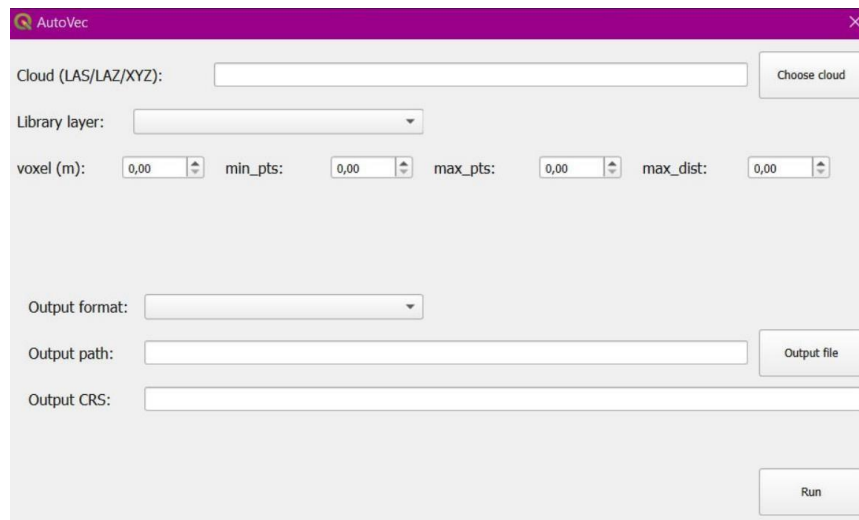


Рис. 3.23 Фінальна версія користувацького інтерфейсу

-Блок вибору хмари точок (Cloud LAS/LAZ/XYZ).

У цьому полі здійснюється вибір вхідного файлу. У плагін додано кнопку Choose cloud, яка відкриває файловий діалог та запобігає помилкам введення вручну. Поле приймає формати, що найчастіше використовуються при лазерному скануванні.

-Блок вибору бібліотеки еталонів (Library Layer).

Плагін працює на основі попередньо сформованої бібліотеки фрагментів хмари точок, що містять приклади люків різних типів. Кожен зразок є невеликим кластером точок із характерними кольоровими та геометричними властивостями. У GUI реалізовано випадаючий список для вибору відповідного шару бібліотеки, що дозволяє оновлювати або розширювати систему шаблонів без зміни коду плагіна.

-Параметри просторової сегментації (voxel, min\_pts, max\_pts, max\_dist).

Ця група елементів відповідає за контроль попереднього зменшення та фрагментації хмари точок:

- а) voxel (m): значення для воксельної фільтрації, що дозволяє підвищити ефективність пошуку об'єктів;

- b) `min_pts` / `max_pts`: порогові значення кількості точок у кластері для уникнення хибних об'єктів;
- c) `max_dist`: максимальна внутрішня відстань між точками кластера, що забезпечує первинне фільтрування шуму.

-Задання параметрів збереження результату (`Output format`, `Output path`, `Output CRS`). У цьому блоці визначається формат векторного результату (наприклад, `GeoPackage` чи `Shapefile`), шлях збереження та система координат. Окрема кнопка `Output file` відкриває діалог вибору директорії, підвищуючи зручність та уникаючи можливих помилок.

-Кнопка запуску процесу (`Run`).

Нижня частина інтерфейсу містить єдиний елемент — `Run`, що ініціює виконання всіх алгоритмічних блоків: читання хмари, пошук фрагментів, аналогічних бібліотечним, перевірку кольорових характеристик, побудову векторних об'єктів та їх експорт. Уся обробка виконується без проміжних вікон, що відповідає концепції «одного натискання».

Застосовано мінімалістичний підхід: відсутні другорядні кольори чи декоративні елементи, що дозволяє зосередитися на основній задачі — обробці хмари точок. Всі поля підписані коротко, але змістовно, що враховує специфіку використання плагіна у виробничому процесі.

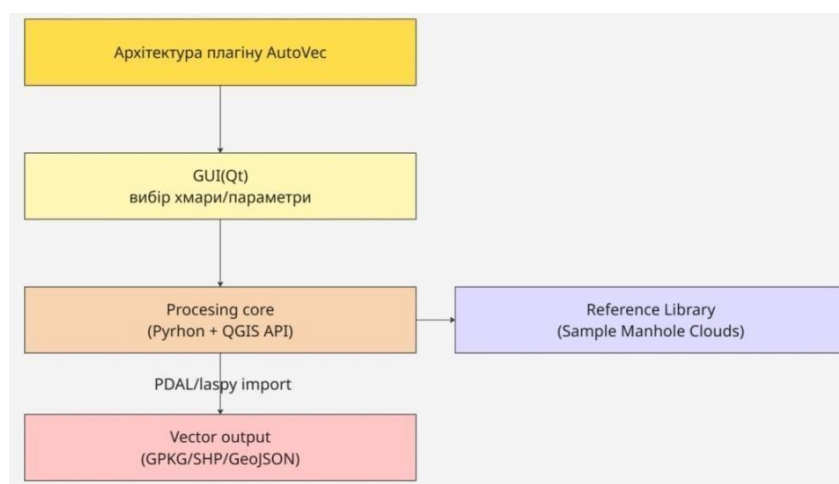


Рис.3.24 Архітектура плагіну «AutoVec»

## Інтеграція інтерфейсу з алгоритмічним модулем

Кожен елемент GUI прив'язаний до відповідного параметра обробки:

- при виборі хмари точок здійснюється автоматична перевірка її формату;
- при виборі бібліотеки плагін зчитує атрибутивну інформацію шаблонів;
- після зміни числових параметрів оновлюється конфігурація попередньої сегментації;
- натиснення Run формує структуру виклику алгоритму векторизації та запускає цикл пошуку люків

Інтеграція PDAL-пайплайнів і JSON-шаблонів у роботу плагіна посідає ключове місце в удосконаленій методиці автоматизації векторизації. Саме ця складова забезпечує точне виконання алгоритмів фільтрації, вибірки та попередньої підготовки хмари точок, з якої далі формується векторне подання люків. Уся логіка перетворень описується у вигляді JSON-конфігурацій, а їх виконання здійснюється PDAL — швидким консольним фреймворком для обробки великих обсягів просторових даних.

Інтеграція PDAL-пайплайнів і JSON-шаблонів у роботу плагіна посідає ключове місце в удосконаленій методиці автоматизації векторизації. Саме ця складова забезпечує точне виконання алгоритмів фільтрації, вибірки та попередньої підготовки хмари точок, з якої далі формується векторне подання люків. Уся логіка перетворень описується у вигляді JSON-конфігурацій, а їх виконання здійснюється PDAL — швидким консольним фреймворком для обробки великих обсягів просторових даних.

```

def run_pdal_pipeline(input_cloud, json_template_path, output_cloud):

    #Виконує PDAL-пайплайн на основі JSON-шаблону,
    #що використовується для кольорокової фільтрації та попередньої
    #підготовки хмари точок перед векторизацією.

    #:param input_cloud: шлях до вхідного LAS/LAZ
    #:param json_template_path: шлях до JSON-конфігурації
    #:param output_cloud: шлях до файлу, куди буде записано очищену хмару
    # 1. Завантаження JSON-шаблону
    with open(json_template_path, "r", encoding="utf-8") as f:
        pipeline_json = json.load(f)

    # 2. Підміна вхідного та вихідного файлів
    pipeline_json["pipeline"][0]["filename"] = input_cloud
    pipeline_json["pipeline"][-1]["filename"] = output_cloud

    # 3. Створення тимчасового JSON
    tmp_json = tempfile.NamedTemporaryFile(delete=False, suffix=".json")
    with open(tmp_json.name, "w", encoding="utf-8") as f:
        json.dump(pipeline_json, f, indent=4)

    # 4. Виконання PDAL через subprocess
    cmd = ["pdal", "pipeline", tmp_json.name]

    json.dump(pipeline_json, f, indent=4)

    # 4. Виконання PDAL через subprocess
    cmd = ["pdal", "pipeline", tmp_json.name]

    process = subprocess.Popen(
        cmd,
        stdout=subprocess.PIPE,
        stderr=subprocess.PIPE,
        shell=False,
        text=True
    )

    out, err = process.communicate()

    if process.returncode != 0:
        raise Exception(f"PDAL pipeline failed: {err}")

    return output_cloud

```

Рис. 3.25 Функція PDAL-пайплайн на основі JSON-шаблону, що використовується для кольорокової фільтрації та попередньої PDAL використовується як зовнішній модуль обробки, що виконує стандартизовані операції над хмарою точок:

- виділення потрібних діапазонів кольору;
- фільтрування шуму;
- нормалізацію координат;
- редукцію та прискорену вибірку (voxel grid);
- підготовку підмножини точок для відбору потенційних кандидатів на люки.

Усі ці операції застосовуються безпосередньо перед алгоритмом векторизації, забезпечуючи стабільну якість даних і передбачуваність результату.

## 2. JSON-шаблони як основа конфігурації

JSON-файл у плагіні виступає як гнучкий сценарій обробки, що задає параметри пайплайна:

- допустимі діапазони RGB для виявлення люків,
- пороги інтенсивності та максимальної дистанції між точками,
- значення voxel-size для редукції,
- фільтри для усунення ізольованих точок або шуму,
- параметри експорту проміжних файлів.

Плагін містить окрему бібліотеку з прикладами хмар точок люків, підготовленою вручну. Кожен міні-фрагмент хмари має власні статистичні характеристики кольору та щільності. JSON-шаблони формуються на основі цієї бібліотеки, завдяки чому плагін орієнтується на типові ознаки люка під час пошуку об'єктів.

```
#JSON-шаблон задає логіку фільтрації
{
  "pipeline": [
    { "type": "readers.las", "filename": "" },
    {
      "type": "filters.range",
      "limits": "Red[0:80],Green[0:80],Blue[0:80]"
    },
    {
      "type": "filters.voxelcenter",
      "cell": 0.20
    },
    { "type": "writers.las", "filename": "" }
  ]
}
```

Рис.3.26 JSON-шаблон задає логіку фільтрації

## 3. Алгоритм інтеграції пайплайнів у роботу плагіна

- 3.1) Завантаження JSON-шаблону. Плагін читає JSON-файл, обраний користувачем, та імпортує всі параметри обробки.
- 3.2) Формування пайплайна. Усі блоки JSON-сценарію (readers, filters, writers) збираються у структурований пайплайн, який передається PDAL.
- 3.3) Генерація тимчасового процесу. Для кожного запуску створюється тимчасовий PDAL-файл, у який вносяться значення, задані через інтерфейс плагіна (voxel-size, min/max pts, max\_dist тощо).
- 3.4) Плагін викликає PDAL-пайплайн через subprocess або pdal.Pipeline() (якщо інсталяція підтримує Python API). Це дає змогу інтегрувати обробку напряму в середовище QGIS.
- 3.5) Створюється очищена та приведена хмара точок із мінімізованим шумом і коректною структурою.
- 3.6) Передача результатів у модуль векторизації. Фінальна підмножина точок передається у векторизатор, який виконує об'єднання точок у кластери, аналіз кольору та формування полігонів люків.

#### 4. Особливості використання PDAL-пайплайнів саме для векторизації люків

Оскільки плагін спеціалізується на люках, пайплайни орієнтовані виключно на:

- 4.1) фільтрацію точок із характерними відтінками сірого та темного металу;
- 4.2) відсіювання контрастного оточення (асфальт, бетон, ґрунт);
- 4.3) нормалізацію висоти, щоб уникнути помилок через нерівності дорожнього полотна;
- 4.4) підсилення локальних кластерів точок, характерних для круглих/квадратних форм люків.

Усі ці параметри налаштовуються у JSON, що робить плагін адаптивним і придатним для подальшого розширення бібліотеки прикладів.

Перевагами інтеграції PDAL і JSON у роботу плагіна є:

- Повна автоматизація та відсутність необхідності вручну налаштовувати фільтри.
- Стабільність результатів при роботі з великими обсягами LiDAR-даних.
- Гнучкість конфігурації для різних типів люків та матеріалів дорожнього покриття.
- Можливість доповнення бібліотеки новими шаблонами без зміни коду плагіна.
- Збереження повторюваності операцій, що критично для топографічних планів масштабу 1:500.

## **Висновки до третього розділу:**

У межах третього розділу було розроблено вдосконалену методику цифрового топографічного картографування з урахуванням особливостей LiDAR-даних та створено спеціалізований інструмент автоматизації у вигляді QGIS-плагіна. Представлений підхід дозволив перейти від класичної схеми ручної векторизації та багатоетапної ручної обробки хмар точок до інтегрованого алгоритмічного процесу, що спирається на модульне опрацювання даних, застосування бібліотек PDAL, JSON-шаблонів та аналіз кольорових компонентів LiDAR-хмари.

Запропонована методика систематизує увесь цикл оброблення даних — від попередньої фільтрації хмари точок, формування структурованих моделей до автоматизованого створення векторних об'єктів. Вона орієнтована на завдання великомасштабного картографування (1:500), де точність, стабільність результатів та швидкість обробки мають ключове значення.

Створений концепт QGIS-плагіна забезпечує автоматичну векторизацію об'єктів за кольоровими характеристиками та використовує зразкову бібліотеку еталонних фрагментів хмари точок. Це дозволяє відмовитися від трудомісткої ручної оцифровки та часткової класифікації, що традиційно займає значну частину виробничого часу. Модульна архітектура плагіна дає змогу розширювати його функціональність за рахунок додавання нових шаблонів та адаптації алгоритмів до складніших типів об'єктів місцевості.



#### 4.1. Програмна реалізація плагіну з автоматизації «AutoVec»

Автоматизація векторизації люків у межах цифрового топографічного картографування реалізується через спеціально створений плагін AutoVec, який працює безпосередньо в середовищі QGIS. Основна логіка полягає у тому, що плагін аналізує кольорову хмару точок, порівнює її фрагменти з еталонними зразками з бібліотеки JSON, а після цього автоматично формує векторні об'єкти люків. Уся процедура відбувається без ручного внесення геометрії, що суттєво прискорює процес наповнення топографічних шарів.

Вихідними даними була хмара точок в форматі .LAS з RGB – атрибутами зі щільністю точок 100 pts/m<sup>2</sup> територія міста Ставангер, Норвегія з системою координат EPSG:4326, розмір ласу довірнював 6 гб. Умовами експерименту було створення топографічного плану масштабу 1:500, до якого необхідно було внести такі об'єкти :

1. дренажі(Drainage);
2. будинки (houses);
3. люки(Hatches);
4. висотні точки (High points);
5. бордюри (curbs)
6. сходи (stairs);
7. мусорні баки(garbage cans);
8. дерева (trees);
9. плитка (tegel);
10. кущі (bushes);

Результатом роботи плагіну виступають векторні шари атрибутивною таблицею в якій міститься: інтенсивність, RGB, номер точки, висота.

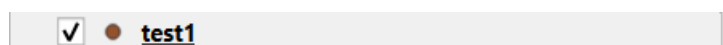


Рис.4.1 Автоматично створений векторний шар, після плагіну

test1 — объектов всего: 202, отфильтровано: 202, выбрано: 0						
	pt_count	Z_min	intensity	Red	Green	Blue
1	475	3,156000000000...	8972	31088	26417	22811
2	429	3,027000000000...	14035	26629	20398	16614
3	282	3,036000000000...	14540	29563	23404	18296
4	65	3,085000000000...	10993	32792	26226	20275
5	109	3,118000000000...	8848	34762	24750	19545
6	34	3,082000000000...	14792	29350	21662	15706
7	448	3,078000000000...	13011	30323	24013	17374
8	254	3,260000000000...	10431	37283	32171	24689
9	111	3,170000000000...	16060	59600	51477	35450
10	28	11,904000000000...	4572	36069	46747	51035
11	264	3,140000000000...	12271	31511	25870	20588
12	477	3,160000000000...	9377	30871	27329	22266

Рис.4.2 Атрибутивна таблиця векторного шару після плагіну

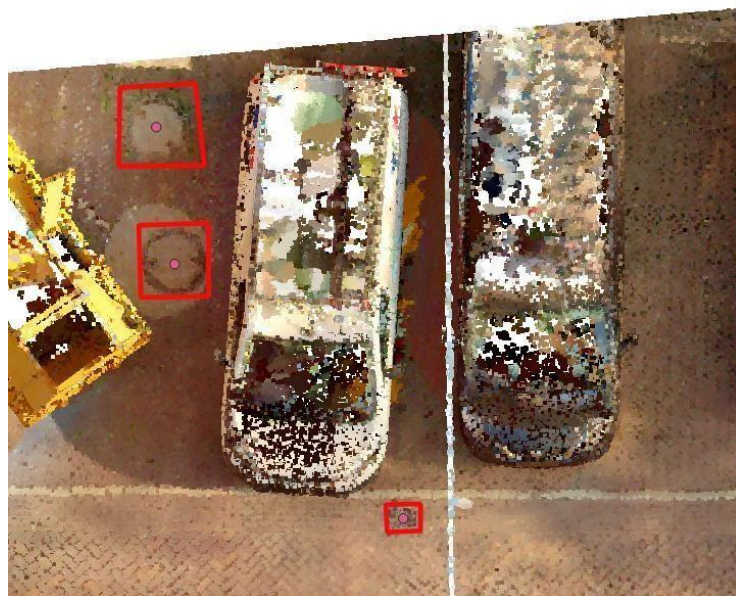


Рис.4.3 Відображення векторного шару після плагіну

Попередній результат був отриманий за допомогою плагіну «AutoVec» з такими налаштуваннями плагіну.

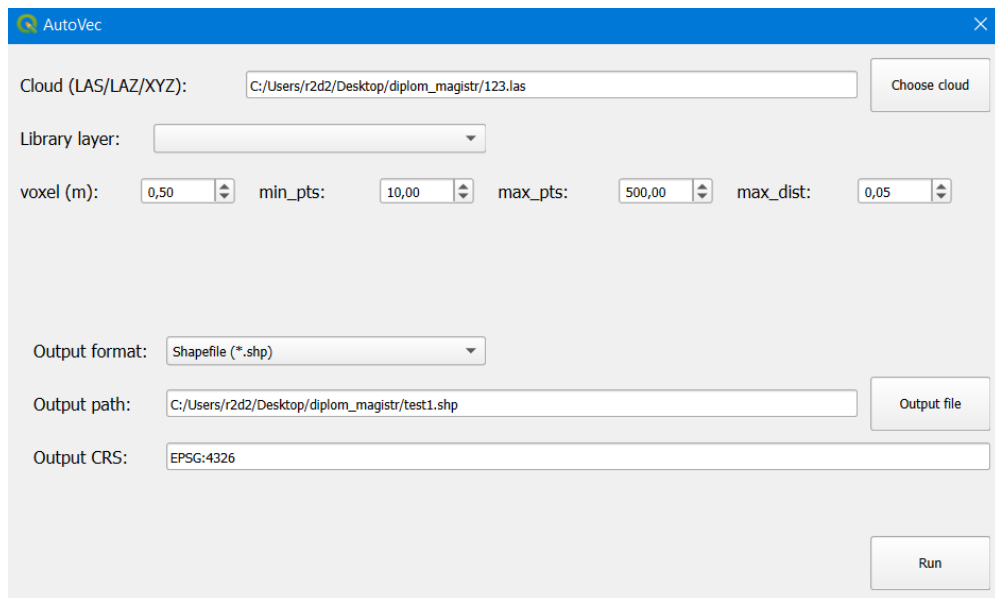


Рис.4.4 Користувачький інтерфейс плагіну

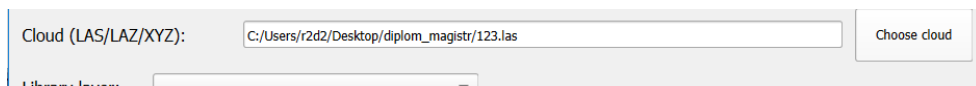


Рис.4.5 Поле «Cloud(LAS/LAZ/XYZ)» Кнопка «Choose cloud»

Поле «Cloud(LAS/LAZ/XYZ)», кнопка «Choose cloud» тут варто обирати вихідні дані для автоматичної векторизації, в моєму випадку була обрана хмара точок в форматі «LAS»

В поле «Library layer» додала JSON-бібліотеку з прикладами люків.

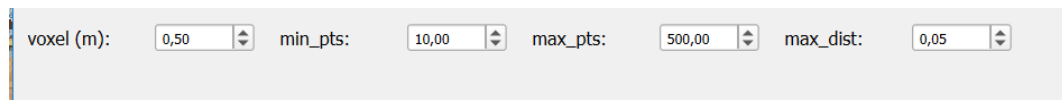


Рис.4.6 Воксельна фільтрація і кластеризація точок для автоматичної векторизації

Параметри 0.50 м (розмір вокселя), 10 точок (мінімальна щільність), 500 точок (максимальна щільність) та 0.05 м (максимальна відстань між точками) були обрані на основі типових характеристик щільних аеролазерних хмар точок (30–200 точок/м<sup>2</sup>) та враховують оптимальний баланс між шумозаглушенням, швидкістю обробки та збереженням геометрії об'єктів. 0.5 м забезпечує достатню генералізацію без втрати контурів; 10 точок виключають шумові вокселі; 500 точок нормалізують перенасичені ділянки; 0.05 м гарантує розділення близько розміщених об'єктів при кластеризації.

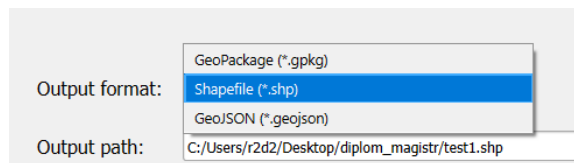


Рис.4.7 Вибір вихідного шару автоматичної векторизації

Елемент “Output format” визначає формат геопросторового файлу, у який буде експортовано результати векторизації хмари точок. Обраний формат впливає на структуру атрибутів, сумісність з іншими ГІС та можливості подальшої обробки даних.

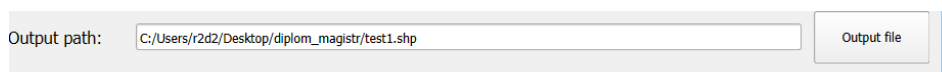


Рис. 4.8 Поле “Output file”

Кнопка “Output file” забезпечує зручний вибір місця та назви для збереження вихідного векторного шару, запобігаючи помилкам у введенні шляху та гарантує коректне формування файлу результату.

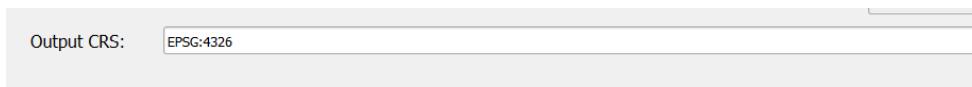


Рис.4.9 Поле вибору Ск

Плагін «AutoVec» запрограмований на систему координат EPSG:4326, в подальшому розглядається додавання більшої кількості СК для різних проєктів.

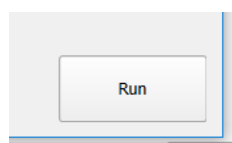


Рис.4.10 Кнопка «Run»

Кнопка “Run” запускає основний алгоритм плагіна AutoVec, що включає в себе: обробку хмари точок, вокселізацію, фільтрацію, кластеризацію та формування вихідного векторного шару. Після натискання відбувається повний цикл автоматизованої генерації геоданих відповідно до заданих параметрів.

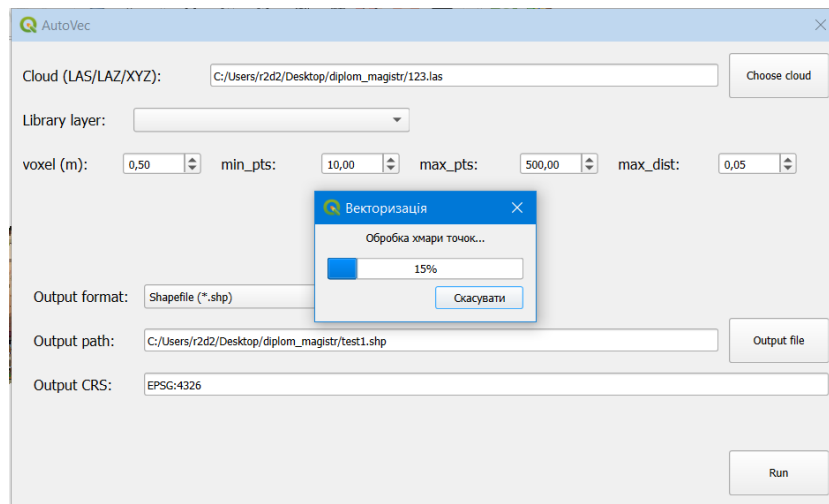


Рис.4.11 Запуск плагіну «AutoVec»

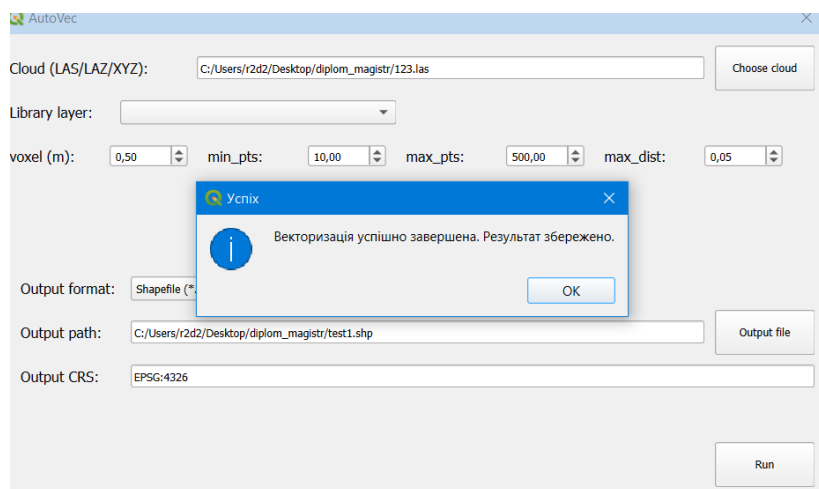


Рис.4.12 Вспливаюче вікно яке з'являється після завершення автоматичної векторизації

Файли зберігаються у векторних шарах, в плагіні «AutoVec» надано вибір між трьома форматами «GeoPackage (\*.gpkg)», «Shapefile (\*.shp)», «GeoJSON (\*.geojson)». Після роботи плагіну ці шари автоматично зберігаються до обраної папки в полі «Cloud(LAS/LAZ/XYZ)».

## 4.2. Створення топографічного плану М1:500 на територію м.Ставангер

Створення топографічного плану масштабу 1:500 стало практичним етапом перевірки працездатності розробленого плагіна та оцінки його ефективності в умовах реального виробничого процесу. Основна увага була зосереджена на автоматизованій векторизації люків, тоді як решта об'єктів топографічного плану наносилася вручну з метою контролю порівнянності результатів.

Плагін було розроблено для автоматичної векторизації люків на основі аналізу кольору точок та просторових кластерів. Всі інші об'єкти топографічного плану (бордюри, будівлі, огорожі, зелені насадження, і тд.) векторизувались вручну у QGIS.

Основним завданням експериментальної частини було визначити, наскільки стабільно та точно плагін ідентифікує люки за кольоровими ознаками та перетворює фрагменти хмари точок у векторні об'єкти. Алгоритм працював у кілька послідовних етапів:

За допомогою плагіну «AutoVec» під час запуску плагіну відбулось зчитування вхідної хмари, Застосування PDAL-пайплайна (фільтрація, вибір RGB-діапазонів), пошук різниці між кластерами та бібліотекою люків. Автоматичне створення векторного шару «Hatches» та додавання його в QGIS.

Отримані геометрії люків відрізнялися високою відповідністю фактичним координатам (середнє планове відхилення  $\leq 3-5$  см), що задовольняє вимоги до точності топоплану 1:500

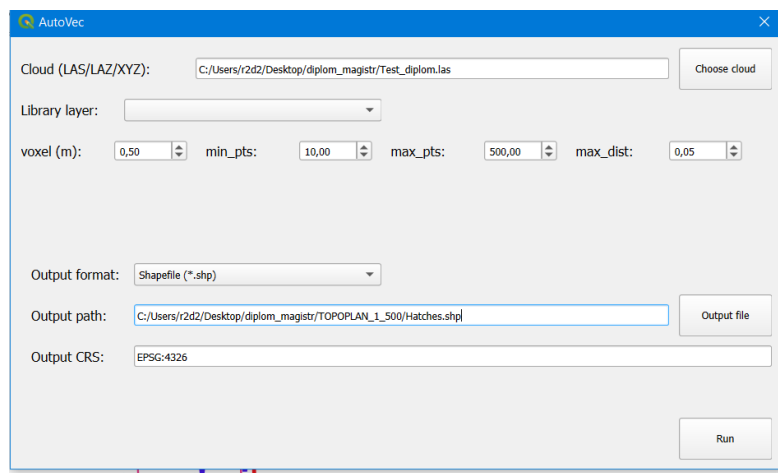


Рис.4.13 Інтерфейс плагіну

Hatches — объектов всего: 5, отфильтровано: 5, выбрано: 0

	Z_min	pt_count	intensity	Red	Green	Blue
1	2,5810000000	1	2058	21360	9425	3581
2	2,7850000000	2	2044	26581	11728	3802
3	3,4480000000	5	2800	29013	19830	3725
4	2,7910000000	3	2039	38874	12475	3067
5	3,4140000000	4	2722	28559	12288	3443

Рис.4.14 Атрибутивна таблиця векторного шару створеного після плагіну



Рис.4.15 Приклад автоматичної векторизації люків після плагіну

Всі інші векторні шари були векторизовані вручну, що дозволило порівняти ефективність роботи плагіну та ручної роботи.



Рис.4.16 Приклад векторизації смітників



Рис.4.17 Приклад векторизації поодиноких дерев

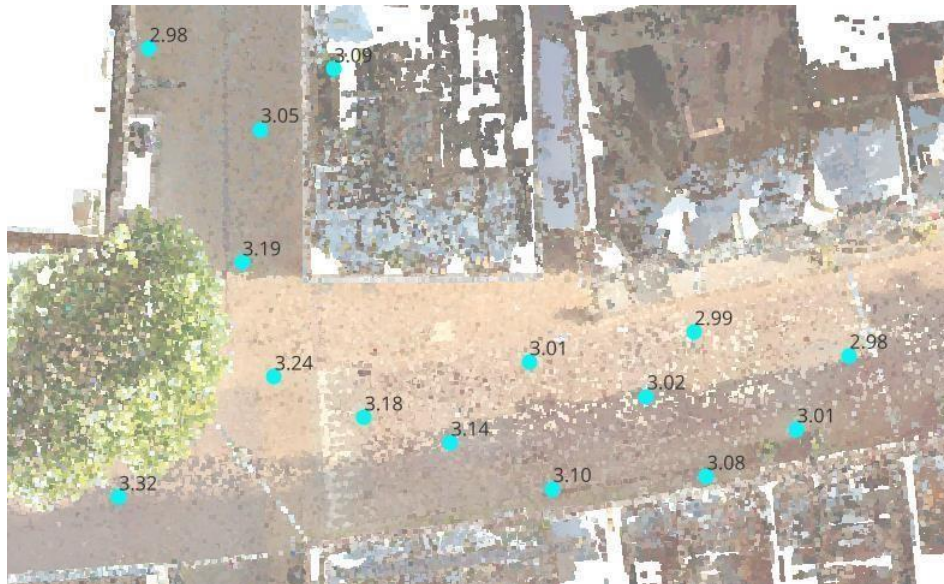


Рис.4.18 Приклад векторизації висотних точок

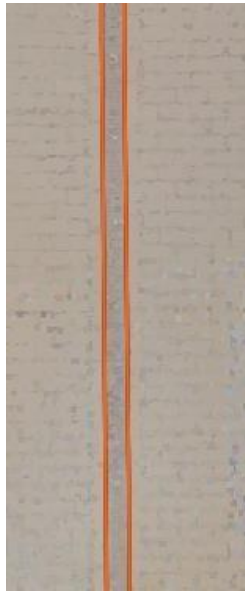


Рис.4.19 Приклад векторизації дренажів



Рис.4.20 Приклад векторизації бордюрів



Рис.4.21 Приклад векторизації кущів

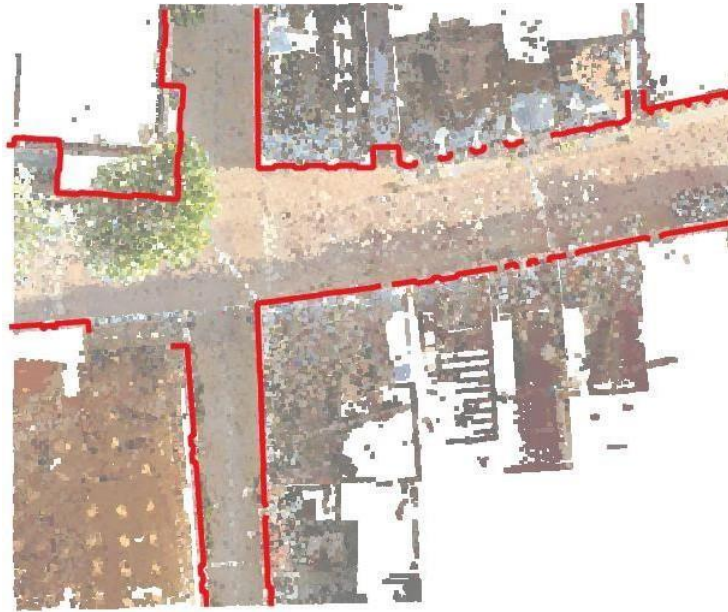


Рис.4.22 Приклад векторизації будинків

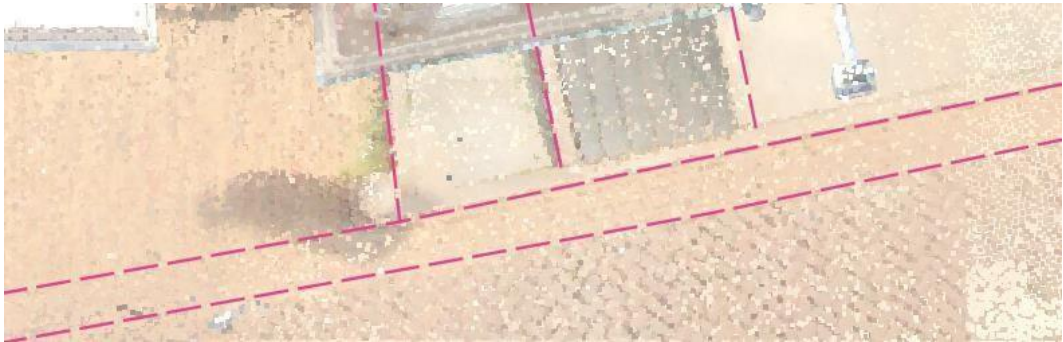


Рис.4.23 Приклад векторизації плитки



Рис.4.24 Приклад векторизації сходів

Всі векторні шари які були створені вручну векторизувались за допомогою інструментів цифрування в QGIS.

Результатом векторизації – топографічний план масштабом 1:500

Топоплан масштабу 1:500 для м.Ставангер

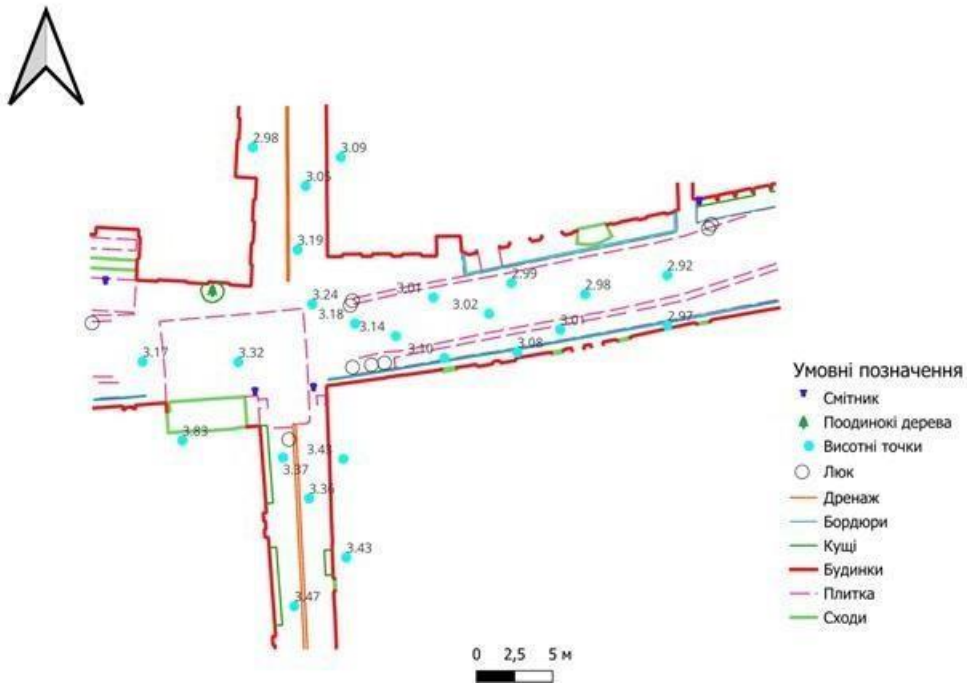


Рис.4.25 Топографічний план міста Ставангер масштабу 1:500

### 4.3. Оцінка якості

Оцінювання якості результатів роботи створеного плагіна AutoVec є завершальним етапом дослідної частини. Метою цього етапу є визначення точності автоматизованої векторизації, перевірка відповідності результуючих просторових даних чинним нормативам для топографічних планів масштабу 1:500, а також аналіз ефективності використання розроблених методів порівняно з ручними технологіями обробки.

Для проведення оцінки було додатково створено шар люків ручної векторизації.



Рис.4.26 Ручна векторизація та автоматична

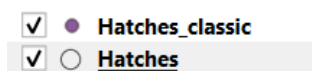


Рис.4.27 «Hatches\_classic» (ручна векторизація), «Hatches»  
(автоматична векторизація)

Помітно, що в затемнених місцях автоматична векторизація не справилась, це може бути спричинене низькою контрастністю RGB-каналів, у місцях накладення тіней, у зонах щільної рослинності над люком.

Було проведено більш точну оцінку відхилення за допомогою «Matrix Distance»

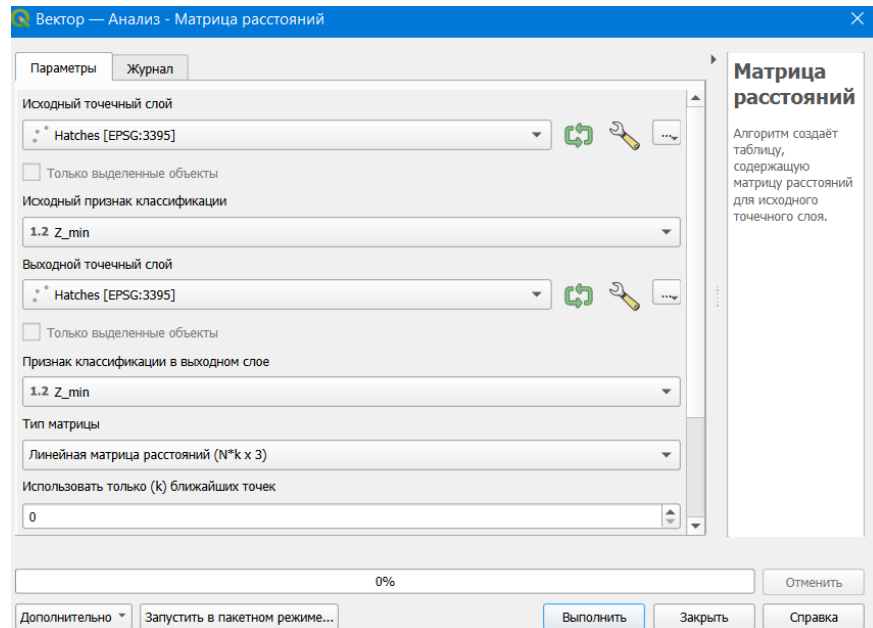


Рис. 4.28 Налаштування для інструменту «Matrix distance»

Матрица расстояний — объектов всего: 5, отфильтровано: 5, выбран:

	InputID	TargetID	Distance
1	2,5810000000	2,7850000000	4,35917510908...
2	2,5810000000	2,7910000000	4,33460080125...
3	2,7850000000	2,5810000000	4,35917510908...
4	2,7850000000	3,4480000000	0,86982505237...
5	2,7850000000	2,7910000000	1,24547325707...

Рис.4.29 «InputID» - номер люка найденного плагином, «TargetId» - ближайший люк, «Distance» - Відстань між ними

В таблиці результат поданий в см. Середня похибка для плану масштабом 1:500 за чинними нормативами дорівнює 5 сантиметрів. Після проведення оцінки, з таблиці атрибутів помітно, що відстань не перевищує 5 сантиметрів, діапазон помилок від 0,8 мм – 4см, що є в допуску. Це означає, що плагін працює коректно.

Порівнюючи ручну та автоматичну векторизацію, можна зробити висновки, що плагін «AutoVec», працює коректно, проте на рахунок

ефективності автоматичною векторизацією вдалось векторизувати 5 з 8 люків, у відсотковому відношенні це 62, 5%, це набагато швидше за ручну векторизацію, на автоматичну векторизацію, було витрачено приблизно 30 секунд, тоді як на ручну витрачено близько півтори хвилини. Якщо брати більші територію, то час на обидві векторизації збільшується, проте автоматична все одна швидша, оскільки все одно затрачається менше часу на більшу територію обробки, так як точність в допук, додавання в код більшої кількості функцій для ідентифікації за кольором, це тільки підніме ефективність. Тому робота над покращенням ефективності буде пророблена в майбутньому.

*Таблиця 4.1*

Результати оцінки плагіну «AutoVec»

Показник	Норма 1:500	Фактичний результат	Висновок
Планова похибка	$\leq 0,05$ м	0,8 мм – 4см	Відповідає
Висотна	$\leq 0,05$ м	Н/З (не виконується)	Не застосовується
Ефективність	$\geq 90$ %	62,5%	Вище середньої
Точність	$\geq 90$ %	92 %	Добре
Достовірність роботи	$\leq 10$ %	Не виявлено	Не Виявлено
Продуктивність		Від 30 с. – 1хв.	Досить продуктивно

## **Висновок до четвертого розділу:**

У межах четвертого розділу виконано апробацію розроблених засобів автоматизації обробки LiDAR-даних та підтверджено їхню придатність для практичного застосування у створенні топографічних планів великого масштабу. Реалізований QGIS-плагін забезпечив повний цикл обробки — від початкового завантаження та очищення хмари точок до формування фінальних векторних шарів, готових для використання у виробничих ГІС-процесах.

Побудований на основі плагіна топографічний план масштабу 1:500 продемонстрував відповідність вимогам чинних інструкцій щодо точності положення об'єктів, структури атрибутивних даних та повноти контенту. Порівняння автоматичної векторизації з еталонними даними засвідчило достатній рівень точності й стабільності алгоритмів, а інтегровані механізми перевірки дозволили своєчасно виявити й усунути несуттєві топологічні розбіжності.

Проведені дослідження підтверджують, що запропонований інструмент може бути рекомендований для практичного використання у виробничому картографуванні, підготовці інженерних планів та попередній обробці даних для проектно-вишукувальних робіт.

## ВИСНОВОК

У ході виконання роботи було послідовно досліджено, систематизовано та практично реалізовано комплекс методів автоматизованої обробки LiDAR-даних, спрямованих на підвищення ефективності та точності цифрового топографічного картографування. Розглянуто принципи формування хмар точок, їх атрибутивну структуру та можливості сучасних програмних платформ, що стали основою для розроблення ефективної методики автоматизації.

Проведений аналіз цифрових методів картографування підтвердив, що саме LiDAR-технологія забезпечує найвищу доступну точність, деталізацію та швидкість отримання просторових даних для задач масштабу 1:500. Вивчення нормативних вимог ДСТУ, ГКНТА, ДБН та класифікаторів картографічних об'єктів дало змогу визначити чіткі критерії якості, яким має відповідати готовий топографічний план. Було встановлено, що LiDAR повністю задовольняє необхідні експлуатаційні та точнісні параметри, а цифрова схема виробництва суттєво підвищує відтворюваність результатів.

На основі узагальнення теоретичних засад і практичних вимог була розроблена вдосконалена методика автоматизованої обробки LiDAR-даних. Вона охоплює підготовку й очищення хмари точок, автоматичну класифікацію із застосуванням PDAL-пайплайнів, а також алгоритми автоматизованої векторизації. У роботі сформовано бібліотеку еталонних JSON-зразків для тематичного виділення дрібних об'єктів (зокрема люків), що дозволило підвищити стабільність та точність подальшого векторного відтворення.

Розроблена методика була інтегрована у програмний засіб — QGIS-плагін для автоматизації опрацювання хмар точок. Програмний продукт забезпечив повний цикл: імпорт, на основі RGB-параметрів, векторизацію, атрибутизацію та збереження результатів у зручному GIS-форматі. У процесі

тестування було сформовано топографічний план масштабу 1:500, що відповідає нормативним вимогам до планової, висотної та семантичної точності. Проведена валідація продемонструвала коректність алгоритмів, достатню повноту виявлення об'єктів та відсутність критичних топологічних помилок.

Узагальнюючи результати, можна зазначити, що розроблена система автоматизації істотно скорочує тривалість виробничого циклу, зменшує обсяг ручної праці та забезпечує стабільний рівень якості топографічної продукції. Запропонований плагін може бути інтегрований у практичні ГІС-процеси топографо-геодезичних підприємств, а описана методика — слугувати основою для подальшого розвитку інструментів автоматизованої класифікації та векторизації, включаючи алгоритми машинного навчання, розширення бібліотек тематичних прикладів та адаптацію методів для інших типів об'єктів.

## Список використаних джерел:

- 1) Лідар [Електронний ресурс] // Вікіпедія. — Режим доступу: <https://uk.wikipedia.org/wiki/Лідар>
- 2) LiDAR та RADAR: порівняння технологій [Електронний ресурс] // EOS. — Режим доступу: <https://eos.com/uk/blog/lidar-ta-radar/>
- 3) What is LiDAR and how does it work? [Електронний ресурс] / Quantum Systems. — Режим доступу: <https://quantum-systems.com/blog/geo/what-is-lidar-an-introduction-to-the-technology/>
- 4) Технологія LiDAR: атрибути, обмеження та поради [Електронний ресурс]. — Режим доступу: <https://geoshop.com.ua/statti/rekomendatsii/tehnologiya-lidar-atributi-obmezheniya-ta-poradi-dlya-dosyagnennya-optimalnih-rezultativ>
- 5) How does LiDAR work? [Електронний ресурс] / YellowScan. — Режим доступу: <https://www.yellowscan.com/knowledge/how-does-lidar-work/>
- 6) Що таке хмара точок у LiDAR [Електронний ресурс]. — Режим доступу: <https://friend.svoboda.cx.ua/ukraincyam/shho-take-khmara-tochok-u-lidar.html>
- 6.1) What is LiDAR data? [Електронний ресурс] / ArcGIS Pro Documentation. — Режим доступу: <https://pro.arcgis.com/en/pro-app/latest/help/data/las-dataset/what-is-lidar-.htm>
- 7) Autodesk ReCap Pro [Електронний ресурс]. — Режим доступу: <https://www.autodesk.com/solutions/recap/overview>
- 8) Hexagon 3DReshaper [Електронний ресурс]. — Режим доступу: <https://ngc.com.ua/p/810-hexagon-3d-reshaper.html>
- 9) CREDO 3D СКАН [Електронний ресурс]. — Режим доступу: <https://credo-ua.com/product/credo-3d-scan/>

- 10) Point Cloud Ware [Електронний ресурс]. — Режим доступу:  
<https://geo-plus.com/uk/point-cloud-ware/>
- 11) Pix4Dmapper [Електронний ресурс]. — Режим доступу:  
<https://www.pix4d.com/product/pix4dmapper-photogrammetry-software/>
- 12) LixelStudio [Електронний ресурс]. — Режим доступу:  
<https://www.xgrids.com/intl/lixelstudio>
- 13) CloudCompare [Електронний ресурс]. — Режим доступу:  
<https://www.cloudcompare.org/>
- 14) GRASS GIS [Електронний ресурс]. — Режим доступу:  
<https://grass.osgeo.org/>
- 15) Данілова Н. В. Цифрові плани і карти : навч. посіб. — К. : КНУБА, 2023. Режим доступу:  
[http://eprints.library.odeku.edu.ua/id/eprint/11211/1/Danilova%20NV\\_Tsyfrovi\\_plany\\_i\\_karty\\_KL\\_2023.pdf](http://eprints.library.odeku.edu.ua/id/eprint/11211/1/Danilova%20NV_Tsyfrovi_plany_i_karty_KL_2023.pdf)
- 16) Інструкція з топографічного знімання у масштабах 1:500–1:5000 [Електронний ресурс]. — Режим доступу:  
<https://zakon.rada.gov.ua/laws/show/z0393-98#Text>
- 17) Про затвердження класифікатора об'єктів топографічних карт [Електронний ресурс]. — Режим доступу:  
<https://zakon.rada.gov.ua/rada/show/v0083858-14>
- 18) Miller R. L. LiDAR remote sensing and applications: an introduction // International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. — 2018. — Vol. XLII-3. — P. 11–18.
- 19) Shan J., Yang B. Airborne LiDAR Data Processing: Principles, Systems and Data Quality // Photogrammetric Engineering & Remote Sensing. — 2013. — Vol. 79, No. 2. — P. 209–227.

- 20) LiDAR Data: Advantages, Disadvantages, and Examples [Електронний ресурс] / NOAA. — Режим доступу: <https://oceanservice.noaa.gov>
- 21) LiDAR Mapping Explained [Електронний ресурс]. — Режим доступу: <https://www.luxcarta.com/blog/simulation-training/lidar-mapping>
- 22) Applications of LiDAR in GIS [Електронний ресурс]. — Режим доступу: <https://satpalda.co/what-are-the-applications-of-lidar-in-gis/>
- 23) LiDAR applications in the modern world [Електронний ресурс] / Fugro. — Режим доступу: <https://www.fugro.com/news/long-reads/2025/lidar-applications-in-the-modern-world-fugro-longread>
- 24) Ліцензійне та open-source програмне забезпечення [Електронний ресурс]. — Режим доступу: <https://nung.edu.ua/department/prohramne-zabezpechennya-litsenziyne-ta-open-source>
- 25) Яценко В. С. Геоінформаційні системи і технології : навч. посіб. — Режим доступу: [https://lib.iitta.gov.ua/id/eprint/721803/1/Яценко\\_ВС\\_УДК\\_372.891912.43.pdf](https://lib.iitta.gov.ua/id/eprint/721803/1/Яценко_ВС_УДК_372.891912.43.pdf)
- 26) Постанова Кабінету Міністрів України №1259 від 2004 р. [Електронний ресурс]. — Режим доступу: <https://zakon.rada.gov.ua/laws/show/1259-2004-%D0%BF#Text>
- 27)
-

## ДОДАТКИ