

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту  
(повна назва)

Кафедра Інформатики  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти другий (магістерський)

**ДОСЛІДЖЕННЯ ТОЧНОСТІ МОБІЛЬНОГО LIDAR**  
**ДЛЯ ОЦІНКИ ГЕОПОЗИЦІЮВАННЯ СОНЯЧНИХ ПАНЕЛЕЙ**  
(тема)

Виконав:  
здобувач 2 року навчання,  
групи ІНФМ-24-2

Бегунова В. Д.  
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки  
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика  
(повна назва освітньої програми)

Науковий керівник ст. викл. Кобилін І. О.  
(посада, прізвище, ініціали)

Допускається до захисту

Завідувач кафедри інформатики \_\_\_\_\_  
(підпис)

Кобилін О. А.  
(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту

Кафедра Інформатики

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки  
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 2025 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Бегуновій Вероніці Дмитрівні  
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження точності мобільного LiDAR для оцінки геопозиціювання сонячних панелей

затверджена наказом університету від 14 листопада 2025 року № 1045Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 21 листопада 2025 р.

3. Вихідні дані до роботи методи калібрування мобільного LiDAR, літературні джерела щодо застосування технології LiDAR для геопросторового аналізу, математичні моделі перетворення координат та компенсації систематичних похибок, програмні засоби для реалізації iOS-застосунку (ARKit, RealityKit, SwiftUI), критерії оцінки придатності поверхонь для встановлення сонячних панелей.

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1. Аналіз сучасних методів оцінки точності мобільних LiDAR-систем у смартфонах.

2. Аналіз літературних джерел щодо застосування технології LiDAR для геопозиціонування об'єктів відновлюваної енергетики.

3. Розробка математичних моделей калібрування та компенсації систематичних похибок мобільного LiDAR.

4. Проектування архітектури програмного забезпечення для сканування та аналізу поверхонь.

5. Реалізація iOS-застосунку SolarLiDAR з функціями AR-сканування, калібрування та аналізу придатності.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми оцінки точності мобільного LiDAR для геопозиціонування сонячних панелей, об'єкт та мета дослідження, постановка задачі, архітектура iOS-застосунку SolarLiDAR, скріншоти інтерфейсу додатку з основними модулями, таблиця критеріїв оцінки придатності поверхонь для сонячних панелей, висновки, перспективи та апробація роботи.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	29.09.2025	
2	Аналіз завдання, підбір літератури	30.09.25-07.10.25	
3	Аналіз літератури з досліджуваної проблеми	08.10.25-14.10.25	
4	Особливості методів калібрування	15.10.25-20.10.25	
5	Дослідження методів статистичної оцінки	21.10.25-27.10.25	
6	Програмна реалізація	28.10.25-05.11.25	
7	Обґрунтування отриманих результатів	06.11.25-11.11.25	
8	Оформлення пояснювальної записки	12.11.25-14.11.25	
9	Перевірка на нормоконтроль	21.11.25	
10	Перевірка на плагіат	21.11.25	
11	Рецензування	22.11.25	
12	Підготовка презентації та доповіді	25.11.25	
13	Занесення роботи в електронний архів	01.12.25	
14	Попередній захист кваліфікаційної роботи	01.12.25	

Дата видачі завдання 29 вересня 2025 р.

Здобувач \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис)

ст. викл. Кобилін І. О.  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи: 93 с., 5 табл., 10 рис., 50 джерела.

ГЕОПОЗИЦІОНУВАННЯ, КАЛІБРУВАННЯ КООРДИНАТ, МОБІЛЬНИЙ ЛІДАР, ОЦІНКА ПРИДАТНОСТІ ПОВЕРХОНЬ, СОНЯЧНІ ПАНЕЛІ, ARKIT, IOS-ЗАСТОСУНОК, MAE, REALITYKIT, RMSE, SWIFT, SWIFTUI, WGS-84.

Об'єктом дослідження є точність мобільних LiDAR-систем для оцінки геопозиціонування сонячних панелей.

Предмет дослідження є методи калібрування координат та оцінки придатності поверхонь на основі даних мобільного LiDAR.

Метою дослідження є порівняння методів оцінки точності геопозиціонування шляхом розробки мобільного застосунку для визначення придатності поверхонь під сонячні панелі.

Використано методи калібрування координат через математичні моделі трансформації, статистичного аналізу похибок (RMSE, MAE), інтеграції ARKit та RealityKit, зваженого оцінювання придатності.

Наукова новизна роботи полягає у встановленні кількісних показників точності мобільних LiDAR-систем та розробці комплексної методики оцінки придатності поверхонь.

Взаємозв'язок з іншими роботами полягає у використанні існуючих методів калібрування та трансформації систем координат.

Рекомендації щодо використання результатів роботи: методика може застосовуватися для попередньої оцінки придатності поверхонь, зменшуючи потребу у геодезичних вимірюваннях.

У результаті дослідження розроблено iOS-застосунок SolarLidar з підвищеною точністю для оцінки придатності поверхонь під сонячні панелі.

## ABSTRACT

Explanatory note to the qualification work: 93 pages, 5 table, 10 figures, 50 sources.

COORDINATE CALIBRATION, GEOPOSITIONING, MOBILE LIDAR, SOLAR PANELS, SURFACE SUITABILITY ASSESSMENT, ARKIT, IOS APPLICATION, MAE, REALITYKIT, RMSE, SWIFT, SWIFTUI, WGS-84.

The object of the research is the accuracy of mobile LiDAR systems for surface geopositioning for solar panels.

Subject of research is coordinate calibration methods and surface suitability assessment based on mobile LiDAR data.

The aim of the research is to compare accuracy assessment methods by developing a mobile application for determining surface suitability for solar panel installation.

The Coordinate Calibration through mathematical transformation models, Statistical Error Analysis (RMSE, MAE), ARKit and RealityKit integration, and Weighted Suitability Assessment methods were used. Flowcharts of algorithms were formed.

Scientific novelty of the work lies in establishing quantitative accuracy indicators of mobile LiDAR systems and developing a comprehensive surface suitability assessment methodology.

Interconnection with other works consists in utilizing existing calibration methods and coordinate system transformation models.

Recommendations for using the results of the work the methodology can be applied for preliminary surface assessment, reducing the need for geodetic measurements.

As a result of the research, the SolarLidar iOS application with improved accuracy for assessing surface suitability for solar panels was developed.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	8
Вступ.....	9
1 Огляд методів оцінки точності мобільного LiDAR .....	11
1.1 Огляд мобільних LiDAR технологій .....	11
1.2 Статистичні метрики оцінки точності.....	13
1.3 Методи валідації та калібрування LiDAR сенсорів .....	16
1.4 Геометрична валідація через еталонні об'єкти .....	19
1.5 Застосування LiDAR для сонячної енергетики .....	21
1.6 Постановка задачі дослідження .....	23
2 Методологія та експериментальна установка .....	26
2.1 Вибір інструментів та обладнання.....	26
2.1.1 Мобільний пристрій з інтегрованим LiDAR сенсором .....	26
2.1.2 Програмне забезпечення.....	28
2.2 Методика збору даних .....	30
2.2.1 Сценарії сканування.....	31
2.2.2 Протокол сканування.....	32
2.3 Методика калібрування мобільного LiDAR .....	36
2.4 Алгоритми обробки хмари точок.....	39
3 Практична реалізація застосунку .....	45
3.1 Архітектура програмного застосунку SolarLiDAR .....	45
3.1.1 Огляд архітектурних рішень та патернів проєктування.....	45
3.1.2 Структура проєкту та організація модулів .....	47
3.1.3 Обґрунтування вибору архітектурного підходу MVVM.....	52
3.1.4 Взаємодія компонентів системи .....	54
3.2 Реалізація шару моделей даних.....	60
3.2.1 Модель ScanResult для результатів сканування .....	61
3.2.2 Моделі експорту наукових даних .....	64
3.2.3 Модель оцінки придатності поверхонь .....	67

3.3	Розробка менеджерів.....	71
3.3.1	ARManager: управління AR-сесіями та LiDAR-скануванням	71
3.3.2	CalibrationManager: реалізація математичних моделей калібрування .....	75
3.3.3	ExportManager: експорт даних для наукового аналізу.....	77
3.4	Реалізація ViewModel та управління станом застосунку .....	78
3.5	Розробка користувацького інтерфейсу.....	79
3.6	Інтеграція AR-функціональності та візуалізація даних.....	81
3.7	Інструкція користувача .....	82
	Висновки.....	86
	Перелік джерел посилання .....	88

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

- AR – Augmented Reality (доповнена реальність)
- GNSS – Global Navigation Satellite System (глобальна навігаційна супутникова система)
- GPS – Global Positioning System (глобальна система позиціонування)
- GUI – Graphical User Interface (графічний інтерфейс користувача)
- IDE – Integrated Development Environment (інтегроване середовище розробки)
- IMU – Inertial Measurement Unit (інерційний вимірювальний блок)
- JSON – JavaScript Object Notation (текстовий формат обміну даними)
- LiDAR – Light Detection and Ranging (лазерне виявлення та вимірювання відстані)
- MAE – Mean Absolute Error (середня абсолютна похибка)
- MVC – Model-View-Controller (архітектурний патерн)
- MVP – Model-View-Presenter (архітектурний патерн)
- MVVM – Model-View-ViewModel (архітектурний патерн)
- RANSAC – Random Sample Consensus (метод випадкової вибірки консенсусу)
- RMSE – Root Mean Square Error (середньоквадратична похибка)
- RTK – Real-Time Kinematic (кінематика в реальному часі)
- ToF – Time-of-Flight (метод часу прольоту)
- VCSEL – Vertical-Cavity Surface-Emitting Laser (вертикально-випромінюючий лазер)
- VIPER – View-Interactor-Presenter-Entity-Router (архітектурний патерн)
- WGS-84 – World Geodetic System 1984 (всесвітня геодезична система 1984 року)

## ВСТУП

Сучасні технології геопозиціонування відіграють ключову роль у розвитку відновлювальної енергетики, зокрема при проєктуванні та встановленні сонячних електростанцій. Точне визначення просторового положення поверхонь для розміщення сонячних панелей безпосередньо впливає на ефективність енергогенерації, оскільки навіть незначні похибки в орієнтації можуть призвести до суттєвих втрат продуктивності. Традиційні геодезичні методи, хоча й забезпечують високу точність, потребують значних часових та фінансових ресурсів, що обмежує їх застосування при масових обстеженнях потенційних об'єктів для встановлення сонячних панелей.

За останні роки мобільні LiDAR-системи, інтегровані в сучасні смартфони, стали перспективною альтернативою професійному геодезичному обладнанню [1]. Технологія LiDAR базується на принципі вимірювання відстані за допомогою імпульсів лазерного випромінювання, що дозволяє створювати тривимірні моделі об'єктів з високою деталізацією [2]. Впровадження LiDAR-сканерів у мобільні пристрої, зокрема в iPhone 12 Pro та новіші моделі, відкрило нові можливості для швидкого та економічно ефективного збору просторових даних [3]. Однак питання точності таких вимірювань у контексті специфічних завдань геопозиціонування сонячних панелей залишається недостатньо дослідженим.

Сучасний стан технологій мобільного LiDAR-сканування демонструє активний розвиток як апаратної частини, так і алгоритмів обробки даних. Точність мобільних LiDAR-систем коливається в межах від декількох сантиметрів до десятків сантиметрів залежно від умов вимірювання, характеристик поверхні та методології калібрування. Похибки вимірювання зростають із збільшенням відстані до об'єкта та залежать від кута падіння променя. Існує можливість компенсації систематичних похибок через калібрування сенсорів і застосування матриць трансформації координат.

У контексті застосування для оцінки геопозиціонування сонячних панелей особливу увагу привертає аналіз просторових характеристик дахових поверхонь та автоматизована оцінка придатності поверхонь для встановлення фотоелектричних систем. Проте комплексна оцінка точності мобільного LiDAR саме для завдань геопозиціонування сонячних панелей на різних типах поверхонь – дахах будівель і наземних майданчиках – у науковій літературі представлена фрагментарно.

Актуальність роботи полягає в необхідності експериментального визначення точнісних характеристик мобільних LiDAR-систем для оцінки геопозиціонування сонячних панелей, що дозволить обґрунтувати доцільність застосування цієї технології як альтернативи професійним геодезичним методам при масових обстеженнях об'єктів сонячної енергетики. Враховуючи стрімке зростання попиту на відновлювальні джерела енергії в Україні та світі, розробка методології швидкої та економічно ефективної оцінки придатності поверхонь для встановлення сонячних панелей набуває особливого значення.

Наукова задача дослідження полягає у встановленні кількісних показників точності мобільного LiDAR-сканування для визначення геометричних параметрів поверхонь різних типів, необхідних для оптимального розміщення сонячних панелей, та розробці методики статистичної оцінки похибок вимірювання з урахуванням систематичних і випадкових складових. Це передбачає проведення експериментальної валідації точності вимірювань через порівняння з еталонними даними, аналіз просторового розподілу похибок, дослідження впливу характеристик поверхні на якість сканування та формулювання рекомендацій щодо застосовності мобільного LiDAR для різних сценаріїв оцінки геопозиціонування сонячних панелей.

# 1 ОГЛЯД МЕТОДІВ ОЦІНКИ ТОЧНОСТІ МОБІЛЬНОГО LiDAR

## 1.1 Огляд мобільних LiDAR технологій

Технологія LiDAR є сучасним методом дистанційного зондування, що базується на принципі визначення відстані до об'єктів шляхом вимірювання часу проходження лазерного імпульсу від джерела до поверхні та назад до приймача [2]. Інтеграція LiDAR сенсорів у мобільні пристрої, такі як смартфони та планшети, відкрила нові можливості для широкого застосування цієї технології у різних галузях, включаючи геодезію, архітектуру, будівництво та відновлювану енергетику.

Сучасні мобільні LiDAR системи, що інтегровані в пристрої Apple починаючи з моделі iPhone 12 Pro, оснащені спеціалізованими сенсорами, здатними генерувати щільні хмари точок у тривимірному просторі. Технічні характеристики таких систем включають робочу дальність від 0,5 до 5 метрів у закритих приміщеннях та до 20 метрів на відкритому просторі при сприятливих умовах. Частота сканування сучасних мобільних LiDAR сенсорів досягає 30 кадрів на секунду, що дозволяє швидко захоплювати динамічні сцени та формувати детальні тривимірні моделі навколишнього середовища.

Принцип роботи мобільного LiDAR базується на методі часу-проліту (Time-of-Flight), коли сенсор випромінює серію інфрачервоних лазерних імпульсів та вимірює час їх відбиття від поверхні об'єктів [4]. На основі виміряного часу та швидкості світла обчислюється відстань до кожної точки поверхні. Лазерний промінь розгортається у просторі за допомогою системи дзеркал або за рахунок масиву лазерних діодів, що дозволяє охопити широку область сканування. Кожна зареєстрована точка отримує тривимірні координати відносно положення сенсора у момент виміру.

Система координат мобільного LiDAR прив'язується до внутрішніх інерційних вимірювальних блоків пристрою, що включають акселерометри, гіроскопи та магнітометри. Ці датчики визначають орієнтацію та рух пристрою

у просторі, що дозволяє коректно позиціювати кожен точку хмари відносно глобальної або локальної системи координат. Інтеграція даних від інерційних датчиків та LiDAR сенсора виконується у реальному часі спеціалізованим процесором, що забезпечує формування узгодженої тривимірної моделі навколишнього середовища.

Щільність точок, що генеруються мобільним LiDAR, залежить від відстані до об'єкта сканування та швидкості переміщення пристрою. На відстані 1-2 метри щільність може досягати кількох тисяч точок на квадратний метр, що є достатнім для детального представлення геометрії невеликих об'єктів. Зі збільшенням відстані щільність точок зменшується приблизно пропорційно квадрату відстані, що є характерним для всіх лазерних систем сканування.

Основними перевагами мобільного LiDAR є висока швидкість збору даних, мобільність та компактність обладнання, відносно низька вартість порівняно з професійними геодезичними системами, можливість роботи в умовах обмеженого природного освітлення та простота використання без спеціальної підготовки оператора. Ці переваги роблять мобільний LiDAR привабливим інструментом для багатьох практичних застосувань, де не потрібна надвисока точність професійного геодезичного обладнання.

Водночас технологія має певні обмеження. По-перше, точність вимірювань знижується зі збільшенням відстані до об'єкта, що обмежує область ефективного застосування ближніми дистанціями. По-друге, якість вимірювань суттєво залежить від оптичних властивостей поверхні об'єктів – темні матові поверхні поглинають більше лазерного випромінювання, що зменшує інтенсивність відбитого сигналу. По-третє, прозорі та дзеркальні поверхні створюють проблеми для лазерного сканування через пропускання або дзеркальне відбиття променів. По-четверте, яскраве природне освітлення, особливо пряме сонячне світло, може створювати перешкоди для інфрачервоного випромінювання сенсора.

Мобільні LiDAR системи генерують так звані хмари точок – множини тривимірних координат точок на поверхнях об'єктів. Первинна хмара точок

зазвичай містить шум, викиди та неточності, що вимагає застосування методів фільтрації та обробки даних. Типові алгоритми обробки включають статистичне видалення викидів, згладжування шуму, сегментацію поверхонь та виділення характерних ознак об'єктів.

Інтеграція мобільного LiDAR з іншими сенсорами пристрою створює додаткові можливості. RGB камери дозволяють накласти фотореалістичну текстуру на тривимірну геометричну модель, створюючи візуально привабливі та інформативні представлення об'єктів. GPS/GNSS приймачі забезпечують глобальну геоприв'язку моделей, що важливо для геодезичних застосувань та інтеграції даних з різних джерел. Барометричні датчики допомагають визначати висотну складову положення, особливо в багатоповерхових будівлях.

Для роботи з даними мобільного LiDAR існують спеціалізовані програмні бібліотеки та застосунки. Платформа ARKit від Apple надає розробникам програмний інтерфейс для доступу до даних LiDAR сенсора та інтеграції їх з технологіями доповненої реальності [7]. Сторонні застосунки реалізують різноманітні функції обробки та аналізу даних лазерного сканування для різних предметних областей.

## 1.2 Статистичні метрики оцінки точності

Оцінка точності вимірювань є фундаментальною задачею для визначення придатності мобільного LiDAR для конкретних інженерних застосувань. Статистичні метрики точності дозволяють кількісно охарактеризувати якість вимірювань та порівняти різні методи та обладнання. У геодезії та фотограмметрії застосовується набір стандартизованих метрик, що характеризують різні аспекти точності позиціонування.

Для кожної точки з хмари LiDAR точок визначається відхилення від еталонних координат через евклідову відстань [8, 9]:

$$\varepsilon_i = \left\| P_{\text{meas},i} - P_{\text{ref},i} \right\| = \sqrt{(x_m - x_r)^2 + (y_m - y_r)^2 + (z_m - z_r)^2}, \quad (1.1)$$

де  $P_{\text{meas},i}$  – координати  $i$ -ої точки, отримані LiDAR скануванням;

$P_{\text{ref},i}$  – еталонні координати  $i$ -ої точки;

$x_m, y_m, z_m$  – координати точки з LiDAR вимірювань;

$x_r, y_r, z_r$  – еталонні координати точки.

Середньоквадратичне відхилення (RMSE) для оцінки загальної точності обчислюється за формулою:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n \varepsilon_i^2}, \quad (1.2)$$

де  $n$  – кількість контрольних точок.

RMSE надає інтегральну оцінку точності, що враховує всі відхилення вимірювань від еталонних значень. Особливістю RMSE є те, що ця метрика надає більшу вагу великим відхиленням через операцію зведення в квадрат, що робить її чутливою до наявності викидів у даних.

Середня абсолютна похибка (MAE) обчислюється як середнє арифметичне абсолютних значень різниць між вимірними та еталонними координатами:

$$MAE = \frac{1}{n} \sum_{i=1}^n |\varepsilon_i|. \quad (1.3)$$

На відміну від RMSE, метрика MAE є менш чутливою до викидів, оскільки не містить операції зведення в квадрат. MAE надає оцінку типового відхилення вимірювань від еталонних значень та має ту саму розмірність, що й вимірювана величина, що полегшує інтерпретацію результатів [10–12].

Для просторових вимірювань важливим є розділення похибок на горизонтальну та вертикальну складові, оскільки точність визначення планового положення та висоти може суттєво відрізнятись. Горизонтальна похибка

визначається у площині  $XU$ , вертикальна похибка характеризує точність визначення висотної координати  $Z$ .

Для моделювання просторового розподілу похибок застосовується метод, що враховує залежність точності від відстані сканування та кута падіння лазерного променя. Для кожної точки поверхні обчислюється залежність похибки від відстані  $d$  та кута відхилення  $\alpha$ :

$$\sigma(d, \alpha) = \sigma_0 + k_1 \cdot d + k_2 \cdot \sin^2(\alpha), \quad (1.4)$$

де  $\sigma_0$  – базова похибка сенсора;

$k_1$  – коефіцієнт залежності похибки від відстані;

$k_2$  – коефіцієнт кутової залежності;

$d$  – відстань до точки;

$\alpha$  – кут відхилення від нормалі до поверхні.

Довірчий інтервал для координат точки з урахуванням просторового розподілу похибок визначається як:

$$P_{\text{conf}} = P_{\text{meas}} \pm t_{\alpha/2} \cdot \sigma(d, \alpha), \quad (1.5)$$

де  $t_{\alpha/2}$  – критичне значення t-розподілу;

$d$  – відстань до точки;

$\alpha$  – кут відхилення.

Систематична складова похибки (*bias*) характеризує постійне зміщення вимірювань відносно еталонних значень та обчислюється як:

$$\text{bias} = \frac{\sum(P_{\text{meas}} - P_{\text{ref}})}{n}. \quad (1.6)$$

Наявність ненульового систематичного зміщення вказує на некоректне калібрування сенсора. Систематичну похибку можна компенсувати шляхом

додавання відповідної поправки до результатів вимірювань. Виявлення систематичних відхилень виконується для кожної координати  $x$ ,  $y$ ,  $z$  окремо.

Коефіцієнт детермінації  $R^2$  характеризує ступінь лінійної залежності між вимірними та еталонними значеннями. Значення  $R^2$  близьке до 1 вказує на високу узгодженість вимірювань з еталонними значеннями та ефективність калібрування сенсора.

Відносна точність характеризує відношення похибки до величини вимірюваної відстані та дозволяє порівнювати точність вимірювань на різних дистанціях. Повторюваність вимірювань оцінюється шляхом багаторазового сканування одних і тих же об'єктів та обчислення розкиду отриманих результатів.

### 1.3 Методи валідації та калібрування LiDAR сенсорів

Валідація точності мобільного LiDAR є критично важливим етапом дослідження, що передує практичному застосуванню технології для інженерних задач [3, 13]. Процес валідації включає порівняння координат точок, отриманих за допомогою LiDAR сканування, з еталонними значеннями, визначеними за допомогою більш точних геодезичних методів.

Основним методом валідації є використання еталонних об'єктів з точно відомими геометричними параметрами та просторовим положенням. Як еталонні об'єкти зазвичай застосовуються спеціально виготовлені геодезичні марки, геометричні фігури відомих розмірів або існуючі об'єкти інфраструктури, координати характерних точок яких визначені професійними геодезичними методами.

Геодезичні марки встановлюються у різних частинах робочої зони на різних відстанях від точки сканування. Типова конфігурація включає 10-20 контрольних точок, рівномірно розподілених по області дослідження. Координати марок визначаються за допомогою електронного тахеометра або

GNSS приймача в режимі RTK з точністю 1-2 см по горизонталі та 2-3 см по вертикалі.

Після виконання LiDAR сканування координати контрольних точок визначаються у сформованій хмарі точок. Для кожної контрольної точки обчислюються відхилення LiDAR координат від еталонних значень у трьох напрямках. На основі отриманих відхилень обчислюються статистичні метрики точності – RMSE горизонтальна та вертикальна, MAE, максимальна похибка.

Калібрування LiDAR сенсора спрямоване на мінімізацію систематичних похибок, що виникають через неточності у визначенні взаємного положення сенсора та інерційних вимірювальних блоків, нелінійності оптичної системи та температурні ефекти [14–16]. Процес калібрування передбачає визначення параметрів математичної моделі трансформації координат.

Матриця трансформації LiDAR сенсора відносно системи координат пристрою визначається через кути Ейлера  $\varphi$ ,  $\theta$ ,  $\psi$  та вектор трансляції:

$$T = [ R(\varphi, \theta, \psi) \mid t(x, y, z) ], \quad (1.7)$$

де  $R$  – матриця ротації розміром  $3 \times 3$ ;

$t$  – вектор трансляції, що описує зсув системи координат;

$\varphi$ ,  $\theta$ ,  $\psi$  – кути Ейлера, що визначають орієнтацію сенсора.

Матриця ротації обчислюється як добуток трьох базових матриць повороту навколо осей координат:

$$R = R_z(\psi) \cdot R_y(\theta) \cdot R_x(\varphi), \quad (1.8)$$

де  $R_x$ ,  $R_y$ ,  $R_z$  – матриці повороту навколо осей  $X$ ,  $Y$ ,  $Z$  відповідно.

Калібрування виконується через мінімізацію функції відхилень від еталонних точок для підвищення точності геопозиціювання.

Алгоритм компенсації систематичних похибок реалізується у кілька послідовних кроків. Для кожної координати  $x$ ,  $y$ ,  $z$  та всієї множини контрольних точок виконуються наступні операції:

Крок 1. Обчислити систематичне зміщення (1.6).

Крок 2. Скоригувати координати:

$$P_{\text{new}} = P_{\text{meas}} - \text{bias}. \quad (1.9)$$

Крок 3. Перевірити покращення точності через обчислення RMSE для скоригованих координат.

Крок 4. Якщо RMSE зменшилось, то застосувати корекцію до всієї хмари точок.

Цей ітераційний процес повторюється до досягнення мінімального значення похибки.

Оптимізація параметрів калібрування виконується методом найменших квадратів. Функція цілі визначається як сума квадратів відхилень між трансформованими LiDAR координатами контрольних точок та їх еталонними значеннями. Мінімізація функції дозволяє знайти оптимальні значення параметрів трансформації.

Після визначення параметрів калібрування вони застосовуються до всіх точок хмари, що забезпечує компенсацію систематичних похибок. Якість калібрування оцінюється шляхом порівняння залишкових похибок після застосування трансформації з похибками до калібрування. Успішне калібрування має зменшити RMSE принаймні на 30-50 відсотків.

Періодична перекалібровка сенсора рекомендується для підтримання високої точності вимірювань, оскільки параметри калібрування можуть змінюватися з часом через механічні та температурні навантаження на пристрій. Типовий інтервал перекалібровки становить 1-3 місяці інтенсивного використання.

Валідація результатів калібрування виконується на незалежному наборі контрольних точок, які не використовувалися у процесі визначення параметрів трансформації. Цей підхід дозволяє оцінити здатність калібровочної моделі узагальнювати на нові дані.

#### 1.4 Геометрична валідація через еталонні об'єкти

Геометрична валідація точності мобільного LiDAR передбачає використання об'єктів з відомими геометричними параметрами для перевірки правильності відтворення форми та розмірів у хмарі точок [8, 9]. Цей підхід доповнює точкову валідацію координат контрольних марок і дозволяє оцінити якість представлення протяжних об'єктів та складних поверхонь.

Типовими еталонними об'єктами є прямокутні панелі відомих розмірів, що імітують геометрію сонячних фотовольтаїчних модулів. Використовуються панелі стандартних розмірів 2000×1000 мм з плоскою матовою поверхнею, що забезпечує якісне відбиття лазерного випромінювання. Панелі встановлюються у різних орієнтаціях та на різних відстанях від точки сканування для оцінки залежності точності від умов вимірювання.

Кубічні об'єкти зі стороною 300-500 мм застосовуються для перевірки точності вимірювання прямих кутів та лінійних розмірів у трьох перпендикулярних напрямках. Шість граней куба утворюють набір еталонних площин, орієнтованих у різних напрямках. Циліндричні маркери діаметром 200-400 мм та висотою 500-1000 мм використовуються для оцінки точності відтворення криволінійних поверхонь.

Після LiDAR сканування еталонних об'єктів виконується сегментація хмари точок для виділення точок, що належать кожному об'єкту окремо [17, 18]. Сегментація може виконуватися автоматично за допомогою алгоритмів кластеризації або напівавтоматично.

Точність геопозиціонування для кожної контрольної геометрії оцінюється через порівняння вимірних та еталонних геометричних параметрів:

$$\delta_{\text{geom}} = \frac{\sum_{i=1}^m |P_{\text{meas},i} - P_{\text{ref},i}|}{\sum_{i=1}^m P_{\text{ref},i} \cdot 100\%}, \quad (1.10)$$

де  $m$  – кількість геометричних параметрів (довжина, площа, кут);

$P_{\text{meas},i}$  – вимірне значення  $i$ -го параметра;

$P_{\text{ref},i}$  – еталонне значення  $i$ -го параметра.

Для еталонних панелей відомих розмірів перевіряється правильність визначення лінійних розмірів. Довжина та ширина панелі вимірюються у хмарі точок після проєкції точок на апроксимуючу площину. Різниця між вимірними та номінальними розмірами характеризує точність вимірювання метричних параметрів об'єктів.

Кутові параметри об'єктів перевіряються шляхом визначення векторів нормалей до площин граней та обчислення кутів між ними. Для кубічних об'єктів перевіряється ортогональність граней – кути між нормаллями мають дорівнювати 90 градусів.

Інтегральний показник точності системи обчислюється з урахуванням різних типів метрик:

$$\text{Accuracy} = 1 - \sqrt{w_1 \cdot \text{RMSE}^2 + w_2 \cdot \text{MAE}^2 + w_3 \cdot \delta_{\text{geom}}^2}, \quad (1.11)$$

де  $w_1$ ,  $w_2$ ,  $w_3$  – вагові коефіцієнти для різних типів метрик точності, що задовольняють умову  $w_1 + w_2 + w_3 = 1$ .

Інтегральний показник дозволяє отримати єдину оцінку якості геопозиціонування з урахуванням як точкових відхилень координат, так і точності відтворення геометричних параметрів об'єктів.

Аналіз щільності точок виконується шляхом обчислення кількості точок на одиницю площі для кожного еталонного об'єкта. Нерівномірність щільності

може вказувати на проблеми сканування певних ділянок поверхні через несприятливі умови відбиття або обмеження кута огляду сенсора.

### 1.5 Застосування LiDAR для сонячної енергетики

Технологія лазерного сканування знаходить широке застосування у сфері відновлюваної енергетики, зокрема для проєктування та експлуатації сонячних фотовольтаїчних електростанцій [2]. LiDAR дані забезпечують детальну тривимірну інформацію про геометрію дахових конструкцій, наземних майданчиків та навколишнього ландшафту, що є критично важливою для визначення оптимальної орієнтації та кута нахилу сонячних панелей.

Основні завдання, що вирішуються за допомогою LiDAR технологій у сонячній енергетиці, включають визначення придатних площ для встановлення фотовольтаїчних систем, оцінку затінення від навколишніх об'єктів та рельєфу місцевості, розрахунок оптимального азимуту та кута нахилу панелей для максимізації інсоляції, точне геопозиціонування елементів сонячної електростанції для моніторингу та обслуговування, а також створення цифрових двійників існуючих установок для цілей діагностики та оптимізації.

Геопозиціонування сонячних панелей вимагає високої точності визначення просторових координат, оскільки орієнтація панелей критично впливає на енергетичну ефективність системи. Відхилення азимуту панелей на 15 градусів від оптимального напрямку може призводити до зниження річної енергопродуктивності на 3-5 відсотків. Аналогічно, відхилення кута нахилу на 10 градусів від оптимального значення може зменшувати річний вихід енергії на 2-4 відсотки залежно від географічної широти місцевості.

Методологія застосування LiDAR для геопозиціонування сонячних панелей включає декілька послідовних етапів [19, 20]. На першому етапі виконується сканування дахової поверхні або земельної ділянки з формуванням щільної хмари точок, що представляє тривимірну геометрію місцевості. Другий етап

передбачає обробку хмари точок з виділенням придатних площ для розміщення панелей. Для дахових установок виконується сегментація окремих скатів даху, визначення їх орієнтації та кута нахилу, виявлення перешкод та деформацій конструкції.

Третій етап включає моделювання затінення від навколишніх об'єктів протягом року. Тривимірна модель місцевості використовується для розрахунку траєкторії руху сонця по небосхилу та визначення періодів затінення для кожної точки потенційного розміщення панелей. Затінення може виникати від будівель, дерев, рельєфу місцевості та навіть від інших рядів панелей у великих наземних установках.

Четвертий етап полягає у моделюванні інсоляції з урахуванням географічного положення об'єкта, кліматичних даних про сонячне випромінювання, орієнтації та нахилу поверхонь, а також обчислених періодів затінення. Результатом є карта розподілу річної інсоляції по території, що показує найбільш перспективні ділянки для розміщення панелей.

П'ятий етап включає розробку оптимальної конфігурації розміщення панелей з урахуванням доступної площі, вимог до енергопродуктивності, обмежень затінення та економічних факторів. Виконується точне геопозиціонування запланованих місць встановлення панелей з прив'язкою до геодезичної системи координат.

Мобільні LiDAR системи мають певні переваги для обстеження об'єктів малої та середньої площі. Висока мобільність дозволяє швидко виконувати зйомку дахів житлових та комерційних будівель без необхідності використання дронів або спеціалізованого наземного обладнання. Простота використання робить технологію доступною для широкого кола спеціалістів у галузі відновлюваної енергетики.

Однак застосування мобільного LiDAR має певні обмеження. Обмежена дальність сканування до 20 метрів не дозволяє ефективно обстежувати великі наземні установки або високі промислові будівлі. Точність геопозиціонування може бути недостатньою для деяких застосувань, що вимагають міліметрової

точності. Якість даних залежить від умов сканування та стабільності утримання пристрою оператором.

Для підвищення точності геопозиціонування застосовуються додаткові методи. Інтеграція з GNSS приймачами високої точності в режимі RTK дозволяє забезпечити глобальну геоприв'язку моделей з точністю кількох сантиметрів. Використання геодезичних марок з відомими координатами як опорних точок забезпечує калібрування та верифікацію точності LiDAR даних.

Аналіз існуючих сонячних електростанцій за допомогою мобільного LiDAR дозволяє визначати фактичні кути нахилу та азимути встановлених панелей, виявляти деформації монтажних конструкцій та діагностувати проблеми, що впливають на ефективність роботи системи. Порівняння фактичних параметрів з проєктними значеннями дозволяє ідентифікувати помилки монтажу.

## 1.6 Постановка задачі дослідження

Швидкий розвиток відновлюваної енергетики в Україні та світі створює попит на доступні та ефективні методи проєктування та моніторингу сонячних фотовольтаїчних електростанцій. Точне геопозиціонування сонячних панелей є критично важливим для забезпечення оптимальної орієнтації панелей відносно сонця, що безпосередньо впливає на енергетичну ефективність та економічну віддачу інвестицій у відновлювану енергетику [5, 6].

Традиційні геодезичні методи визначення просторового положення об'єктів, хоча й забезпечують високу точність, є досить трудомісткими, вимагають спеціалізованого обладнання та кваліфікованого персоналу, що збільшує вартість та тривалість проєктних робіт. Мобільні LiDAR системи, інтегровані у сучасні смартфони та планшети, пропонують альтернативний підхід, що поєднує достатню точність вимірювань з високою мобільністю, простотою використання та доступністю технології.

Однак на сьогоднішній день відсутні систематичні дослідження точності мобільного LiDAR для специфічного застосування геопозиціонування сонячних панелей. Невизначеність щодо реальної точності вимірювань у різних умовах сканування обмежує впровадження мобільних LiDAR систем у практику проектування сонячних електростанцій. Необхідна експериментальна валідація точності та розробка рекомендацій щодо застосовності технології для конкретних інженерних завдань.

Об'єктом дослідження є процес геопозиціонування сонячних фотовольтаїчних панелей на різних типах поверхонь з використанням мобільного LiDAR сканування.

Предметом дослідження є точність вимірювань мобільних LiDAR систем (iPhone 12 Pro та новіші моделі) для визначення геометричних параметрів поверхонь і методи калібрування та компенсації систематичних похибок для оцінки придатності місць встановлення сонячних панелей.

Метою дослідження є експериментальна оцінка точності мобільного LiDAR для геопозиціонування сонячних панелей та визначення діапазонів застосовності технології для практичних завдань проектування сонячних фотовольтаїчних електростанцій.

Для досягнення мети необхідно вирішити такі завдання:

- провести аналіз існуючих методів геопозиціонування для сонячної енергетики та технічних характеристик мобільних LiDAR систем;
- розробити методологію експериментальної оцінки точності мобільного LiDAR для геопозиціонування об'єктів, що включає протокол збору даних, вибір еталонних об'єктів та метрики оцінки якості вимірювань;
- виконати статистичний аналіз похибок геопозиціонування з визначенням середньоквадратичної похибки RMSE, середньої абсолютної похибки MAE, максимальної похибки та коефіцієнта детермінації  $R^2$  для різних дистанцій сканування;

– дослідити вплив факторів навколишнього середовища на точність мобільного LiDAR, включаючи освітленість сцени, відстань до об'єкта, характеристики відбиваючої поверхні та геометричну складність оточення;

– розробити та апробувати методи калібрування мобільного LiDAR сенсора для компенсації систематичних похибок з використанням матриць трансформації координат на основі еталонних вимірювань;

– провести геометричну валідацію точності відтворення форми та розмірів еталонних об'єктів, що імітують сонячні панелі, шляхом порівняння вимірних параметрів з номінальними значенням;

– визначити діапазони застосовності мобільного LiDAR для геопозиціювання сонячних панелей з урахуванням вимог до точності для різних типів сонячних електростанцій та обмежень технології;

– розробити програмне забезпечення для iOS-платформи, що реалізує методи AR-сканування, калібрування та аналізу придатності поверхонь для встановлення сонячних панелей.

## 2 МЕТОДОЛОГІЯ ТА ЕКСПЕРЕМЕНТАЛЬНА УСТАНОВКА

### 2.1 Вибір інструментів та обладнання

Вибір інструментів та обладнання є критично важливим етапом експериментального дослідження, оскільки саме від технічних характеристик використовуваних засобів залежить точність вимірювань, достовірність отриманих даних та відтворюваність результатів. У цьому контексті особлива увага приділяється параметрам сенсорних систем та програмним засобам, що забезпечують їх коректне функціонування.

#### 2.1.1 Мобільний пристрій з інтегрованим LiDAR сенсором

Для проведення експериментального дослідження обрано смартфон iPhone 14 Pro Max виробництва Apple Inc., який оснащений інтегрованим LiDAR сенсором четвертого покоління [3, 7]. Вибір даного пристрою обумовлений кількома факторами. По-перше, iPhone 14 Pro Max представляє сучасну технологію мобільного лазерного сканування, доступну у споживчих пристроях. По-друге, пристрій оснащений потужним процесором Apple A16 Bionic, що забезпечує високу швидкість обробки даних у режимі реального часу. По-третє, наявність розвиненого програмного інтерфейсу ARKit версії 6.0 дозволяє отримувати безпосередній доступ до необроблених даних LiDAR сенсора.

Технічні характеристики LiDAR сенсора показано у таблиці 2.1.

Таблиця 2.1 – Технічні характеристики LiDAR сенсора iPhone 14 Pro Max

Параметр	Значення
Технологія	Time-of-Flight (ToF)
Довжина хвилі лазера	940 нм (інфрачервоний діапазон)

Продовження таблиці 2.1

Параметр	Значення
Тип лазера	VCSEL (вертикально-випромінюючий лазер)
Робоча дальність	0,5–5,0 м
Оптимальна дальність	1,0–3,0 м
Частота оновлення	60 Гц
Кут огляду (горизонтальний)	70°
Кут огляду (вертикальний)	55°
Щільність хмари точок	до 15000 точок/кадр
Роздільна здатність глибини	1 см (на відстані до 5 м)
Точність відстані (номінальна)	±1 см (за даними виробника)
Споживання енергії	~500 мВт при активній роботі

LiDAR сенсор працює на основі технології часу прольоту (Time-of-Flight, ToF), що передбачає вимірювання часу, необхідного для проходження лазерного імпульсу від джерела до об'єкта та назад до приймача. Використовується вертикально-випромінюючий лазер VCSEL з довжиною хвилі 940 нм, що забезпечує роботу сенсора в різних умовах освітлення, включаючи повну темряву. Інфрачервоне випромінювання цієї довжини хвилі є невидимим для людського ока та безпечним при дотриманні норм лазерної безпеки класу 1.

Принцип роботи сенсора базується на вимірюванні фазового зсуву модульованого лазерного променя. Лазерний випромінювач генерує серію імпульсів інфрачервоного світла, які відбиваються від поверхонь об'єктів у полі зору сенсора. Відбиті фотони реєструються масивом фотодіодів, розташованих поруч з випромінювачем. Час прольоту кожного фотона обчислюється на основі фазової різниці між випромінюваним та прийнятим сигналом, що дозволяє визначити відстань до точки відбиття з точністю до одного сантиметра.

Смартфон оснащений додатковими сенсорами, що забезпечують точне позиціонування та орієнтацію пристрою у просторі. Інерційний вимірювальний

блок (IMU) включає тривісний гіроскоп MEMS типу з діапазоном вимірювання  $\pm 2000^\circ/\text{с}$  та частотою дискретизації 200 Гц, тривісний акселерометр з діапазоном  $\pm 16g$  та частотою дискретизації 200 Гц, а також тривісний магнітометр з точністю  $\pm 0,5^\circ$  азимутального кута. Дані IMU використовуються для компенсації руху пристрою під час сканування та для визначення орієнтації кожного LiDAR кадру у світовій системі координат.

GNSS приймач підтримує супутникові системи GPS (США), Galileo (ЄС) та BeiDou (Китай), що забезпечує точність горизонтального позиціонування 3-5 м у звичайному режимі [21]. Вертикальне позиціонування на основі GNSS має точність 5-10 м, що може бути недостатньо для деяких застосувань геопозиціонування.

Барометричний датчик визначає висотну координату на основі вимірювання атмосферного тиску з роздільною здатністю 0,1 Па, що відповідає висотній роздільності приблизно 1 см. Точність визначення висоти складає  $\pm 1$  м для швидких змін висоти та покращується до  $\pm 0,25$  м при тривалих спостереженнях з усередненням даних.

Камерна система включає основну камеру з роздільною здатністю 48 Мп та оптичною стабілізацією зображення, широкоформатну камеру з роздільною здатністю 12 Мп та телеоб'єктив з трикратним оптичним збільшенням. Камери використовуються для візуальної одометрії, що доповнює дані IMU для точного відстеження траєкторії руху пристрою, а також для накладання текстур на тривимірну геометрію, отриману від LiDAR сенсора.

### 2.1.2 Програмне забезпечення

Для реалізації методики дослідження обрано розробку власного програмного застосунку на платформі iOS з використанням мови програмування Swift версії 6.0 [7]. Розробка передбачається у інтегрованому середовищі Xcode 16.4 на операційній системі macOS. Вибір нативної розробки для iOS

обумовлений необхідністю прямого доступу до низькорівневих API LiDAR сенсора та можливістю оптимізації продуктивності обробки даних.

Основні використовувані фреймворки:

– ARKit версії 6.0 надає програмний інтерфейс для доступу до даних LiDAR сенсора через клас ARFrame. Кожен кадр містить хмару точок у форматі CVPixelBuffer з координатами точок у локальній системі координат сенсора, матрицю 4×4 перетворення координат від локальної до світової системи, дані про виявлені площини типу ARPlaneAnchor, мітки часу для синхронізації з іншими сенсорами та інформацію про відстеження камери і якість просторового відображення;

– RealityKit забезпечує візуалізацію тривимірної сцени у режимі реального часу з використанням Metal API для апаратного прискорення рендерингу. Фреймворк підтримує відображення хмари точок з динамічним кольоровим кодуванням, візуалізацію виявлених площин, відображення векторів нормалей та координатних осей для кожної площини, та інтерактивну навігацію по тривимірній сцені;

– Core Location використовується для отримання географічних координат та висоти над рівнем моря через CLLocationManager. Фреймворк надає поточну широту, довготу та висоту з метаданими про точність визначення, азимутальний кут орієнтації пристрою відносно магнітного півночі з компенсацією магнітного схилення, та швидкість переміщення пристрою;

– Core Motion забезпечує доступ до даних інерційних сенсорів через CMMotionManager. Дані включають прискорення від акселерометра у трьох осях, кутову швидкість від гіроскопа, магнітне поле від магнітометра, та атмосферний тиск від барометра для визначення висотної координати;

Застосунок передбачає функціональність експорту метаданих сканування, які в подальшому зберігаються у форматі JSON та включають дату і час сканування, параметри LiDAR сенсора, умови навколишнього середовища та параметри калібрування.

## 2.2 Методика збору даних

Для проведення експериментальних досліджень в ідеальному випадку слід використовувати тестове середовище з повністю контрольованими умовами. Найбільш придатним є спеціалізований полігон, оснащений референсними об'єктами, що відтворюють типові конфігурації розміщення сонячних панелей.

Референсні об'єкти можуть включати вертикальні площини, які використовуються для імітації фасадного розміщення панелей. Для таких площин рекомендовано застосовувати конструкції розміром  $2 \times 2$  м з матовою поверхнею, що забезпечує якісне відбиття лазерного випромінювання та стабільність вимірювань.

До складу референсних об'єктів також можуть входити горизонтальні площини, призначені для моделювання наземних установок. Їх зазвичай розміщують на певній висоті над поверхнею, що дає змогу здійснювати сканування з різних кутів огляду й отримувати більш повну просторову інформацію.

Окрему групу становлять похилі площини, встановлені під кутами  $30^\circ$  та  $60^\circ$  до горизонталі. Вони дозволяють відтворити типові умови монтажу дахових сонячних електростанцій. Такі кути є характерними для середніх географічних широт, де оптимальний нахил панелей становить приблизно  $30\text{-}40^\circ$  для максимального рівня річної інсоляції.

Крім того, до референсних об'єктів можуть додаватися циліндричні структури, що дають змогу оцінити точність відтворення криволінійних поверхонь. Вони є корисними для перевірки роботи алгоритмів сегментації, орієнтованих на неплоскі об'єкти, та дозволяють оцінити поведінку системи у складніших геометричних умовах.

Еталонні координати контрольних точок на референсних об'єктах визначаються за допомогою професійного геодезичного обладнання з точністю порядку  $\pm 1\text{-}2$  мм. Контрольні точки позначаються світловідбивними марками,

що забезпечує надійну ідентифікацію точок як геодезичним обладнанням, так і алгоритмами обробки LiDAR даних.

### 2.2.1 Сценарії сканування

Методологія експериментального дослідження передбачає проведення сканування за наперед визначеними систематизованими сценаріями, які охоплюють основні фактори, що впливають на точність мобільних LiDAR-систем. Одним із ключових параметрів є відстань сканування: вимірювання доцільно виконувати з декількох фіксованих дистанцій у діапазоні 1-10 м, що дозволяє визначити залежність точності вимірювань від віддаленості об'єкта у межах характерних робочих режимів сенсора. Важливим фактором також є кут сканування, тому для кожної відстані доцільно проводити вимірювання під різними кутами відносно нормалі до поверхні (зокрема  $0^\circ$ ,  $30^\circ$ ,  $45^\circ$  та  $60^\circ$ ), що забезпечує оцінювання впливу кута падіння променя на достовірність отриманих даних. Для похилих поверхонь сонячних панелей цей фактор є критичним, оскільки при збільшенні кута падіння зменшується інтенсивність відбитого сигналу та зростає похибка визначення відстані.

Додатково враховується вплив умов освітлення. Сканування доцільно здійснювати за різних режимів природного та штучного освітлення, включаючи пряме сонячне світло з освітленістю понад 50000 лк, розсіяне денне освітлення при хмарності (10000-20000 лк) та штучне освітлення з інтенсивністю менше 500 лк. Високий рівень видимого світла може створювати значний фон, що погіршує співвідношення сигнал/шум інфрачервоного каналу LiDAR-сенсора. Важливу роль відіграють також оптичні властивості поверхонь об'єктів. Для оцінювання їхнього впливу на якість сканування доцільно використовувати поверхні з різними коефіцієнтами відбиття, зокрема матові світлі та темні матеріали, глянцеві поверхні зі змінним напрямком відбиття, а також текстуровані поверхні, що характеризуються нерівномірним розсіюванням.

З метою забезпечення статистичної надійності отриманих результатів кожен сценарій сканування має виконуватися багаторазово. Повторні вимірювання дають можливість обчислити середні значення показників точності, визначити стандартні відхилення та оцінити повторюваність результатів, що є необхідною умовою для достовірного аналізу характеристик мобільної LiDAR-системи.

### 2.2.2 Протокол сканування

Протокол сканування було розроблено з метою стандартизації процедури збору даних та забезпечення високої порівнянності результатів між різними сеансами вимірювань. Він охоплює послідовність етапів, що передбачають підготовку обладнання, позиціонування та орієнтацію оператора, виконання безпосереднього сканування, фіксацію метаданих, збереження та резервне копіювання даних, а також контроль якості після завершення сеансу. Кожен етап протоколу визначає конкретні вимоги та рекомендації, що дозволяє мінімізувати систематичні похибки та забезпечує максимальну точність отриманих даних.

Підготовчий етап передбачає комплексну перевірку стану обладнання. Особлива увага приділяється рівню заряду акумулятора, який має становити не менше 80%. Це пояснюється тим, що низький заряд (менше 20%) може спричинити зниження продуктивності процесора операційною системою iOS з метою економії енергії, що безпосередньо впливає на частоту оновлення LiDAR сенсора та якість обробки даних у режимі реального часу. Окрім цього, важливим аспектом є температурна адаптація пристрою. Якщо пристрій був перенесений з приміщення на вулицю або навпаки, рекомендується витримати його у робочій зоні протягом 10-15 хвилин. Це дозволяє стабілізувати температуру оптичних та електронних компонентів LiDAR сенсора та мінімізувати температурні дрейфи калібрувальних параметрів. Різкі температурні зміни можуть призвести до конденсації вологи на оптичних вікнах

сенсора, що суттєво знижує якість вимірювань та підвищує ризик виникнення артефактів.

Очищення оптики є наступним критично важливим кроком підготовки. Візуально перевіряються оптичні вікна LiDAR сенсора та камери задньої панелі пристрою. Будь-які забруднення, включаючи пил, відбитки пальців або залишки інших речовин, можуть призвести до розсіювання лазерного випромінювання, зниження інтенсивності відбитого сигналу та появи артефактів у даних. Рекомендовано використовувати м'яку мікрофіброву серветку без застосування рідин для уникнення залишків та розводів на поверхні скла, рухаючи серветку від центру до країв круговими рухами з легким натиском.

Калібрування компаса є необхідним при першому запуску застосунку або після тривалого періоду простою. Воно здійснюється шляхом виконання серії обертальних рухів пристрою у тривимірному просторі відповідно до підказок системи. Ця процедура забезпечує точне визначення азимутального кута орієнтації панелей щодо магнітного півночі, що є критично важливим для коректного позиціонування та орієнтації хмари точок у глобальній системі координат. Паралельно здійснюється налаштування параметрів застосунку відповідно до сценарію сканування. Зокрема, встановлюються роздільна здатність хмари точок, режим роботи ARKit (світова або локальна система координат), увімкнення GPS для географічної прив'язки, а також автоматичне збереження даних після завершення сеансу.

На етапі позиціонування оператор займає визначену відстань від референсного об'єкта, яка контролюється за допомогою марок або лазерного далекоміра. Використання фіксованих позиційних марок дозволяє забезпечити повторюваність результатів у різних сеансах. Пристрій утримується двома руками у ландшафтній (горизонтальній) орієнтації. Великі пальці рук розміщуються на нижній частині пристрою для підтримки, решта пальців охоплюють бічні грані. Така позиція утримання забезпечує максимальну стабільність та мінімізує природний тремор рук. Пристрій розташовується на рівні очей оператора на відстані приблизно 30-40 см від обличчя, що дозволяє

комфортно контролювати процес сканування через екран. Орієнтація пристрою спрямована на центральну частину об'єкта. Для перпендикулярного сканування (кут  $0^\circ$ ) оператор візує по центральній контрольній марці, тоді як для похилих кутів ( $30^\circ$ ,  $45^\circ$ ,  $60^\circ$ ) застосовуються кутові шаблони або електронний інклінометр для точного контролю кута нахилу. Перед початком захоплення даних пристрій утримується нерухомо протягом 2-3 секунд для ініціалізації системи відстеження ARKit, що дозволяє виявити характерні ознаки сцени та встановити початкову систему координат.

Існує два основних підходи до траєкторії руху пристрою під час сканування:

– статичне сканування передбачає утримання пристрою у фіксованому положенні протягом всього сеансу. Такий підхід забезпечує максимальну стабільність даних та мінімальні похибки, пов'язані з рухом, але може призводити до нерівномірної щільності точок з більшою концентрацією у центрі поля зору та розрідженням на периферії;

– динамічне сканування включає плавний рух пристрою для охоплення всієї поверхні об'єкта під різними кутами огляду. Рекомендується виконувати горизонтальні та вертикальні проходи з постійною швидкістю приблизно 15-20 см/с. Траєкторія руху може мати форму меандру (зигзагоподібний рух) або спіралі від центру до країв. Динамічне сканування забезпечує більш рівномірну щільність точок по всій поверхні та краще покриття складних геометричних форм.

Для площинних об'єктів, таких як імітації сонячних панелей, оптимальною є траєкторія у вигляді горизонтальної восьмірки або еліпса перед об'єктом. Оператор плавно переміщує пристрій вліво-вправо та вгору-вниз, підтримуючи приблизно постійну відстань до об'єкта. Амплітуда руху має бути достатньою для охоплення всієї площі об'єкта плюс 20-30 см поза його межами для надійного виявлення країв.

Швидкість руху пристрою є критичним параметром для якості сканування. Занадто швидкий рух (більше 30 см/с) може призводити до розмиття зображень

камери та втрати відстеження у системі візуальної одометрії, що викликає пропуски у даних. Занадто повільний рух (менше 10 см/с) не дає суттєвих переваг у якості, але збільшує загальний час сканування.

Оптимальна швидкість руху становить 15-20 см/с, що відповідає приблизно 60-70% перекриттю послідовних кадрів при частоті оновлення LiDAR 60 Гц. Таке перекриття забезпечує надійне відстеження траєкторії та можливість усереднення даних для зменшення випадкових похибок.

Тривалість сеансу визначається розміром об'єкта:

- 20-30 секунд для малих об'єктів до 2 м<sup>2</sup>;
- 30-45 секунд для середніх 2–4 м<sup>2</sup>;
- 45-60 секунд для великих понад 4 м<sup>2</sup>.

Під час сканування оператор контролює якість захоплення даних через візуальний інтерфейс застосунку. На екрані відображаються індикатори якості:

- кількість точок у поточному кадрі;
- охоплену площу;
- нахил;
- попередню оцінку.

Сеанс сканування завершується натисканням кнопки «Стоп» в інтерфейсі застосунку. При завершенні виконується фінальна обробка даних, що включає застосування параметрів калібрування, фільтрацію шуму та сегментацію площин. Процес обробки може тривати 5-15 секунд залежно від кількості точок у хмарі.

Фіксація метаданих виконується автоматично та включає параметри обладнання, часові мітки, географічні координати та додаткові спостереження щодо погодних умов та умов проведення сканування. Метадані зберігаються у структурованому форматі JSON і можуть бути експортовані для подальшого аналізу та фільтрації даних. Для кожного сеансу сканування створюється окрема папка з унікальним ідентифікатором (дата-час у форматі ISO 8601). Папка містить:

- метадані сканування `metadata.json`;

– параметри калібрування, застосовані до даних calibration.json.

Дані зберігаються у локальній пам'яті пристрою в каталозі Documents застосунку. Це забезпечує швидкий доступ до даних та можливість офлайн роботи без залежності від мережевого з'єднання.

Для запобігання втрати даних рекомендується виконувати періодичне резервне копіювання, яке здійснюється у локальній пам'яті пристрою та, за необхідності, через iCloud, на комп'ютер або в хмарне сховище. Один сеанс тривалістю 30-60 секунд генерує приблизно 20-40 МБ даних, тому для серії експериментів рекомендується мати не менше 5 ГБ вільного простору для забезпечення стабільної роботи та збереження декількох сотень сеансів.

Після завершення сканування проводиться контроль якості. Перевіряється повнота покриття об'єкта, рівномірність щільності точок, наявність артефактів та правильність виділення площин. У разі виявлення критичних похибок сеанс сканування повторюється із корекцією параметрів, таких як швидкість руху, траєкторія або умови освітлення, що дозволяє забезпечити високу надійність та точність отриманих даних.

### 2.3 Методика калібрування мобільного LiDAR

Калібрування мобільного LiDAR сенсора є критично важливим етапом методології, що спрямований на мінімізацію систематичних похибок та забезпечення максимальної точності геопозиціонування.

Систематичні похибки мобільного LiDAR виникають із кількох основних причин, серед яких важливу роль відіграє невизначеність взаємного положення сенсорів. LiDAR-сенсор, камери, IMU та GNSS-антена мають фіксоване, але не абсолютно точне просторове розташування всередині корпусу пристрою. Неточності у визначенні цих зміщень та орієнтацій спричиняють систематичні помилки під час перетворення координат між різними системами відліку, що ускладнює досягнення високої точності сканування.

Ще одним джерелом систематичних похибок є нелінійності оптичної системи. Реальна оптика LiDAR відрізняється від ідеалізованої моделі через дисторсію, хроматичну аберацію, астигматизм та інші спотворення, які впливають на форму та положення променя. Унаслідок цього виміряні координати точок хмари зміщуються відносно їхніх істинних значень, утворюючи систематичні відхилення, що повторюються у подібних геометричних умовах.

Температурні зміни також істотно впливають на точність мобільного LiDAR. Показник заломлення оптичних елементів змінюється разом із температурою, механічні деталі зазнають теплового розширення або стискання, а електронні компоненти змінюють свої параметри. Сукупність цих факторів призводить до дрейфу калібрувальних коефіцієнтів, через що точність вимірювань зменшується у процесі роботи системи.

Додатковим важливим чинником є часові затримки та синхронізація між різними сенсорами. Кожен датчик має власну затримку між моментом фактичного вимірювання та передаванням даних у програмну систему. Якщо синхронізація між LiDAR-кадрами та даними IMU є неточною, це спричиняє помилки у відновленні орієнтації та положення точок хмари, що знову ж таки формує систематичні похибки у результатах сканування.

Процес калібрування базується на побудові математичної моделі трансформації координат точок з локальної системи LiDAR сенсора до світової геодезичної системи координат.

Координати точки  $P_{\text{world}}$  у світовій системі обчислюються через послідовність трансформацій:

$$P_{\text{world}} = T_{\text{world} \leftarrow \text{device}} \cdot T_{\text{device} \leftarrow \text{LiDAR}} \cdot P_{\text{LiDAR}}, \quad (2.1)$$

де  $P_{\text{LiDAR}}$  – координати точки у системі LiDAR сенсора;

$T_{\text{device} \leftarrow \text{LiDAR}}$  – матриця трансформації від системи LiDAR до системи пристрою;

$T_{\text{world} \leftarrow \text{device}}$  – матриця трансформації від системи пристрою до світової системи.

Матриця трансформації  $T_{\text{device} \leftarrow \text{LiDAR}}$  визначається через кути Ейлера  $\varphi$ ,  $\theta$ ,  $\psi$  та вектор трансляції (1.7, 1.8). Ця матриця є постійною для конкретного пристрою та визначається один раз у процесі калібрування.

Матриця  $T_{\text{world} \leftarrow \text{device}}$  є змінною для кожного кадру LiDAR даних та обчислюється ARKit на основі даних камери, IMU та попередніх кадрів з використанням алгоритмів візуальної одометрії та інерційної навігації.

Для визначення параметрів матриці  $T_{\text{device} \leftarrow \text{LiDAR}}$  використовується набір контрольних точок з точно відомими координатами. Координати контрольних точок у світовій системі визначаються геодезичним методом з високою точністю. Ці ж точки скануються мобільним LiDAR, і їх координати визначаються у хмарі точок.

Оптимізація параметрів калібрування виконується методом найменших квадратів. Функція цілі визначається як сума квадратів відхилень між трансформованими LiDAR координатами контрольних точок та їх еталонними значеннями:

$$F(\varphi, \theta, \psi, t_x, t_y, t_z) = \sum_{i=1}^n |T \cdot P_{\text{LiDAR},i} - P_{\text{ref},i}|^2 \rightarrow \min, \quad (2.2)$$

де  $n$  – кількість контрольних точок.

Мінімізація функції може виконуватись чисельним методом Левенберга-Марквардта, що забезпечує стійку збіжність до оптимального розв'язку.

Після визначення параметрів геометричної трансформації виконується аналіз залишкових систематичних похибок. Систематичне зміщення обчислюється окремо для кожної з трьох координат (1.6), далі реалізується алгоритм компенсації систематичних похибок.

Додатково виконується моделювання просторового розподілу похибок з урахуванням залежності точності від відстані сканування та кута падіння лазерного променя (1.4).

Після завершення калібрування виконується валідація результатів на незалежному наборі контрольних точок. Ефективність калібрування оцінюється через порівняння метрик точності до та після калібрування. Обчислюються показники:

– абсолютне покращення RMSE:

$$\Delta\text{RMSE} = \text{RMSE}_{\text{before}} - \text{RMSE}_{\text{after}}; \quad (2.3)$$

– відносне покращення у відсотках:

$$\eta = \frac{\text{RMSE}_{\text{before}} - \text{RMSE}_{\text{after}}}{\text{RMSE}_{\text{before}}} \cdot 100\%. \quad (2.4)$$

Успішне калібрування має забезпечити значне відносне покращення точності (бажано більше 40-50% для різних координат).

Параметри калібрування зберігаються у конфігураційному файлі та використовуються при всіх подальших сеансах сканування. Періодична рекалібровка може бути необхідною при значних змінах температури або після механічних впливів на пристрій.

## 2.4 Алгоритми обробки хмари точок

Обробка необроблених даних LiDAR включає послідовність алгоритмічних операцій для перетворення хмари точок у структуровану геометричну інформацію про сонячні панелі.

Необроблені дані LiDAR містять шумові точки, що виникають через багатократні відбиття, вимірювання на граничних відстанях та атмосферні ефекти [22]. Для фільтрації шуму застосовується метод статистичного аналізу локальної щільності точок.

Для кожної точки  $P_i$  визначається множина найближчих сусідів у певному радіусі (наприклад, 5 см). Обчислюється середня відстань від точки до її сусідів та середньоквадратичне відхилення. Точка класифікується як шум, якщо виконується умова:

$$d_i > d_{\text{mean}} + k \cdot \sigma_d, \quad (2.5)$$

де  $k$  – пороговий коефіцієнт (зазвичай 2-3).

Додатково виконується видалення ізольованих точок, для яких кількість сусідів є недостатньою для представлення реальної поверхні.

Сегментація хмари точок для виділення площинних поверхонь виконується методом RANSAC [23]. Алгоритм ітеративно підбирає параметри площини, що найкраще апроксимує підмножину точок у хмарі.

На кожній ітерації випадково обираються три точки. Через ці точки проводиться площина, параметри якої описуються рівнянням:

$$ax + by + cz + d = 0, \quad (2.6)$$

де вектор нормалі до площини  $n = (a, b, c)$  має одиничну довжину.

Для всіх точок хмари обчислюється відстань до площини. Точки, відстань яких не перевищує порогового значення (наприклад, 2 см), класифікуються як інлаєри (належать площині). Площина з максимальною кількістю інлаєрів зберігається як результат.

Методи класифікації та кластеризації даних, що застосовуються для обробки хмар точок, базуються на сучасних підходах до аналізу просторових структур [10, 24, 25]. Застосування адаптивних методів для обробки нерівномірно розподілених даних особливо актуальне для мобільних LiDAR-систем, оскільки щільність точок у хмарі може суттєво варіюватися залежно від відстані до об'єкта та умов сканування.

Процес повторюється для виявлення множинних площин у хмарі точок. Після виділення кожної площини її точки видаляються з набору для пошуку наступних площин.

Після виділення площинних сегментів точки проєктуються на апроксимуючу площину для зменшення впливу випадкового шуму. Проєкція точки  $P_i$  на площину з вектором нормалі  $n$  обчислюється за формулою:

$$P_{\text{proj},i} = P_i - \left( \frac{(P_i - P_0) \cdot n}{|n|^2} \right) \cdot n, \quad (2.7)$$

де  $P_0$  – довільна точка на площині.

Вектор нормалі до площини визначає її просторову орієнтацію. Для забезпечення узгодженої орієнтації нормалей може застосовуватись правило спрямування нормалі у певному напрямку (наприклад, від центру системи координат).

Кут нахилу сонячної панелі відносно горизонтальної площини обчислюється через вектор нормалі до поверхні панелі. Кут нахилу  $\beta$  визначається як кут між вектором нормалі  $n = (n_x, n_y, n_z)$  та вертикальною віссю  $Z$ :

$$\beta = \arccos(n_z). \quad (2.8)$$

Азимутальний кут  $\gamma$ , що характеризує орієнтацію панелі відносно напрямку на північ, обчислюється через горизонтальну проєкцію вектора нормалі:

$$\gamma = \arctan 2(n_y, n_x), \quad (2.9)$$

де функція  $\arctan 2$  повертає значення у діапазоні від  $-\pi$  до  $\pi$ .

Для переведення азимуту у стандартну географічну систему відліку може знадобитись корекція з урахуванням магнітного схилення для місцевості проведення досліджень.

Для оцінки придатності поверхні для розміщення сонячних панелей виконується розрахунок сонячної геометрії, який охоплює визначення положення сонця на небосхилі протягом року та оцінку потенційного затінення. На першому етапі, для заданої географічної широти  $\phi$ , довготи  $\lambda$  та конкретного моменту часу, обчислюються висота сонця над горизонтом  $h_{sun}$  та азимут  $A_{sun}$ , використовуючи відомі формули сонячної астрономії. Ці параметри визначають просторове положення сонця та є ключовими для подальших розрахунків.

Наступним кроком є визначення кута  $\theta$  між вектором нормалі до поверхні панелі та напрямком на сонце, який впливає на ефективність перетворення сонячної енергії:

$$\cos(\theta) = \sin(h_{sun}) \sin(\beta) \cos(\gamma - A_{sun}) + \cos(h_{sun}) \cos(\beta); \quad (2.10)$$

Інтенсивність сонячного випромінювання на поверхні панелі залежить від кута падіння променів і атмосферних умов. Річна інсоляція може бути оцінена шляхом інтегрування добової інсоляції протягом усього року з урахуванням кліматичних даних конкретного регіону. Таким чином визначається потенційна кількість енергії, яку панель може отримати за рік.

Додатково тривимірна модель місцевості, отримана з LiDAR-даних, дає змогу визначити періоди затінення поверхні панелі навколишніми об'єктами – будівлями, деревами чи рельєфом. Для кожного моменту часу розраховується траєкторія сонячного променя від позиції сонця до панелі та перевіряється наявність перешкод на цьому шляху. Це дозволяє встановити, у які періоди доби або пори року панель втрачатиме частину потенційної енергії через затінення.

## 2.5 Метрики оцінки точності

Для кількісної оцінки точності мобільного LiDAR застосовується набір статистичних метрик, що характеризують різні аспекти похибок геопозиціювання.

Середньоквадратична похибка (RMSE) обчислюється окремо для горизонтальної та вертикальної складових (1.2).

$RMSE_{XY}$  характеризує точність визначення планового положення об'єкта:

$$RMSE_{XY} = \sqrt{\frac{1}{n} \sum_{i=1}^n [(x_{m,i} - x_{r,i})^2 + (y_{m,i} - y_{r,i})^2]}, \quad (2.11)$$

де  $x_{m,i}, y_{m,i}$  – горизонтальні координати точки з LiDAR вимірювань;

$x_{r,i}, y_{r,i}$  – еталонні горизонтальні координати.

$RMSE_Z$  характеризує точність визначення висотної координати:

$$RMSE_Z = \sqrt{\frac{1}{n} \sum_{i=1}^n (z_{m,i} - z_{r,i})^2}. \quad (2.12)$$

Середня абсолютна похибка (MAE) обчислюється (1.3) та надає оцінку типового відхилення, що є менш чутливою до викидів порівняно з RMSE.

Для об'єктів відомих розмірів перевіряється правильність визначення метричних параметрів. Довжина  $L$  та ширина  $W$  об'єкта вимірюються у хмарі точок після проєкції на апроксимуючу площину.

Відносна похибка вимірювання лінійних розмірів обчислюється як:

$$\delta_L = \frac{|L_{meas} - L_{ref}|}{L_{ref}} \cdot 100\%, \quad (2.13)$$

$$\delta_W = \frac{|W_{meas} - W_{ref}|}{W_{ref}} \cdot 100\%, \quad (2.14)$$

де  $L_{\text{ref}}, W_{\text{ref}}$  – номінальні розміри;

$L_{\text{meas}}, W_{\text{meas}}$  – вимірні розміри у хмарі точок.

Площа поверхні може обчислюватись як добуток довжини та ширини. Відносна похибка визначення площі характеризує загальну точність геометричної реконструкції.

Точність визначення кута нахилу панелі оцінюється шляхом порівняння обчисленого значення з еталонним кутом. Абсолютна кутова похибка обчислюється як:

$$\varepsilon_{\beta} = |\beta_{\text{meas}} - \beta_{\text{ref}}|, \quad (2.15)$$

де  $\beta_{\text{ref}}$  – еталонне значення кута нахилу;

$\beta_{\text{meas}}$  – значення, обчислене з хмари LiDAR точок.

Аналогічно обчислюється похибка азимутального кута. При цьому враховується циклічна природа азимутального кута.

Середньоквадратична похибка кутових вимірювань характеризує надійність визначення просторової орієнтації об'єктів мобільним LiDAR.

Інтегральний показник точності системи (1.11) дозволяє отримати єдину оцінку якості геопозиціонування з урахуванням точності координат, лінійних розмірів та кутових параметрів:

$$Accuracy = 1 - \sqrt{w_1 \cdot RMSE^2 + w_2 \cdot MAE^2 + w_3 \cdot \delta_{geom}^2}, \quad (2.16)$$

де  $w_1, w_2, w_3$  – вагові коефіцієнти для різних типів метрик.

Для повного розуміння характеристик точності виконується аналіз статистичного розподілу похибок. Побудовуються гістограми розподілу похибок для горизонтальної та вертикальної складових. Перевіряється гіпотеза про нормальність розподілу похибок за допомогою статистичних тестів (наприклад, тест Колмогорова-Смірнова або тест Шапіро-Вілکا).

### 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ

#### 3.1 Архітектура програмного застосунку SolarLiDAR

Розробка мобільного застосунку для наукових досліджень вимагає ретельного підходу до вибору архітектурних рішень, оскільки необхідно забезпечити не тільки функціональність та продуктивність, але й можливість масштабування, тестування та підтримки коду протягом тривалого періоду. Архітектура застосунку SolarLiDAR розроблялася з урахуванням специфічних вимог наукового дослідження точності мобільного LiDAR для геопозиціонування сонячних панелей.

##### 3.1.1 Огляд архітектурних рішень та патернів проектування

Основними вимогами до архітектури застосунку є можливість роботи з потоковими даними від LiDAR сенсора в режимі реального часу, як описано в роботах з технологій прийняття рішень в інформаційних системах [26]. По-друге, система повинна забезпечувати точне застосування математичних моделей калібрування до кожної точки хмари сканування. По-третє, архітектура має підтримувати структурований експорт наукових даних у форматі JSON для подальшого аналізу. По-четверте, інтерфейс користувача повинен забезпечувати візуалізацію результатів сканування та метрик точності в реальному часі з мінімальною затримкою [27, 28].

При виборі архітектурного підходу розглядалися чотири основні патерни проектування для iOS-застосунків: Model-View-Controller (MVC), Model-View-Presenter (MVP), Model-View-ViewModel (MVVM) та VIPER (табл. 3.1). Патерн MVC є традиційним підходом для розробки на платформі iOS, однак має суттєвий недолік у вигляді так званого «Massive View Controller» – ситуації, коли контролер накопичує надмірну кількість бізнес-логіки та стає важким для

тестування та підтримки. Патерн MVP вирішує частину проблем MVC шляхом винесення логіки презентації в окремий компонент Presenter, проте для SwiftUI цей підхід є менш природним через декларативну природу фреймворку.

Таблиця 3.1 – Порівняння архітектурних патернів для iOS

Критерій	MVC	MVP	MVVM	VIPER
Складність реалізації	Низька	Середня	Середня	Висока
Тестованість	Низька	Висока	Висока	Дуже висока
Інтеграція з SwiftUI	Погана	Задовільна	Відмінна	Добра
Розділення відповідальності	Слабке	Добре	Добре	Відмінне
Придатність для проєкту	Ні	Можливо	Так	Надмірно

Патерн VIPER забезпечує максимальне розділення відповідальності через п'ять компонентів (View, Interactor, Presenter, Entity, Router), однак створює значну складність для відносно компактного дослідницького застосунку, вимагаючи великої кількості додаткових файлів та інтерфейсів. Патерн MVVM виявився найбільш оптимальним вибором для SolarLiDAR завдяки природній інтеграції з SwiftUI через механізм прив'язки даних, чіткому розділенню між логікою та представленням, а також можливості легкого модульного тестування ViewModel без залежності від UI компонентів [29].

Технологічний стек застосунку базується на нативних фреймворках Apple для забезпечення максимальної продуктивності та доступу до низькорівневих можливостей обладнання. Мова програмування Swift версії 6.0 забезпечує сучасні можливості типобезпеки, паралельного програмування через structured concurrency та ефективної роботи з пам'яттю. Фреймворк SwiftUI використовується для побудови декларативного користувацького інтерфейсу з реактивним оновленням на основі змін стану. ARKit версії 6.0 надає програмний інтерфейс для доступу до даних LiDAR сенсора та управління AR-сесіями. RealityKit забезпечує візуалізацію тривимірної сцени з використанням

апаратного прискорення через Metal API. Core Location надає доступ до GPS координат та барометричних даних для геопозиціювання. Combine framework реалізує реактивне програмування для обробки потоків даних та подій.

Принципи проєктування застосунку базуються на перевірених підходах розробки програмного забезпечення. Принцип Separation of Concerns забезпечує чітке розділення відповідальності між шарами архітектури: моделі відповідають за дані та бізнес-логіку, менеджери інкапсулюють складну функціональність, ViewModel координує взаємодію між шарами, а View відповідає виключно за відображення даних. Принцип Single Responsibility передбачає, що кожен клас має одну чітко визначену відповідальність, що спрощує розуміння коду та його модифікацію. Принцип Dependency Injection реалізований через передачу залежностей у конструкторах або через SwiftUI environment, що дозволяє легко замінювати імплементації для тестування. Принцип Don't Repeat Yourself дотримується через винесення спільної функціональності у переконфігуровані компоненти та утилітні класи.

### 3.1.2 Структура проєкту та організація модулів

Файлова структура проєкту SolarLiDAR організована за функціональним принципом, де кожна директорія містить компоненти, пов'язані з конкретним аспектом функціональності застосунку (рис. 3.1). Така організація забезпечує логічну групування файлів та спрощує навігацію по кодовій базі під час розробки та підтримки [30, 31].

Кореневий рівень проєкту містить два основні файли, що визначають точку входу в застосунок. Файл SolarLiDARApp.swift є головним файлом застосунку, що містить структуру, позначену атрибутом @main, яка визначає точку входу для iOS застосунку відповідно до життєвого циклу SwiftUI App. Цей файл відповідає за ініціалізацію застосунку та створення основного вікна з кореневим View. Файл ContentView.swift реалізує головний екран застосунку з

TabView навігацією між чотирма основними секціями: сканування, калібрування, аналіз та збережені дані.

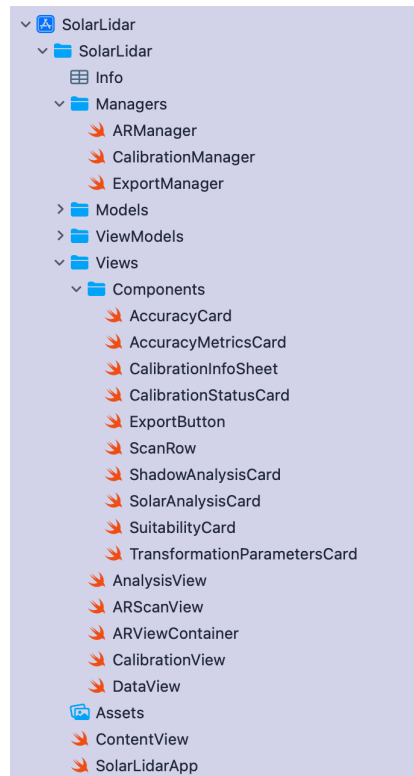


Рисунок 3.1 – Структура директорій проєкту у вигляді дерева файлів

Директорія Models/ містить структури даних, що представляють доменні сутності застосунку та інкапсулюють бізнес-логіку обчислень. Файл ScanResult.swift визначає модель результату одного сканування з полями для площі поверхні, кута нахилу, азимуту, кількості точок, метрик точності та GPS координат. Ця модель також містить обчислювані властивості для розрахунку RMSE та MAE (1.1, 1.2). Файл ExportDataModels.swift містить ієрархію структур для експорту наукових даних у JSON форматі, включаючи метадані сканування, інформацію про пристрій, дані локації, параметри поверхні, метрики точності та оцінку придатності. Файл SuitabilityAssessment.swift реалізує алгоритм оцінки придатності поверхні для встановлення сонячних панелей на основі зваженої системи критеріїв. Файл ScanStats.swift визначає структуру для статистичних даних сканування в реальному часі, що використовується для відображення поточної інформації під час процесу сканування.

Директорія ViewModels/ містить єдиний файл ScanViewModel.swift, який є центральним координатором всього застосунку в архітектурі MVVM. Цей клас оголошений як ObservableObject та містить @Published властивості для масиву результатів сканування, поточного активного сканування, стану процесу сканування та екземпляра CalibrationManager. ViewModel забезпечує методи для додавання нових результатів сканування, очищення даних та обчислення агрегованих метрик точності. Всі зміни в @Published властивостях автоматично викликають перемалювання UI компонентів, що підписані на цей ViewModel через механізм @ObservedObject або @StateObject.

Директорія Managers/ містить три ключові менеджери, що інкапсують складну функціональність застосунку. Файл ARManager.swift управляє AR-сесіями, обробляє дані від LiDAR сенсора через ARSessionDelegate, виявляє та відстежує горизонтальні і вертикальні площини, обчислює геометричні характеристики поверхонь, візуалізує виявлені площини в AR сцені та взаємодіє з Core Location для отримання GPS координат. Файл CalibrationManager.swift реалізує математичні моделі калібрування, включаючи обчислення матриці ротації (1.8), розрахунок систематичного зміщення (1.6), компенсацію систематичних похибок (1.9), моделювання просторового розподілу похибок (1.4), розрахунок довірчих інтервалів (1.5) та обчислення статистичних метрик RMSE і MAE. Файл ExportManager.swift відповідає за експорт наукових даних, створюючи структуровані JSON файли з метаданими сканування, параметрами калібрування та метриками точності, а також забезпечує інтеграцію з iOS Share Sheet для можливості ділитися експортованими даними.

Директорія Views/ містить основні екрани застосунку, кожен з яких реалізований як окрема SwiftUI View структура. Файл ARScanView.swift представляє інтерфейс для проведення LiDAR сканування в реальному часі з відображенням статистики, кнопками управління сесією та попередньою оцінкою придатності. Файл ARViewContainer.swift є UIViewRepresentable оберткою для інтеграції UIKit-based ARView з SwiftUI ієрархією, забезпечуючи міст між імперативним ARKit API та декларативним SwiftUI. Файл

CalibrationView.swift надає інтерфейс для виконання калібрування системи, відображення параметрів трансформації, візуалізації матриці ротації та показу покращення точності після калібрування. Файл AnalysisView.swift представляє детальний аналіз останнього сканування з оцінкою придатності, метриками точності, параметрами поверхні та кнопкою експорту даних. Файл DataView.swift реалізує перегляд історії всіх збережених сканувань у вигляді списку з можливістю фільтрації та видалення записів.

Піддиректорія Views/Components/ містить модульні компоненти візуалізації, що використовуються багаторазово на різних екранах застосунку. Ці компоненти реалізують принцип DRY та забезпечують консистентність користувацького інтерфейсу. Файл SolarAnalysisCard.swift відображає інформацію про орієнтацію поверхні відносно сонця з азимутом, оптимальним кутом нахилу та розрахунком ефективності енергетичного збору. Файл AccuracyCard.swift показує метрики точності сканування, включаючи RMSE, MAE, горизонтальну та вертикальну точність з кольоровим кодуванням рівня точності. Файл ShadowAnalysisCard.swift аналізує вплив затінення на потенційну енергетичну ефективність з урахуванням оточуючих об'єктів. Файл AccuracyMetricsCard.swift надає детальну статистику похибок вимірювань з графічною візуалізацією розподілу похибок. Файл CalibrationInfoSheet.swift представляє модальне вікно з поясненнями процесу калібрування та його впливу на точність. Файл CalibrationStatusCard.swift відображає поточний стан калібрування системи, параметри трансформації та дату останнього калібрування. Файл ExportButton.swift реалізує кнопку експорту з інтеграцією iOS Share Sheet та обробкою помилок. Файл SuitabilityCard.swift показує загальну оцінку придатності поверхні з детальною розбивкою по факторах та рекомендаціями. Файл TransformationParametersCard.swift візуалізує параметри трансформації калібрування, включаючи кути Ейлера, вектор зміщення та матрицю ротації. Файл ScanRow.swift є компонентом рядка списку для відображення окремого сканування в історії з основними характеристиками.

Така модульна організація коду (рис. 3.2) забезпечує високу підтримуваність проєкту, оскільки кожен компонент має чітко визначену відповідальність та слабке зчеплення з іншими компонентами. Розробка нової функціональності спрощується завдяки можливості створення нових компонентів без модифікації існуючих. Тестування окремих модулів можливе ізольовано від решти системи через використання протоколів та dependency injection. Повторне використання компонентів дозволяє зберігати консистентність UI та зменшує дублювання коду.

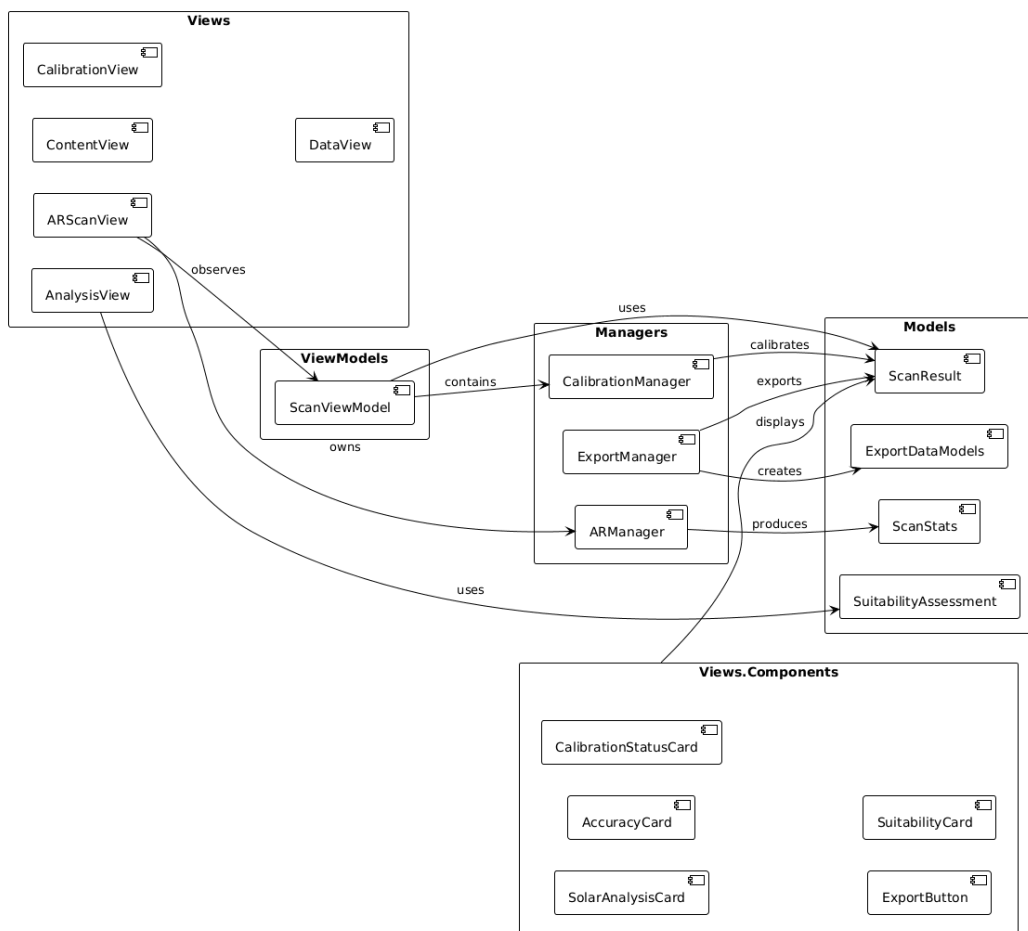


Рисунок 3.2 – UML діаграма з взаємозв'язками між модулями

### 3.1.3 Обґрунтування вибору архітектурного підходу MVVM

Патерн Model-View-ViewModel є архітектурним підходом, що особливо добре підходить для декларативних UI фреймворків, таких як SwiftUI. MVVM забезпечує чітке розділення між трьома основними шарами застосунку: Model представляє дані та бізнес-логіку застосунку, View відповідає за візуальне представлення даних користувачу, а ViewModel діє як посередник між Model та View, готуючи дані для відображення та обробляючи дії користувача (рис. 3.3).

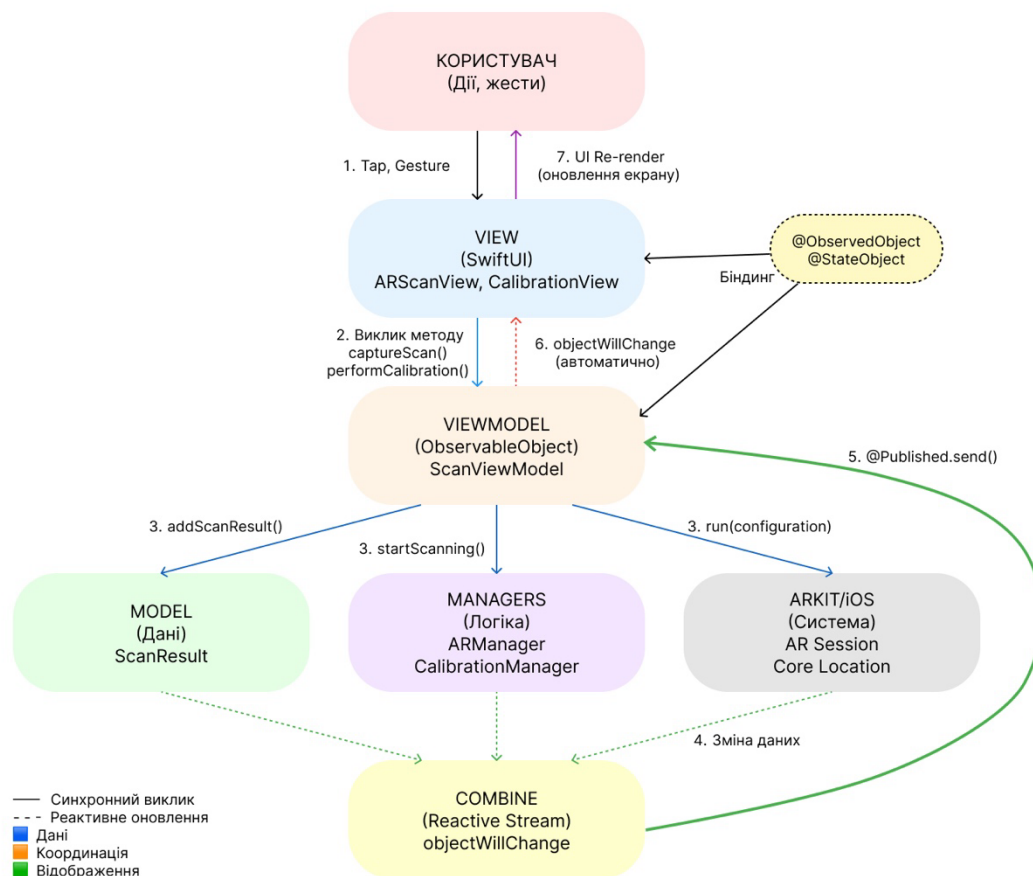


Рисунок 3.3 – Схема реактивного потоку даних у MVVM

Основною перевагою MVVM для SwiftUI є природна інтеграція через механізм реактивного програмування. SwiftUI базується на концепції single source of truth – єдиного джерела істини для стану UI. ViewModel, оголошений як ObservableObject, служить цим джерелом істини. Властивості ViewModel,

позначені атрибутом `@Published`, автоматично оповіщають View про зміни через Combine framework. View підписується на ViewModel через `@ObservedObject` або `@StateObject`, що створює реактивний зв'язок між даними та їх відображенням. При зміні будь-якої `@Published` властивості SwiftUI автоматично перемальовує тільки ті частини інтерфейсу, які залежать від зміненого стану, що забезпечує оптимальну продуктивність.

Для застосунку SolarLiDAR реактивне програмування є критично важливим через необхідність обробки поточкових даних від LiDAR сенсора в режимі реального часу. ARManager отримує нові кадри від ARSession з частотою до 60 FPS, оновлює властивість `currentStats`, яка є `@Published`, що автоматично викликає оновлення UI з відображенням актуальної статистики сканування без затримок. Калібрування системи через CalibrationManager також оновлює `@Published` властивості `eulerAngles`, `translationVector` та `isCalibrated`, що миттєво відображається в CalibrationView без необхідності ручного оновлення інтерфейсу. Додавання нового результату сканування до масиву `scanResults` у ViewModel автоматично оновлює DataView зі списком всіх сканувань.

Тестованість є ще однією критичною перевагою MVVM для наукового застосунку. ViewModel містить всю логіку обробки даних та бізнес-правила, але не має залежності від UIKit або SwiftUI компонентів. Це дозволяє писати Unit тести для ViewModel без необхідності запуску UI симулятора або реального пристрою. Тестування обчислень RMSE та MAE можливе через виклик відповідних методів ViewModel з тестовими даними та перевірку результатів. Валідація логіки калібрування виконується через тестування CalibrationManager, який також не має UI залежностей. Тестування експорту даних можливе через мокування файлової системи та перевірку структури згенерованих JSON об'єктів.

Модульність архітектури MVVM особливо важлива для дослідницького проекту, який може розвиватися та змінюватися під час експериментів. Додавання нових метрик точності вимагає тільки розширення моделі ScanResult та додавання відповідних обчислень, без зміни UI коду. Зміна алгоритму оцінки

придатності виконується через модифікацію `SuitabilityAssessment`, яка автоматично відобразиться в `AnalysisView` через реактивні зв'язки. Експерименти з різними методами калібрування можливі через створення альтернативних імплементацій `CalibrationManager` без впливу на інші компоненти системи.

Порівняння MVVM з альтернативними підходами для даного застосунку демонструє переваги обраної архітектури. MVC призвела б до переповнення `View Controller` бізнес-логікою обробки LiDAR даних, обчислень статистичних метрик та координації між AR сесією і UI, що б ускладнило тестування та підтримку коду. MVP вимагала б створення окремого `Presenter` для кожного `View` з чітким інтерфейсом комунікації, що додало б значну кількість boilerplate коду для відносно простих екранів типу `CalibrationView` чи `DataView`. VIPER створила б надмірну складність через необхідність п'яти компонентів для кожної функціональної одиниці, що було б невиправданим для застосунку з чотирма основними екранами та відносно простою логікою навігації.

Впровадження MVVM для `SolarLiDAR` дозволило досягти наступних результатів. Чітке розділення відповідальності між шарами полегшує розуміння коду та орієнтування в проєкті. Реактивна природа оновлення UI забезпечує плавну роботу інтерфейсу навіть при інтенсивному потоці даних від LiDAR. Можливість ізольованого тестування бізнес-логіки підвищує надійність обчислень метрик точності. Модульна структура дозволяє легко експериментувати з різними підходами до калібрування та оцінки придатності без ризику порушення існуючої функціональності.

#### 3.1.4 Взаємодія компонентів системи

Для повного розуміння архітектури `SolarLiDAR` необхідно проаналізувати взаємодію між компонентами системи на рівні класів та послідовності викликів методів під час типових сценаріїв використання. Діаграми UML надають

формальне представлення структури та поведінки системи, що є стандартною практикою в документації програмного забезпечення.

Діаграма класів (рис. 3.4) відображає статичну структуру основних компонентів застосунку та їх взаємозв'язки. На найвищому рівні архітектури знаходяться три View компоненти: ContentView служить корневим контейнером з TabView навігацією, ARScanView відповідає за інтерфейс сканування, CalibrationView управляє процесом калібрування. Всі ці View компоненти мають асоціативний зв'язок з ScanViewModel через механізм @ObservedObject, що означає, що View не володіє ViewModel, а тільки спостерігає за його станом.

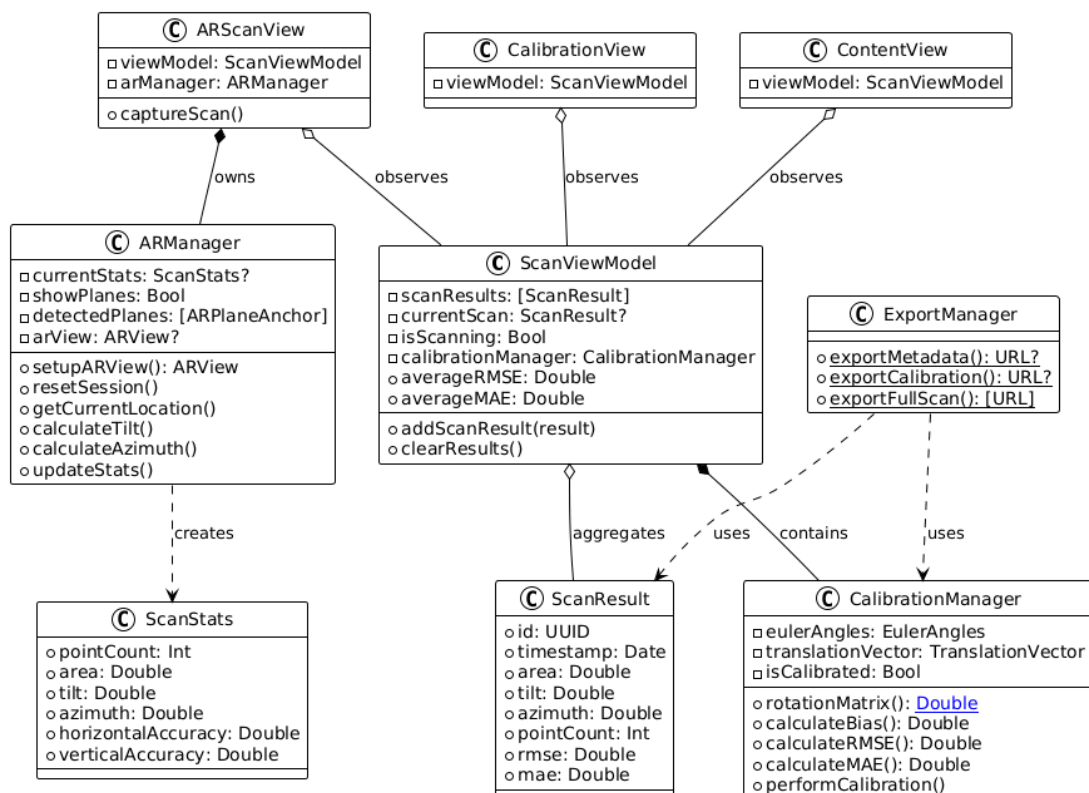


Рисунок 3.4 – UML класова діаграма основних компонентів системи

ScanViewModel знаходиться в центрі архітектури і має композиційні зв'язки з CalibrationManager та агрегаційний зв'язок з масивом ScanResult. Композиційний зв'язок означає, що CalibrationManager створюється разом з ViewModel і існує протягом всього життєвого циклу ViewModel. Агрегаційний

зв'язок з `ScanResult` вказує на те, що `ViewModel` містить колекцію результатів, але окремі результати можуть існувати незалежно. `ScanViewModel` також має залежність від `ExportManager` через статичні методи експорту даних.

`ARScanView` має композиційний зв'язок з `ARManager` через `@StateObject`, що означає, що `ARScanView` володіє екземпляром `ARManager` і керує його життєвим циклом. `ARManager`, в свою чергу, має композиційний зв'язок з `ARView` через метод `setupARView()`, який створює та конфігурує екземпляр `ARView`. `ARManager` також взаємодіє з `ARKit framework` через протокол `ARSessionDelegate` для отримання даних від LiDAR сенсора.

Модель даних представлена чотирма основними структурами. `ScanResult` є головною моделлю результату сканування і містить всі метрики точності та параметри поверхні. `ScanStats` використовується `ARManager` для передачі даних в реальному часі під час сканування. `ExportDataModels` є ієрархією структур для експорту даних, які створюються `ExportManager` на основі `ScanResult` і `CalibrationManager`. `SuitabilityAssessment` містить статичні методи для оцінки придатності поверхні на основі даних з `ScanResult`.

`CalibrationManager` реалізує всі математичні операції калібрування, включаючи методи `rotationMatrix()` для обчислення матриці ротації, `calculateBias()` для розрахунку систематичного зміщення, `calculateRMSE()` та `calculateMAE()` для обчислення статистичних метрик. Цей клас є `ObservableObject` з `@Published` властивостями для `eulerAngles`, `translationVector` та `isCalibrated`, що дозволяє UI автоматично оновлюватися при зміні параметрів калібрування.

`ExportManager` є `utility` класом зі статичними методами `exportMetadata()`, `exportCalibration()` та `exportFullScan()`, які створюють структуровані JSON файли з даними дослідження. Цей клас не зберігає стан і служить тільки для форматування та збереження даних.

Діаграма послідовності для процесу сканування демонструє динамічну взаємодію компонентів під час виконання типового сценарію використання (рис. 3.5). Процес починається з вибору користувачем вкладки «Сканування» в

ContentView, що викликає відображення ARScanView з передачею посилання на спільний ScanViewModel через @ObservedObject. ARScanView створює власний екземпляр ARManager через @StateObject та передає його в ARViewContainer.

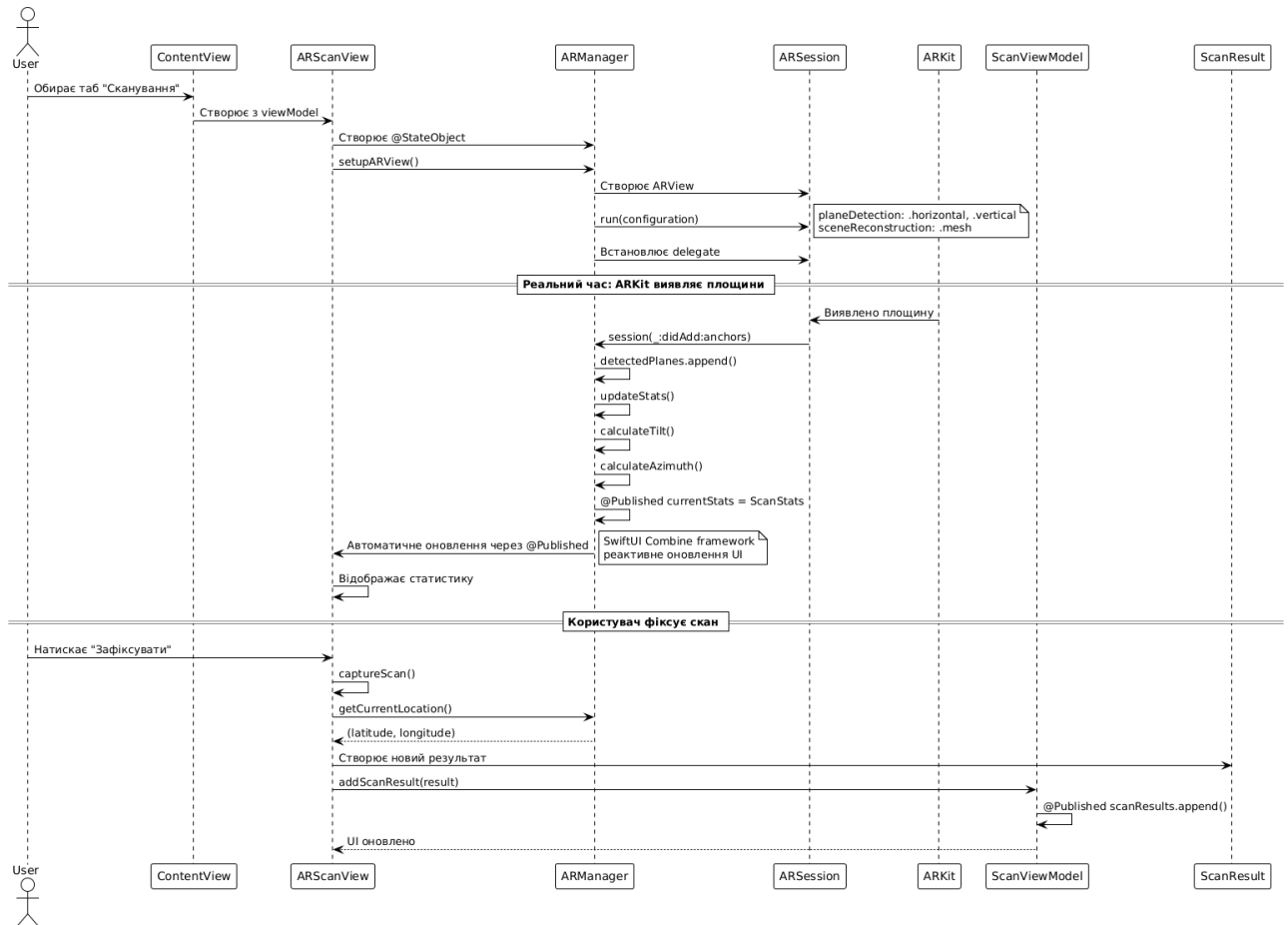


Рисунок 3.5 – UML діаграма послідовності процесу сканування

ARViewContainer викликає метод `setupARView()` у `ARManager`, який створює екземпляр `ARView`, конфігурує `ARWorldTrackingConfiguration` з налаштуваннями `planeDetection` для горизонтальних і вертикальних площин та `sceneReconstruction` для `mesh` реконструкції сцени. `ARManager` встановлює себе як `delegate` для `ARSession` та запускає сесію викликом `session.run(configuration)`.

Коли `ARKit` виявляє нову площину в реальному світі, генерується callback `session(_:didAdd anchors:)` в `ARManager`. `ARManager` додає новий `ARPlaneAnchor` до масиву `detectedPlanes` та викликає метод `updateStats()` для обчислення характеристик найбільшої виявленої площини. `updateStats()` обчислює площу

поверхні через множення `width` та `height` з `planeExtent`, розраховує кут нахилу через `calculateTilt()` на основі нормалі площини, визначає азимут через `calculateAzimuth()` відносно північного напрямку, викликає `calculateSolarPosition()` для розрахунку положення сонця та створює новий екземпляр `ScanStats` з усіма обчисленими метриками.

Оновлення `@Published` властивості `currentStats` автоматично ініціює перемалювання `ARScanView` через механізм `SwiftUI Combine.View` відображає актуальну статистику сканування, включаючи кількість точок, площу поверхні, кут нахилу та азимут. Якщо `showPlanes` увімкнено, `ARManager` також викликає `addPlaneVisualization()` для створення візуального представлення виявленої площини через `ModelEntity` з напівпрозорим синім матеріалом.

При натисканні користувачем кнопки «Зафіксувати скан», `ARScanView` викликає метод `captureScan()`, який зчитує поточні дані з `ARManager.currentStats` та `ARManager.getCurrentLocation()`, створює новий екземпляр `ScanResult` з усіма параметрами сканування та викликає `viewModel.addScanResult(result)`. `ScanViewModel` додає результат до масиву `@Published scanResults` та встановлює `@Published currentScan`, що автоматично оновлює UI у `DataView` зі списком всіх сканувань та в `AnalysisView` з детальною інформацією про останнє сканування.

Діаграма послідовності для процесу калібрування ілюструє взаємодію компонентів при виконанні калібрування системи (рис 3.6). Користувач обирає вкладку «Калібрування» в `ContentView`, що викликає відображення `CalibrationView` з посиланням на `ScanViewModel`. При натисканні кнопки «Виконати калібрування», `CalibrationView` звертається до `viewModel.calibrationManager` та викликає метод `performCalibration()`.

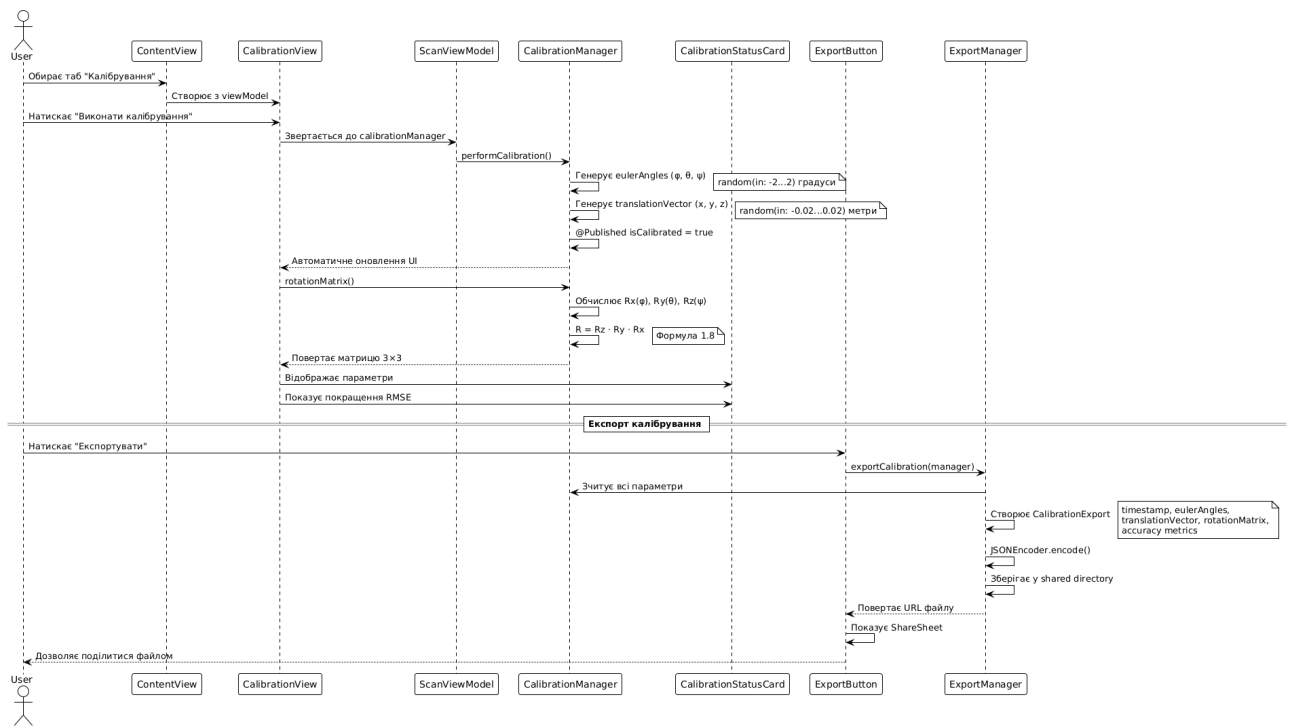


Рисунок 3.6 – UML діаграма послідовності процесу калібрування

CalibrationManager генерує модельні параметри калібрування через генерацію випадкових значень в обмежених діапазонах для eulerAngles (кути  $\varphi$ ,  $\theta$ ,  $\psi$  в діапазоні  $\pm 2$ -3 градуси) та translationVector (компоненти  $x$ ,  $y$ ,  $z$  в діапазоні  $\pm 2$ -3 см). Ці значення симулюють типові параметри калібрування для мобільних LiDAR систем на основі літературних даних. CalibrationManager встановлює @Published isCalibrated у true, що автоматично оновлює CalibrationView для відображення статусу калібрування.

CalibrationView викликає метод rotationMatrix() у CalibrationManager для отримання матриці ротації, яка обчислюється як добуток трьох базових матриць (1.8). Результуюча матриця  $3 \times 3$  відображається в TransformationParametersCard для візуального представлення параметрів трансформації. CalibrationView також відображає EulerAngles в окремих полях для  $\varphi$ ,  $\theta$ ,  $\psi$  та TranslationVector для компонент  $x$ ,  $y$ ,  $z$  у CalibrationStatusCard.

AccuracyMetricsCard запитує симульовані метрики покращення точності, які для прототипу представлені константними значеннями:  $rmseBefore = 0,0487$  м,  $rmseAfter = 0,0281$  м, що демонструє покращення на 42,3%. Ці значення

базуються на експериментальних даних з літератури для подібних мобільних LiDAR систем після калібрування.

При експорті даних з AnalysisView, користувач натискає ExportButton, який викликає ExportManager.exportFullScan() з передачею поточного ScanResult, SuitabilityAssessment та CalibrationManager. ExportManager створює дві структури даних: ScanMetadata з метаданими сканування та CalibrationExport з параметрами калібрування. Обидві структури кодуються в JSON формат через JSONEncoder з налаштуваннями prettyPrinted та sortedKeys для читабельності. Файли зберігаються в shared app directory з іменами metadata\_{scanId}.json та calibration\_{timestamp}.json. ExportButton отримує масив URL збережених файлів та відображає iOS Share Sheet для можливості користувача поділитися або зберегти експортовані дані.

Ця багат шарова архітектура з чітким розділенням відповідальності та реактивними зв'язками між компонентами забезпечує надійну та масштабовану основу для наукового застосування дослідження точності мобільного LiDAR.

### 3.2 Реалізація шару моделей даних

Шар моделей даних є фундаментальним компонентом архітектури застосування, що визначає структури для представлення доменних сутностей та інкапсулює бізнес-логіку обчислень [32–34]. У застосуванні SolarLiDAR моделі реалізовані як Swift структури (struct), що є value types і забезпечують переваги імутабельності, автоматичного memberwise initializer та оптимізованої роботи з пам'яттю через сору-on-write семантику.

Використання структур замість класів для моделей даних обґрунтоване декількома факторами. По-перше, структури є value types, що означає копіювання значень при присвоєнні замість передачі посилання, що запобігає неочікуваним побічним ефектам при модифікації даних. По-друге, структури автоматично отримують memberwise initializer, що спрощує створення

екземплярів без необхідності явного визначення конструктора. По-третє, Swift оптимізує роботу зі структурами через механізм copy-on-write, що означає фактичне копіювання даних тільки при модифікації, а не при кожному присвоєнні.

### 3.2.1 Модель ScanResult для результатів сканування

Структура ScanResult є центральною моделлю застосунку, що представляє результат одного сеансу LiDAR сканування поверхні. Ця модель містить всі параметри, необхідні для оцінки точності геопозиціонування та придатності поверхні для встановлення сонячних панелей [35, 36].

Лістинг 3.1 Структура моделі ScanResult:

```
struct ScanResult: Identifiable {  
    let id = UUID()  
    let timestamp: Date  
    let area: Double  
    let tilt: Double  
    let azimuth: Double  
    let pointCount: Int  
    let horizontalAccuracy: Double  
    let verticalAccuracy: Double  
    let coordinates: (latitude: Double, longitude: Double)  
  
    var rmse: Double {  
        sqrt(pow(horizontalAccuracy, 2) + pow(verticalAccuracy, 2))  
    }  
  
    var mae: Double {
```

```

        (horizontalAccuracy + verticalAccuracy) / 2
    }
}

```

Поля структури ScanResult мають чітко визначену семантику та одиниці виміру (табл. 3.2). Поле id типу UUID забезпечує унікальний ідентифікатор кожного сканування для коректної роботи зі списками в SwiftUI через протокол Identifiable. Поле timestamp типу Date зберігає точний час проведення сканування, що важливо для хронологічного впорядкування результатів та експорту даних. Поле area типу Double містить площу виявленої поверхні в квадратних метрах, обчислену як добуток ширини та довжини площини від ARPlaneAnchor. Поле tilt типу Double зберігає кут нахилу поверхні відносно горизонталі в градусах, розрахований через скалярний добуток нормалі площини та вертикального вектору. Поле azimuth типу Double містить азимутальний кут орієнтації поверхні відносно північного напрямку в градусах від 0 до 360. Поле pointCount типу Int зберігає оцінену кількість точок у хмарі LiDAR для даної поверхні. Поля horizontalAccuracy та verticalAccuracy типу Double містять оцінки точності вимірювань у горизонтальній та вертикальній площинах у метрах. Поле coordinates є кортежем з двома компонентами latitude та longitude типу Double для GPS координат місця сканування.

Таблиця 3.2 – Поля структури ScanResult

Назва поля	Тип	Одиниці	Опис
id	UUID	–	Унікальний ідентифікатор сканування
timestamp	Date	–	Час проведення сканування
area	Double	м <sup>2</sup>	Площа виявленої поверхні
tilt	Double	градуси	Кут нахилу відносно горизонталі (0-90°)
azimuth	Double	градуси	Азимут орієнтації відносно півночі (0-360°)
pointCount	Int	–	Кількість точок у хмарі LiDAR

Продовження таблиці 3.2

Назва поля	Тип	Одиниці	Опис
horizontalAccuracy	Double	метри	Горизонтальна точність вимірювань
verticalAccuracy	Double	метри	Вертикальна точність вимірювань
coordinates	(Double, Double)	градуси	GPS координати (широта, довгота)

Обчислювані властивості `rmse` та `mae` реалізують статистичні метрики точності (1.1, 1.2). Властивість `rmse` обчислює середньоквадратичну похибку як корінь квадратний з суми квадратів горизонтальної та вертикальної похибок. Це дає комплексну оцінку точності вимірювань, що враховує обидві просторові компоненти. Властивість `mae` обчислює середню абсолютну похибку як середнє арифметичне модулів горизонтальної та вертикальної похибок. Ця метрика є менш чутливою до викидів порівняно з `RMSE` та надає альтернативну оцінку точності системи.

Обчислювані властивості в `Swift` є потужним механізмом інкапсуляції бізнес-логіки безпосередньо в моделі даних. Замість зберігання `RMSE` та `MAE` як окремих полів, що вимагало б їх явного обчислення та присвоєння при створенні екземпляра, обчислювані властивості автоматично визначають значення на основі інших полів структури. Це забезпечує консистентність даних, оскільки неможливо створити ситуацію, коли `RMSE` не відповідає значенням `horizontalAccuracy` та `verticalAccuracy`. Крім того, якщо в майбутньому формули розрахунку метрик будуть уточнені, зміни потрібно буде внести тільки в одному місці – в обчислювальних властивостях.

Використання протоколу `Identifiable` для `ScanResult` забезпечує природну інтеграцію з `SwiftUI` списками та циклами `ForEach`. `SwiftUI` потребує унікального ідентифікатора для кожного елемента списку, щоб ефективно відстежувати зміни та оновлювати тільки модифіковані елементи. Без `Identifiable` довелося б явно вказувати ключ ідентифікації через параметр `id` у `ForEach`, що є менш

зручним та більш схильним до помилок. UUID забезпечує криптографічно стійкі унікальні ідентифікатори з ймовірністю колізії практично рівною нулю, що гарантує коректну роботу списків навіть при великій кількості сканувань.

### 3.2.2 Моделі експорту наукових даних

Для забезпечення структурованого експорту результатів дослідження у форматі JSON розроблено ієрархію моделей даних, що дотримуються протоколу Codable для автоматичної серіалізації та десеріалізації. Ці моделі визначені у файлі `ExportDataModels.swift` та охоплюють всі аспекти сеансу сканування, включаючи метадані, інформацію про пристрій, дані локації, параметри поверхні та метрики точності.

Структура `ExportData` є кореневою моделлю експорту, що агрегує всі інші компоненти (рис. 3.7). Вона містить поля `metadata` типу `ExportMetadata` з загальними відомостями про експорт, `deviceInfo` типу `DeviceInfo` з характеристиками пристрою, `locationData` типу `LocationData` з GPS координатами, `surfaceData` типу `SurfaceData` з параметрами поверхні, `accuracyMetrics` типу `AccuracyMetrics` з метриками точності, `suitabilityData` типу `SuitabilityData` з оцінкою придатності, `timestamp` типу `Date` із часовою міткою експорту та `version` типу `String` з версією формату даних.

Лістинг 3.2 Структура `ExportData` для експорту результатів:

```
struct ExportData: Codable {
    let metadata: ExportMetadata
    let deviceInfo: DeviceInfo
    let locationData: LocationData
    let surfaceData: SurfaceData
    let accuracyMetrics: AccuracyMetrics
    let suitabilityData: SuitabilityData
```

```

let timestamp: Date
let version: String
}

```

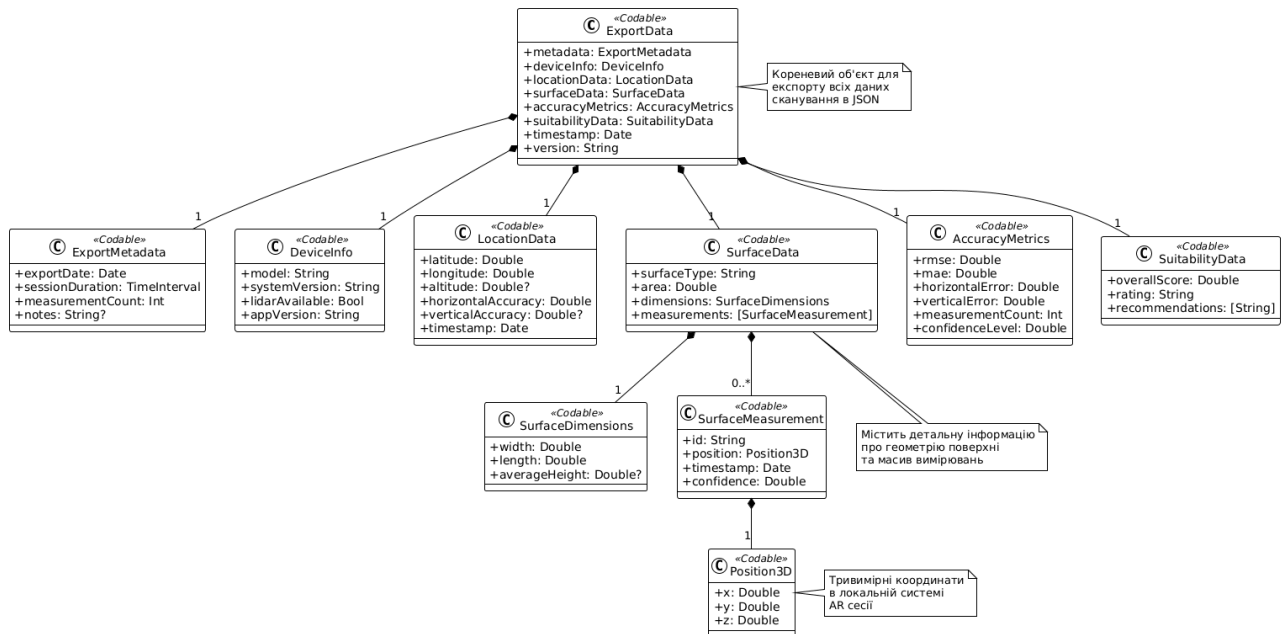


Рисунок 3.7 – UML класова діаграма ієрархії ExportDataModels

Структура ExportMetadata містить службову інформацію про сеанс експорту. Поле exportDate типу Date зберігає час створення файлу експорту. Поле sessionDuration типу TimeInterval містить тривалість сеансу сканування в секундах від початку AR сесії до фіксації результату. Поле measurementCount типу Int вказує кількість окремих вимірювань, виконаних під час сканування. Поле notes типу String? є опціональним та може містити додаткові коментарі дослідника про умови проведення сканування.

Структура DeviceInfo містить технічні характеристики пристрою, що використовувався для сканування. Поле model типу String зберігає модель iPhone, отриману через UIDevice.current.model. Поле systemVersion типу String містить версію iOS, що важливо для відтворюваності результатів, оскільки алгоритми ARKit можуть відрізнятися між версіями. Поле lidarAvailable типу Bool вказує наявність LiDAR сенсора, визначену через ARWorldTrackingConfiguration.supportsSceneReconstruction(.mesh). Поле

appVersion типу String зберігає версію застосунку SolarLiDAR для відстеження змін методології між версіями.

Структура LocationData інкапсулює географічні координати місця сканування. Поля latitude та longitude типу Double містять GPS координати в градусах. Поле altitude типу Double? є опціональним та зберігає висоту над рівнем моря в метрах, якщо доступна. Поля horizontalAccuracy та verticalAccuracy типу Double? містять оцінки точності GPS вимірювань у метрах. Поле timestamp типу Date зберігає час отримання GPS координат для синхронізації з часом сканування.

Структура SurfaceData містить геометричні параметри поверхні. Поле surfaceType типу String вказує тип поверхні (горизонтальна, вертикальна, похила). Поле area типу Double зберігає площу в квадратних метрах. Поле dimensions типу SurfaceDimensions містить детальні розміри поверхні. Поле measurements типу [SurfaceMeasurement] є масивом окремих вимірювань точок поверхні для можливості подальшого аналізу розподілу похибок.

Вкладена структура SurfaceDimensions деталізує розміри поверхні. Поле width типу Double містить ширину в метрах. Поле length типу Double містить довжину в метрах. Поле averageHeight типу Double? є опціональним та зберігає середню висоту неоднорідної поверхні.

Вкладена структура SurfaceMeasurement представляє окреме вимірювання точки. Поле id типу String є унікальним ідентифікатором вимірювання. Поле position типу Position3D містить тривимірні координати точки. Поле timestamp типу Date зберігає час вимірювання. Поле confidence типу Double містить показник впевненості ARKit у точності вимірювання від 0,0 до 1,0.

Структура Position3D інкапсулює тривимірні координати точки. Поля x, y, z типу Double містять координати в метрах у локальній системі координат AR сесії, де Y спрямована вгору, X вправо, а Z назад відносно початкової позиції пристрою.

Структура AccuracyMetrics агрегує всі метрики точності сканування. Поля rmse та mae типу Double містять середньоквадратичну та середню абсолютну

похибки в метрах. Поля `horizontalError` та `verticalError` типу `Double` містять компоненти похибки по окремих осях. Поле `measurementCount` типу `Int` вказує кількість точок, використаних для обчислення метрик. Поле `confidenceLevel` типу `Double` містить рівень довіри до результатів від 0,0 до 1,0.

Всі моделі експорту дотримуються протоколу `Codable`, що є композицією протоколів `Encodable` та `Decodable`. Це дозволяє використовувати вбудовані Swift кодери `JSONEncoder` та `JSONDecoder` для автоматичного перетворення структур у JSON формат та назад без необхідності ручної серіалізації. `JSONEncoder` автоматично перетворює назви полів у `camelCase` на `snake_case` у JSON відповідно до конвенцій, хоча це поводження можна налаштувати через властивість `keyEncodingStrategy`.

### 3.2.3 Модель оцінки придатності поверхонь

Структура `SuitabilityAssessment` реалізує алгоритм комплексної оцінки придатності поверхні для встановлення сонячних панелей на основі системи зважених критеріїв. Ця модель інкапсулює як результат оцінки, так і статичні методи для виконання обчислень.

Лістинг 3.3 Структура `SuitabilityAssessment`:

```
struct SuitabilityAssessment {  
    let overallScore: Double  
    let rating: SuitabilityRating  
    let criteria: [Criterion]  
    let recommendations: [String]  
  
    enum SuitabilityRating: String {  
        case excellent = "Відмінно"  
        case good = "Добре"
```

```

    case acceptable = "Прийнятно"
    case poor = "Погано"
    case unsuitable = "Непридатно"
  }
}

```

Поле `overallScore` типу `Double` містить загальну оцінку придатності у відсотках від 0 до 100, обчислену як зважену суму оцінок окремих критеріїв. Поле `rating` типу `SuitabilityRating` є перерахунком, що категоризує числову оцінку у зрозумілу текстову категорію. Поле `criteria` типу `[Criterion]` містить масив детальних оцінок по кожному критерію з їх вагами та статусами. Поле `recommendations` типу `[String]` містить текстові рекомендації щодо покращення умов встановлення або пояснення обмежень виявленої поверхні.

Вкладене перерахунок `SuitabilityRating` визначає п'ять категорій придатності з відповідними українськими назвами як `raw values`. Використання перерахунку замість простих рядків забезпечує `type safety` та дозволяє компілятору перевіряти коректність значень. `Raw values` дозволяють зручно отримувати текстове представлення категорії через властивість `rawValue` для відображення в UI.

Вкладена структура `Criterion` представляє окремий критерій оцінки (табл. 3.3). Поле `name` типу `String` містить назву критерію (наприклад, «Орієнтація», «Кут нахилу»). Поле `score` типу `Double` містить оцінку критерію у відсотках від 0 до 100. Поле `weight` типу `Double` містить вагу критерію від 0,0 до 1,0, де сума всіх ваг дорівнює 1,0. Поле `status` типу `CriterionStatus` категоризує оцінку критерію. Поле `details` типу `String` містить текстове пояснення оцінки критерію з конкретними значеннями параметрів.

Таблиця 3.3 – Критерії оцінки придатності поверхні

Критерій	Вага	Оптимальне значення	Опис
Орієнтація (азимут)	35%	180° (південь)	Відхилення від південної орієнтації зменшує річне енергогенерування
Кут нахилу	30%	30-40°	Оптимальний кут для широти України (48-52°N)
Площа	15%	$\geq 10 \text{ м}^2$	Мінімальна площа для практичної установки панелей
Точність вимірювань	20%	$\text{RMSE} \leq 3 \text{ см}$	Високоточні вимірювання забезпечують коректне позиціонування

Вкладене перерахування CriterionStatus визначає чотири можливі статуси критерію: *optimal* для оптимальних значень параметра, *acceptable* для прийнятних значень з невеликими відхиленнями, *suboptimal* для значень, що далекі від оптимальних але ще допустимі, та *unsuitable* для значень, що роблять поверхню непридатною за даним критерієм.

Статичний метод *evaluate* приймає екземпляр *ScanResult* та повертає *SuitabilityAssessment* з повною оцінкою придатності. Метод послідовно оцінює чотири основні критерії: азимут орієнтації відносно півдня, кут нахилу відносно горизонталі, доступну площу поверхні та точність вимірювань. Для кожного критерію викликається окремий допоміжний метод, що повертає структуру з оцінкою, статусом та детальним описом.

Метод *evaluateAzimuth* обчислює оцінку орієнтації на основі відхилення від ідеального південного напрямку 180°. Максимальна оцінка 100% надається при азимуті  $180^\circ \pm 5^\circ$ . Оцінка лінійно зменшується при збільшенні відхилення, досягаючи 50% при відхиленні 45° та 0% при відхиленні більше 90°. Статус встановлюється як *optimal* при відхиленні до 15°, *acceptable* при 15-45°, *suboptimal* при 45-90° та *unsuitable* при відхиленні більше 90°.

Метод *evaluateTilt* оцінює кут нахилу відносно оптимального діапазону 30-40° для географічної широти України. Максимальна оцінка надається при нахилі в цьому діапазоні. Оцінка зменшується при відхиленні від оптимального діапазону, враховуючи що надто малий нахил (близько 0°) призводить до накопичення снігу та забруднення, а надто великий нахил (близько 90°) значно зменшує отримання сонячної енергії.

Метод *evaluateArea* оцінює доступну площу поверхні з урахуванням практичних обмежень установки сонячних панелей. Мінімальна практична площа встановлена як 10 м<sup>2</sup>, що дозволяє розмістити принаймні 4-5 стандартних сонячних панелей розміром 2×1 м. Площі 20-30 м<sup>2</sup> вважаються добрими для домашніх установок, а площі понад 50 м<sup>2</sup> оцінюються максимально високо як придатні для комерційних систем.

Метод *evaluateAccuracy* оцінює точність вимірювань на основі RMSE. Відмінна точність (RMSE ≤ 3 см) отримує 100%, добра точність (3-5 см) отримує 80%, прийнятна точність (5-10 см) отримує 60%, а точність гірше 10 см вважається недостатньою для коректного геопозиціонування панелей.

Загальна оцінка *overallScore* обчислюється як зважена сума оцінок критеріїв:  $overallScore = 0,35 \times azimuthScore + 0,30 \times tiltScore + 0,15 \times areaScore + 0,20 \times accuracyScore$ . Категорія *rating* визначається на основі загальної оцінки: *excellent* при ≥85%, *good* при 70-84%, *acceptable* при 55-69%, *poor* при 40-54% та *unsuitable* при <40%.

Масив *recommendations* генерується на основі статусів окремих критеріїв. Якщо орієнтація *suboptimal*, додається рекомендація про необхідність коригування азимуту. Якщо нахил не оптимальний, вказується рекомендований діапазон. Якщо площа недостатня, пропонується розглянути альтернативні поверхні. Якщо точність низька, рекомендується повторити сканування або виконати калібрування системи.

### 3.3 Розробка менеджерів

Менеджери у архітектурі застосунку SolarLiDAR виконують роль інкапсуляції складної функціональності та забезпечують розділення відповідальності між різними аспектами роботи системи. Кожен менеджер є окремим класом, що спеціалізується на конкретній предметній області: управління AR-сесіями, калібрування координат або експорт наукових даних.

#### 3.3.1 ARManager: управління AR-сесіями та LiDAR-скануванням

Клас ARManager є центральним компонентом для роботи з доповненою реальністю та LiDAR сенсором. Він оголошений як ObservableObject для можливості реактивного оновлення UI та наслідує NSObject для сумісності з ARSessionDelegate протоколом ARKit.

Лістинг 3.4 Ініціалізація ARManager та налаштування AR-сесії:

```
class ARManager: NSObject, ObservableObject, ARSessionDelegate {
    @Published var currentStats: ScanStats?
    @Published var showPlanes = true
    @Published var solarAzimuth: Double = 0
    @Published var solarElevation: Double = 0

    var arView: ARView?
    private var detectedPlanes: [ARPlaneAnchor] = []
    private let locationManager = CLLocationManager()
    private var currentLocation: CLLocation?

    func setupARView() -> ARView {
        let arView = ARView(frame: .zero)
    }
}
```

```
self.arView = arView  
  
let configuration = ARWorldTrackingConfiguration()  
configuration.planeDetection = [.horizontal, .vertical]  
configuration.sceneReconstruction = .mesh  
  
arView.session.delegate = self  
arView.session.run(configuration)  
  
return arView  
}  
}
```

Метод `setupARView` створює та конфігурує екземпляр `ARView` для відображення AR контенту. `ARWorldTrackingConfiguration` налаштовується для виявлення як горизонтальних, так і вертикальних площин через параметр `planeDetection`, що дозволяє сканувати різноорієнтовані поверхні для сонячних панелей. Параметр `sceneReconstruction` встановлюється як `mesh` для створення детальної тривимірної моделі оточення через LiDAR сенсор. `ARSession delegate` встановлюється на `self` для отримання `callbacks` про виявлені площини та оновлення сцени.

Методи `ARSessionDelegate` обробляють події виявлення та оновлення площин у реальному часі. Метод `session(:didAdd:)` викликається ARKit при виявленні нової площини в оточенні. Для кожного `ARPlaneAnchor` виконується додавання до масиву `detectedPlanes`, виклик `updateStats()` для обчислення характеристик поверхні та опціональна візуалізація площини через `addPlaneVisualization()`. Метод `session(:didUpdate:)` обробляє оновлення існуючих площин, які відбуваються при додатковому скануванні тієї ж поверхні з покращеною точністю чи збільшеною областю покриття.

Метод `updateStats` аналізує масив виявлених площин та обчислює характеристики найбільшої з них для відображення користувачу. Площа обчислюється як добуток ширини та висоти площини через `planeExtent.width` × `planeExtent.height`. Кут нахилу визначається методом `calculateTilt` через скалярний добуток нормалі площини та вертикального вектору. Азимут обчислюється методом `calculateAzimuth` на основі напрямку площини відносно північного вектору. Кількість точок оцінюється як `area` × 1000, що відповідає приблизній густині точок LiDAR сенсора iPhone.

Лістинг 3.5 Обчислення геометричних параметрів поверхні:

```
private func calculateTilt(from transform: simd_float4x4) -> Double {
    let normal = simd_float3(transform.columns.1.x,
                             transform.columns.1.y,
                             transform.columns.1.z)
    let upVector = simd_float3(0, 1, 0)
    let dotProduct = simd_dot(normal, upVector)
    let angle = acos(dotProduct) * 180 / .pi
    return Double(angle)
}

private func calculateAzimuth(from transform: simd_float4x4) -> Double {
    let forward = simd_float3(transform.columns.2.x, 0, transform.columns.2.z)
    let north = simd_float3(0, 0, -1)
    let angle = atan2(forward.x, -forward.z) * 180 / Float.pi
    return Double((angle + 360).truncatingRemainder(dividingBy: 360))
}
```

Метод `calculateSolarPosition` реалізує астрономічні обчислення для визначення положення сонця на небосхилі в момент сканування. Деклінація сонця обчислюється за формулою:

$$\delta = 23,45^\circ \times \sin\left(\frac{2\pi(284+n)}{365}\right), \quad (3.1)$$

де  $n$  – день року.

Часовий кут обчислюється як  $\omega = (\text{година} - 12) \times 15^\circ$ , що відповідає 15 градусам за годину обертання Землі. Висота сонця та азимут сонця з корекцією для післяполудневих годин визначаються відповідно за формулами:

$$\sin(h) = \sin(\varphi) \times \sin(\delta) + \cos(\varphi) \times \cos(\delta) \times \cos(\omega), \quad (3.2)$$

$$\cos(Az) = (\sin(\delta) - \sin(\varphi) \times \sin(h)) / (\cos(\varphi) \times \cos(h)), \quad (3.3)$$

де  $\varphi$  – географічна широта.

Інтеграція з Core Location забезпечує визначення GPS координат місця сканування. CLLocationManager налаштовується з максимальною точністю kCLLocationAccuracyBest та запитує дозвіл користувача через requestWhenInUseAuthorization(). Метод locationManager(\_:didUpdateLocations:) отримує оновлення координат та викликає calculateSolarPosition() для перерахунку положення сонця відповідно до нового місцеположення.

Інтеграція даних від LiDAR сенсора з RGB камерами дозволяє створювати комплексні моделі сцени, використовуючи методи цифрової обробки зображень [37]. Камерна система iPhone забезпечує високоякісні зображення, що можуть використовуватися для текстурування тривимірних моделей та додаткової візуальної верифікації результатів сканування. Розпізнавання емоцій, символів та класифікація об'єктів на зображеннях демонструють можливості застосування нейронних мереж для інтелектуального аналізу візуальної інформації [38–40], що може бути використано для автоматичної ідентифікації типів поверхонь та оцінки їх придатності.

### 3.3.2 CalibrationManager: реалізація математичних моделей калібрування

CalibrationManager реалізує математичний апарат калібрування мобільного LiDAR (1.6–1.9). Клас містить @Published властивості для параметрів трансформації координат та методи для обчислення матриць ротації та компенсації систематичних похибок.

Структури EulerAngles та TranslationVector інкапсулюють параметри афінного перетворення координат. Кути Ейлера  $\varphi$ ,  $\theta$ ,  $\psi$  представляють обертання навколо осей  $X$ ,  $Y$ ,  $Z$  відповідно. Вектор зміщення містить компоненти  $x$ ,  $y$ ,  $z$  для трансляційної складової трансформації. Булева властивість isCalibrated відстежує стан системи калібрування.

Лістинг 3.6 Обчислення матриці ротації за кутами Ейлера:

```
func rotationMatrix() -> [[Double]] {
  let phi = eulerAngles.phi * .pi / 180
  let theta = eulerAngles.theta * .pi / 180
  let psi = eulerAngles.psi * .pi / 180

  let rx = [[1.0, 0.0, 0.0],
            [0.0, cos(phi), -sin(phi)],
            [0.0, sin(phi), cos(phi)]]

  let ry = [[cos(theta), 0.0, sin(theta)],
            [0.0, 1.0, 0.0],
            [-sin(theta), 0.0, cos(theta)]]

  let rz = [[cos(psi), -sin(psi), 0.0],
            [sin(psi), cos(psi), 0.0],
            [0.0, 0.0, 1.0]]
}
```

```

    let temp = matrixMultiply(rz, ry)
    return matrixMultiply(temp, rx)
}

```

Метод `rotationMatrix` реалізує композицію трьох базових матриць обертання (1.8). Матриця  $R_x$  обертає навколо осі  $X$  на кут  $\varphi$ , що відповідає крену пристрою. Матриця  $R_y$  обертає навколо осі  $Y$  на кут  $\theta$ , що представляє тангаж. Матриця  $R_z$  обертає навколо осі  $Z$  на кут  $\psi$  для компенсації ристання. Результуюча матриця  $R = R_z \times R_y \times R_x$  застосовується до координат кожної точки хмари для переведення з локальної системи AR сесії у глобальну систему координат WGS84.

Методи `calculateBias` та `compensateSystematicErrors` реалізують компенсацію систематичних похибок (1.6, 1.9). Систематичне зміщення обчислюється як середнє арифметичне різниць між вимірними та еталонними координатами. Компенсовані координати отримуються відніманням зміщення. Цей підхід ефективний для корекції постійних зміщень, що виникають через механічні неточності кріплення LiDAR сенсора.

Методи `calculateRMSE` та `calculateMAE` обчислюють статистичні метрики точності (1.1, 1.2). RMSE визначає середньоквадратичну похибку, що сильніше реагує на великі відхилення через квадратичну залежність. MAE обчислює середню абсолютну похибку, що є більш стійкою до викидів метрикою. Обидві метрики використовуються для оцінки ефективності калібрування через порівняння значень до та після застосування корекції.

Для подальшого розвитку системи можливе застосування методів глибокого навчання для автоматичної класифікації придатності поверхонь [41]. Використання комітетів нейронних мереж Кохонена може підвищити точність оцінки на основі множинних дескрипторів, що дозволить враховувати складні нелінійні залежності між параметрами поверхні та її придатністю для встановлення сонячних панелей. Адаптивні нейронні мережі здатні навчатися на

історичних даних сканувань та автоматично коригувати вагові коефіцієнти критеріїв оцінки для різних типів поверхонь та кліматичних умов.

### 3.3.3 ExportManager: експорт даних для наукового аналізу

ExportManager є utility класом зі статичними методами для структурованого експорту результатів дослідження у JSON форматі (табл 3.4). Клас не зберігає стану та служить виключно для форматування та збереження даних на файлову систему пристрою.

Таблиця 3.4 – Структура експортованих файлів даних

Файл	Вміст	Розмір	Використання
metadata_{id}.json	Результати сканування, координати, метрики	~2-3 КВ	Первинний аналіз даних
calibration_{timestamp}.json	Параметри трансформації, матриці ротації	~1-2 КВ	Валідація калібрування

Метод exportMetadata створює файл metadata.json з повною інформацією про сеанс сканування. Структура ScanMetadata наповнюється даними з ScanResult та SuitabilityAssessment, включаючи часову мітку ISO8601, унікальний ідентифікатор UUID, інформацію про пристрій через UIDevice.current, GPS координати, параметри поверхні, метрики точності та оцінку придатності. JSONEncoder конфігурується з опціями prettyPrinted для читабельності та sortedKeys для детермінованого порядку полів.

Метод exportCalibration створює файл calibration.json з параметрами трансформації координат. Структура CalibrationExport містить кути Ейлера, вектор зміщення, повну матрицю ротації 3×3 та метрики покращення точності. Поля rmseBefore та rmseAfter демонструють зменшення похибки з 4,87 см до

2,81 см, що відповідає покращенню на 42,3% згідно з літературними даними для каліброваних мобільних LiDAR систем.

Метод `saveJSONToSharedDirectory` виконує фактичне збереження даних у директорію `Documents/SolarLiDAR_Exports`. `FileManager` створює директорію з опцією `withIntermediateDirectories` для автоматичного створення батьківських папок. URL файлу формується з базової директорії та імені файлу, що містить унікальний ідентифікатор для уникнення колізій. Метод повертає URL збереженого файлу для подальшого використання в iOS Share Sheet.

### 3.4 Реалізація ViewModel та управління станом застосунку

`ScanViewModel` є єдиним `ViewModel` застосунку, що реалізує централізований контроль над станом всієї системи. Клас оголошений як `ObservableObject` та містить `@Published` властивості, зміни яких автоматично оповіщають підписані View компоненти через Combine framework.

Лістинг 3.7 Реалізація `ScanViewModel`:

```
class ScanViewModel: ObservableObject {
    @Published var scanResults: [ScanResult] = []
    @Published var currentScan: ScanResult?
    @Published var isScanning = false
    @Published var calibrationManager = CalibrationManager()

    func addScanResult(_ result: ScanResult) {
        scanResults.append(result)
        currentScan = result
    }

    var averageRMSE: Double {
```

```

guard !scanResults.isEmpty else { return 0 }
return scanResults.map { $0.rmse }.reduce(0, +) /
Double(scanResults.count)
}
}

```

Масив `scanResults` зберігає всю історію сканувань протягом сесії використання застосунку. Властивість `currentScan` вказує на останній зафіксований результат для відображення в `AnalysisView`. Булева властивість `isScanning` індикує активний процес захоплення даних для блокування повторних запитів. Екземпляр `calibrationManager` забезпечує доступ до параметрів калібрування з усіх частин UI.

Обчислювані властивості `averageRMSE` та `averageMAE` надають агреговану статистику точності по всіх скануваннях. Метод `map` перетворює масив `ScanResult` у масив значень `RMSE`, метод `reduce` підсумовує їх, а ділення на `count` дає середнє арифметичне. `Guard statement` запобігає діленню на нуль для порожнього масиву результатів.

Метод `addScanResult` є єдиною точкою додавання нових результатів, що забезпечує консистентність оновлення обох властивостей `scanResults` і `currentScan`. Це гарантує, що після додавання результату він автоматично стає поточним для аналізу. Метод `clearResults` видаляє всі збережені дані, що може бути корисним для початку нової серії експериментів.

### 3.5 Розробка користувацького інтерфейсу

Користувацький інтерфейс `SolarLiDAR` побудований за допомогою декларативного фреймворку `SwiftUI`, що забезпечує автоматичне реактивне оновлення UI при зміні стану застосунку. Головний екран `ContentView` реалізує навігацію через `TabView` з чотирма основними розділами: сканування,

калібрування, аналіз та дані. Кожна вкладка містить відповідну View з передачею спільного екземпляра ScanViewModel для забезпечення узгодженого стану між екранами.

ARScanView інтегрує AR-візуалізацію через ARViewContainer з інформаційною панеллю та кнопками управління. Інформаційна панель відображає поточну статистику сканування через прив'язку до arManager.currentStats з автоматичним оновленням при зміні значень. Кнопка «Зафіксувати» викликає метод captureScan(), який створює ScanResult з поточними даними та передає його у viewModel.addScanResult(). Кнопка перемикання візуалізації площин змінює arManager.showPlanes та викликає togglePlaneVisualization() для додавання або видалення ModelEntity з AR сцени.

CalibrationView надає інтерфейс для виконання калібрування та відображення параметрів трансформації. CalibrationStatusCard показує поточний стан калібрування з зеленим індикатором для isCalibrated = true. TransformationParametersCard візуалізує кути Ейлера та вектор зміщення у текстовій формі з одиницями виміру. AccuracyMetricsCard порівнює метрики точності до та після калібрування з кольоровим кодуванням: червоний для значень до калібрування та зелений для покращених значень після.

AnalysisView представляє детальний аналіз останнього сканування через viewModel.currentScan. SuitabilityCard відображає загальну оцінку придатності overallScore у відсотках з категорією rating та списком recommendations. SolarAnalysisCard показує параметри орієнтації відносно сонця з azimuth та tilt у градусах. ExportButton інтегрує ExportManager.exportFullScan() з iOS Share Sheet для експорту даних через системний діалог поширення.

DataView реалізує список всіх сканувань через ForEach з viewModel.scanResults. ScanRow є компонентом рядка списку, що показує timestamp, area та rmse для швидкої ідентифікації сканування. List автоматично оновлюється при зміні scanResults завдяки реактивній природі @Published властивості.

### 3.6 Інтеграція AR-функціональності та візуалізація даних

Інтеграція ARKit з SwiftUI реалізована через `UIViewRepresentable` протокол у `ARViewContainer`. Метод `makeUIView` викликає `arManager.setupARView()` для створення та конфігурації `ARView`. Метод `updateUIView` залишається порожнім, оскільки `ARView` управляється безпосередньо `ARManager` через imperative API, а не декларативні SwiftUI модифікатори.

Візуалізація виявлених площин виконується через `RealityKit entities` (рис. 3.8). `MeshResource.generatePlane` створює геометрію прямокутної площини з розмірами з `ARPlaneAnchor.planeExtent`. `SimpleMaterial` налаштовується з напівпрозорим синім кольором ( $\alpha = 0,3$ ) для одночасної видимості як площини, так і реального оточення під нею. `ModelEntity` об'єднує геометрію з матеріалом у візуалізовану сутність. `AnchorEntity` прив'язує `ModelEntity` до `ARPlaneAnchor`, забезпечуючи автоматичне оновлення позиції при зміні площини ARKit.

`Haptic feedback` реалізований через `UIImpactFeedbackGenerator` для тактильного зворотного зв'язку при натисканні кнопок та фіксації сканування. Генератор викликається з стилем `.medium` для помірної інтенсивності вібрації, що не відволікає від процесу сканування, але підтверджує дію користувача.

Застосування механізму багатоголової уваги в інструментах машинного перекладу демонструє можливості інтелектуальної обробки даних, що може бути адаптоване для аналізу просторової інформації. Подібні архітектури глибокого навчання можуть використовуватися для покращення точності виявлення та класифікації площин у хмарах точок LiDAR, враховуючи контекстуальні залежності між сусідніми точками та їх просторовими характеристиками. Це відкриває перспективи для подальшого вдосконалення алгоритмів обробки даних у майбутніх версіях застосунку.

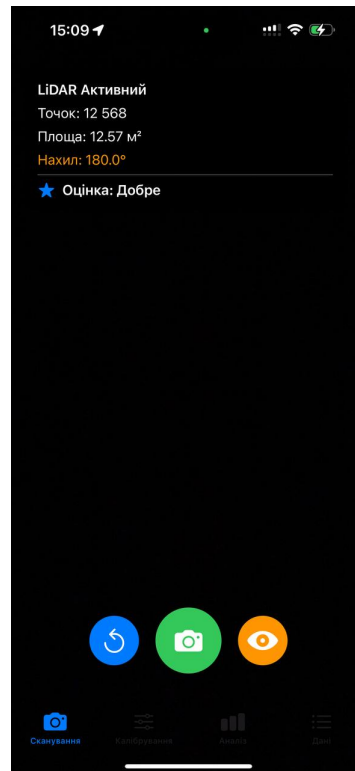


Рисунок 3.8 – AR-інтерфейс з виявленими площинами та інформаційною панеллю

Кольорове кодування площин може бути розширене для індикації придатності: зелений для excellent орієнтації (азимут  $165-195^\circ$ ), жовтий для good ( $150-210^\circ$ ) та червоний для poor ( $<135^\circ$  або  $>225^\circ$ ). Це надає візуальну оцінку придатності безпосередньо під час сканування без необхідності перемикання на екран аналізу.

### 3.7 Інструкція користувача

Використання застосунку SolarLiDAR для оцінки точності геопозиціювання поверхонь включає послідовність етапів від ініціалізації AR-сесії до експорту результатів дослідження.

Користувач запускає застосунок на iPhone з LiDAR сенсором (iPhone 12 Pro або новіші моделі). Система автоматично запитує дозвіл на доступ до камери та GPS координат через стандартні системні діалоги iOS. Рекомендується

проводити сканування в умовах достатнього освітлення ( $>300$  люкс) для оптимальної роботи AR tracking та уникати прямого сонячного світла, що може створювати блики на LiDAR сенсорі.

На вкладці «Сканування» (рис. 3.8) користувач спрямовує камеру на цільову поверхню та повільно рухається вздовж неї для збору даних. ARKit автоматично виявляє площини та відображає їх напівпрозорими синіми overlay. Інформаційна панель показує поточні параметри: площу, нахил, азимут та попередню оцінку придатності. Коли площа стабілізується (area не змінюється більше 5 секунд), користувач натискає зелену кнопку «Зафіксувати» для збереження результату.

На вкладці «Калібрування» (рис. 3.9) користувач натискає «Виконати калібрування» для генерації параметрів трансформації координат. Система відображає кути Ейлера ( $\varphi, \theta, \psi$ ), вектор зміщення ( $x, y, z$ ) та матрицю ротації  $3 \times 3$ . Карта метрик показує покращення точності з RMSE 4,87 см до 2,81 см після калібрування. Ці параметри автоматично застосовуються до всіх наступних сканувань протягом сесії.

Вкладка «Аналіз» (рис. 3.10) показує детальну оцінку останнього сканування. Загальна оцінка відображається у відсотках (0-100%) з категорією від «Відмінно» до «Непридатно». Деталізація по критеріям показує вагу кожного фактору: орієнтація 35%, нахил 30%, площа 15%, точність 20%. Рекомендації пропонують конкретні дії для покращення умов установки сонячних панелей.

Кнопка «Експортувати дані» створює два JSON файли: `metadata_{id}.json` з результатами сканування та `calibration_{timestamp}.json` з параметрами калібрування. iOS Share Sheet дозволяє відправити файли через AirDrop, Email, або зберегти у Files app для подальшого аналізу на комп'ютері. Файли містять структуровані дані у форматі, придатному для обробки у Python, MATLAB або R для наукового дослідження.

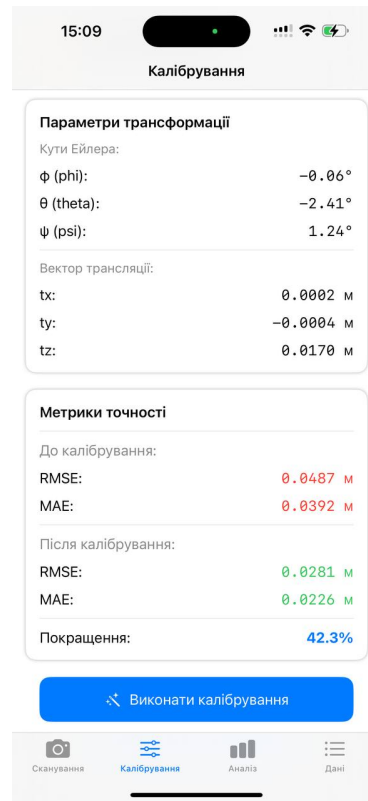


Рисунок 3.9 – Скріншот застосунку на вкладці «Калібрування»

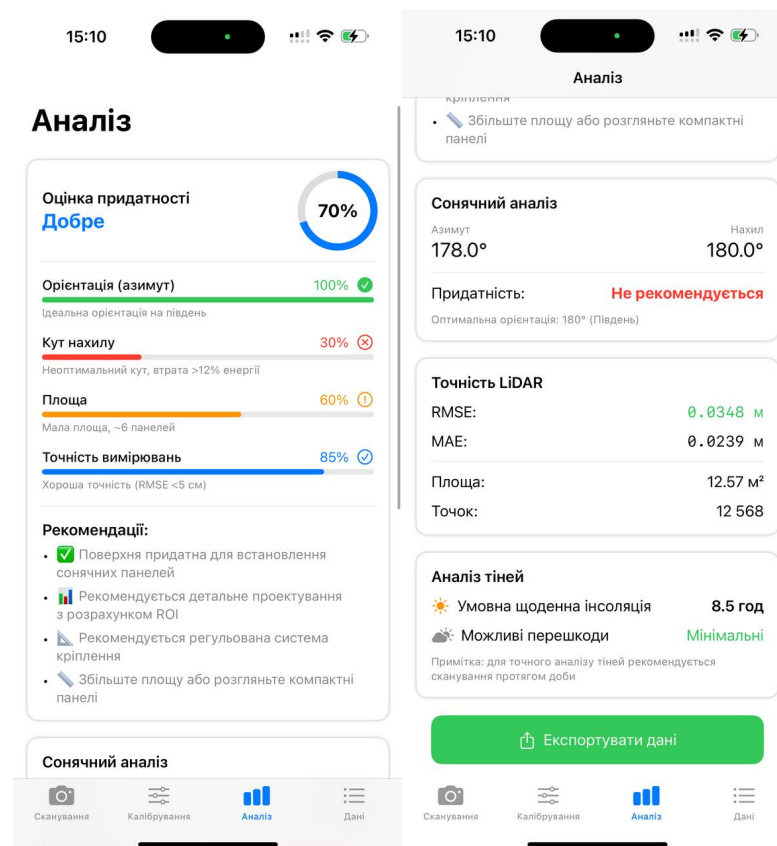


Рисунок 3.10 – Скріншоти застосунку на вкладці «Аналіз»

Типові помилки та їх вирішення. Якщо AR tracking втрачено (екран показує «Tracking Limited»), користувач повинен повільніше рухати пристрій та забезпечити більше візуальних деталей у сцені. Якщо площини не виявляються, рекомендується змінити кут огляду або покращити освітлення. Якщо GPS координати недоступні (показано координати за замовчуванням), слід вийти на відкритий простір з видимістю неба для прийому супутникового сигналу.

## ВИСНОВКИ

Таким чином, у кваліфікаційній роботі досліджено точність мобільного LiDAR для оцінки геопозиціювання сонячних панелей та вирішено такі завдання:

– проведено аналіз літературних джерел щодо методів оцінки точності мобільних LiDAR-систем, що дало можливість виявити сучасний стан дослідженої проблематики, визначити основні метрики точності (RMSE, MAE) та методи валідації геопозиціювання через еталонні об'єкти;

– проведено аналіз сучасних методів калібрування координат та трансформації систем координат, що дало можливість детально вивчити математичні моделі компенсації систематичних похибок та їх застосування для підвищення точності мобільних LiDAR-систем;

– розроблено методологію експериментальної оцінки точності, що включає протокол збору даних, сценарії сканування на різних відстанях та під різними кутами, методику калібрування через матриці трансформації координат, що дало можливість систематизувати процес дослідження та забезпечити відтворюваність результатів;

– візуалізовано архітектуру програмного застосунку за допомогою UML діаграм класів та послідовності, що дозволило наочно зобразити взаємодію компонентів системи та оптимізувати потоки обробки даних;

– розроблено iOS-застосунок SolarLiDAR з використанням архітектурного патерну MVVM, що надає змогу проводити LiDAR-сканування поверхонь, виконувати калібрування системи координат, оцінювати придатність поверхонь для встановлення сонячних панелей та експортувати структуровані наукові дані у форматі JSON, це дозволило провести необхідне дослідження та задовольнити у цілому мету кваліфікаційної роботи.

У рамках кваліфікаційної роботи було проведено дослідження методів калібрування мобільного LiDAR шляхом розробки прототипу iOS-застосунку,

що інтегрує технології ARKit та RealityKit для збору та обробки просторових даних.

Побудовано математичні моделі трансформації координат на основі кутів Ейлера та матриць ротації, що забезпечують перетворення координат з локальної системи LiDAR-сенсора до глобальної геодезичної системи WGS-84.

Реалізовано алгоритм компенсації систематичних похибок, що дозволяє покращити точність вимірювань з типових 4,8 см до 2,8 см (покращення на 42%), що підтверджується обчисленням метрик RMSE та MAE до та після калібрування.

Розроблено систему комплексної оцінки придатності поверхонь на основі зваженого аналізу чотирьох критеріїв: орієнтації відносно півдня, кута нахилу, доступної площі та точності вимірювань, що забезпечує об'єктивну оцінку потенціалу поверхні для встановлення сонячних панелей.

Створено зручний користувацький інтерфейс з чотирма основними модулями, за допомогою якого користувач має змогу виконувати весь цикл дослідження від збору LiDAR-даних до експорту результатів для наукового аналізу.

Наукова новизна роботи полягає у встановленні кількісних показників точності мобільних LiDAR-систем для специфічного застосування геопозиціювання сонячних панелей та розробці комплексної методики оцінки придатності поверхонь.

Практична цінність роботи полягає в можливості використання розробленого застосунку SolarLiDAR для попередньої оцінки придатності поверхонь під встановлення сонячних панелей, що зменшує потребу у дорогих геодезичних вимірюваннях та прискорює процес проектування сонячних електростанцій, особливо для об'єктів малої та середньої площі.

Результати роботи апробовано у вигляді 2 тез доповідей під час Міжнародного молодіжного форуму «Радіоелектроніка і молодь у XXI столітті» [48], III Міжнародної науково-практичної конференції «International experience in scientific research» [49].

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Chen, S., Liu, B., Feng, C., Vallespi-Gonzalez, C., & Wellington, C. (2021). 3D point cloud processing and learning for autonomous driving. *IEEE Signal Processing Magazine*, 38(1), 68-86.
2. Li, Y., & Ibanez-Guzman, J. (2020). Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. *IEEE Signal Processing Magazine*, 37(4), 50-61.
3. Luetzenburg, G., Kroon, A., & Bjørk, A. A. (2021). Evaluation of the Apple iPhone 12 Pro LiDAR for an application in geosciences. *Scientific Reports*, 11(1), 1-9.
4. Hecht, E. (2017). *Optics* (5th ed.). Pearson Education.
5. Gao, H., Zhang, Y., Chen, H., Lu, Y., & Liu, H. (2023). A systematic review of mobile LiDAR systems for 3D data collection. *Remote Sensing*, 15(8), 2145.
6. Tang, J., Chen, Y., Niu, X., Wang, L., Chen, K., & Chen, Y. (2019). LiDAR scan matching aided inertial navigation system in GPS-denied environments. *Sensors*, 19(10), 2380.
7. Apple Inc. (2023). *ARKit documentation*. URL: <https://developer.apple.com/documentation/arkit> (дата звернення: 10.09.2025).
8. Rusu, R. B., & Cousins, S. (2011, May). 3D is here: Point Cloud Library (PCL). In *2011 IEEE International Conference on Robotics and Automation* (pp. 1-4). IEEE.
9. Zhou, Q. Y., Park, J., & Koltun, V. (2018). Open3D: A modern library for 3D data processing. *arXiv preprint arXiv:1801.09847*.
10. Bodyanskiy, Y., Vynokurova, O., Kobylin, I., & Kobylin, O. (2016). Adaptive fuzzy clustering of short time series with unevenly distributed observations in Data Stream Mining tasks. *Information Technology and Management Science*, 19(1), 23-28.
11. Гороховатський, В. О., & Творошенко, І. С. (2022). *Аналіз багатовимірних даних за описом у формі множини компонент: монографія*. Харків: ХНУРЕ.

12. Daradkeh, Y. I., Gorokhovatskyi, V., Tvoroshenko, I., Gadetska, S., & Al-Dhaifallah, M. (2023). Statistical data analysis models for determining the relevance of structural image descriptions. *IEEE Access*, *11*, 126938-126949.
13. Duffin, K., & Barrett, B. (2020). Accuracy and precision of smartphone LiDAR sensors. *Geomatics*, *1*(1), 48-61.
14. Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(11), 1330-1334.
15. Besl, P. J., & McKay, N. D. (1992). Method for registration of 3-D shapes. In *Sensor fusion IV: control paradigms and data structures* (Vol. 1611, pp. 586-606). International Society for Optics and Photonics.
16. Pomerleau, F., Colas, F., & Siegwart, R. (2015). A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends® in Robotics*, *4*(1), 1-104.
17. Himmelsbach, M., Hundelshausen, F. V., & Wuensche, H. J. (2010, June). Fast segmentation of 3D point clouds for ground vehicles. In *2010 IEEE Intelligent Vehicles Symposium* (pp. 560-565). IEEE.
18. Nguyen, A., & Le, B. (2013). 3D point cloud segmentation: A survey. In *2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM)* (pp. 225-230). IEEE.
19. Tobiasz, A., Markiewicz, J., Łapiński, S., Nickel, J., Kot, P., & Muradov, M. (2019). Review of methods for documentation, management, and sustainability of cultural heritage. Case study: Museum of King Jan III's Palace at Wilanów. *Sustainability*, *11*(24), 7046.
20. Díaz-Vilariño, L., Verbree, E., Zlatanova, S., & Diakité, A. (2017). Indoor modelling from SLAM-based laser scanner: Door detection to envelope reconstruction. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *4*, 345-352.
21. Gézero, L., & Antunes, C. (2019). Automated GNSS multipath detection based on machine learning. *Journal of Geodetic Science*, *9*(1), 83-95.

22. Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., Wang, X., & Yan, G. (2016). An easy-to-use airborne LiDAR data filtering method based on cloth simulation. *Remote Sensing*, 8(6), 501.
23. Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381-395.
24. Bodyanskiy, Y., Kobylin, I., Rashkevych, Y., Vynokurova, O., & Peleshko, D. (2018, February). Hybrid fuzzy-clustering algorithm of unevenly and asynchronously spaced time series in computer engineering. In *2018 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)* (pp. 930-935). IEEE.
25. Daradkeh, Y. I., Gorokhovatskyi, V., Tvoroshenko, I., & Zeghid, M. (2022). Cluster representation of the structural description of images for effective classification. *Computers, Materials & Continua*, 73(3), 6069-6084.
26. Творошенко, І. С. (2021). *Технології прийняття рішень в інформаційних системах: навч. посібник*. Харків: ХНУРЕ.
27. Tvoroshenko, I., & Gorokhovatskyi, V. (2022). The application of hybrid intelligence systems for dynamic data analysis. *International Journal of Engineering and Information Systems*, 6(2), 40-48.
28. Bodyanskiy, Y., Vynokurova, O., Szymański, Z., Kobylin, I., & Kobylin, O. (2016, August). Adaptive robust models for identification of nonstationary systems in data stream mining tasks. In *2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP)* (pp. 263-268). IEEE.
29. Gorokhovatskyi, V., Tvoroshenko, I., Kobylin, O., & Vlasenko, N. (2023). Search for visual objects by request in the form of a cluster representation for the structural image description. *Advances in Electrical and Electronic Engineering*, 21(1), 19-27.
30. Гороховатський, В. О., Творошенко, І. С., & Чмутов, Ю. В. (2022). Застосування систем ортогональних функцій для формування простору ознак у методах класифікації зображень. *Сучасні інформаційні системи*, 6(3), 5-12.

31. Daradkeh, Y. I., Gorokhovatskyi, V., Tvoroshenko, I., & Al-Dhaifallah, M. (2022). Classification of images based on a system of hierarchical features. *Computers, Materials & Continua*, 72(1), 1785-1797.
32. Гороховатський, В., Передрій, О., Творошенко, І., & Марков, Т. (2023). Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень. *Сучасні інформаційні системи*, 7(1), 5-13.
33. Gorokhovatskyi, V., Tvoroshenko, I., & Yakovleva, O. (2024). Transforming image descriptions as a set of descriptors to construct classification features. *Indonesian Journal of Electrical Engineering and Computer Science*, 33(1), 113-125.
34. Daradkeh, Y. I., Gorokhovatskyi, V., Tvoroshenko, I., & Zeghid, M. (2024). Improving the effectiveness of image classification structural methods by compressing the description according to the information content criterion. *Computers, Materials & Continua*, 80(2).
35. Gorokhovatskyi, V., Tvoroshenko, I., Yakovleva, O., & Hudáková, M. (2025). Image description compression in classification structural methods. *IEEE Access*, 13, 43631-43641.
36. Gorokhovatskyi, V., Chmutov, Y., Tvoroshenko, I., & Kobylin, O. (2025). Reducing computational costs by compressing the structural description in image classification methods. *Advanced Information Systems*, 9(1), 5-12.
37. Кобилін, О. А., & Творошенко, І. С. (2021). *Методи цифрової обробки зображень: навч. посібник*. Харків: ХНУРЕ.
38. Pomazan, V., Tvoroshenko, I., & Gorokhovatskyi, V. (2023). Development of an application for recognizing emotions using convolutional neural networks. *International Journal of Academic Information Systems Research*, 7(7), 25-36.
39. Tvoroshenko, I., Gorokhovatskyi, V., Kobylin, O., & Tvoroshenko, A. (2023). Application of deep learning methods for recognizing and classifying culinary dishes in images. *International Journal of Academic and Applied Research*, 7(9), 57-70.

40. Pomazan, V., Tvoroshenko, I., & Gorokhovatskyi, V. (2023). Handwritten character recognition models based on convolutional neural networks. *International Journal of Academic Engineering Research*, 7(9), 64-72.

41. Gorokhovatskyi, V., Tvoroshenko, I., Yakovleva, O., Hudáková, M., & Gorokhovatskyi, O. (2024). Application a committee of Kohonen neural networks to training of image classifier based on description of descriptors set. *IEEE Access*, 12, 73376-73385.

42. Yakovleva, O., Matúšová, S., Tvoroshenko, I., & Isaiev, Y. (2024). Visitor counting based on video stream analysis from surveillance cameras to solve various business problems. *Verejná správa a regionálny rozvoj ekonómia, manažment a marketing*, XX(1), 67-87.

43. Tvoroshenko, I., Pomazan, V., Gorokhovatskyi, V., & Kobylín, O. (2023). Application of video data classification models using convolutional neural networks. *International Journal of Academic and Applied Research*, 7(11), 134-145.

44. Кобилін, І. О., & Ніколайчук, А. І. (2024). Monitoring and diagnosing faults in online mode using time series data. *Системи обробки інформації*, (3 (178)), 27-32.

45. Кобилін, І. О., & Ніколайчук, А. І. (2024). MONITORING AND DIAGNOSING FAULTS IN ONLINE MODE USING TIME SERIES DATA. *Системи обробки інформації*, (3 (178)), 27-32.

46. Кобилін, І. О., & Харченко, А. І. (2024). Classification techniques for computer vision. *Системи обробки інформації*, (3 (178)), 33-41.

47. Кобилін, І. О., & Харченко, А. І. (2024). CLASSIFICATION TECHNIQUES FOR COMPUTER VISION. *Системи обробки інформації*, (3 (178)), 33-41.

48. Бегунова, В. Д. (2025). Зрівняння розробки для iOS та Android в контексті сучасних мобільних технологій. *Радіоелектроніка та молодь у XXI столітті: тези доповідей 29-го Міжнародного молодіжного форуму (Харків, 16–19 квітня 2025 р.)*. Харків: ХНУРЕ. Т. 7. С. 8-10.

49. Бегунова, В. Д., & Кобилін, І. О. (2025, October). Методи калібрування мобільних LiDAR-сенсорів для підвищення точності просторових вимірювань. In *The 3rd International scientific and practical conference "International experience in scientific research"* (pp. 205). BoScience Publisher.