

Київський національний університет імені Тараса Шевченка
Факультет інформаційних технологій
Кафедра програмних систем і технологій

На правах рукопису

УДК 004.42

ВИПУСКНА КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

Тема Автономна ANDROID-система визначення координат та поступового
нарощування їх точності в умовах відсутності радіосигналів
Спеціальність 121 “Інженерія програмного забезпечення”

ПОЯСНЮВАЛЬНА ЗАПИСКА

Студент

_____ Антон Табунов
(підпис) (розшифровка підпису) (дата)

Науковий керівник

_____ д.т.н., проф. Віктор Шевченко
(посада) (підпис) (розшифровка підпису) (дата)

Допускається до захисту
з питань нормоконтролю
Завідувач кафедри

(підпис) (розшифровка підпису) (дата)

2022

Рішенням Екзаменаційної комісії
випускна кваліфікаційна робота студента

захищена з оцінкою

Голова Екзаменаційної комісії
професор, доктор техн. наук Бондарчук Андрій

Київський національний університет імені Тараса Шевченка
Факультет інформаційних технологій
Кафедра програмних систем і технологій
Спеціальність 121 “Інженерія програмного забезпечення”

ЗАТВЕРДЖУЮ:

Завідувач кафедри програмних систем і технологій

_____ (О.С.Бичков)

“ _____ ” _____ 20__ р.

**ЗАВДАННЯ
НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ МАГІСТЕРСЬКУ РОБОТУ
СТУДЕНТУ**

_____ Табунову Антону Анатолійовичу

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної магістерської роботи “Автономна ANDROID-система визначення координат та поступового нарощування їх точності в умовах відсутності радіосигналів”

затверджені наказом вищого навчального закладу від „__” _____ 20__ р. № _____

2. Строк здачі студентом закінченої роботи _____

3. Вихідні дані до роботи теоретичні концепції алгоритму геолокації, опис методів поступового підвищення результатів обчислень, програмне забезпечення для мобільних пристроїв під управлінням операційної системи Android, що імплементує наведену у роботі методику.

4. Зміст пояснювальної записки (перелік питань, що їх належить розробити)

1. Аналіз існуючих рішень геолокації за астрономічними об'єктами.

2. Опис алгоритму геолокації, допоміжних технологій та інструментів вирішення поставленої задачі.

3. Опис методів підвищення точності роботи системи.

4. Розробка програмного забезпечення.

5. Результати досліджень і розробки, аналіз та пошук можливих рішень вдосконалення розробленої системи.

5. Перелік графічного матеріалу (з точним забезпеченням обов'язкових креслень)

1. Залежність між географічною широтою та довготою, часом сходу/заходу Сонця (рис. 1.1)

2. Зв'язок між кутом піднесення Сонця та довжиною тіні об'єкта (рис. 1.2)

3. Представлення небесної сфери у певній точці (рис. 2.1)

4. Діаграма пакетів (вищий рівень) (рис. 3.1)

5. Діаграма бази даних (таблиці) (рис. 3.2)

6. Діаграма класів пакету “logic” (скорочений варіант, основні сутності та зв'язки) (рис. 3.3)

7. Візуальне відображення інтерфейсу користувача (рис. 3.4)

8. Приклади відображення повідомлень (рис. 3.5)

9. Представлення для ручного пошуку об'єкта на зображенні (рис. 3.6)

6. Консультанти з роботи із зазначенням розділів роботи, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Аналітико-теоретична частина: розділ 1 та 2	В.Л. Шевченко		
Практична частина: розділ 3	В.Л. Шевченко		
Аналіз та перспектива: розділ 4	В.Л. Шевченко		

7. Дата видачі завдання _____

Керівник _____

(підпис)

В.Л. Шевченко

(розшифровка підпису)

Завдання прийняв до виконання _____

(підпис)

А.А. Табунов

(розшифровка підпису)

КАЛЕНДАРНИЙ ПЛАН

Номер і назва етапів роботи	Термін виконання етапів роботи	Примітка
1. Вивчення на аналіз наявних рішень геолокації		
2. Опис алгоритму визначення координат		
3. Пошук методів підвищення точності результатів		
5. Побудова архітектури програмного забезпечення		
6. Розробка першого наближення додатку		
7. Розробка інструментів до алгоритму геолокації		
8. Аналіз, пошук напрямків вдосконалення системи		
9. Оформлення пояснювальної записки		

Студент – магістр _____

(підпис)

А.А. Табунов

(розшифровка підпису)

Керівник роботи _____

(підпис)

В.Л. Шевченко

(розшифровка підпису)

АНОТАЦІЯ

Випускна кваліфікаційна магістерська робота: 77 с., 9 рис., 6 додат., 31 джерела.

Тема: Автономна ANDROID-система визначення координат та поступового нарощування їх точності в умовах відсутності радіосигналів

Об'єкт дослідження: зображення небосхилу, датчики, рух певних астрономічних об'єктів, розташування по відношенню до Землі.

Мета роботи: розширення можливостей неспеціалізованих мобільних пристроїв щодо визначення географічних координат в позаштатних ситуаціях за умови зникнення доступу до стандартної мережі GPS з впровадженням методів поступового підвищення точності.

Предмет дослідження: обчислювальні методи; технології та методи обробки зображень; технології отримання показань сенсорів та методи їх обробки; моделі руху астрономічних об'єктів, які застосовуються для визначення розташування цих об'єктів по відношенню до Землі.

Результати дослідження: запропонована модифікація математичної моделі геолокації з урахуванням її реалізації на неспеціалізованих мобільних пристроях телекомунікації під управлінням операційної системи Android; надано опис методів підвищення точності результатів; розроблено відповідне програмне забезпечення.

Висновок

Методика геолокації можлива до застосування в процесі вирішення задач, що не вимагають високої точності. Система залишається працездатною в умовах відсутності доступу до стандартних систем геолокації та мережі Інтернет.

ГЕОЛОКАЦІЯ ЗА НЕБЕСНИМИ ОБ'ЄКТАМИ, ГЕОГРАФІЧНІ
КООРДИНАТИ, ШИРОТА, ДОВГОТА, ОРІЄНТАЦІЯ У ПРОСТОРІ,
КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ

АННОТАЦИЯ

Выпускная квалификационная магистерская работа: 77 с., 9 рис., 6 прил., 31 источник.

Тема: Автономная ANDROID-система определения координат и постепенного наращивания их точности в условиях отсутствия радиосигналов

Объект исследования: изображения небосклона, датчики, движение некоторых астрономических объектов, расположение по отношению к Земле.

Цель работы: расширение возможностей неспециализированных мобильных устройств по определению географических координат во внештатных ситуациях при условии исчезновения доступа к стандартной сети GPS с внедрением методов постепенного повышения точности.

Предмет исследования: численные методы; технологии и методы обработки изображений; технологии получения показаний датчиков и методы их обработки; модели движения астрономических объектов, применяемые для определения положения этих объектов по отношению к Земле.

Результаты исследования: предложена модификация математической модели геолокации с учётом реализации на неспециализированных мобильных устройствах телекоммуникации под управлением операционной системы Android; приведено описание методов повышения точности результатов; разработано соответствующее программное обеспечение.

Вывод

Методика геолокации возможна к использованию в процессе решения задач, которые не требуют высокой точности. Система сохраняет работоспособность в условиях отсутствия доступа к стандартным системам геолокации и сети Интернет.

ГЕОЛОКАЦИЯ ПО НЕБЕСНЫМ ОБЪЕКТАМ, ГЕОГРАФИЧЕСКИЕ
КООРДИНАТЫ, ШИРОТА, ДОЛГОТА, ОРИЕНТАЦИЯ В ПРОСТРАНСТВЕ,
КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ

ANNOTATION

Graduation qualifying master's thesis: 77 pages, 9 figures, 6 applications, 31 sources.

Topic: Autonomous ANDROID-system for determining coordinates and gradually increasing their accuracy in the absence of radio signals

The research object: images of the sky, sensors, motion of objects some astronomical objects and their location in relation to the Earth.

The purpose of work is to expand the capabilities of non-specialized mobile devices to determine geographical coordinates in emergency situations when access to the standard GPS network has been lost and to introduce methods of gradual increase accuracy.

The research subject: computational methods; technologies and methods of image processing; sensor reading technology and methods of processing these values; models of astronomical objects' motion used to determine the location of these objects in relation to the Earth.

The results of the research: modification of the mathematical model of geolocation according to implementation on non-specialized mobile telecommunications devices running the Android operating system has been proposed; the description of the methods for improving the accuracy of the results is given; appropriate software has been developed.

Conclusions

The method of geolocation can be used in solving problems that do not require high accuracy. The system remains operational in the absence of access to standard geolocation systems and the Internet.

GEOLOCATION BY HEAVENLY BODIES, GEOGRAPHICAL COORDINATES, LATITUDE, LONGITUDE, ORIENTATION IN SPACE, COMPUTATIONAL MODELING

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ПОСТАНОВКА ЗАДАЧІ	13
1.1. Опис існуючих рішень геолокації	13
1.1.1. Система позиціонування на основі інтенсивності сонячного світла	13
1.1.2. Визначення географічного розташування об'єктів на основі довжини тіні	14
1.2. Визначення необхідності розробки нової системи	15
РОЗДІЛ 2 ОПИС АЛГОРИТМУ ГЕОЛОКАЦІЇ, ДОПОМІЖНИХ ТЕХНОЛОГІЙ ТА ВАРІАЦІЙ РІШЕНЬ	16
2.1. Вимоги для застосування алгоритму	16
2.2. Опис методики геолокації	16
2.2.1. Положення, що використовуються під час проведення геолокації	16
2.2.2. Перший етап. Визначення даних за небесним об'єктом: азимут та кут піднесення	18
2.2.3. Другий етап. Визначення географічних координат	20
2.2.4. Підвищення точності на другому етапі	21
2.3. Запропонований інструментарій для роботи системи	22
2.3.1. Застосування моделювання руху астрономічних об'єктів	22
2.3.2. Отримання орієнтації пристрою: робота з даними сенсорів	23
2.3.3. Автоматизований пошук небесного об'єкта на фотознімку: примітивний алгоритм	24
2.3.4. Автоматизований пошук небесного об'єкта на фотознімку: використання нейронних мереж	25
2.3.5. Підвищення точності обчислень: реальні координати пристрою фіксовані	26

2.3.6. Прогнозування координат в умовах переміщення пристрою за певним правилом	27
2.3.7. Підвищення точності обчислень: реальні координати пристрою змінюються	28
2.4. Використання алгоритму в умовах підвищеної хмарності	29
2.5. Висновки за алгоритмом визначення географічних координат у поєднанні з запропонованим інструментарієм	30
РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	32
3.1. Цільова платформа та основні технології для розробки мобільного додатку	32
3.2. Опис архітектури мобільного додатку (клієнтська частина)	32
3.2.1. Загальні відомості за архітектурою	32
3.2.2. Основні поняття системи та допоміжна функціональність	35
3.2.3. База даних	35
3.2.4. Опис модулю, що реалізує основну логіку системи	37
3.2.5. Налаштування, що доступні користувачу	41
3.2.6. Інтерфейс користувача та його компоненти	42
3.2.7. Опис основних взаємодій між компонентами	47
3.3. Побудова допоміжних модулів: комп'ютерне бачення на основі нейронних мереж	48
3.4. Узагальнення результатів розробки	49
РОЗДІЛ 4 РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ ТА РОЗРОБКИ, ПЕРСПЕКТИВИ РОЗВИТКУ	51
4.1. Базові відомості для користувача мобільного додатку	51
4.2. Галузь застосування	52
4.3. Узагальнення результатів досліджень та перспектива розвитку	52

ВИСНОВКИ	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	56
ДОДАТКИ	60

ВСТУП

Існує велика кількість ситуацій, в яких необхідно мати можливість визначити географічні координати об'єкта точно, чи хоча б отримати приблизні значення, як, наприклад, у випадках екстрених умов, коли особа загубилася, чи втратила зв'язок з навігаційною системою у горах, лісах, джунглях, тощо. Наявність додаткових підручних засобів геолокації у деяких ситуаціях може допомогти зберегти людські життя, пришвидшити пошук осіб, чи скоротити час реабілітації. Це демонструє актуальність проблеми орієнтування у критичних умовах.

Актуальність теми

Сучасне людство звикло до щоденного використання смартфонів. Відмова у роботі мережі здатна викликати у суспільстві глобальний хаос і паніку. А отже, постає питання забезпечення людства додатковими джерелами інформації щодо географічних координат в умовах зникнення зв'язку із сотовою мережею та системою глобального позиціонування. Крім того, точність при застосуванні лише одного способу геолокації не завжди є задовільною. Тому, можливо використовувати різні методи та системи, розробка і впровадження яких є актуальною задачею гарантування стійкого розвитку людства у будь-яких несподіваних ситуаціях.

Мета і задачі дослідження

Метою роботи є розширення можливостей на прикладі звичайних смартфонів (неспеціалізованих пристроїв телекомунікації) під управлінням операційної системи Android щодо визначення географічних координат в позаштатних ситуаціях за умови зникнення доступу до стандартної мережі GPS (системи глобального позиціонування) і зв'язку із сотовою мережею з впровадженням методів поступового підвищення точності.

Задачі та завдання дослідження, що дозволяють досягти поставленої мети:

1. огляд існуючих методів визначення географічних координат;

2. теоретична розробка алгоритму геолокації;
3. визначення можливостей отримання показників сенсорів у мобільному пристрої та подальшої обробки отриманої інформації;
4. визначення підходів отримання необхідної мета-інформації фотознімку небосхилу та їх реалізація;
5. дослідження та впровадження (реалізація) моделей руху небесних об'єктів;
6. реалізація алгоритму геолокації на основі попередньої теоретичної обробки для використання у Android-застосунку.

Об'єкт дослідження - зображення небосхилу, датчики, рух певних астрономічних об'єктів (Земля, Сонце, Місяць тощо) та їх взаємодія, розташування по відношенню до Землі.

Предмет досліджень - обчислювальні методи; технології та методи обробки зображень; технології отримання показань сенсорів та методи їх обробки; моделі руху астрономічних об'єктів, які застосовуються для визначення розташування цих об'єктів по відношенню до Землі.

Методи дослідження

Під час виконання досліджень та прийнятті рішень використовувались підходи та методи: теорії кватерніонів та матричного аналізу, методи чисельних обчислень, методи пошуку об'єктів на зображенні, методи астрономічного моделювання.

Новизна одержаних результатів

1. Запропонована модифікація математичної моделі геолокації з урахуванням її реалізації на неспеціалізованому мобільному пристрої телекомунікацій. Універсальність такого рішення полягає в можливості адаптації моделі під використання даних різних небесних тіл (Сонця, Місяця, полярної зірки, тощо).

2. Вдосконалена математична модель обробки даних за рахунок впровадження методів поступового підвищення точності обчислень.

Практичне значення одержаних результатів

1. Програмне забезпечення дозволяє обчислювати приблизне географічне положення побутового мобільного пристрою на базі штатних сенсорів в умовах зникнення доступу до системи глобального позиціонування та мережі Інтернет.

2. Математична модель та її реалізація дозволяє адаптацію рішення під різні неспеціалізовані мобільні пристрої.

3. Представлено прототип програмного забезпечення для використання на неспеціалізованих мобільних пристроях під управлінням платформи Android. Реалізація виконувалася з урахуваннями подальшого розширення функціоналу.

Особистий внесок студента

Стислий опис результатів дослідження: розроблено методикку визначення географічного положення, що реалізується на неспеціалізованих мобільних пристроях та дозволяє проводити геолокацію в автономному режимі. Представлені методи поступового підвищення точності географічних координат, що можуть застосовуватися у поєднанні з наведеною методикою. Розроблено мобільний застосунок під операційну систему Android, у якому імплементовано зазначений метод. Проведені теоретичні дослідження з розширення сфери застосування алгоритму.

Апробація результатів випускної кваліфікаційної магістерської роботи

Результати дослідження представлені та схвалені на другому турі Всеукраїнського конкурсу студентських наукових робіт зі спеціальності “Комп’ютерні науки”. Робота посіла друге місце (додаток А).

Впровадження результатів дослідження засвідчене відповідними актами (додаток Б та додаток В).

Публікації

- 1) Табунов А., Шевченко В., Програмне забезпечення для визначення координат за допомогою сенсорів смартфона без використання GPS, - XII всеукраїнська науково-практична конференція «Актуальні проблеми комп'ютерних наук АПКН-2020», м. Хмельницький, 9-10 листопада 2020. - с. 292-294
- 2) Табунов А., Шевченко В., Програмне забезпечення визначення координат за допомогою сенсорів смартфона без використання GPS, - VII Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Об'єднані наукою: перспективи міждисциплінарних досліджень», 12-13 листопада 2020. - с. 221-223
- 3) Табунов А., Шевченко В., Автономна Android-система визначення координат та поступового нарощування їх точності в умовах відсутності радіосигналів, - 7-ма Східно-Європейська конференція Математичні та програмні технології Internet of Everything, м. Київ, 2020. - с. 61
- 4) Табунов А., Шевченко В., Забезпечення геолокації з підтримкою стабільної роботи в позаштатних умовах відсутності радіозв'язку, - 7-ма Східно-Європейська конференція Математичні та програмні технології Internet of Everything, м. Київ, 2020. - с. 62-63

РОЗДІЛ 1

АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Опис існуючих рішень геолокації

1.1.1. Система позиціонування на основі інтенсивності сонячного світла

Географічні широта та довгота визначаються на основі часу сходу та заходу Сонця (рис. 1.1).

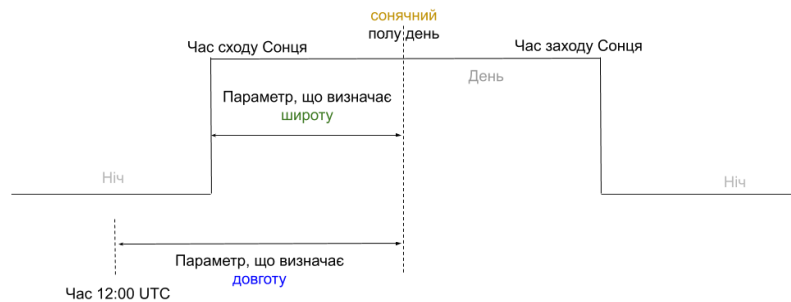


Рис. 1.1. Залежність між географічною широтою та довготою, часом сходу/заходу Сонця

Схід та захід Сонця позначаються такими у моменти, коли інтенсивність освітлення досягає порогових значень [1-2]. Система передбачає наявність точного годинника (для визначення часу (UTC) і дати) та датчика інтенсивності освітлення.

Розглянута система становить просту та енергоефективну систему локалізації датчиків, які, наприклад, закріплені на поверхні об'єкту. На основі довжини дня можливо проводити моніторинг якості води [1], відслідковувати міграцію тварин (наприклад, північного морського слона [3]).

Проте, до недоліків використання такої системи (як окремого компоненту) належить той факт, що інтенсивність освітлення розглядається як постійна

лише в межах певної зони, і наявність значного руху об'єкта (між сходом та заходом) матиме вплив на коректність результатів. Також погода, особливості флори (наприклад, рослини затримуватимуть частину сонячних променів), інші фактори можуть призвести до появи похибок.

1.1.2. Визначення географічного розташування об'єктів на основі довжини тіні

Стратегія геолокації спирається на те, що за допомогою інформації про довжину тіней від об'єктів можна вивести значення кута піднесення Сонця (рис. 1.2) [4]. Для ідентифікації на зображенні вертикальних об'єктів та їх тіней пропонується застосовувати нейронні мережі.

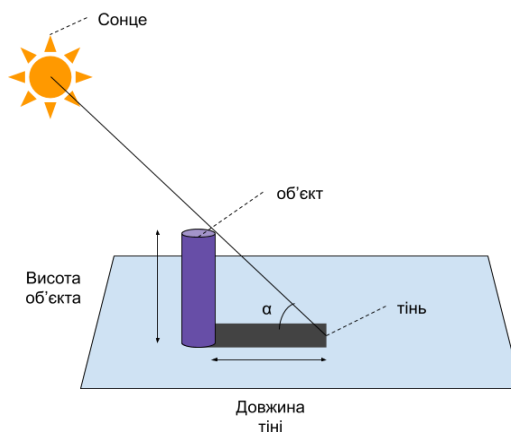


Рис. 1.2. Зв'язок між кутом піднесення Сонця та довжиною тіні об'єкта

У роботах [4] та [5] встановлюється зв'язок між кутом піднесення та географічною широтою, сонячним часом (що в свою чергу пов'язаний з кутом падіння сонячних променів на Землю) та значенням довготи.

Система майже не потребує додаткових дій (калібрування тощо) від користувача, а вся необхідна інформація для визначення локації, зазвичай, зберігається у метаданих фотознімку. Недоліком даної системи є похибка, яку

можливо отримати при знаходженні довжини тіней, через що результатом буде визначена некоректна зона.

1.2. Визначення необхідності розробки нової системи

Існує велика кількість ситуацій, в яких необхідно мати можливість визначити географічні координати об'єкта точно [6] (наприклад, для управління польотом літака), чи хоча б отримати приблизні значення (щоб зберегти життя), наприклад, у випадках: у Карпатах загубилися декілька груп на початку 2021 року, у лісі на Тернопільщині загубився та загинув грибник у 2017 році, миротворець Організації Об'єднаних Націй втратив зв'язок зі своїм підрозділом в джунглях Конго у 2018 році, пошукова операція тривала більше 15 діб [7-9]. Це не одиничні випадки, вони трапляються в усьому світі. У таких ситуаціях можливе використання допоміжних систем геолокації, проте відомі технічні рішення зазвичай не надають достатньо точних результатів для користувачів (можливе усереднення та комбінація даних, що отримані з різних джерел), або використовуються в якості бортових програмних комплексів [10-12]. Недоліками таких систем є громіздкість, вимога щодо розташування на борту рухомого засобу, великі вартість та енергоспоживання, недоступність для пересічних громадян.

Саме тому виникає необхідність у розробці та впровадженні додаткового підручного засобу геолокації, що буде застосовуватися для уточнення [13] даних основних систем (глобальна система позиціонування [14]) та матиме можливість працювати як автономна система в умовах, коли визначення географічних координат ускладнене чи неможливе. В якості додаткового засобу визначення географічного положення можна використовувати інформацію про розташування Сонця, Місяця та зірок на небосхилі.

РОЗДІЛ 2

ОПИС АЛГОРИТМУ ГЕОЛОКАЦІЇ, ДОПОМІЖНИХ ТЕХНОЛОГІЙ ТА ВАРІАЦІЙ РІШЕНЬ

2.1. Вимоги для застосування алгоритму

Методика геолокації, яка запропонована у роботі, можлива для використання на неспеціалізованому мобільному пристрої, до оснащення якого входять:

- годинник, який забезпечує точне визначення дати та часу (за Гринвічем);
- датчики [15] акселерометр та магнітометр;
- камера, що дозволяє проводити фотозйомку.

Вказані сенсори здебільшого представлені у штатній комплектації смартфонів.

Використання часу за Гринвічем обумовлено певним спрощенням при виконанні обчислень (ігнорування часового поясу, урахування переходу на літній час [1] тощо).

2.2. Опис методики геолокації

2.2.1. Положення, що використовуються під час проведення геолокації

У роботі застосовується принцип дії, що подібний до вимірювань секстанту (оптичний прилад, за допомогою якого можливо провести визначення величини піднесення небесного тіла (наприклад, Сонця) [16]).

Методика геолокації побудована на використанні небесної сфери - уявної сфери, на яку проектуються небесні об'єкти [17] (рис. 2.1), та базується на

визначенні близьких до реальних значень кутів азимуту та піднесення (схилення) [18].

При проведенні визначення географічних координат вважається, що азимутом є горизонтальний кут між напрямками на північ та на проекцію небесного об'єкта (відлік від 0 до 360 градусів за годинниковою стрілкою: північ, схід, південь, захід).

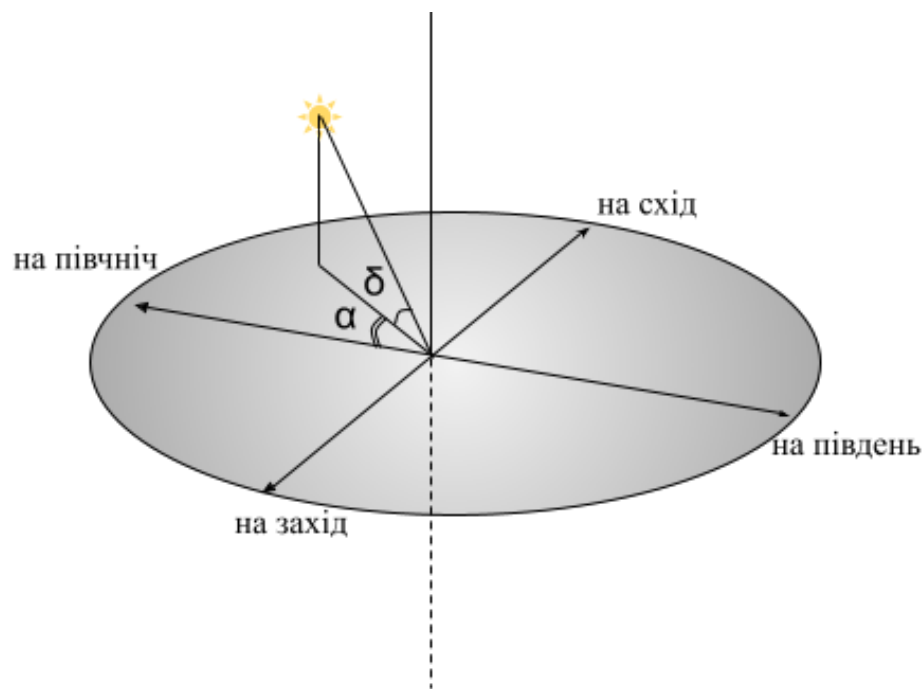


Рис. 2.1. Представлення небесної сфери у певній точці, де куту “ α ” відповідає азимут; “ δ ” – схилення

При використанні методики вважається, що у кожен момент часу можливо однозначно визначити координати (з певним околom) у певній зоні на поверхні Землі за кутами азимуту та схилення для встановленого небесного об'єкту. Висота над рівнем моря у такій зоні приймається незмінною.

2.2.2. Перший етап. Визначення даних за небесним об'єктом: азимут та кут піднесення

Визначення кутів азимуту та піднесення (над горизонтом) за небесним об'єктом можливо на основі показників, що надають датчики [15] акселерометр та магнітометр. Для цього проводиться обчислення кутів обертання системи координат мобільного пристрою відносно опорної системи координат XYZ (точка відліку знаходиться у поточному розташуванні), де [19]:

- вісі X та Y лежать у дотичній до Землі площині у поточному розташуванні;
- позитивний напрямок вісі Y вказує на північний магнітний полюс;
- вісь Z спрямована у небо (догори) та перпендикулярна дотичній до Землі площині у поточному розташуванні;
- вісь X вказує на орієнтовний схід (визначається на основі векторного добутку одиничних значень за Y та Z).

Подальше отримання шуканих на етапі значень забезпечується проведенням певних дій, які пов'язані з урахуванням положення мобільного пристрою. Слід зазначити, що система координат, яка відповідає мобільному пристрою, співпадає з опорною при нульових значеннях кутів обертання. Забезпечення дотримання необхідних правил розташування виконується за допомогою штатної фотокамери. Надалі вважається, що наявна відповідність між системами координат мобільного пристрою та зображення, яке є результатом фотозйомки (інакше, необхідно враховувати додаткові перетворення - співставлення систем координат).

Кут піднесення небесного об'єкту визначається як зміна кута обертання мобільного пристрою навколо опорної вісі X. Для цього необхідно обчислити різницю двох відповідних кутів:

- перше значення визначається при фотозйомці небесного об'єкту, коли той розташований по центру зображення;

- друге значення отримується при фотозйомці за умови суміщення центральної точки зображення з лінією горизонту (загалом можливо прийняти значення другого показника як нуль).

Вважаємо, що, при застосуванні методики, значення за модулем кута обертання навколо вісі X не може перевищувати 90 градусів (для забезпечення цього користувачу достатньо виконати додаткові обертання навколо вісі Z). Обертання навколо вісі Y при визначенні кута піднесення передбачено на рівні похибки.

Необхідне значення кута азимуту визначається як кут обертання мобільного пристрою навколо вісі Z , коли на зображенні, що отримано з фотокамери, небесний об'єкт розташований на лінії, яка проходить через центральну точку зображення та є перпендикуляром до поверхні Землі у поточному розташуванні. Така лінія визначається за кутом обертання навколо вісі Y .

При одночасній наявності значного обертання навколо вісі X та навколо вісі Y при визначенні азимуту, є потреба у проведенні додаткових досліджень, адже можливе виникнення похибки (наприклад, через спотворення дистанцій на фотознімку). Такий варіант наразі не передбачений для обробки.

Дії, що виконуються при визначенні азимуту та кута піднесення за небесним об'єктом, можливо поєднати задля поліпшення досвіду користування методикою. Наявний варіант, коли користувачу досить зробити один фотознімок, де небесний об'єкт буде розташований у центрі зображення, а обертання мобільного пристрою навколо вісі Y буде на рівні похибки.

Момент фотозйомки небосхилу повинен супроводжуватися збереженням поточної інформації за часом та необхідних показників сенсорів, щоб мінімізувати похибку (не враховувати час обробки зображення). Валідація наявності та відповідності положення небесного об'єкту на зображенні відбувається через його пошук за допомогою додаткових інструментів, наприклад з використанням нейронних мереж.

2.2.3. Другий етап. Визначення географічних координат

Опорні географічні координати - це набір точок (широта та довгота), що відповідають певній зоні на поверхні Землі, де передбачається може знаходитися користувач методики. При вирішенні задачі вважається, що, в загальному випадку, опорні координати покривають зону з встановленим значенням зміни за широтою та зміни за довготою на вибраних ділянках.

Другим етапом визначення географічного положення є співставлення знайдених (на попередньому етапі) значень азимуту та кута піднесення зі значеннями кутів, які обчислені за опорними географічними координатами для того самого небесного тіла і у момент часу, що відповідає виконанню першого етапу (чи усереднене значення). Серед визначених опорних значень кутів відбувається пошук за методом найменших квадратів (формула 2.1):

$$(\alpha_{ref_i} - \alpha)^2 + (\beta_{ref_i} - \beta)^2 \rightarrow min \quad (2.1)$$

де α - значення азимуту, що отримано на першому етапі;

β - значення кута піднесення, що отримано на першому етапі;

α_{ref_i} - значення азимуту для опорних координат з індексом i ;

β_{ref_i} - значення кута піднесення для опорних координат з індексом i .

Результатом застосування методу буде значення індексу опорних координат, які вважаються результатом роботи алгоритму.

Знайдене значення географічних координат є приблизним та може відрізнятись від реального. Проте, можливо визначити окіл, у якому скоріше за все буде знаходитись користувач, в якості простору між найближчими сусідніми точками (у напрямках: на схід, на захід, на північ, на південь, на північний захід, на південний захід, на північний схід, на південний схід).

Швидкодія даного етапу залежить від кількості опорних точок: чим їх більше, тим більше необхідно часу на проведення допоміжних обчислень. Проте, коли точністю можна знехтувати, можливо зменшити кількість опорних географічних координат, що призведе до підвищення швидкодії. Також зменшити час виконання етапу можливо за рахунок програмної реалізації з виконанням обчислень у декількох потоках.

2.2.4. Підвищення точності на другому етапі

Для випадку, коли опорні географічні координати покривають зону “сіткоподібно” (тобто повністю покривають певні значення широти та довготи), можливо досягти підвищення точності на другому етапі за рахунок цільового стиснення простору:

- 1) Певна зона покривається “сіткоподібно” опорними географічними координатами зі встановленою частотою.
- 2) За методами, що описані раніше, визначається певний окіл, у якому скоріш за все знаходяться реальні значення географічного положення.
- 3) Знайдений окіл визначає нову зону менших розмірів.
- 4) Дії повторюються з першого етапу, доки не буде досягнута необхідна точність (окіл набуде потрібних розмірів).

Такий підхід позбавляє необхідності попереднього визначення усіх опорних координат для певної зони та зменшує кількість “непотрібних” обчислень (у частинах зони, що виходять за окіл на кожній ітерації). Перша ітерація не потребує розміщення великої кількості опорних географічних координат, а подальші ітерації використовують опорні координати, що обчислюються автоматично за встановленою частотою.

Загалом, точність результатів на другому етапу залежить від точності отриманих значень азимуту та кута піднесення на першому етапі та точності математичної моделі руху небесного тіла для визначення опорних значень кутів.

2.3. Запропонований інструментарій для роботи системи

2.3.1. Застосування моделювання руху астрономічних об'єктів

Для обчислення значень азимуту та кута піднесення небесних об'єктів (Сонце, Місяць, комети тощо), за географічними координатами на поверхні Землі, пропонується застосування алгоритмів визначення положення астрономічних об'єктів на основі орбітальних елементів [20-21], базові з яких визначають:

- форму та розмір орбіти, положення об'єкта на орбіті (середня дистанція (велика піввісь), ексцентриситет, час в перигелії);
- орієнтацію орбіти у просторі (нахил орбіти, довгота висхідного вузла, аргумент перицентра).

Точність застосування моделей (Сонце, Місяць) складає до двох кутових хвилин (1 кутова хвилина дорівнює $1/60$ градуса [21]) та поширюється на будь-яку дату у межах двадцять першого сторіччя. Для цього використанні певні спрощення [20-21]:

- ігнорування нутації та відхилень;
- ігнорування часу переміщення світла;
- ігнорування між земним часом та всесвітнім;
- обчислення прецесії у спрощений спосіб;
- ігнорування показників вищих порядків для орбітальних елементів;
- ігнорування більшості планетарних збурень, тощо.

Проте, наведена схема надає можливість впровадження обчислень за додатковими астрономічними об'єктами (наприклад, можливе визначення за кометами).

2.3.2. Отримання орієнтації пристрою: робота з даними сенсорів

Визначення положення мобільного пристрою у просторі відбувається на основі показників, що отримані з сенсорів акселерометру та магнітометру. Ці датчики представлені у значній кількості сучасних смартфонів (мобільних пристроїв) та можуть надавати значення у формі кватерніона, що описує обертання системи координат мобільного пристрою відносно опорної системи координат [19].

Значення, отримані у формі кватерніона, необхідно перетворити у формат кутів обертання - тривимірний вектор, що розраховується за формулою 2.2 (для подальшого зручного опису положення мобільного пристрою та використання при застосуванні методики геолокації):

$$\left\{ \begin{array}{l} \arcsin(2 \cdot (q_x \cdot q_z - q_w \cdot q_y)) \cdot 180 / \pi; \\ \operatorname{atan2}(2 \cdot (q_x \cdot q_w + q_y \cdot q_z), 1 - 2 \cdot (q_z^2 + q_w^2)) \cdot 180 / \pi; \\ \operatorname{atan2}(2 \cdot (q_x \cdot q_y + q_z \cdot q_w), 1 - 2 \cdot (q_y^2 + q_z^2)) \cdot 180 / \pi; \end{array} \right\} \quad (2.2)$$

де π - математична константа;

$q = \{X, Y, Z, W\}$ - попередньо сформований кватерніон.

Для загального випадку використання смартфонів під управлінням операційної системи Android, система координат мобільного пристрою описується наступним чином [22]:

- додатна вісь X спрямована перпендикулярно до правої сторони екрану смартфона (портретний режим);
- додатна вісь Y спрямована перпендикулярно до верхньої частини смартфона (портретний режим);
- вісь Z перпендикулярна поверхні мобільного пристрою (екрану смартфона) та спрямова від зовнішньої частини екрану.

Проте, слід зауважити, що якщо система координат зображення, яке надає фотокамера, не співпадає з системою координат мобільного пристрою, то результуючі значення кутів обертання, що будуть використані при виконанні методики геолокації, обчислюються при виконанні співставлення вищезазначених систем координат (зокрема може змінитися послідовність застосування обертання за вісями).

2.3.3. Автоматизований пошук небесного об'єкта на фотознімку: примітивний алгоритм

Для небесних світил, які формують зону значного розміру певної інтенсивності (вдень - Сонце, вночі - Місяць) на зображенні, що надає фотокамера, можливо застосувати примітивний алгоритм пошуку (за умови, що така зона одна на зображенні), який складається з етапів:

- 1) Обробка зображення з застосуванням чорно-білого фільтру.
- 2) Визначення “найяскравішого” пікселю на чорно-білому зображенні.
- 3) Бінаризація зображення - кожному пікселю ставиться у відповідність одне зі значень:
 - 1 - якщо інтенсивність пікселю відповідає значенню, що визначено на другому етапі, з певним оточенням;
 - 0 - у іншому випадку.
- 4) Пошук на бінаризованому зображенні скупчення одиниць, що має найбільшу площу.
- 5) Визначення центральної точки скупчення, що є результатом роботи алгоритму.

2.3.4. Автоматизований пошук небесного об'єкта на фотознімку: використання нейронних мереж

Реалізація автоматизованого пошуку небесного об'єкта на фотознімку пропонується при виконанні двох етапів.

На першому етапі, проходить перевірка факту наявності на зображенні шуканого небесного об'єкта (відповідність до певного класу). Для цього пропонується використання згорткової нейронної мережі визначення.

Згорткові нейронні мережі (Convolutional Neural Network) відрізняються високим показником розпізнавання патернів на зображенні, тобто класифікації [23], що підходить для поставленої на етапі мети. Згорткові нейронні мережі працюють на основі фільтрів, що займаються розпізнавання певних характеристик зображення (наприклад, вигнуті лінії Місяця). Фільтром є колекція “кернелів” - матриць чисел з вагами, що навчаються для пошуку певних характеристик на зображенні. Фільтр переміщується вздовж зображення та визначає, чи присутня певна характеристика на цьому проміжку. Для визначення такого роду застосовується згортка, що еквівалентно сумі всіх добутків чисел матриць фільтру та вхідного сигналу. Приблизно можна описати роботу так: якщо певна характеристика наявна на зображенні, то результатом буде велике число; за відсутності - маленьке число [23-24].

При розробці нейронної мережі слід визначити, що:

- використовується три канали фільтру [23], оскільки процес відбувається над зображеннями, які описуються у форматі RGB;
- застосовується функція активації ReLU (Rectified Linear Unit) [23] для формування певної похибки з метою кращого навчання: значення, що менші, чи дорівнюють нулю - перетворюються у нуль, усі інші значення залишаються без змін;
- класи класифікації відповідають небесним тілам (Сонце, Місяць), за якими можливий пошук.

Опис обраної моделі:

- вхідний слой приймає зображення 250x250;
- представлено 2 згорткових слоя;
- за кожним згортковим слоєм застосовується слой максимального об'єднання [23]: з пікселів, що потрапили до поля зору (2x2) обирається максимальний, та переходить у результуючу матрицю;
- для визначення класифікації раніше отриманих характеристик застосовується багатосаровий перцептрон (Multi-Layer Perceptron, MLP) [23];
- вихідний слой відповідає за формування ймовірностей прилежності об'єкту до класу;
- кількість нейронів вихідного слою дорівнює кількості класів;
- на виході сигнали модифікуються на основі функції активації softmax (перетворення у вектор, де кожен параметр знаходиться у межах від 0 до 1, та сума усіх параметрів дорівнює 1).

Другий етап передбачає застосування нейронної мережі типу архітектури “Autoencoder” для пошуку положення визначеного об'єкту на зображенні (за умови його наявності).

Нейронна мережа приймає зображення розміру 250 пікселів (висота) на 250 пікселів (ширина). Тому, при її тренуванні та при її використанні необхідно проводити попереднє масштабування зображення зі збереженням пропорцій, після якого усі вільні пікселі заповнюються середнім кольором по зображенню (для уникнення збоїв у роботі, коли вільні пікселі якогось одного кольору будуть визначати характеристику).

2.3.5. Підвищення точності обчислень: реальні координати пристрою фіксовані

Коли неспеціалізований мобільний пристрій знаходиться у фіксованому положенні (відсутній рух та значна зміна географічних координат) можливо

провести поступове підвищення точності геолокації шляхом проведення повторних вимірів (ітерацій уточнення, де i - індекс, номер ітерації):

1) На першій ітерації: результуюче значення географічних координат дорівнює поточному визначенню ($i = 0$) місцезнаходження (широта та довгота).

2) За необхідністю продовження уточнення, кожна наступна ітерація (значення i збільшується на одиницю) супроводжується виконанням дій:

- нове визначення географічного положення (з індексом i);
- оновлення результуючого значення як середнього (за кожною координатою) між поточним результуючим значенням та місцезнаходженням з індексом i .

Також, визначення остаточних географічних координат можливо як усереднення значень усіх вимірів з проведених попередньо ітерацій.

2.3.6. Прогнозування координат в умовах переміщення пристрою за певним правилом

Нехай користувач разом з мобільним пристроєм переміщується у просторі (змінюється реальне географічне положення). Тоді, після проведення певної кількості визначень, можливо обчислити приблизні майбутні географічні координати, де буде знаходитись користувач через певний час, на основі екстраполяції, наприклад, ступеневим поліномом [25], за кожною географічною координатою (формула 2.3):

$$\begin{aligned} \text{Long}(t) &= \sum_{i=0}^{m-1} \text{along}_i \cdot t^i; \\ \text{Lat}(t) &= \sum_{i=0}^{m-1} \text{alat}_i \cdot t^i \end{aligned} \quad (2.3)$$

де m - кількість даних;

t - змінна часу;

Long - функція екстраполяційного поліному для довготи;
Lat - функція екстраполяційного поліному для широти;
 $along_i$ та $alat_i$ - коефіцієнти з індексом i відповідних поліномів.

Пошук коефіцієнтів поліному можливо виконати на основі методу найменших квадратів: сума квадратів відхилень реальних значень широти та довготи від обчислених за відповідною функцією повинна бути мінімальна. Якщо коефіцієнти були однозначно визначені, то можливо обчислити орієнтовні географічні координати користувача через певний час (за умови, що користувач переміщується за якимось правилом).

Слід додати, що не у всіх випадках поліном необхідно будувати за всіма даними визначень, можливо використовувати останні актуальні значення. А функцію екстраполяції, загалом, краще визначати в залежності від виду руху.

2.3.7. Підвищення точності обчислень: реальні координати пристрою змінюються

При переміщенні мобільного пристрою (зміні географічного положення) можливо підвищити точність обчислень, використовуючи експоненційне згладжування на основі фільтру Калмана [26]. Для цього необхідно провести певну кількість визначень географічних координат, після чого виконати наступні дії:

Крок 1. Встановити проміжок часу, через який буде відбуватися повторне визначення місцезнаходження, - *<певний час>*.

Крок 2. Провести прогнозування (на основі попередніх геолокацій) значень географічних координат через *<певний час>*: {*широта_прогнозована*; *довгота_прогнозована*}.

Крок 3. Через *<певний час>* повторно визначити місцезнаходження: {широта_обчислена; довгота_обчислена}.

Крок 4. Обчислити результуюче значення географічного положення за формулами 2.4:

$$\begin{aligned} \text{широта} &= V * \text{широта_прогнозована} + (1 - V) * \text{широта_обчислена}; \\ \text{довгота} &= V * \text{довгота_прогнозована} + (1 - V) * \text{довгота_обчислена} \end{aligned} \quad (2.4)$$

де *широта* – результуюче значення широти поточного місцезнаходження;

довгота – результуюче значення довготи поточного місцезнаходження;

V – коефіцієнт довіри до прогнозованих даних;

широта_прогнозована, *довгота_прогнозована*, *широта_обчислена*, *довгота_обчислена* – значення, що отримані на попередніх етапах.

Наведені дії можливо повторювати за необхідністю.

2.4. Використання алгоритму в умовах підвищеної хмарності

Підвищена захмареність небосхилу може спричинити ряд ускладнень при застосуванні описаної у роботі методики (неможливість визначити положення небесного об'єкту на зображенні та правильно розташувати мобільний пристрій).

Це питання може бути частково вирішено при застосуванні поляризаційного фільтру для штатної фотокамери (наприклад, у якості окремого компоненту). За його допомогою можливо визначити зони найбільшої та найменшої поляризованості світла, що допоможе отримати приблизне положення небесного світила (Сонця) [27-28] на фотозображенні.

Слід додати, що певні результати можуть бути отримані і при застосуванні побутових поляризаційних сонцезахисних окулярів у якості фільтру (в тому числі при великій інтенсивності сонячного світла). Проте,

гідним варіантом фільтру може слугувати кристал ісландського шпату (прозорий крупнокристалічний різновид кальциту) [29].

Спостереження за хмарним небосхилом через ісландський шпат та виконання певних обертів останнім надають можливість визначення зон, звідки, в результаті релеєвського розсіювання, проходить повністю поляризоване світло. Положення схованого за хмарами Сонця визначається на основі перпендикулярів до ліній, що з'єднують вищевказані області [29].

Застосування ісландського шпату потенційно може забезпечити розширення варіантів використання методики, адже під час сутінок небосхил випромінює поляризоване світло, що надає змоги визначити напрямок на джерело світла навіть, коли Сонце сховалося більше ніж на 5 градусів за горизонт [27-28].

Співпадіння та неспівпадіння площин поляризації у розсіяного світла та у кристала виражається затемненням та висвітленням небосхилу.

Точність використання кристалу: похибка у кілька градусів [27, 28, 30]. Точність підвищується зі зменшенням гущини хмар та появою розривів між хмарами. Застосування описаного фільтру найкраще у високих широтах [27-28].

2.5. Висновки за алгоритмом визначення географічних координат у поєднанні з запропонованим інструментарієм

У роботі представлена методика геолокації, що може бути використана для задач, які не вимагають високої точності. Представлені можливості підвищення точності результатів обчислень.

Загалом процес геолокації накопичує певні похибки, які пов'язані з:

- точністю моделювання руху небесних об'єктів;
- точністю значень отриманих з сенсорів;
- не врахуванням явища рефракції (заломлення сонячних променів при проходженні через атмосферу Землі) [31];

- чутливістю методики до місцевості, у межах якої проходить значна зміна висоти над рівнем моря;
- ракурсом фотозйомки.

Найбільш точні результати на даному етапі розробки методики можливо отримати при проведенні геолокації на мінімальній висоті над рівнем моря.

Для пошуку небесного об'єкту на зображенні можливо застосовувати комбінацію представлених методів, щоб забезпечити більш стабільну роботу методики (системи). При проведенні геолокації за сузір'ями, пропонується розглядати одну зірку сузір'я як опорну.

Можливі варіанти подальшого розвитку процесу геолокації за наведеною методикою:

- підвищення точності представлених моделей та методів;
- використання показань датчику тиску (за наявності) для врахування висоти над рівнем моря;
- адаптація алгоритмів для використання під час навігації на інших планетах, орієнтуючись на відповідні світила.

РОЗДІЛ 3

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Цільова платформа та основні технології для розробки мобільного додатку

У роботі представлена версія програмного забезпечення для роботи на мобільних пристроях під управлінням операційної системи Android. Цільова версія операційної системи для використання програмного продукту починається з версії 5.1 (API level 22).

Розробка мобільного додатку відбувається з використанням мови програмування Java та інтегрованої середовища розробки Android Studio. Для забезпечення рівня абстракції при роботі з клієнтською базою даних використовується бібліотека Room (версія 2.4.2). Процес взаємодії з камерою базується на використанні бібліотеки CameraX.

Слід зауважити, що загальні викладки по визначенню географічних координат, які наведено у роботі, можливо реалізувати для використання на неспеціалізованих мобільних пристроях, що відповідають технічним вимогам, під управлінням ряду операційних систем, що відмінні від Android.

3.2. Опис архітектури мобільного додатку (клієнтська частина)

3.2.1. Загальні відомості за архітектурою

Архітектура мобільного застосунку (додаток Г) представлена набором модулів, кожен з яких описаний у відповідному пакеті (рис. 3.1):

1) Пакет “common”. Модуль містить реалізацію загальних доменів, що можуть використовуватися при розробці інших систем. Наприклад, опис шаблонної логіки для роботи з кутами у різних форматах (градуси, радіани).

2) Пакет “domain”. Модуль представляє опис доменних об’єктів: сутності моделі, функції зворотного виклику, опис можливих налаштувань системи.

3) Пакет “database”. Модуль представляє опис взаємодії з базою даних мобільного додатку. У пакеті наведено опис схеми: таблиць (сутностей) та зв’язків між ними, представлень; опис запитів до бази даних (додавання нових записів, видалення, оновлення тощо); опис конвертацій для роботи з певними об’єктами у зручному форматі (наприклад, датами).

4) Пакет “logic”. Модуль представляє опис головної логіки роботи системи (визначення географічних координат) з використанням певного набору інструментів та реалізацій.

5) Пакет “ui”. Модуль містить програмний опис інтерфейсу користувача.

6) Пакет “uextensions”. Узагальнені розширення для роботи з інтерфейсом користувача: функціонал по відображенню повідомлень різних типів - про помилку, успішно виконану дію тощо; генерація локалізованих рядкових літералів з динамічною інформацією; опис визначень методів (інтерфейсів), що застосовуються у програмному коді інтерфейсу користувача.

7) Пакет “logicdependent”. Модуль, що представляє додаткові реалізації інструментів до пакету “logic”, що орієнтовані на використання тільки під управлінням даної операційної системи.

8) Пакет “infrastructure”. У модулі наведений опис пласту між платформою та основною логікою системи, що містить: репозиторії для зберігання даних, платформи-залежні реалізації (наприклад, отримання значень налаштувань та інструментарію), опис конвертацій сутностей бази даних у доменні об’єкти (та навпаки), менеджери (наприклад, по виклику зовнішніх сервісів) тощо.

9) Пакет “managment”. Модуль представляє пласт можливостей (дій) по керуванню основною логікою системи з інтерфейсу користувача. Зокрема у

модулі проходить: налаштування прослуховувачів значень сенсорів (їх запуск, вимкнення, отримання показників); запуск та підтримка процесу геолокації; виклик менеджера зв'язку з зовнішніми сервісами.

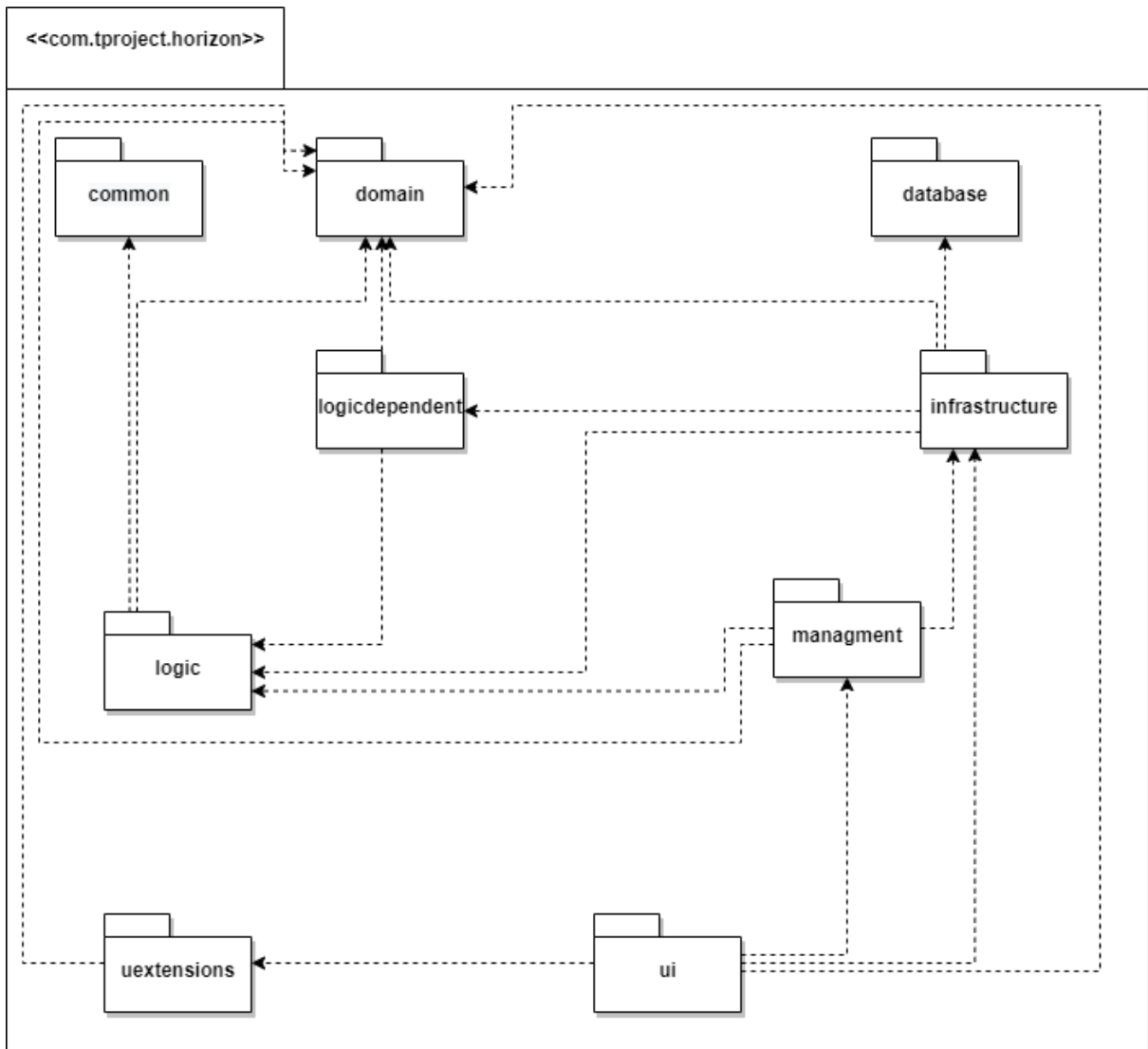


Рис. 3.1. Діаграма пакетів (вищій рівень)

Виокремлення модулів дозволяє: полегшити процес програмування; провести поділ роботи з розробки різних компонентів системи; проводити незалежно від інших модулів вдосконалення; повторно використовувати різні модулі при розробці іншого програмного забезпечення.

3.2.2. Основні поняття системи та допоміжна функціональність

Основні поняття, що використовуються при побудові архітектури системи:

1) “Визначення” - це одна ітерація повного визначення географічних координат (представляє алгоритм геолокації).

2) “Сесія” - це поєднання пов’язаних визначень, на основі яких обчислюються результуючі дані за якоюсь логікою (наприклад, усереднення значень географічних координат при проведенні повторних вимірів).

Програмна система покликана (у тому числі) допомагати людині в екстреній ситуації. Тому, пропонується використання email-інформування довіреної особи: визначені географічні координати надсилаються на вказану електронну адресу при появі доступу до мережі Інтернет (наприклад, у ситуаціях, коли людина не може постійно перевіряти доступність мережі, близька до втрати свідомості тощо).

3.2.3. База даних

База даних була реалізована для мобільного застосунку з метою поліпшити досвід користування системою. Її впровадження надає можливість завчасного завантаження та зберігання усіх необхідних даних за місцевістю, у якій буде знаходитися користувач (на випадок відсутності зв’язку), що дозволяє не тримати усю ключову інформацію у мобільному пристрої. Окрім цього, основні дані про процес геолокації зберігаються до бази даних, що дозволяє реалізовувати різні дії над ними, зокрема перегляд історії проведених раніше геолокацій.

Схема бази даних (рис. 3.2) включає об’єкти, які описують певні поняття системи.

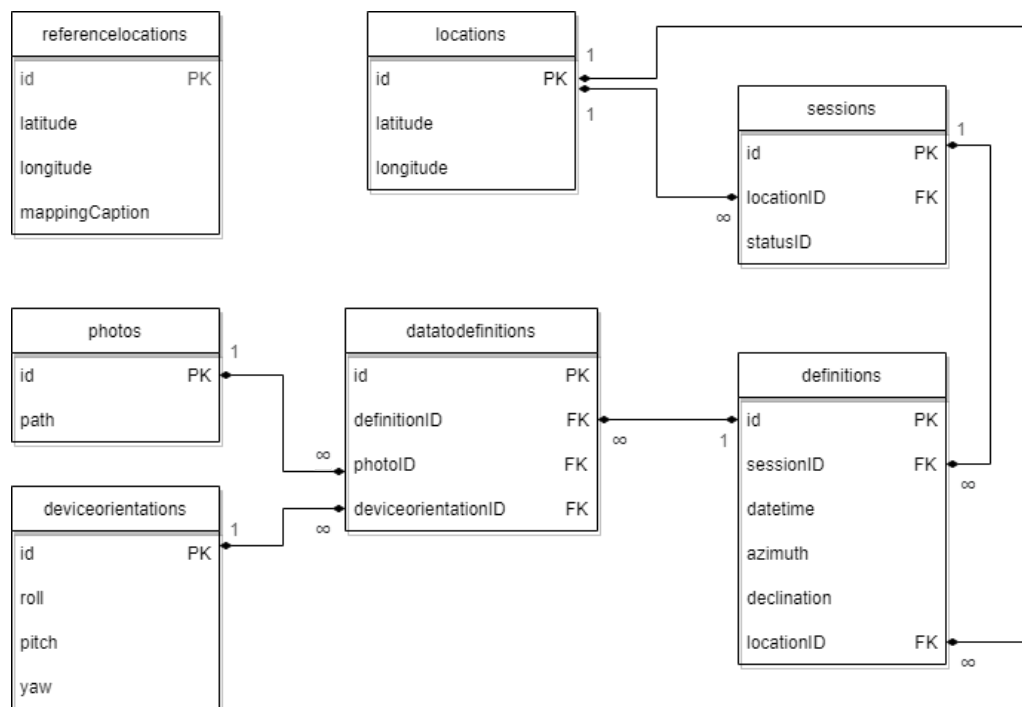


Рис. 3.2. Діаграма бази даних (таблиці)

1) Таблиця “referencelocations”. У таблиці зберігається інформація за географічними координатами, що є опорними точками під час процесу геолокації за наведеною у роботі методикою. Поля, що описують запис у таблиці (стовпці): ідентифікатор, значення широти, значення довготи, назва місцевості (для відображення користувачу).

2) Таблиця “locations”. У таблиці зберігається інформація за географічними координатами, які були визначені у процесі геолокації. Поля, що описують запис у таблиці (стовпці): ідентифікатор, значення широти, значення довготи.

3) Таблиця “sessions”. Таблиця описує сутність “сесія геолокації”. Поля, що описують запис у таблиці (стовпці): ідентифікатор, ідентифікатор локації (зовнішній ключ до таблиці “locations”), ідентифікатор статусу сесії.

4) Таблиця “definitions”. Таблиця описує сутність “визначення”. Поля, що описують запис у таблиці (стовпці): ідентифікатор, ідентифікатор сесії (зовнішній ключ до таблиці “sessions”), дата та час проведення визначення

географічних координат, значення азимуту, значення кута піднесення небесного об'єкту, ідентифікатор локації (зовнішній ключ до таблиці “locations”).

5) Таблиця “photos”. У таблиці зберігається інформація про зроблені знімки. Поля, що описують запис у таблиці (стовпці): ідентифікатор, шлях зберігання зображення у файловій системі.

6) Таблиця “deviceorientations”. У таблиці зберігається інформація про положення мобільного пристрою у просторі при проведенні визначення. Поля, що описують запис у таблиці (стовпці): ідентифікатор, крен, ристання, тангаж.

7) Таблиця “datatodefinitions”. Таблиця представляє зв'язку типу “багато до багатьох” між сутностями “визначення” та даними, що використовувалися при цьому визначенні. Поля, що описують запис у таблиці (стовпці): ідентифікатор, ідентифікатор визначення (зовнішній ключ до таблиці “definitions”), ідентифікатор фотознімку (зовнішній ключ до таблиці “photos”), ідентифікатор положення пристрою (зовнішній ключ до таблиці “deviceorientations”).

8) Представлення “SessionDetail”. Представлення описується запитом, який надає необхідну інформацію за кожною сесією геолокації, для відображення на панелі історії.

Пакет “database” містить опис об'єктів доступу до даних (шаблон “data access object”), що дозволяє забезпечити відокремлення безпосередньо логіки роботи з базою даних від інших компонентів системи.

3.2.4. Опис модулю, що реалізує основну логіку системи

Опис основної логіки та алгоритмів, що необхідні для проведення процесу геолокації, наведено у пакеті “logic”.

Кожен окремий компонент логіки будується з дотриманням певних концепцій, що дозволяють зробити розширення функціональності системи більш гнучким:

- 1) наявний уніфікований опис основних функцій для взаємодії з компонентом (у вигляді інтерфейсу);
- 2) представлено набір реалізацій цих методів (класи, що імплементують інтерфейс; можливо створювати нові реалізації, не змінюючи попередні);
- 3) використовується патерн проектування “Фабрика” для доступу до різних реалізацій компонентів.

У пакеті представлено опис методів фабрики на основі абстрактного класу. При ініціалізації модулю відбувається запис зовнішньої реалізації класу-фабрики (встановлення необхідної конфігурації) у спеціальне внутрішнє сховище.

Основні структурні компоненти логіки (рис. 3.3):

- 1) Менеджер сесії геолокації.

Загальні методи: реєстрація зображення; отримання даних сесії (географічні координати тощо); функція перевірки активності сесії; функція перевірки успішності останньої реєстрації зображення (чи були визначені географічні координати).

Реалізації (додаток Д): `BaseSessionManager` (визначає положення на основі одного визначення географічних координат), `StaticSessionManager` (застосовує повторні визначення для покращення точності), `DynamicSessionManager` (застосовує експоненційне згладжування для покращення точності).

- 2) Менеджер визначення географічних координат.

Загальні методи: реєстрація зображення; отримання даних визначення (в тому числі географічних координат); функція перевірки успішності визначення географічних координат.

Реалізації (додаток E): `SingleDefinitionManager` (необхідно одне зображення, визначення за опорними координатами); `MultiDefinitionManager` (необхідно три зображення, визначення за опорними координатами); `AutoAccuracyDefinitionManager` (необхідно одне зображення, визначення за опорними координатами зі збільшенням точності).

3) Менеджер по визначенню положення мобільного пристрою у просторі.

Загальні методи: отримання положення мобільного пристрою у просторі (кутів) на основі переданих з сенсорів показань.

Реалізації: `AccMagnDeviceOrientationManager` (базовий клас отримання положення у просторі на основі показників сенсорів акселерометру та магнітометру)

4) Менеджер пошуку об'єктів на зображенні (комп'ютерний зір).

Загальні методи: отримання знайденої опорної точки об'єкту на зображенні.

Реалізації: `BasePrimitiveSearchManager` (примітивний пошук на зображенні зони однакової інтенсивності); `BaseNeuralNetworkVisionManager` (пошук об'єкта на зображенні з використанням нейронних мереж); та реалізації для різних небесних об'єктів (Сонце та Місяць), що розширюють попередні (за необхідністю).

5) Менеджер, що надає доступ до даних моделювання руху небесних тіл.

Загальні методи: обчислення значень азимуту та піднесення небесного тіла на основі географічних координат та часу.

Реалізації базуються на використанні різних моделей за певними небесними тілами.

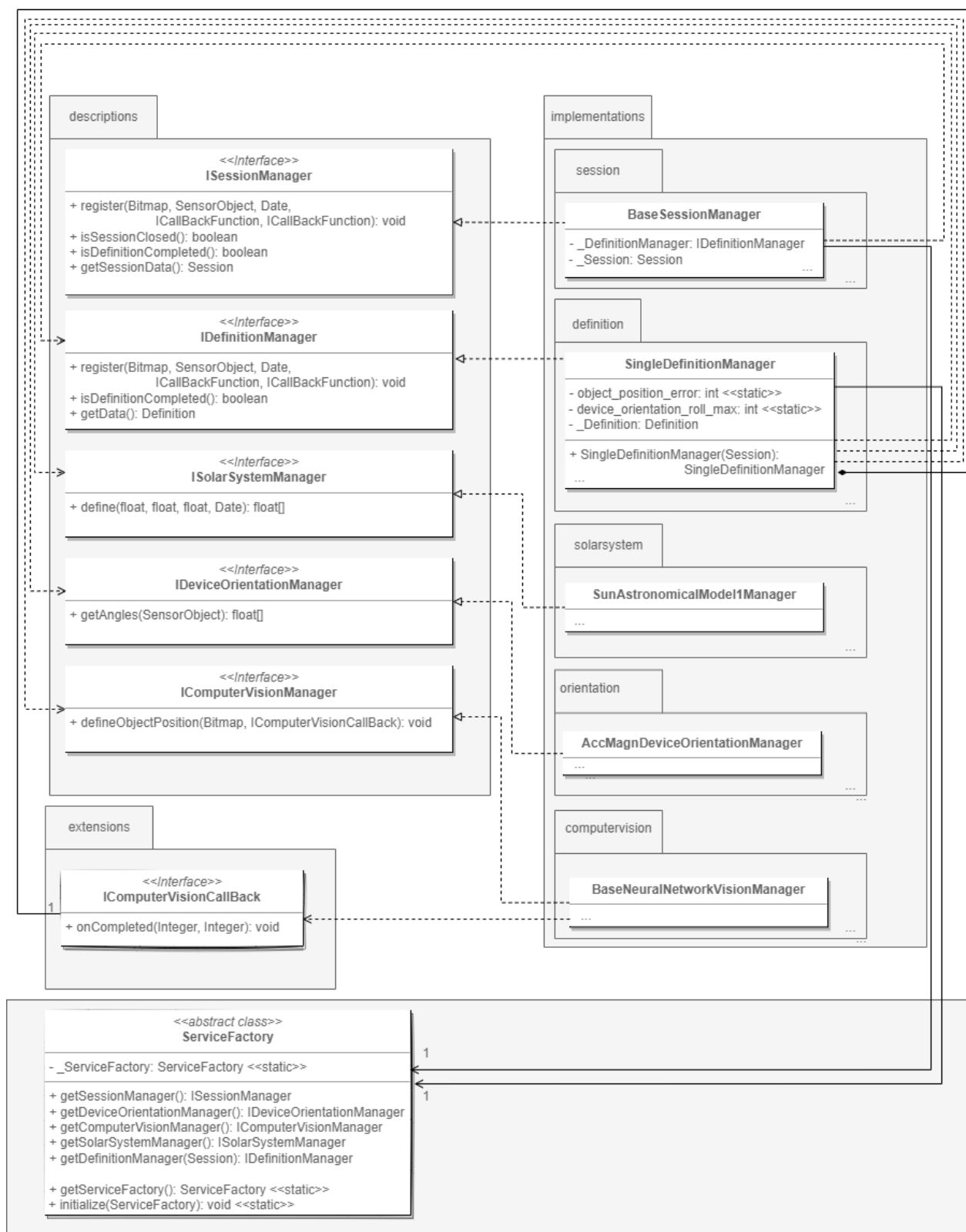


Рис. 3.3. Діаграма класів пакету "logic" (скорочений варіант, основні сутності та зв'язки)

3.2.5. Налаштування, що доступні користувачу

Налаштування у мобільному додатку представлені трьох типів:

1) Налаштування додатку.

Пункти налаштувань:

- “Мова” - вибір мови мобільного додатку.

2) Налаштування системної взаємодії. Цей тип включає усі налаштування системи, що не пов’язані з основною логікою (визначення географічних координат).

Пункти налаштувань:

- “Ваше ім'я для ідентифікації” - ім'я, що буде використовуватися при email-інформуванні довіреної особи;

- “Використовувати електронну адресу довіреної особи” - для активації цього пункту необхідно дозволити зберігання історії геолокації;

- “Електронна адреса довіреної особи” - електронна адреса, на яку будуть надходити листи з географічними координатами (якщо наявний доступ до мережі Інтернет);

- “Зберігати історію геолокації” - відповідає за збереження інформації (про успішні визначення географічних координат) у локальну базу даних.

3) Налаштування логіки.

Пункти налаштувань:

- “Визначення за” - вибір типу небесного об'єкта для процесу геолокації;

- “Наявність ускладнених умов” - вибір ускладнених умов з переліку (можливі значення - “підвищена хмарність”);

- “Приблизна місцевість” - назва приблизної місцевості для визначення опорних географічних координат;

- “Підвищення точності результатів” - вибір режиму сесії для підвищення точності (мобільний пристрій не змінює положення, мобільний пристрій переміщується у просторі);
- “Алгоритм комп'ютерного зору (<назва небесного тіла>)” - можливі значення: “Нейронна мережа”, “Примітивний пошук”, “Ручний пошук”;
- “Використання наближення моделі руху небесних тіл (<назва небесного тіла>)” - перелік доступних реалізацій моделей руху;
- “Підвищення точності обчислення (статичний режим)” - вибір алгоритму підвищення точності визначення географічних координат у статичному режимі;
- “Підвищення точності обчислення (динамічний режим)” - вибір алгоритму підвищення точності визначення географічних координат у динамічному режимі.

Кожен тип представлений окремою сутністю (інтерфейсом), у якій відображені усі можливі параметри налаштувань. Реалізації цих інтерфейсів (за стратегіями отримання та зберігання значень) представлені у пакеті “infrastructure”. Поточні значення отримуються за допомогою менеджера. Налаштування логіки кешується, щоб забезпечити незмінюваність (у тому числі у рамках однієї сесії). Для налаштувань системної взаємодії значення завжди актуальні. Це пов'язано зі зручністю користування при email-інформуванні.

3.2.6. Інтерфейс користувача та його компоненти

При розробці мобільного додатку, задля забезпечення взаємодії користувача з системою, було впроваджено три основні представлення:

1) Панель “Геолокація” (рис. 3.4). Представлення надає доступ користувачу до елементів, що необхідні для зручного проведення геолокації за представленим у роботі алгоритмом.

2) Панель “Історія” (рис. 3.4). Представлення відображає історію сесій геолокацій (у вигляді списку), що були проведені та збережені у базу даних.

3) Панель “Налаштування”. Представлення відображає список можливих налаштувань алгоритмів та системи, що підлягають зміні з боку користувача.

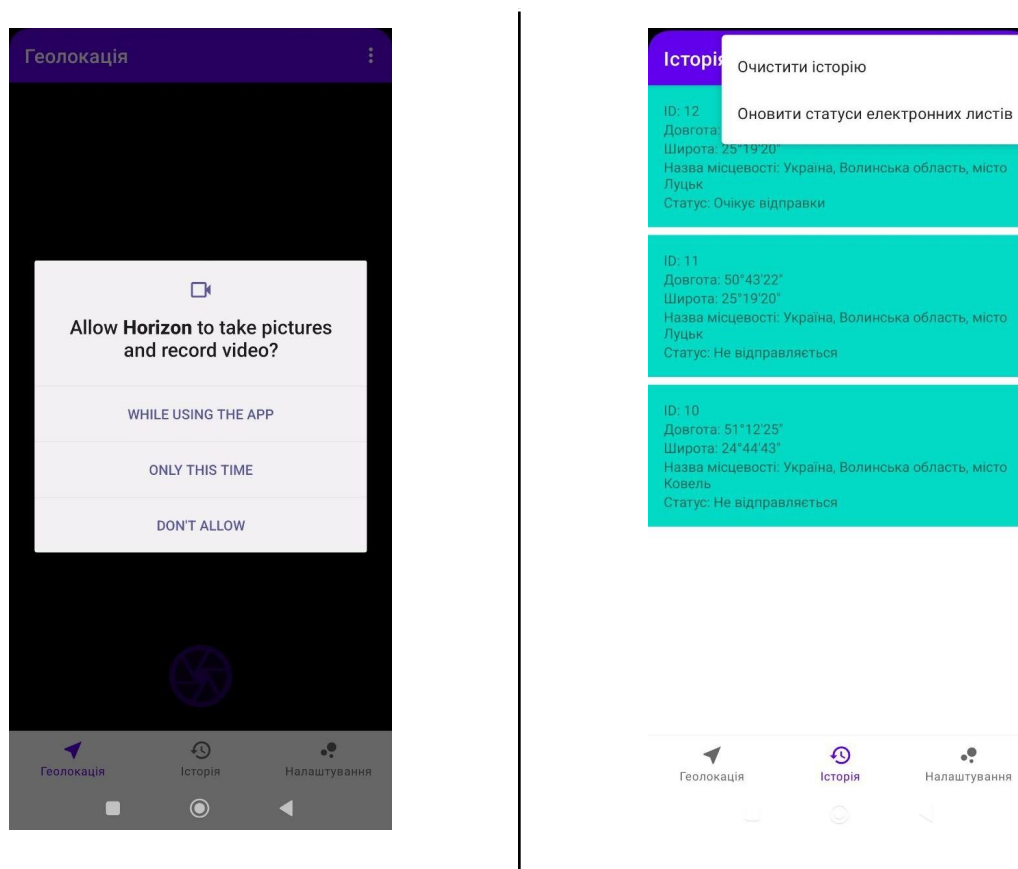


Рис. 3.4. Візуальне відображення інтерфейсу користувача: а) зліва - панель “Геолокація” (запит дозволів на зйомку); б) справа - панель “Історія” (з відображенням можливих дій)

Забезпечення переключення між представленнями реалізується за допомогою навігаційних кнопок внизу екрану. При відображенні представлення користувачу надається можливість виконання певних пов’язаних дій за допомогою контекстного меню у верхньому куті екрану:

- 1) Дії, які доступні на панелі “Геолокація”:
 - “Зупинити сесію геолокації” - при успішному виконанні дії поточна сесія буде завершена (без збереження результатів, якщо визначення часткове), буде створений новий менеджер сесії, що відповідає поточним налаштуванням.
 - “Завантажити дані” - при виконанні цієї дії проходить завантаження опорних географічних координат з зовнішнього сервісу та збереження їх до локальної бази даних.
- 2) Дії, які доступні на панелі “Історія” (див. рис. 3.4):
 - “Очистити історію” - виконання дії запускає очищення бази даних зі збереженою інформацією за всіма визначеннями та сесіями.
 - “Оновити статуси електронних листів” - при виконанні цієї дії проходить: виклик спеціального сервісу (при наявності мережі Інтернет) з метою спроби відправки усіх електронних листів, що очікують; отримання інформації про успішність відправлень електронних листів; оновлення відповідних статусів у базі даних.
- 3) Дії, які доступні на панелі “Налаштування”:
 - “Застосувати нові налаштування” - дія проводить очищення кешу налаштувань та перезавантажує додаток.

Опис елементів, які використовуються при побудові інтерфейсу користувача, відбувається у xml-файлах, що розташовані у ресурсах додатку (каталог “res”):

- каталог “layout” містить файли опису розмітки інтерфейсу користувача;
- каталог “menu” містить файли опису доступних користувачу дій (контексні меню) та переходів між представленнями (меню внизу екрану);
- файл “settings.xml” описує відображення налаштувань, доступних користувачу;

- файл “keys.xml” містить опис неперекладаємих літералів, на основі яких визначаються певні “ключі доступу” у системі (наприклад, для визначення певного пункту налаштувань, зручного мапінгу списку значень налаштувань);
- файл “arrays.xml” представляє опис доступних масивів, зокрема, для опису варіантів опцій у налаштуваннях;
- файли з каталогу “strings” містять опис рядкових літералів, що мають переклад на різні мови (зادля забезпечення можливості локалізації додатку).

Для більш гнучкої побудови інтерфейсу користувача було впроваджено пакет “uextensions” що містить ряд допоміжних сутностей:

- 1) StringHelper - використовується для формування локалізованих рядкових літералів з динамічними даними;
- 2) NotificationHelper - використовується для виведення інформаційних повідомлень системи у інтерфейс користувача (наприклад, про успішність виконання дій у системі (рис. 3.5));

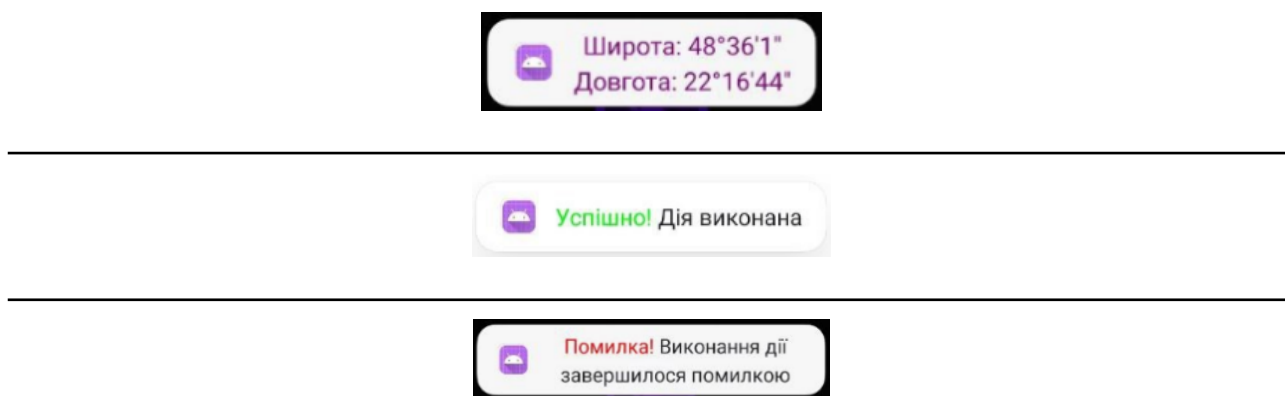


Рис. 3.5. Приклади відображення повідомлень: а) зверху - після проведення успішного визначення з відображенням знайдених географічних координат; б) посередині - успішне виконання дії (наприклад, очищення історії); в) знизу - повідомлення про помилку

3) StatusHelper - використовується для роботи зі статусами даних сесії (процес відправки електронних листів) в інтерфейсі користувача, зокрема, для отримання локалізованої назви статусу за ідентифікатором.

Імплементація основної логіки системи може включати платформу залежні компоненти. Їх впровадження у систему відбувається у пакеті “logicdependent”. Поточна реалізація включає компонент ручного пошуку об’єкта на зображенні. Реалізація відповідного менеджера базується на створенні нового представлення, де користувач може вибрати положення об’єкта (рис. 3.6).

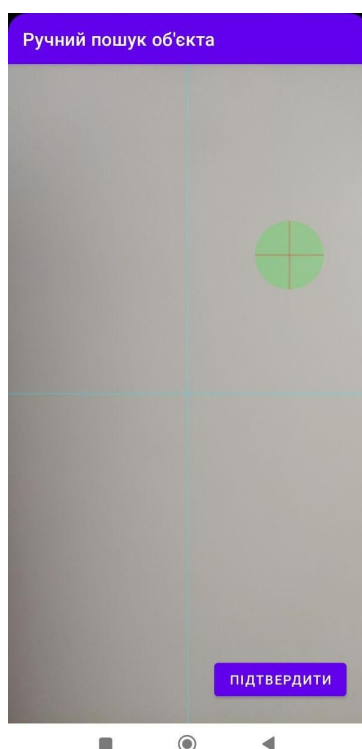


Рис. 3.6. Представлення для ручного пошуку об’єкта на зображенні

Використання ручного пошуку об’єкта не має чітких часових обмежень, тому для забезпечення взаємодії усіх компонентів логіки впроваджені функції зворотного виклику (про успішність та неуспішність виконання операції).

3.2.7. Опис основних взаємодій між компонентами

Архітектура програмного забезпечення передбачає певні правила взаємодії компонентів (див. рис. 3.1):

1) Запуск мобільного додатку супроводжується ініціалізацією менеджера взаємодії HorizonManager (пакет “managment”) та ініціалізацією необхідних сутностей рівня “infrastructure”: з визначенням реалізацій управління налаштуваннями; з отриманням доступу до сенсорів системи; зі створенням доступу до бази даних; зі збереженням важливих властивостей у відповідних сховищах.

2) Основна взаємодія користувача через графічний інтерфейс “ui” з компонентами системи (логіки, бази даних тощо) відбувається на основі пакету “managment”.

3) Усі дії пов’язані з внутрішніми компонентами системи проходять через виклик менеджера взаємодії (HorizonManager): виклик зовнішніх сервісів, початок та закінчення моніторингу показників сенсорів, визначення географічних координат тощо.

4) Менеджер взаємодії не використовує безпосередньо реалізації компонентів “logic”, а працює з “логікою” на основі класу-фабрики та опису доступних зовнішніх методів.

5) Рівень “infrastructure” надає доступ до:

- платформи-залежних реалізацій: отримання значень з панелі налаштувань, правил одержання компонентів логіки на основі налаштувань (реалізація класу-фабрики ServiceFactory);

- правил конвертації (сутностей бази даних у доменну модель та навпаки);

- реалізації менеджера взаємодії з зовнішнім сервісом;

- репозитаріїв доступу: ApplicationRepository - інформація функціонування додатку; HorizonRepository - інформація за налаштуваннями

логіки, доступ до сенсорів тощо; `UserRepository` - дані за поточним користувачем; `DatabaseRepository` - взаємодія системи з базою даних на основі доменної моделі (за необхідністю внутрішньо відбувається конвертація сутностей та виклик виконання певних дій у базі даних).

б) Сутності пакету “`domain`” використовуються по всій системі, крім модулю загальної логіки “`common`” та рівня безпосередньої взаємодії з базою даних “`database`”.

3.3. Побудова допоміжних модулів: комп’ютерне бачення на основі нейронних мереж

Технології, що застосовуються, при побудові моделей нейронних мереж:

- Python 3.7.* - версія, що підтримує пакет `tensorflow`;
- `numpy` - для роботи з великими масивами даних;
- `tensorflow` та модуль `keras` - бібліотека для машинного навчання та розробки штучних нейронних мереж;
- `sklearn` - застосовується на підготовчих етапах для подальшої зручності роботи системи, надає метри порівняння.

Для тренування нейронної мережі був сформований набір даних (датасет) за класами об’єктів (для Сонця, для Місяця):

- об’єкти легко визначаємі (великий масштаб об’єкту);
- відсутнє захмарення, чи перекриття іншими тілами;
- з датасету виключено усі зображення, на яких наявні перетини за класами.

Для збільшення розмірів датасету можливо використання автоматичної генерації даних (розроблені спеціальні компоненти): зашумлення існуючих зображень з випадковою силою, обертання навколо центральної точки, масштабування.

Після тренування моделі необхідно провести її збереження та конвертацію у необхідний формат, що буде використаний при побудові відповідних компонентів мобільного додатку (використання створює обмеження по реалізації на певних версіях операційної системи Android).

Підвищення успішності визначення в ускладнених ситуаціях можливо забезпечити шляхом збільшення розміру вибірки зображень (датасету) для тренування; врахування часткового перекриття небесних об'єктів предметами, рослинами, хмарами; впровадження класів, за якими хибно може бути визначений об'єкт. При визначенні за сузір'ями можливо розглядати певні опорні точки, на основі яких користувачу буде надана необхідна інформація.

3.4. Узагальнення результатів розробки

Архітектура мобільного додатку представлена набором односторонньо зв'язаних між собою модулів та побудована з урахуванням:

1) можливостей по розширенню функціональності системи: архітектура дозволяє впровадження та розробку нових методів і компонентів логіки з внесенням мінімальних змін у існуючому програмному коді (додавання нових реалізацій інструментарію тощо);

2) можливостей повторного використання програмного коду, компонентів: модульність архітектури дозволяє використовувати частину функціоналу повторно, в тому числі будувати на основі окремих модулів нові програмні системи;

3) можливостей для впровадження паралельного процесу розробки: архітектура дозволяє працювати незалежно над різними частинами інфраструктури: інтерфейс, база даних, логіка системи тощо; також, наявна можливість незалежної розробки окремих компонентів логіки: визначення положення мобільного пристрою, алгоритми комп'ютерного бачення, підвищення точності у рамках сесії тощо.

4) підтримки супроводжуваності програмного коду: програмний код є структурованим, найменування (змінних, інтерфейсів, класів) є інтуїтивно зрозумілими, наявні коментарі.

Покращення досвіду користування мобільним додатком можливо з:

- відображенням знайдених географічних координат на мапі за допомогою певного ключа API;
- реалізацією менеджерів підказок в інтерфейсі;
- впровадженням методів валідації у розроблені компоненти (наприклад, при використанні методів комп'ютерного бачення: користувач за допомогою фотокамери "сканує" небосхил, а система у цей час автоматично визначає яке положення мобільного пристрою відповідає правилам методики та буде використано у процесі геолокації).

Загалом, архітектура програмного забезпечення мобільного додатку відповідає поставленим на етапі проектування цілям і на її основі можливо проводити подальше удосконалення системи, зокрема логічних складових.

РОЗДІЛ 4

РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ ТА РОЗРОБКИ, ПЕРСПЕКТИВИ РОЗВИТКУ

4.1. Базові відомості для користувача мобільного додатку

Перший запуск мобільного додатку супроводжується запитом на надання доступу до камери та запису інформації у сховища даних. Користувачу представлено інтерфейс (див. рис. 3.4), у якому він може взаємодіяти з трьома основними панелями: Геолокація, Історія, Налаштування.

Перед початком проведення геолокації користувач може змінити налаштування алгоритмів та системи. За необхідністю вказати дані довіреної особи, на електронну пошту якої будуть надходити результати визначення географічних координат (за наявності доступу до мережі Інтернет).

Процес геолокації побудований на основі сесій, у рамках яких відбувається поступове покращення точності значень географічних координат за вказаними алгоритмами. Якщо користувач хоче розпочати нову сесію, він може викликати виконання відповідної дії у контекстному меню - “Зупинити сесію геолокації”.

Визначення географічних координат відбувається при проведенні фотозйомки за вказаними у роботі правилами. Для захоплення зображення небосхилу та запуску процесу визначення необхідно натиснути на круглу кнопку на панелі “Геолокація”. Виконання всіх дій супроводжується виведенням результату у спеціальних повідомленнях.

Користувач може переглянути історію даних за збереженими сесіями на відповідній панелі.

Додаток доступний на двох мовах - англійська та українська.

4.2. Галузь застосування

У порівнянні із сучасними методами геолокації представлена у роботі система не може використовуватися при вирішенні задач, що вимагають високої точності. Проте, вона дозволяє проводити уточнення даних основних систем та може стати пріоритетною при вирішенні певних спеціалізованих задач у випадках, коли відсутня потреба у високій точності.

Можливі варіанти застосування розробленого програмного додатку:

- у позаштатних ситуаціях за умови зникнення доступу до мережі GPS (системи глобального позиціонування);
- під час руху автомобіля по невідомій трасі;
- визначення географічного положення у морському плаванні.

При внесенні певних модифікацій у роботу системи (проведення необхідних розрахунків, перенесення на спеціалізовані мобільні пристрої, тощо) можливо реалізувати:

- систему відслідковування міграції тварин;
- додаткову бортову систему на літаку;
- систему навігації на інших планетах.

4.3. Узагальнення результатів досліджень та перспектива розвитку

У роботі представлена допоміжна (уточнююча) методика геолокації за небесними об'єктами. Універсальність методу полягає в тому, що його можна використовувати для визначення географічних координат за різними небесними об'єктами, зокрема Сонцем, Місяцем, полярною зіркою, сузір'ями тощо (проте це потребує моделювання руху відповідних астрономічних об'єктів). У роботі наведено підходи, що дозволяють зробити систему повністю незалежною від попереднього завантаження даних за опорними координатами.

Архітектура мобільного додатку представлена набором модулів, що дозволяє полегшити процес впровадження нового функціоналу у систему (масштабування) та працювати над окремими компонентами незалежно.

Система накопичує низку похибок, проте у екстрених ситуаціях її застосування може мати вирішальне значення для користувача. Представлені методи підвищення точності за різних умов.

Основні напрямки подальших досліджень:

- уточнення математичної моделі сонячної системи, врахування особливостей земної атмосфери;
- визначення висоти над рівнем моря на основі сенсору тиску;
- підвищення точності роботи методів комп'ютерного зору;
- підвищення зручності використання системи користувачем - автоматичне захоплення зображення, використання різних камер мобільного пристрою.

ВИСНОВКИ

З початком використанням глобальної системи позиціонування навігація на основі положення небесних тіл стала менше використовуватись серед населення. Проте, проведений аналіз свідчить про необхідність розробки додаткових систем геолокації. У ясні дні астрономічні об'єкти забезпечують гарний орієнтир для визначення географічного місцезнаходження. Мореходці використовували це явище за допомогою секстанта та отримували невеликі відхилення від дійсних значень. Подібний до цього метод застосований при розробці системи у даній роботі.

Практичне значення базується на:

- відносно простій адаптації та можливості впровадження математичної моделі у якості прикладної системи для роботи на неспеціалізованих пристроях загального користування (смартфонах);
- розширенні можливостей мобільних пристроїв щодо визначення географічних координат в позаштатних ситуаціях (зникнення доступу до стандартної мережі GPS тощо).

Розроблене програмне забезпечення дозволяє визначати приблизне географічне положення неспеціалізованого мобільного пристрою під управлінням платформи Android на базі штатних сенсорів в умовах зникнення доступу до мережі GPS та Інтернет.

Система накопичує низку похибок та не може використовуватися як основна, проте у екстрених ситуаціях її застосування може мати вирішальне значення для користувача. Запропоновані методи уточнення при роботі системи. Розглянуто використання системи в ускладнених умовах.

Окремо, слід обумовити універсальність застосованого методу, що дозволяє при корекції певних параметрів визначати координати за різними небесними об'єктами (Сонце, Місяць, полярна зірка, різні сузір'я тощо).

Напрямки подальших досліджень: уточнення математичної моделі руху астрономічних об'єктів; визначення висоти над рівнем моря на основі сенсору тиску; підвищення точності роботи методів комп'ютерного зору.

Одержані в даному дослідженні результати представлені та схвалені на другому турі Всеукраїнського конкурсу студентських наукових робіт зі спеціальності "Комп'ютерні науки". Захист проходив дистанційно 21 квітня 2021 року, робота посіла друге місце.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Gómez J. V., Sandnes F. E., Fernández B., Sunlight Intensity Based Global Positioning System for Near-Surface Underwater Sensors. // *Sensors*, 2012. - 1930–1949с.
2. Lisovski S., Hewson C. M., Klaassen R. H. G., Korner-Nievergelt F., Kristensen M. W., Hahn S, Geolocation by light: accuracy and precision affected by environmental factors // *Methods in Ecology and Evolution*, 2012. - 603–612с.
3. Delong, R.L., Steward, B.S., Hill, R.D., Documenting migrations of northern elephant seals using day length // *Marine Mammal Science*, 8, 1992. - 155–159с.
4. Sandnes F. E., Determining the Geographical Location of Image Scenes based on Object Shadow Lengths. // *Journal of Signal Processing Systems*, 2010. - 35-47с.
5. Sandnes F. E., Towards calibration-free geo-localization of stationary outdoor webcams // *NIK conference*, 2010. - 12с.
6. Galina S. Panayotova, Georgi P. Dimitrov, Dimitar A. Dimitrov, Wireless Sensors for analysis transport systems. - Florence, Italy // 10th International Conference on Computer Science and Information Technology (ICCSIT 2017), ICCSIT 2017 in Florence Italy, October 23-25, 2017.
7. У Карпатах третій день шукають зниклого в снігах киянина. Зв'язку з ним немає - BBC News | Україна. – [Електронний ресурс]. – Режим доступу: <https://www.bbc.com/ukrainian/news-56069047>
8. Грибника, який загубився в лісі на Тернопільщині, знайшли мертвим – ПроЗахід. – [Електронний ресурс]. – Режим доступу: <https://prozahid.com/content-52061-html>
9. Українські військові врятували миротворця ООН у джунглях Конго: фото – 24 канал. – [Електронний ресурс]. – Режим доступу:

https://24tv.ua/ukrayinski_viyskovi_vryatuvati_mirotvortsya_oon_u_dzhunglyah_ko_ngo_foto_n1075334

10. Выборнов А.А. Оптико-электронные приборы астроориентации и навигации космических аппаратов: учебное пособие; Южный федеральный университет. Таганрог: Издательство южного федерального университета. Ростов-на-Дону, 2018. - 120 с.

11. Федосеев В.И., Колосов М.П. Оптико-электронные приборы ориентации и навигации космических аппаратов: учебное пособие. М.: Логос. 2007. - 248 с.

12. Зямзин Е.Д., Кожеуров В.Н. Определение состава и структурной схемы бортового комплекса управления для осуществления посадки космического аппарата. Пояснительная записка к выпускной квалификационной работе. Челябинск, 2017. - 63 с.

13. Shevchenko V.L. Optimization Modeling in Strategic Planning. - К.: CVSD NUOU, 2011. – 283р.

14. What is GPS? - Official U.S. government information about the Global Positioning System (GPS) and related topics. – [Электронный ресурс]. – Режим доступа: <https://www.gps.gov/systems/gps/>

15. Shevchenko V. Dynamic Objects Emergency State Monitoring by Means of Smartphone Dynamic Data / Shevchenko A., Bychkov O., Shevchenko V. // 2017 14-th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM). Proceeding. - Polyana, February 21-25, 2017. - p.292-294. DOI: 10.1109/CADSM.2017.7916138.

16. Черный М. А. Авиационная астрономия. М.: Транспорт. 1978. - 208 с.

17. Celestial sphere - Wikipedia, the free encyclopedia. – [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Celestial_sphere

18. Horizontal coordinate system - Wikipedia, the free encyclopedia. – [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Horizontal_coordinate_system
19. SensorEvent - developers. – [Електронний ресурс]. – Режим доступу: <https://developer.android.com/reference/android/hardware/SensorEvent>
20. Computing planetary positions - a tutorial with worked examples - Paul Schlyter, Stockholm, Sweden. – [Електронний ресурс]. – Режим доступу: <http://stjarnhimlen.se/comp/tutorial.html>
21. How to compute planetary positions - Paul Schlyter, Stockholm, Sweden. – [Електронний ресурс]. – Режим доступу: <http://stjarnhimlen.se/comp/ppcomp.html>
22. Sensor Coordinate System - developers. – [Електронний ресурс]. – Режим доступу: https://developer.android.com/guide/topics/sensors/sensors_overview#sensors-coords
23. Convolutional Neural Networks from the ground up - towards data science. – [Електронний ресурс]. – Режим доступу: <https://towardsdatascience.com/convolutional-neural-networks-from-the-ground-up-c67bb41454e1>
24. Intuitively Understanding Convolutions for Deep Learning - towards data science. – [Електронний ресурс]. – Режим доступу: <https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1>
25. Ситніков О.В. Числові методи. Конспект лекцій. Навчальний посібник. КПІ ім. Ігоря Сікорського. Київ, 2019. - 165 с.
26. Kalman filter - Wikipedia, the free encyclopedia. – [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Kalman_filter
27. «Солнечные камни» викингов или как ходить в море без компаса - Feldgrau.info. Военно-исторический сайт. – [Електронний ресурс]. – Режим доступу:

<https://feldgrau.info/2010-09-02-14-48-28/76-uncategorised/19591-solnechnye-kamn-i-vikingov-ili-kak-khodit-v-more-bez-kompasa>

28. Horvath, G., Barta, A., Pomozi, I., Suhai, B., Hegedus, R., Akesson, S., Wehner, R., On the trail of Vikings with polarized skylight: experimental study of the atmospheric optical prerequisites allowing polarimetric navigation by Viking seafarers. - // Philosophical Transactions of the Royal Society B: Biological Sciences, 2011. - 772–782с.

29. Iceland spar - Wikipedia, the free encyclopedia. – [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Iceland_spar

30. A depolarizer as a possible precise sunstone for Viking navigation by polarized skylight - The royal society. Publishing. – [Электронный ресурс]. – Режим доступа: <https://royalsocietypublishing.org/doi/full/10.1098/rspa.2011.0369>

31. Department of air force, Air Navigation (Air Force Pamphlet 11-216 1 MARCH 2001 Flying Operations). - // Publisher: CreateSpace Independent Publishing Platform, 2013. - 428с.

ДОДАТКИ

Додаток А

Диплом II ступеня (Всеукраїнський конкурс студентських наукових робіт зі спеціальності «Комп'ютерні науки» в 2020/2021 навчальному році)



Додаток Б

Акт впровадження результатів дослідження №1

ЗАТВЕРДЖУЮ
Директор Державного підприємства
Український центр
розвитку інноваційних технологій
Міністерства освіти та науки України



В.Б.ПОЛІЩУК

АКТ

впровадження результатів дослідження
студента Київського національного університету імені Тараса Шевченка
Табунова Антона Анатолійовича
на тему:

“Автономна Android-система визначення координат та поступового нарощування їх точності в умовах відсутності радіосигналів”

Комісія у складі:

голова комісії – Директор ДП УкрНЦ РІТ МОНУ Поліщук В.Б.;

члени комісії:

Провідний науковий співробітник Нетесін І.Є.,

Головний інженер Коконцев С.О.

Встановила та цим актом засвідчує, що нижчеперелічені результати досліджень:

1. Запропонована модифікація метода співставлення поточного стану математичної моделі руху Землі з положенням проекції Сонця на фотознімку на основі просторової орієнтації з урахуванням реалізації на НМІІ телекомунікацій.
2. Розроблена технологія обчислення приблизного географічного положення побутового мобільного пристрою на базі штатних сенсорів з урахуванням підтримки безперервної роботи в умовах зникнення доступу до мережі GPS та Інтернет.
3. Дослідження, щодо роботи системи в складних ситуаціях, зокрема у випадку підвищеної хмарності.

впроваджені в програмі перспективних інновацій ДП УкрНЦ РІТ МОН України в розділі «прототипи зразків програмних та програмно-апаратних комплексів для перспективних стартапів».

Голова комісії
кандидат технічних наук
Члени комісії
Кандидат фізико-математичних наук

В.Б.ПОЛІЩУК

І.Є.НЕТЕСІН
С.О.КОКОНЦЕВ

17 грудня 2020 року

Додаток В
Акт впровадження результатів дослідження №2

АКТ
впровадження результатів дослідження
студента Київського національного університету імені Тараса Шевченка
Табунова Антона Анатолійовича
на тему:
“Програмне забезпечення визначення координат за допомогою сенсорів смартфона
без використання GPS”

Акт засвідчує, що нижчеперелічені результати досліджень:

1. Запропонована модифікація математичної моделі з урахуванням її реалізації на неспеціалізованих мобільних пристроях телекомунікацій та з урахуванням наявного набору вбудованих сенсорів.
2. Вдосконалена математична модель обробки інформації, що отримана з сенсорів неспеціалізованого мобільного пристрою та знімку небосхила, за рахунок перетворення даних у зручний формат, який дозволяє провести співставлення результатів зазначених обчислень з даними моделі руху планети Земля навколо Сонця.

впроваджені при розробці системи визначення координат за допомогою сенсорів смартфона без використання GPS.

Головний інженер,
ДП «Антонов»

Начальник управління
програмного технічного



С.А.Бичков

Ю.Р.Бойко

Додаток Г

Документування архітектури розробленого програмного забезпечення у відповідності до стандарту Software Architecture Document

Taras Shevchenko National University of Kyiv Faculty of Information Technology

Horizon Software Architecture Document (SAD)

CONTENT OWNER: Tabunov Anton

DOCUMENT NUMBER:

1.0

RELEASE/REVISION:

Horizon ANDROID-app 1.0

RELEASE/REVISION DATE:

18.04.2022

Table of Contents

1	Documentation Roadmap	4
1.1	Document Management and Configuration Control Information	4
1.2	Purpose and Scope of the SAD	4
1.3	Viewpoint Definitions	4
2	Architecture Background	5
2.1	Problem background	5
	2.1.1 System Overview	5
	2.1.2 Goals and Context	5
	2.1.3 Significant Driving Requirements	5
2.2	Solution Background	6
	2.2.1 Architectural Approaches	6
	2.2.2 Analysis Results	8
	2.2.3 Requirements Coverage	8
3	Referenced Materials	9
4	Directory	10
4.1	Glossary	10
4.2	Acronym List	10
5	Samples	11

List of Figures

Figure 1: Package diagram	6
Figure 2: Database diagram	7
Figure 3: Architecturally significant use-case diagram	9
Figure 4: Sample panel “Geolocation”	11
Figure 5: Sample panel “Settings”	11

1. Documentation Roadmap

1.1. Document Management and Configuration Control Information

- Revision Number: 1.0
- Revision Release Date: 18.04.2022
- Purpose of Revision: This Software Architecture Document provides a basic architectural overview of the ANDROID System.
- Scope of Revision: This document is designed to cover and convey the significant architectural decisions that have been made in the system. This document includes an architecture overview, database overview and use-cases overview.

1.2. Purpose and Scope of the SAD

This SAD specifies the software architecture for Horizon ANDROID-app 1.0. All information regarding the software architecture may be found in this document, although much information is incorporated by reference to other documents.

Software architecture has emerged as an important subdiscipline of software engineering, especially in the field of large system development.

The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them. [Bass 1998]. Architecture gives us intellectual control over the complex, allowing us to replace the complex with a set of interacting pieces, each one of which is much simpler than the whole.

1.3. Viewpoint Definitions

The SAD employs a stakeholder-focused, multiple view approach to architecture documentation, as required by ANSI/IEEE 1471-2000, the recommended best practice for documenting the architecture of software-intensive systems [IEEE 1471].

A view is the specification of one or more of these structures, and documenting a software architecture, then, is a matter of documenting the relevant views and then documenting information that applies to more than one view [Clements 2002].

ANSI/IEEE 1471-2000 provides guidance for choosing the best set of views to document, by bringing stakeholder interests to bear. It prescribes defining a set of viewpoints to satisfy the stakeholder community. A viewpoint identifies the set of concerns to be addressed and identifies the modeling techniques, evaluation techniques, consistency checking techniques, etc., used by any conforming view. A view, then, is a viewpoint applied to a system. It is a representation of a set of software elements, their properties, and the relationships among them that conform to a defining viewpoint. Together, the chosen set of views shows the entire architecture and its relevant properties. A SAD contains the viewpoints, relevant views, and information that applies to more than one view to give a holistic description of the system.

2. Architecture Background

2.1. Problem background

2.1.1 System Overview

The software is designed to expand the capability of non-specialized mobile devices to determine the geographical coordinates in an emergency situation when access to the standard GPS network has been lost. The software allows calculating the approximate geographical location of a NSMD running the Android operating system on the basis of standard sensors. The model is based on comparing the position of a celestial body's projection on the photograph from the current state of the Earth's motion model with the help of spatial orientation, which standard sensors of the mobile device can provide.

To start the process of calculating GC, you need to take some photos according to the rules. At the moment when a photograph of the sky is taken, the necessary time data and the information from sensors are read. The software displays the approximate GC of the user (the latitude and the longitude) as a message after taking photos successfully.

2.1.2 Goals and Context

Properly constructed architecture can:

- facilitate the programming process;
- divide the work on the development of various components of the system;
- carry out independent of other modules of improvement;
- reuse different modules when developing other software.

2.1.3 Significant Driving Requirements

A non-specialized mobile device must be equipped with:

- An accurate clock (to determine the date and time in Greenwich Mean Time).
- A camera for photography.
- A set of sensors (an accelerometer, a magnetic field sensor).

Most mobile devices have built-in digital watches that can run properly on a single battery for many years. So we can assume that it is possible to get the value of the current time and date at any time. The system will use the time set by Greenwich (UTC), which allows for avoiding some calculations, such as the transition to summertime [Sensors 2012].

2.2. Solution Background

2.2.1 Architectural Approaches

The architecture of the mobile application is represented by a set of modules, each of which is described in the corresponding package (Figure 1).

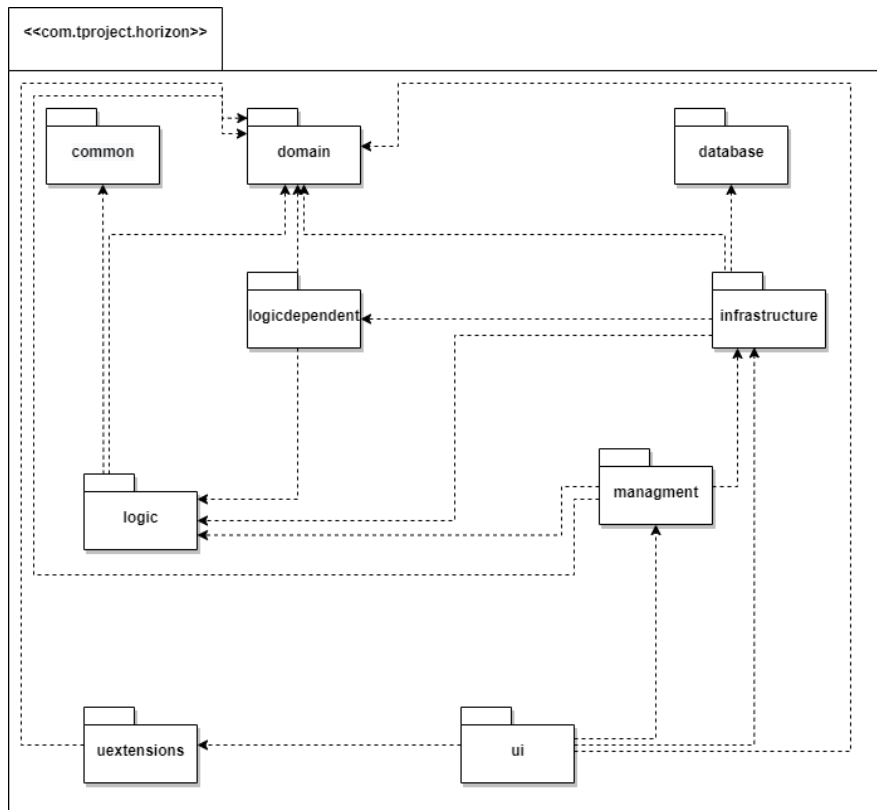


Figure 1: Package diagram

1) Package “common”. The module contains the implementation of common domains that can be used in the development of other systems. For example, a description of logic for working with angles in different formats (degrees, radians).

2) Package “domain”. The module is a description of domain objects: models, callback functions, and a description of possible system settings.

3) Package “database”. The module provides a description of the interaction with the DB of the mobile application. The package describes the DB scheme: tables (entities) and relationships between them, views; queries to the database (adding new records, deleting, updating, etc.). The module describes conversions to work with specific objects in a convenient format (such as dates).

4) Package “logic”. The module is a description of the main logic of the system (determination of GCs) using a set of tools and implementations.

5) Package "ui". The module contains a program description of the user interface.

6) Package "uextensions". Generalized extensions for working with the user interface: functionality for displaying messages of different types - about an error, a successful action, etc .; generation of localized string literals with dynamic information; a description of the definitions of the methods (interfaces) used in the program code of the user interface.

7) Package “logicdependent”. A module that represents additional implementations of tools to the "logic" package, which are focused on use only under the control of the Android OS.

8) Package “infrastructure”. The module describes the layer between the platform and the basic logic of the system, which includes: repositories for data storage, platform-dependent implementations (for example, obtaining settings and tools), a description of converting database entities to domain objects (and vice versa), managers (for example, for calling external services), etc.

9) Package “managment”. The module represents a layer of capabilities (actions) to control the basic logic of the system from the user interface. In particular, the module includes: setting the listeners to the values of the sensors (start them, turn them off, get values); launching and supporting the geolocation process; calling the external communications manager.

The database schema includes objects according to figure 2. The database package contains a description of data access objects, which allows you to separate the logic of the database from other components of the system.

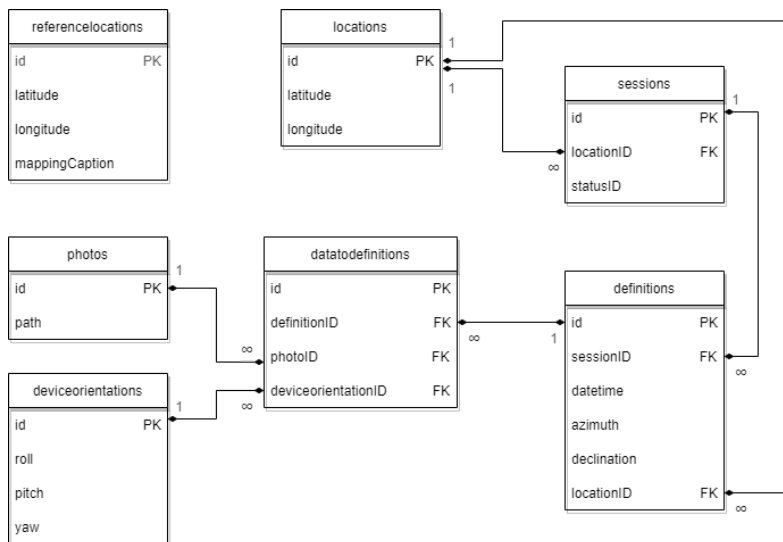


Figure 2: Database diagram

The description of the basic logic and algorithms required for the geolocation process is given in the "logic" package. Each individual component of “logic” is built in accordance with certain concepts:

- there is a unified description of the main functions for interaction with the component (in the form of an interface);
- a set of implementations of these methods is presented (classes that implement the interface);
- the Factory design pattern is used to access various component implementations.

The package describes the “factory”-methods based on the abstract class. When the module is initialized, the external implementation of the “factory”-class is recorded in special internal storage.

The software architecture provides certain rules for the interaction of components:

1) The launch of the mobile application is accompanied by the initialization of the HorizonManager interaction manager (“management” package) and the initialization of the necessary entities of the “infrastructure” level: with the definition of settings management implementations; with access to system sensors; with the creation of access to the database; with preservation of important properties in the corresponding storages.

2) The main interaction of the user through the graphical interface "ui" with the components of the system (logic, database, etc.) is based on the package "managment".

3) All actions related to internal components of the system go through a call to the interaction manager (HorizonManager): call external APIs, start and end monitoring of sensor indicators, determination of geographical coordinates, etc.

4) The interaction manager does not directly use the implementation of "logic" components, but works with "logic" based on the "factory"-class and the description of available external methods.

5) The "infrastructure" level provides access to platform-dependent implementations; rules of conversion; implementation of the manager of interaction with external services; access repositories.

6) Domain entities are used throughout the system, except for the "common"-logic module and the level of direct interaction with the database (package "database").

2.2.2 Analysis Results

The architecture of the developed application consists of modules that allow making it easier to scale and change the system. The main components that implement the logic of the system are separated into independent modules, which allows the developers to work at them independently.

The database package contains a description of data access objects, which allows you to separate the logic of the database from other components of the system.

Each individual component of "logic" is built in accordance with certain concepts that allow you to expand the functionality of the system more flexibly. It is possible to create new logic component implementations without changing the previous ones.

2.2.3 Requirements Coverage

During the development of the mobile application, in order to ensure user interaction with the system, three main representations were introduced:

1) Geolocation panel (figure 5 in samples). The view provides the user with access to the elements necessary for easy geolocation according to the algorithm presented in the paper.

2) History panel (figure 6 in samples). The view displays the history of geolocation sessions (as a list) that were conducted and saved to the database.

3) Settings panel. The view displays a list of possible algorithm and system settings that need to be changed by the user.

In general, a number of basic actions are available to the user (figure 3):

- 1) put necessary data (photo, sensor values, etc) to the geolocation module;
- 2) get the result of geolocation;
- 3) download location reference data;
- 4) view the history of geolocations;
- 5) change settings of the algorithms.

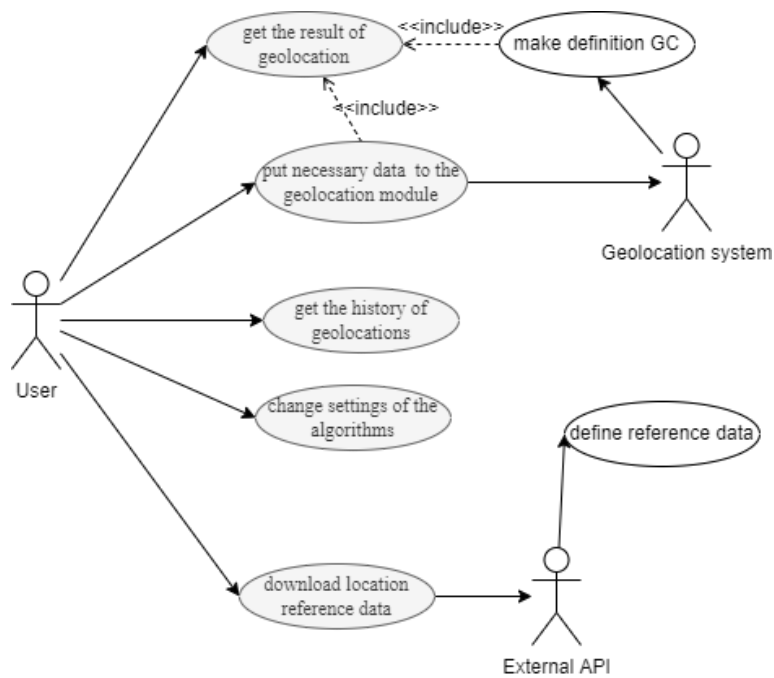


Figure 3: Architecturally significant use-case diagram

The database was implemented for a mobile application in order to improve the experience of using the system. Its implementation allows you to pre-download and store all the necessary data about the location where the user will be (in case of the absence of GPS and radio signal), which allows not to keep all the key information in the mobile device. In addition, basic data on the geolocation process is stored in a database, which allows you to perform various actions on them, including reviewing the history of previous geolocations.

3. Referenced Materials

[Bass 1998]	Bass, L.; Clements, P.; & Kazman, R. Software Architecture in Practice. Reading, MA: Addison-Wesley Longman, 1998.
[IEEE 1471]	ANSI/IEEE-1471-2000, IEEE Recommended Practice for Architectural Description of Software-Intensive Systems, 21 September 2000.
[Clements 2002]	Clements, Kazman, Klein, Evaluating Software Architectures: Methods and Case Studies, Addison Wesley Longman, 2001.
[Sensors 2012]	Gómez J. V., Sandnes F. E., Fernández B., “Sunlight Intensity Based Global Positioning System for Near-Surface Underwater Sensors”, Sensors, 2012, pp. 1930–1949.

4. Directory

4.1. Glossary

Term	Definition
software architecture	The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them. By “externally visible” properties, we are referring to those assumptions other components can make of a component, such as its provided services, performance characteristics, fault handling, shared resource usage, and so on [Bass 1998]
package	A package is a namespace that organizes a set of related classes and interfaces.

4.2. Acronym List

NSMD	Non-specialized mobile device
OS	Operating System
GC	Geographical coordinates
API	Application Program Interface
DB	Database

5. Samples

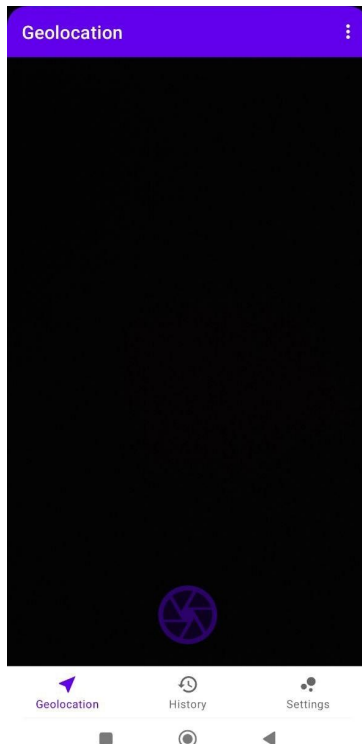


Figure 4: Sample panel “Geolocation”

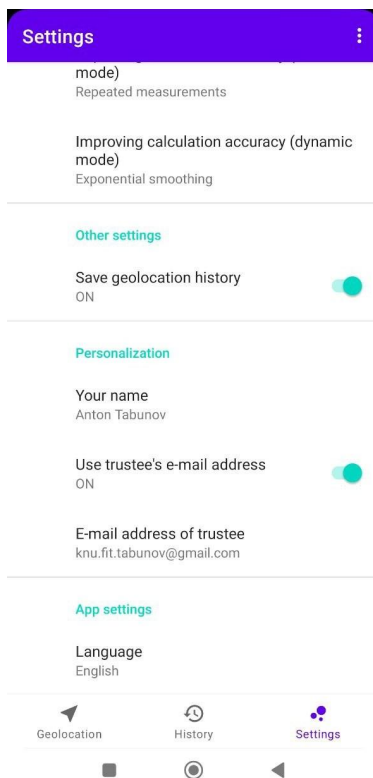


Figure 5: Sample panel “Settings”

Додаток Д

Витяг програмного коду: реалізація менеджерів сесії

```
public class BaseSessionManager implements ISessionManager {
    protected IDefinitionManager _DefinitionManager;
    ...
    protected final Session _Session;

    protected boolean isSessionClosed = false;

    public BaseSessionManager(){
        this._Session = new Session();
        createDefinitionManager();
    }

    @Override
    public void register(Bitmap bitmap, SensorObject sensorObject, Date datetime,
        ICallbackFunction successCallBack,
        ICallbackFunction failCallBack) throws Exception {
        definitionCompleted = false;
        _DefinitionManager.register(bitmap, sensorObject, datetime, new ICallbackFunction() {
            @Override
            public void onCompleted() {
                if(definitionCompleted = _DefinitionManager.isDefinitionCompleted()){
                    updateSessionData(_DefinitionManager.getData());
                    createDefinitionManager();
                    updateSessionClosed();
                }
                successCallBack.onCompleted();
            }
        }, new ICallbackFunction() {
            @Override
            public void onCompleted() {
                failCallBack.onCompleted();
            }
        });
    }

    @Override
    public boolean isSessionClosed(){... }

    @Override
    public boolean isDefinitionCompleted(){...}

    @Override
    public Session getSessionData(){... }

    private void createDefinitionManager(){
        _DefinitionManager = ServiceFactory.getServiceFactory().getDefinitionManager(this._Session);
    }

    protected void updateSessionData(Definition definition){
        if(definition != null && definition.getLocation() != null){
            this._Session.addDefinition(definition);
            updateLocationData();
        }
    }
}
```

```

protected void updateLocationData() {
    Definition actualDefinition = this._Session.getDefinitions().get(0);
    _Session.setLocation(actualDefinition.getLocation());
}

protected void updateSessionClosed(){
    isSessionClosed = true;
}

}

public class StaticSessionManager extends BaseSessionManager{
    public StaticSessionManager(){
        super();
    }

    @Override
    protected void updateLocationData() {
        if (this._Session.getDefinitions().size() > 0) {
            float longitude_summary = 0;
            float latitude_summary = 0;
            for (Definition definition : this._Session.getDefinitions()) {
                longitude_summary += definition.getLocation().getLongitude();
                latitude_summary += definition.getLocation().getLatitude();
            }

            float longitude = longitude_summary / this._Session.getDefinitions().size();
            float latitude = latitude_summary / this._Session.getDefinitions().size();

            _Session.setLocation(new Location(latitude, longitude));
        }
    }

    @Override
    protected void updateSessionClosed(){ }
}

```

Додаток Е

Витяг програмного коду: реалізація менеджера визначення за одним зображенням

```
public class SingleDefinitionManager implements IDefinitionManager {
    ...

    private Definition _Definition = null;

    public SingleDefinitionManager(Session session){
        this._Definition = new Definition(session);
    }

    @Override
    public void register(Bitmap bitmap, SensorObject sensorObject, Date datetime,
        ICallbackFunction successCallBack,
        ICallbackFunction failCallBack) throws Exception {
        ServiceFactory.getServiceFactory().getComputerVisionManager().defineObjectPosition(bitmap, new
IComputerVisionCallBack() {
        @Override
        public void onCompleted(Integer x, Integer y) {
            try{
                float ref_x = bitmap.getWidth() / 2;
                float ref_y = bitmap.getHeight() / 2;
                if((x == null || y == null)
                    || (x < ref_x - object_position_error) || (x > ref_x + object_position_error)
                    || (y < ref_y - object_position_error) || (y > ref_y + object_position_error)){
                    throw new Exception(...);
                }

                float[] deviceOrientation =
                    ServiceFactory.getServiceFactory().getDeviceOrientationManager().getAngles(sensorObject);

                if(deviceOrientation[2] > device_orientation_roll_max &&
                    deviceOrientation[2] < (360 - device_orientation_roll_max)){
                    throw new Exception(...);
                }
            }
            ...
            Float latitude = null;
            Float longitude = null;

            double min_val = Double.MAX_VALUE;
            for (ReferenceLocation refLocation:
                ServiceRepository.getReferenceLocations()) {
                float[] result = ServiceFactory.getServiceFactory().getSolarSystemManager().define(
                    refLocation.getLatitude(),
                    refLocation.getLongitude(),
                    0,
                    datetime);

                float diff_azimuth = result[0] - azimuth;
                float diff_declination = result[1] - declination;

                double val = diff_azimuth * diff_azimuth + diff_declination * diff_declination;

                if(val < min_val){
                    min_val = val;
                    latitude = refLocation.getLatitude();
                }
            }
        }
    }
}
```

```

        longitude = refLocation.getLongitude(); } }

boolean isDefinitionSucceeded = latitude != null && longitude != null;

if(isDefinitionSucceeded){
    _Definition.addDefinitionData(new DefinitionData(_Definition,
        bitmap,
        deviceOrientation[2], deviceOrientation[1], deviceOrientation[0]));
    _Definition.setDatetime(datetime);

    _Definition.setAzimuth(azimuth);
    _Definition.setDeclination(declination);

    _Definition.setLocation(new Location(latitude, longitude));
}
else{
    throw new Exception();
}

    successCallBack.onCompleted();
}
catch (Exception e){
    failCallBack.onCompleted();
}
}
});
}

@Override
public boolean isDefinitionCompleted() {
    return _Definition.getLocation() != null;
}

@Override
public Definition getData() {
    return _Definition;
}
}
}

```