Всеукраїнській конкурс студентських наукових робіт
з природничих, технічних та гуманітарних наук

Спеціальність: Інформаційні системи і технології

Конкурсна робота

Шифр роботи «Сонце»

ПЗ визначення координат за допомогою сенсорів смартфону без використання GPS

2019-2020 навчальний рік

Зміст

Анотація

Актуальність роботи

Люди занадто звикли до щоденного використання смартфонів. Проста відмова мережи може викликати глобальний хаос і паніку. Питання забезпечення людей додатковими джерелами інформації щодо географічних координат в умовах зникнення зв’язку із сотовою мережею та системою глобального позиціонування є **актуальною** задачею гарантування стійкого розвитку людства за умов будь-яких несподіваних ситуацій.

Окрім того, щоб покращити точність результату роботи основних систем геолокації, можливо застосування допоміжних даних, які отримані за допомогою інших методів.

Мета наукової роботи

Розширити можливості звичайних смартфонів (неспеціалізованих телекомунікаційних пристроїв) щодо визначення географічних координат в нештатній ситуації, в умовах зникнення доступу до стандартної мережі GPS (системи глобального позиціонування) та зв’язку із сотовою мережею.

Об’єкт дослідження - зображення небосхилу, датчики; рух об’єктів сонячної системи та їх взаємодія, розміщення по відношенню один до одного.

Предмет досліджень - технології та методи обробки зображень; технології отримання показань сенсорів та методи їх обробки; модель руху сонячної системи та визначення кутів падіння сонячних променів на поверхню Землі.

Завдання дослідження, які дозволяють досягти поставленої мети

1. Зчитування показників сенсорів з неспеціалізованих мобільних пристроїв під управлінням платформи Android.
2. Отримання мета-інформації фотознімку небосхилу.
3. Обробка отриманих даних: показників сенсорів, метаінформації.
4. Обчислення даних контрольних точок на основі моделювання руху Землі навколо Сонця.

5. Співставлення результатів обробки показників сенсорів та метаінформації з обчисленими даними на контрольних точках (визначення приблизних географічних координат місця, де проводилась фотозйомка).

При виконанні досліджень та прийнятті рішень використовувались такі **підходи та методи**: теорія кватерніонів, теорія матричного аналізу, методи чисельних обчислень, методи моделювання руху Землі навколо Сонця, засновані на ньютонівській взаємодії двох тіл.

Наукова новизна

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.

Практична значущість

1. Математична модель та її реалізація дозволяє легку адаптацію рішення під будь-які неспеціалізовані мобільні пристрої.

2. Розроблене універсальне програмне забезпечення дозволяє обчислювати приблизне географічне положення неспеціалізованого (побутового) мобільного пристрою на базі штатних сенсорів в умовах зникнення доступу до мережі GPS та Інтернет (на основі метаданих, які були попередньо завантажені).

3. Основна версія програмного забезпечення реалізована для використання на неспеціалізованих мобільних пристроях під управлінням платформи Android. Реалізація виконувалася з використанням фреймворку

Xamarin, що містить достатньо програмних інструментів для подальшого вдосконалення системи, додавання до неї нових функцій та легкого переносу під різні платформи.

Стислий опис результатів дослідження: розроблено метод співставлення поточного стану моделі руху Землі з положенням проекції Сонця на фотознімку на основі просторової орієнтації. Система надає можливість: уточнення даних основних систем (наприклад, глобальної системи позиціонування); створення дешевих систем для ситуацій, коли точність методу буде задовільною.

Робота має обсяг 27 аркушів основного тексту, має 9 малюнків, 2 таблиці, 17 джерела посилань, додатки на 3 аркушах.

Ключові слова: геолокація за сонцем, географічні координати, сенсори, орієнтація у просторі.

Вступ

Геолокація - визначення реального місцезнаходження (географічних координат) електронного пристрою [1]. Питання забезпечення людей додатковими джерелами інформації щодо географічних координат в умовах зникнення зв’язку із сотовою мережею та системою глобального позиціонування є *актуальною* задачею гарантування стійкого розвитку людства за умов настання несподіваних ситуацій.

Також, необхідно мати можливість точно визначити географічні координати об’єкта (наприклад, для управління польотом літака). Саме тому не завжди достатньо застосовувати лише один спосіб і ситуація вимагає комбінацію та усереднення даних, отриманих за допомогою різних методів. Запропонований у роботі метод є допоміжним і, коли задача вимагає високої точності, може застосовуватись лише для покращення результатів роботи основних систем геопозиціонування.

Але слід зазначити, що, наприклад, при русі автомобіля незнайомою трасою, чи для відслідковування міграції тварин, коли геопозицію можна визначити приблизно, геолокація на основі Сонця стане дешевим вирішенням поставленої задачі.

Також, в окремих випадках неприпустима затримка в декілька хвилин для отримання коректної інформації, як це інколи відбувається при використанні системи глобального позиціонування (зі спутників) [2].

1. Аналіз існуючих рішень, постановка задачі та опис можливих варіантів застосування

Існує велика кількість ситуацій, коли необхідно мати можливість точно визначити географічні координати об’єкта (наприклад, для управління польотом літака). Саме тому не завжди достатньо застосовувати лише один спосіб і ситуація вимагає комбінацію та усереднення даних, отриманих за допомогою різних методів.

Також, визначення географічного положення в нештатних ситуаціях (зникнення доступу до стандартної мережі GPS (системи глобального позиціонування), мережі Інтернет, зв’язку із сотовою мережею) є актуальною задачею, наприклад, при русі автомобіля незнайомою територією, чи в умовах ведення бойових дій.

В якості додаткового засобу визначення координат можна використовувати інформацію про розташування сонця та зірок на небі. Відомі технічні рішення найчастіше використовують в якості бортових систем [3-5]. **Недоліками** таких систем є громіздкість, вимога щодо розташування на борту рухомого засобу, велика вартість, енергоспоживання та недоступність для пересічних громадян.

Постановка задачі: необхідно розробити алгоритм роботи системи визначення географічних координат на основі фотозйомки небосхилу та показників сенсорів, що вбудовані в середньостатистичний смартфон (неспеціалізований мобільний пристрій). Результатом обчислень повинен бути діапазон координат, де орієнтовно знаходиться пристрій (користувач).

Система може застосовуватися у ситуаціях, коли точність методики є задовільною (наприклад, відстеження міграції тварин); для уточнення даних основних систем (наприклад, глобальної системи позиціонування). Враховуючи вартість обладнання глобальних навігаційних систем, при модифікації

(проведенні необхідних розрахунків), можливе застосування для навігації на інших планетах.

2. Вирішення поставленої задачі

2.1. Припущення при вирішенні поставленої задачі

Неспеціалізований мобільний пристрій має бути оснащений: точним годинником (для визначення дати та часу за Гринвічем); камерою; набором сенсорів (акселерометр, датчик магнітного поля, гіроскоп (опціонально, для більшої точності)).

Цифрові годинники, що вбудовані у більшість мобільних девайсів, можуть працювати протягом багатьох років від однієї батареї. Тому можна припустити, що у будь-який момент часу можливо отримати поточний час та дату. Час, що встановлений за Гринвічем (UTC), спрощує обчислення, бо відпадають декілька проблем, наприклад урахування літнього часу [6].

Передбачається, що система буде застосовувати подібний до секстанта (оптичний прилад для вимірювання величини кута між двома видимими об'єктами [7]) принцип дії. Принцип дії авіаційного секстанта передбачає визначення кутової висоти (піднесення) небесного тіла відносно бульбашкового штучного горизонту. Прилад оснащений спиртовим бульбашковим рівнем, що дає змогу під час вимірювань фіксувати точний горизонт (рівень) без спостереження за лінією горизонту [6].

Щоб ефективно застосовувати систему в умовах відсутнього доступу до мережі Інтернет, необхідно перед можливим використанням програмного забезпечення проводити завантаження певних метаданих (орієнтовно один раз на добу), що обчислюються на серверній частині.

2.2. Загальний алгоритм роботи системи

Принцип дії:

1) За допомогою камери проводиться фотозйомка небосхилу, на якому наявний необхідний небесний об’єкт (Сонце).

2) У момент, коли робиться знімок, проходить зчитування необхідних даних часу та інформації з датчиків, що дозволяє отримати орієнтацію пристрою у просторі.

3) Наступним етапом, знімок аналізується: пошук мета-інформації з зображення (розпізнавання патернів, визначення знаходження Сонця на знімку).

4) На основі отриманих даних, система обчислює азимут та кут, що визначає висоту небесного об’єкта над горизонтом (схилення Сонця).

5) Для контрольних точок на поверхні (Землі) у момент часу фотознімку обчислюється значення кутів (описані у пункті 4). Дані інтерполюються та проходить пошук ділянки, де наявне співпадіння з інформацією, що отримана у пункті 4 (з урахуванням похибки) - результатом є орієнтовні координати цієї місцевості.

2.3. Використання показників сенсорів: орієнтація у просторі

Завдяки сенсорам акселерометра та магнітного поля можливо отримати дані орієнтації пристрою у просторі (кути повороту пристрою у трьох вимірах). Наведені датчики наявні у більшості сучасних неспеціалізованих мобільних пристроях, доступ до них наданий на основі спеціальних інтерфейсів.

Надалі, розрахунки та формат отримання даних описуються на основі використання мобільних пристроїв під управлінням операційної системи Android, проте даний опис можливо застосувати до будь-якого типу пристроїв, що відповідають технічним вимогам.

Нехай, пристрій має 3-D систему координат (рис. 2.1) з наступними вісями [8]:

- додатна вісь X вказує на праву сторону дисплея у портретному режимі;

- додатна вісь Y вказує на верхню частину пристрою у портретному режимі;

- додатна вісь Z вказує на екран.

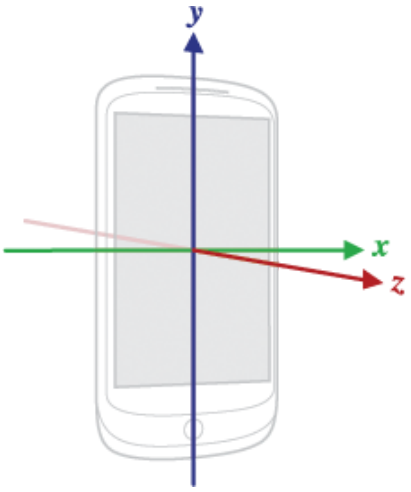

Рис. 2.1 - Система координат мобільного пристрою [https://developer.android.com/reference/android/hardware/SensorEvent.html]

Водночас, опорна система координат визначається як прямий ортонормований базис [9] (рис. 2.2):

- вісь X є дотичною до Землі в поточному розташуванні та вказує на схід;

- вісь Y є дотичною до Землі в поточному розташуванні та вказує на магнітний север;

- вісь Z вказує на небосхил та перпендикулярна до Землі.

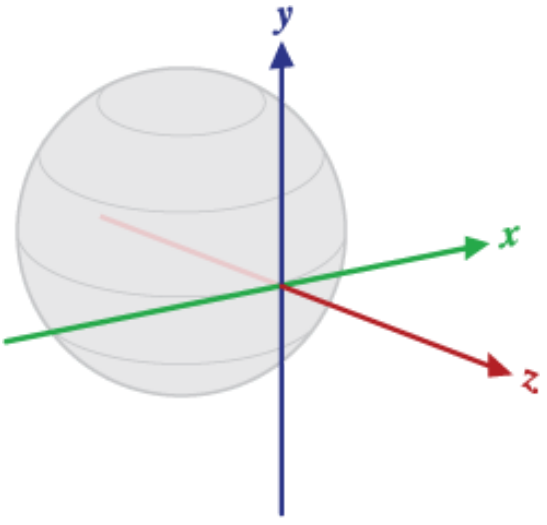

Рис. 2.2 - Опорна система координат [https://developer.android.com/reference/android/hardware/SensorEvent.html]

Показання отримуються у формі кватерніона, що описує обертання системи координат пристрою відносно системи координат Землі. Якщо вісь обертання є нормалізованим вектором (ax, ay, az), кут обертання (правобічний) α, то кватерніон надається у вигляді (X, Y, Z, W) [8], де:

$$X = ax \cdot sin(\alpha / 2);$$

$$Y = ay \cdot sin(\alpha / 2);$$

$$Z = az \cdot sin(\alpha / 2);$$

$$W = cos(\alpha / 2). \qquad (2.1)$$

Якщо вибрати відповідну вісь та кут, то можливо задати будь-яку орієнтацію об’єкта (рис. 2.3). Кватерніон описує цю властивість за формулою 2.1.

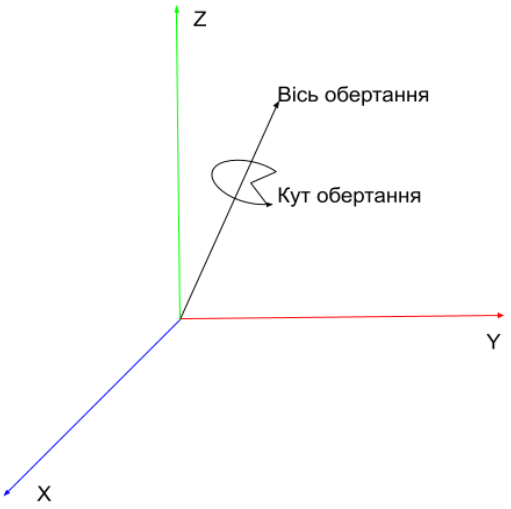

Рис. 2.3 - Представлення вісі та кута обертання

Проте необхідно перетворити отримані значення у три кута для зручного опису орієнтації пристрою та їх подальшого застосування при розрахунках.

Виведення формули перетворення кватерніонів на матрицю обертів наведено далі.

Перетворення точки P_1 у точку P_2 проходить на базі формули [10] (норма векторного компоненту зберігається):

$$P_2 = q \cdot P_1 \cdot q^{-1} \qquad (2.2)$$

, де q - кватерніон, що описує обертання;

q^{-1} - обернений кватерніон.

Також відомо, що за допомогою матриці поворотів [M] можливо зробити подібні перетворення [11]:

$$P_2 = [M] \cdot P_1 \qquad (2.3)$$

Якщо $P_1 = (0, x, y, z)$, тоді:

$$P_2 = q \cdot P_1 \cdot q^{-1} = (q_w + q_x \cdot i + q_y \cdot j + q_z \cdot k) \cdot (0 + x \cdot i + y \cdot j + z \cdot k) \cdot$$
$$(q_w - q_x \cdot i - q_y \cdot j - q_z \cdot k) =$$
$$((q_w \cdot x \cdot i + q_x \cdot i \cdot x \cdot i + q_y \cdot j \cdot x \cdot i + q_z \cdot k \cdot x \cdot i) +$$
$$(q_w \cdot y \cdot j + q_x \cdot i \cdot y \cdot j + q_y \cdot j \cdot y \cdot j + q_z \cdot k \cdot y \cdot j) +$$
$$(q_w \cdot z \cdot k + q_x \cdot i \cdot z \cdot k + q_y \cdot j \cdot z \cdot k + q_z \cdot k \cdot z \cdot k)) \cdot$$
$$(q_w - q_x \cdot i - q_y \cdot j - q_z \cdot k);$$

Властивості кватерніону [12]:

$$i^2 = j^2 = k^2 = i \cdot j \cdot k = -1 \qquad (2.4)$$

Таблиця 2.1

Таблиця множення базисних кватерніонів (1, i, j, k)

x	1	i	j	k
1	1	i	j	k
i	i	-1	k	-j
j	j	-k	-1	i
k	k	j	-i	-1

Тоді:

$$P_2 = ((q_w \cdot x \cdot i + q_x \cdot x \cdot (-1) + q_y \cdot x \cdot (-k) + q_z \cdot x \cdot j) +$$

$$(q_w \cdot y \cdot j + q_x \cdot y \cdot k + q_y \cdot y \cdot (-1) + q_z \cdot y \cdot (-i)) +$$

$$(q_w \cdot z \cdot k + q_x \cdot z \cdot (-j) + q_y \cdot z \cdot i + q_z \cdot z \cdot (-1))) \cdot (q_w - q_x \cdot i - q_y \cdot j - q_z \cdot k) =$$

$$(-q_x \cdot x - q_y \cdot y - q_z \cdot z + i \cdot (q_w \cdot x - q_z \cdot y + q_y \cdot z) + j \cdot (q_z \cdot x + q_w \cdot y -$$

$$q_x \cdot z) + k \cdot (-q_y \cdot x + q_x \cdot y + q_w \cdot z)) \cdot (q_w - q_x \cdot i - q_y \cdot j - q_z \cdot k) =$$

$$-q_x \cdot q_w \cdot x - q_y \cdot q_w \cdot y - q_z \cdot q_w \cdot z + i \cdot ({q_w}^2 \cdot x - q_z \cdot q_w \cdot y + q_y \cdot q_w \cdot z) +$$

$$j \cdot (q_z \cdot q_w \cdot x + {q_w}^2 \cdot y - q_x \cdot q_w \cdot z) + k \cdot (-q_y \cdot q_w \cdot x + q_x \cdot q_w \cdot y + {q_w}^2 \cdot z) +$$

$${q_x}^2 \cdot x \cdot i + q_y \cdot q_x \cdot y \cdot i + q_z \cdot q_x \cdot z \cdot i + q_x \cdot q_y \cdot x \cdot j + {q_y}^2 \cdot y \cdot j + q_z \cdot q_y \cdot z \cdot j$$

$$+ q_x \cdot q_z \cdot x \cdot k + q_y \cdot q_z \cdot y \cdot k + {q_z}^2 \cdot z \cdot k - (-1) \cdot (q_w \cdot q_x \cdot x - q_z \cdot q_x \cdot y +$$

$$q_y \cdot q_x \cdot z) -$$

$$(-k) \cdot (q_z \cdot q_x \cdot x + q_w \cdot q_x \cdot y - {q_x}^2 \cdot z) - j \cdot \left(-q_y \cdot q_x \cdot x + {q_x}^2 \cdot y + q_w \cdot q_x \cdot z\right)$$

$$- k \cdot (q_w \cdot q_y \cdot x - q_z \cdot q_y \cdot y + {q_y}^2 \cdot z) - (-1) \cdot (q_z \cdot q_y \cdot x + q_w \cdot q_y \cdot y -$$

$$q_x \cdot q_y \cdot z) - (-i) \cdot (-{q_y}^2 \cdot x + q_x \cdot q_y \cdot y + q_w \cdot q_y \cdot z) - (-j) \cdot (q_w \cdot q_z \cdot x - {q_z}^2$$

$$\cdot y + q_y \cdot q_z \cdot z) - i \cdot ({q_z}^2 \cdot x + q_w \cdot q_z \cdot y - q_x \cdot q_z \cdot z) - (-1) \cdot (-q_y \cdot q_z \cdot x +$$

$$q_x \cdot q_z \cdot y + q_w \cdot q_z \cdot z) =$$

$$(-q_x \cdot q_w \cdot x - q_y \cdot q_w \cdot y - q_z \cdot q_w \cdot z + q_w \cdot q_x \cdot x - q_z \cdot q_x \cdot y + q_y \cdot q_x \cdot z +$$

$$q_z \cdot q_y \cdot x + q_w \cdot q_y \cdot y - q_x \cdot q_y \cdot z - q_y \cdot q_z \cdot x + q_x \cdot q_z \cdot y + q_w \cdot q_z \cdot z) +$$

$$i \cdot ({q_w}^2 \cdot x - q_z \cdot q_w \cdot y + q_y \cdot q_w \cdot z + {q_x}^2 \cdot x + q_y \cdot q_x \cdot y + q_z \cdot q_x \cdot z -$$

$${q_y}^2 \cdot x + q_x \cdot q_y \cdot y + q_w \cdot q_y \cdot z - {q_z}^2 \cdot x - q_w \cdot q_z \cdot y + q_x \cdot q_z \cdot z) +$$

$$j \cdot (q_z \cdot q_w \cdot x + {q_w}^2 \cdot y - q_x \cdot q_w \cdot z + q_x \cdot q_y \cdot x + {q_y}^2 \cdot y + q_z \cdot q_y \cdot z +$$

$$q_y \cdot q_x \cdot x - {q_x}^2 \cdot y - q_w \cdot q_x \cdot z + q_w \cdot q_z \cdot x - {q_z}^2 \cdot y + q_y \cdot q_z \cdot z) +$$

$$k \cdot (-q_y \cdot q_w \cdot x + q_x \cdot q_w \cdot y + {q_w}^2 \cdot z + q_x \cdot q_z \cdot x + q_y \cdot q_z \cdot y + {q_z}^2 \cdot z +$$
$$q_z \cdot q_x \cdot x + q_w \cdot q_x \cdot y - {q_x}^2 \cdot z - q_w \cdot q_y \cdot x + q_z \cdot q_y \cdot y - {q_y}^2 \cdot z) =$$

$0 +$

$$i \cdot (x \cdot ({q_w}^2 + {q_x}^2 - {q_y}^2 - {q_z}^2) + y(-2 \cdot q_z \cdot q_w + 2 \cdot q_y \cdot q_x) + z \cdot (2 \cdot q_y \cdot q_w + 2 \cdot q_z \cdot q_x) +$$
$$j \cdot (x \cdot (2 \cdot q_z \cdot q_w + 2 \cdot q_x \cdot q_y) + y \cdot ({q_w}^2 + {q_y}^2 - {q_x}^2 - {q_z}^2) + z \cdot (2 \cdot q_z \cdot q_y - 2 \cdot q_x \cdot q_w) +$$
$$k \cdot (x \cdot (2 \cdot q_x \cdot q_z - 2 \cdot q_y \cdot q_w) + y \cdot (2 \cdot q_x \cdot q_w + 2 \cdot q_y \cdot q_z) + z \cdot ({q_w}^2 + {q_z}^2 - {q_x}^2 - {q_y}^2).$$

Враховуючи, що норма кватерніона дорівнює 1 (${q_w}^2 + {q_x}^2 + {q_y}^2 + {q_z}^2 = 1$) отримуємо матрицю M (таблиця 2.2).

Таблиця 2.2

Матриця обертів

$1 - 2 \cdot {q_z}^2 - 2 \cdot {q_y}^2$	$-2 \cdot q_z \cdot q_w + 2 \cdot q_y \cdot q_x$	$2 \cdot q_y \cdot q_w + 2 \cdot q_z \cdot q_x$
$2 \cdot q_z \cdot q_w + 2 \cdot q_x \cdot q_y$	$1 - 2 \cdot {q_x}^2 - 2 \cdot {q_z}^2$	$2 \cdot q_z \cdot q_y - 2 \cdot q_x \cdot q_w$
$2 \cdot q_x \cdot q_z - 2 \cdot q_y \cdot q_{w)}$	$2 \cdot q_x \cdot q_w + 2 \cdot q_y \cdot q_z$	$1 - 2 \cdot {q_x}^2 - 2 \cdot {q_y}^2$

Тобто перетворення кватерніону відбувається у тривимірний вектор кутів (у градусах) за наступними формулами:

$$\{$$
$$\arcsin(2 \cdot (q_x \cdot q_z - q_w \cdot q_y)) \cdot 180 / \mathrm{PI};$$
$$\mathrm{atan2}(2 \cdot (q_x \cdot q_w + q_y \cdot q_z),\ 1 - 2 \cdot ({q_z}^2 + {q_w}^2)) \cdot 180 / \mathrm{PI};$$
$$\mathrm{atan2}(2 \cdot (q_x \cdot q_y + q_z \cdot q_w),\ 1 - 2 \cdot ({q_y}^2 + {q_z}^2)) \cdot 180 / \mathrm{PI};$$
$$\} \qquad (2.5)$$

, де:

PI - математична константа;

q - отриманий кватерніон.

Приклад встановлення відповідності між матрицею обертів та кутами наведено у додатку А.

Один з необхідних надалі кутів (азимут) - це третій вимір вектору у формулі 2.5.

Слід зазначити, що випадок сингулярності не розглядається у даній роботі.

Спосіб обробки даних кватерніону (та перетворення його у кути обертання) у коді програмного забезпечення наведено у додатку Б; спосіб зчитування та відправки кватерніона на обробку наведено у додатку В.

2.4. Застосування моделі сонячної системи

Модель руху Землі навколо Сонця описується на основі небесної сфери (уявна сфера, на яку проектуються небесні тіла) (рисунок 2.4.). Завдяки цьому представленню можливо описати положення Сонця щодо шуканої точки на основі двох кутів, яким однозначно відповідає місцезнаходження (координати).

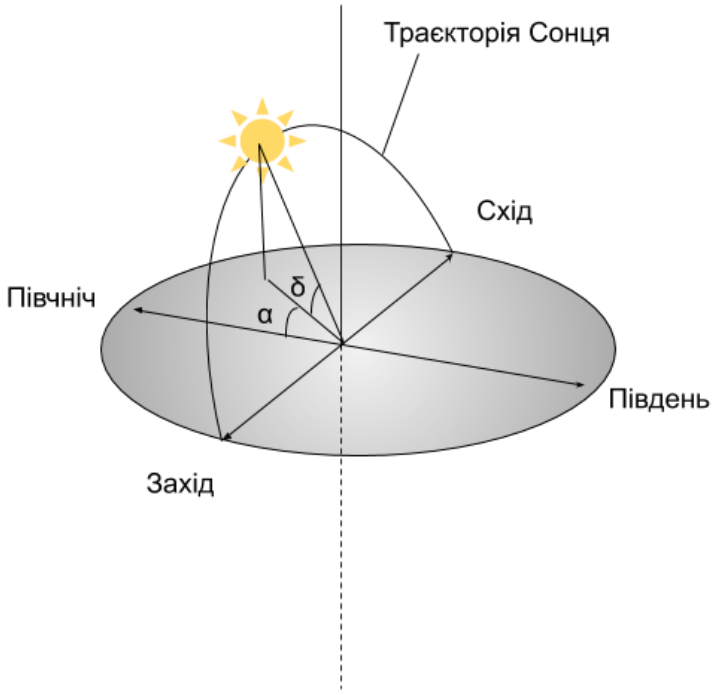

Рис. 2.4 - Представлення небесної сфери у певній точці.

Зображено кути, що визначають положення точки на Землі відносно сонячних променів: α − азимут; δ − схилення

Схилення Сонця - це кут між площиною небесного екватору та напрямком на світило [13 - 14]. Кут нахилу площини екватора Землі до площини її орбіти дорівнює приблизно 23.45°, тому в різні часи протягом року, коли Земля обертається навколо Сонця, відхилення змінюється від 23.45° на північ до 23.45° на південь [14].

Зв'язок між кутом піднесення (ε) та географічною широтою встановлюється за формулою [2]:

$$sin(\varepsilon) = sin(\delta) \cdot sin(\varphi) + cos(\delta) \cdot cos(\varphi) \cdot cos(\omega) \qquad (2.6)$$

, де φ - параметр, за яким визначається географічна широта;

ω - кут падіння сонячних променів на Землю;

δ - схилення Сонця, що можна апроксимувати в залежності від дати (d) [15]:

$$\delta = -0.4092797 \cdot cos(2 \cdot \pi/365 \cdot (d + 10)) \quad (2.7)$$

Значення довготи (λ) пов’язано з сонячним часом [2]:

$$t_{sun} = t_{utc} - 12/\pi \cdot \lambda \quad (2.8)$$

, де t_{sun} - сонячний час на певній довготі;

t_{utc} - час за Грінвічем (UTC).

У свою чергу сонячний час пов’язаний з кутом падіння сонячних променів на Землю [2] (формула 2.9).

$$\omega(t_{sun}) = 180 / 12 \cdot (t_{sun} - 12) \quad (2.9)$$

2.5. Інформація, що отримується з зображення-фотознімку

Якщо розглянути проекцію об'єктів небесної сфери (Сонця) на площину (рис. 2.5), що перпендикулярна до площини, яка описує кут схилення та проходить через діаметр кола в основі півсфери, то можливо розробити залежність між відношенням висоти над горизонтом Сонця та висоти над горизонтом крайньої верхньої точки і кутом схилення.

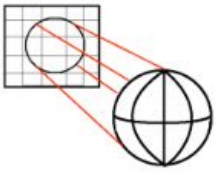

Рис. 2.5 - Проекція сфероїдної фігури (Сонця) на площину [16]

Для коректного визначення відношення необхідно враховувати різні кути огляду (зору) камер мобільних пристроїв. Якщо інформацію про кут огляду не надано, то необхідно провести збір даних про камеру на основі пробних знімків (наприклад, зробити певну вибірку знімків).

Для спрощення, проекція Сонця повинна бути центрована (рисунок 2.6) для визначення приблизного кута, що описує висоту зірки над горизонтом.

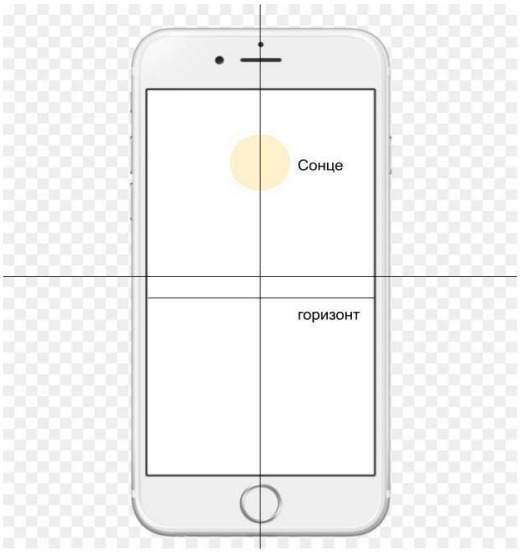

Рис. 2.6 - Центрування проекції Сонця посередині та чітко виділений горизонт

Модель передбачає наявність чітко виділеного горизонту на фотознімку. Дане обмеження (та обмеження з центруванням Сонця) можна усунути, застосувавши, наприклад, нейронні мережі для знаходження кута, що описує висоту над горизонтом.

3. Результати досліджень та перспективи розвитку

3.1. Опис версії додатку

Цільова платформа: Android (від версії 5.0).

Технології на базі якої відбувається розробка: мова програмування C#, технологія .Net та фреймворк Xamarin.

Xamarin - фреймворк, що надає можливість кросплатформної розробки (iOS, Android, Windows Phone) з використанням мови програмування C# (рис. 3.1) [17].

Рис. 3.1 - Слої програмного забезпечення на базі Xamarin

Даний фреймворк обрано з урахуванням можливості розширення додатку для використання на інших платформах, без значної зміни функціонального коду. Він містить бібліотеку класів, що дозволяє описати єдиний функціонал для декількох платформ; бібліотеку класів, що надає доступ до Android SDK [17] (розробка для цільової платформи), тощо.

Потік даних (взаємодія інформації) від отримання показників сенсорів до визначення географічного положення відображено на рис.3.2.

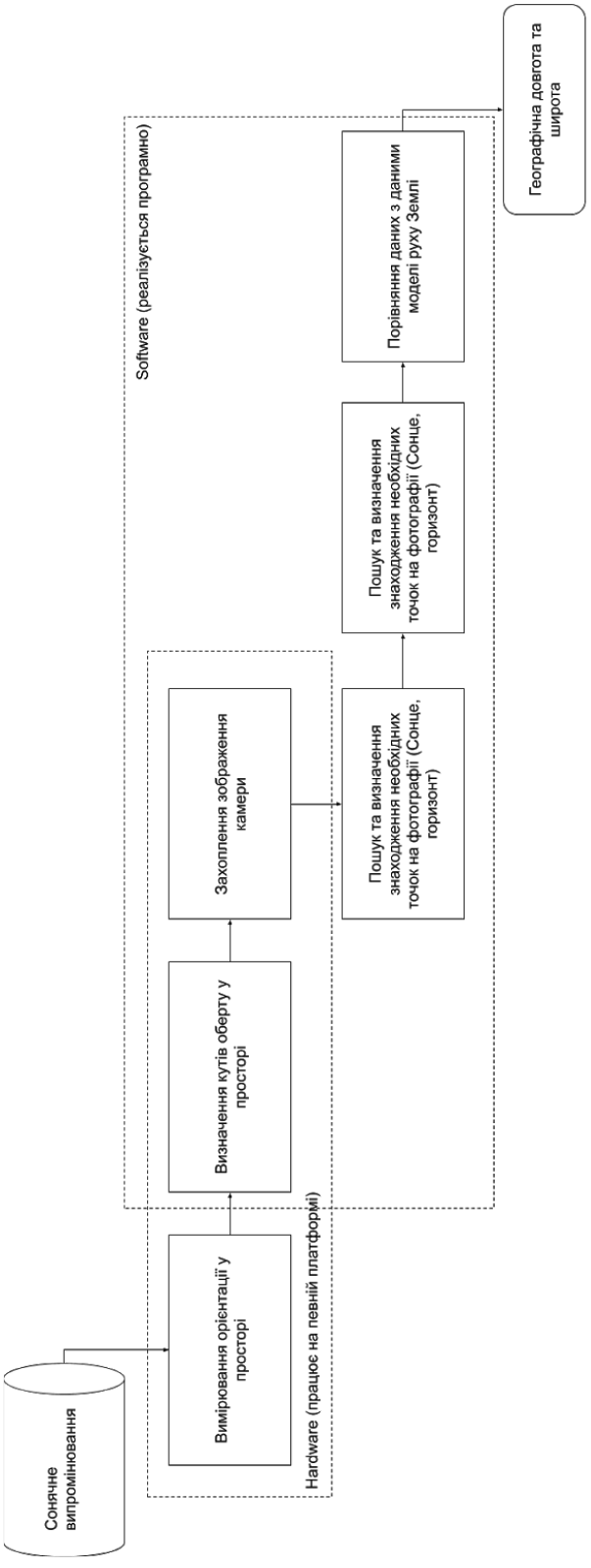

Рис. 3.2 - Блок-схема відображає за які кроки в роботі програмного забезпечення відповідає «залізо» (Hardware) та програмний додаток (Software)

Архітектура розробленого додатку складається з модулів, що дозволяє зробити її простішою для масштабування та змін.

Наявні наступні модулі:

- Horizon – визначення орієнтації у просторі (робота з кватерніонами);
- Helix – підтримка моделі руху Землі та надання функціоналу для визначення необхідних даних;
- LagoLix – модуль, що відповідає за пошук просторової інформації на зображенні (пошук координат проекції Сонця);
- Tenzor – модуль, що є основним для всіх платформ, відповідає за зчитування даних сенсорів, відображення платформо-незалежної інформації, тощо;
- Модуль Android – використовує Tenzor, та є платформо-залежним модулем.

Після запуску застосування, користувач може побачити кути орієнтації у просторі (рис. 3.3).

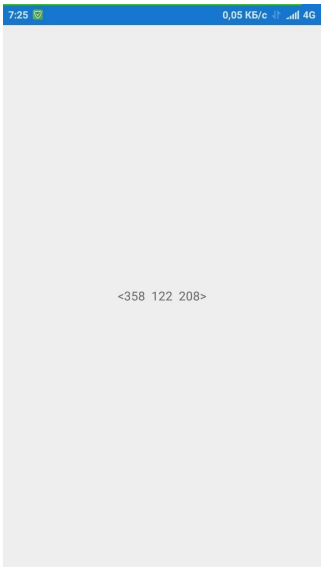

Рис. 3.3 - Знімок екрану застосування після запуску

Коли кути відповідають наступному шаблону: (0, 90, *), телефон починає вібрувати та у користувача з’являється можливість зробити знімок

(використовуючи основну камеру). Користувач повинен центрувати Сонце посередині екрану, для покращення геолокації.

Після успішно зробленого фото, програмне забезпечення видає приблизне географічне положення користувача.

Користувач може розпочати визначення спочатку за допомогою кнопки «Reset», чи запустивши програмне забезпечення наново.

3.2. Перспектива розвитку системи

Подальші дослідження:

- підвищення точності з використанням технологій, таких як комп’ютерний зір;
- побудова більш точної моделі сонячної системи;
- розрахунок географічного положення, враховуючи особливості земної атмосфери;
- позбавлення користувача від необхідності позиціонувати Сонце відносно камери (фотознімок під будь-яким кутом).

При моделюванні руху Землі навколо Сонця необхідно враховувати:

- еліптичну орбіту навколо Сонця;
- обертання Землі навколо власної вісі під кутом;
- прецесію;
- нутацію.

Висновки

До використання таких технологій, як GPS, навігація та позиціонування виконувалось на основі положення небесних тіл (наприклад, Сонце). У ясні дні Сонце забезпечує гарний орієнтир для виявлення своєї позиції. Дуже часто мореходці використовували це, користувавшись секстатнтом, та отримували похибку приблизно у 200 метрів. Подібний метод був застосований у даній роботі.

Практичне значення:

1. Розроблено метод співставлення поточного стану моделі руху Землі з інформацією, що отримана на основі фотознімку та показників сенсорів (просторової орієнтації фотознімку).

2. Розроблене програмне забезпечення дозволяє обчислювати приблизне географічне положення неспеціалізованого (побутового) мобільного пристрою, враховуючи можливі збої при використанні мережі Інтернет, системи глобального позиціонування GPS та зв’язку з сотовою мережею (наприклад, при втраті зв’язку з вищезазначеними ресурсами під час руху автомобіля по трасі).

3. Метод можливо застосовувати для уточнення даних основних систем (наприклад, глобальної системи позиціонування).

Наукова новизна полягає в:

1. Модифікації математичної моделі з урахуванням її реалізації на неспеціалізованих мобільних пристроях телекомунікацій та з урахуванням наявного набору вбудованих сенсорів.

2. Вдосконалені математичної 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.

У порівнянні із сучасними методами геолокації наведелений метод не може застосовуватись до задач, що вимагають високої точності. Однак, при поліпшенні моделі руху Землі навколо Сонця та розпізнавання об’єктів і геопросторових параметрів на фотознімку, даний метод може стати пріоритетним для вирішення певних спеціалізованих задач.

Напрямки подальших досліджень: підвищення точності з використанням технологій, таких як комп’ютерний зір; побудова більш точної моделі сонячної системи; розрахунок географічного положення, враховуючи особливості земної атмосфери.

Література

1. Geolocation - Wikipedia, the free encyclopedia. – [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Geolocation

2. Sandnes F. E., Determining the Geographical Location of Image Scenes based on Object Shadow Lengths. // Journal of Signal Processing Systems, 2010 - 35-47c.

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

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

5. 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.

6. Gómez J. V., Sandnes F. E., Fernández B., Sunlight Intensity Based Global Positioning System for Near-Surface Underwater Sensors. // Sensors, 2012 - 1930–1949c.

7. Секстант - Вікіпедія, вільна енциклопедія – [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Секстант

8. Xamarin.Essentials: OrientationSensor - Microsoft Docs - [Електронний ресурс]. – Режим доступу: https://docs.microsoft.com/en-us/xamarin/essentials/orientation-sensor?context=xamarin/xamarin-forms

9. SensorEvent - Android Developers - [Електронний ресурс]. – Режим доступу: https://developer.android.com/reference/android/hardware/SensorEvent.html

10. Кватерніони і повороти простору - Вікіпедія, вільна енциклопедія – [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Кватерніони_і_повороти_простору

11. Матриця повороту - Вікіпедія, вільна енциклопедія – [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Матриця_повороту

12. Кватерніони - Вікіпедія, вільна енциклопедія – [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Кватерніони

13. Екваторіальна система координат - Вікіпедія, вільна енциклопедія – [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Екваторіальна_система_координат

14. Declination - Sustainable By Design – [Електронний ресурс]. – Режим доступу: https://susdesign.com/popups/sunangle/declination.php

15. Sandnes F. E., Towards calibration-free geo-localization of stationary outdoor webcams // NIK conference, 2010 - 12c.

16. Karin Blank, EPIC Geolocation and Color Imagery Algorithm Revision 5 - Greenbelt. // Goddard Space Flight Center Greenbelt, Maryland, 2017. - 39c.

17. Подробно о Xamarin - Хабр - [Електронний ресурс]. – Режим доступу: https://habr.com/ru/post/188130/

Додатки

Додаток А *Встановлення залежності при перетворенні матриці обертів у кути Ейлера*

Дано матриця М:

$1-2\cdot {q_z}^2-2\cdot {q_y}^2$	$-2\cdot q_z\cdot q_w+2\cdot q_y\cdot q_x$	$2\cdot q_y\cdot q_w+2\cdot q_z\cdot q_x$
$2\cdot q_z\cdot q_w+2\cdot q_x\cdot q_y$	$1-2\cdot {q_x}^2-2\cdot {q_z}^2$	$2\cdot q_z\cdot q_y-2\cdot q_x\cdot q_w$
$2\cdot q_x\cdot q_z-2\cdot q_y\cdot q_w$	$2\cdot q_x\cdot q_w+2\cdot q_y\cdot q_z$	$1-2\cdot {q_x}^2-2\cdot {q_y}^2$

Перетворення кутів yaw (y), pitch (p) та roll (r) у матрицю обертання (наприклад, yzx) (MO) відбувається наступним чином:

cos(y)*cos(p)	-cos(y)*sin(p)*cos(r) +sin(y)*sin(r)	cos(y)*sin(p)*sin(r)+ sin(y)*cos(r)
sin(p)	cos(p)*cos(r)	-cos(p)*sin(r)
-sin(y)*cos(p)	sin(y)*sin(p)*cos(r)+ cos(y)*sin(r)	-sin(y)*sin(p)*sin(r) +cos(y)*cos(r)

Звідси, можна зробити висновок, що:

за MO_{10} можна визначити sin(p);

за MO_{00} таM_{20} можна визначити *tan*(*y*);

за MO_{11} та M_{12} можна визначити *tan*(*r*);

Додаток Б *Витяг коду для обробки даних кватерніона (отриманих даних орієнтації пристрою у просторі)*

Мова програмування: C#.

```
using System;
using System.Numerics;
…
 public static Vector3 FromQ2(Quaternion q) {
        Vector3 v;
        v.Y = (float)(Math.Atan2(2f * q.X * q.W + 2f * q.Y * q.Z, 1 - 2f * (q.Z * q.Z + q.W * q.W)) * 180 / Math.PI);
   v.X = (float)(Math.Asin(2f * (q.X * q.Z - q.W * q.Y)) * 180 / Math.PI);
        v.Z = (float)(Math.Atan2(2f * q.X * q.Y + 2f * q.Z * q.W, 1 - 2f * (q.Y * q.Y + q.Z * q.Z)) * 180 / Math.PI);
        return NormalizeAngles(v);
    }
    static Vector3 NormalizeAngles(Vector3 angles){
        angles.X = NormalizeAngle(angles.X);
        angles.Y = NormalizeAngle(angles.Y);
        angles.Z = NormalizeAngle(angles.Z);
        return angles;
    }
    static float NormalizeAngle(float angle){
  angle = angle % 360;
if(angle < 0) angle += 360;
        angle = (float)Math.Round(angle);
        return angle;
    }          ...
```

Додаток В *Витяг коду для зчитування показників сенсорів, виведення показників на екран та створення вібрацій (як підказка для користувача)*

Мова програмування: C#.

Фреймворк: Xamarin.

Застосування для неспеціалізованого мобільного пристрою під управлінням операційної системи Android.

```
OrientationSensor.ReadingChanged += (sender, args) =>
{
    Quaternion q = args.Reading.Orientation;
    Vector3 vec = FromQ2(q);
    label.Text = q.ToString();
    label.Text += "\n";
    label.Text += vec.ToString();
    if ((vec.X < 5 || vec.X > 355) && vec.Y< 95 && vec.Y> 85)
    {
        Vibration.Vibrate();
    }
};
```