

*А.А. СОКОЛОВ, О.Г АВРУНІН, д-р техн. наук*

## **ВИКОРИСТАННЯ БІБЛІОТЕКИ ARCORE ДЛЯ ВІЗУАЛІЗАЦІЇ ХМАРИ КЛЮЧОВИХ ТОЧОК У НАВІГАЦІЙНИХ СИСТЕМАХ**

### **Вступ**

Сліпота та серйозні порушення зору створюють безліч труднощів у повсякденному житті, включаючи орієнтацію у просторі та уникнення перешкод для людей з обмеженими можливостями. Такі проблеми значно обмежують незалежність незрячих людей, особливо в незнайомому середовищі. Вирішення цих завдань має глибокий вплив на суспільство, починаючи від підвищення рівня безпеки та комфорту незрячих людей і закінчуючи позитивним ефектом на економіку, зокрема через інтеграцію таких осіб у соціальні та трудові процеси [1].

Сучасні технології відкривають нові перспективи для вирішення цих проблем. Зокрема, комбінація алгоритмів SLAM та нейронних мереж створює передумови для розробки ефективних навігаційних систем. Такі системи здатні значно полегшити орієнтацію в просторі, автоматизуючи процеси аналізу середовища, і тим самим повністю або частково вирішити ключові проблеми, з якими стикаються незрячі люди [2, 3]. Прикладом подібного рішення є Seeing AI від Microsoft.

### **Архітектура навігаційної системи. Основні принципи**

Створення ефективної навігаційної системи для незрячих людей вимагає врахування їх матеріальних можливостей та доступності компонентів кінцевого продукту. На жаль, у сучасних умовах більшість незрячих людей в Україні не має високого рівня матеріального забезпечення. Це вимагає розробки технологій на основі максимально доступних і масових апаратних засобів, які не потребують значних фінансових витрат.

Враховуючи це, ми пропонуємо використовувати сучасний смартфон як основний обчислювальний модуль системи. Смартфони мають достатній рівень продуктивності для реалізації алгоритмів одночасної локалізації і побудови карти (SLAM) та інференсу нейронних мереж. Крім того, смартфон є поширеним пристроєм, яким вже володіє велика кількість людей, включаючи незрячих.

Запропонована концепція передбачає інтеграцію нейронної мережі та SLAM-алгоритмів для створення навігаційного пристрою, здатного аналізувати оточення через камеру смартфона, обробляти дані локально та передавати їх на тактильний модуль через Bluetooth. Такий підхід забезпечує компактність і доступність кінцевого продукту, що є ключовими факторами для розширення його застосування [4, 5].

Система також передбачає класифікацію перешкод за типами, що дозволяє інформувати користувача про небезпечні об'єкти, такі як автомобілі чи тварини, а також визначати важливі елементи інфраструктури, наприклад світлофори, пішохідні переходи чи пошкодження дорожнього покриття. Це підвищує рівень безпеки та зручності користувача, сприяючи його незалежності та інтеграції в суспільство.

Для реалізації тактильного модуля передбачається використання доступного мікроконтролера з підтримкою Bluetooth, наприклад, ESP32 [6 – 8]. Такий модуль забезпечуватиме тактильний зворотний зв'язок із користувачем через компактну матрицю вібромоторів. Інформація про відстань до перешкод та їх тип буде передаватися у формі тактильних сигналів різної інтенсивності, частоти або патернів, що дозволить користувачу швидко орієнтуватися в просторі.

Таким чином, запропонована система має потенціал значно спростити процес навігації для незрячих людей, зберігаючи її доступність і функціональність, а також забезпечити незалежність користувача за рахунок використання масових і недорогих компонентів.

## Загальні принципи роботи ARCore

ARCore – це пропрієтарна технологія і повний алгоритм якої невідомий, хоча і ґрунтується на відомих методах. Основними є методи візуальної одометрії, оцінки переміщення на основі IMU і нейронної мережі, що зменшує накопичення помилки від IMU. Остання є пропрієтарною розробкою Google. Приблизний алгоритм роботи ARCore запропонований на рис. 1. Розглянемо основні технології більш детально[9,10].

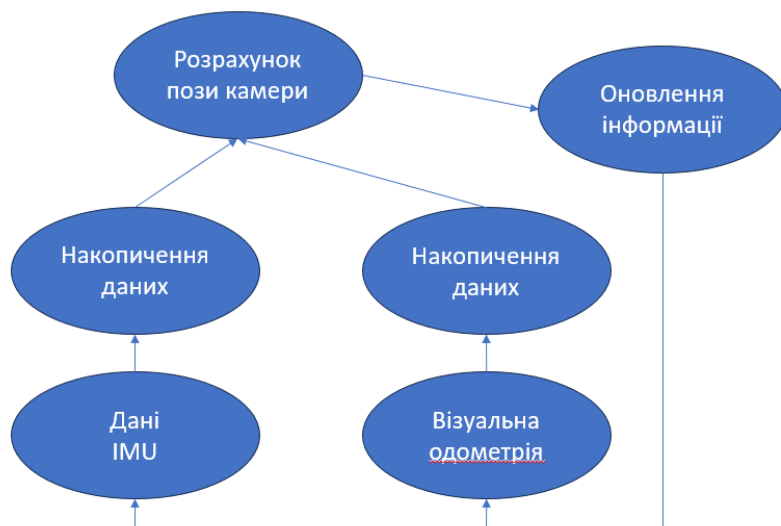


Рис. 1. Алгоритм роботи ARCore

Основна ідея візуальної одометрії полягає в зіставленні зображень: ми повинні знайти ключові точки, які можна знайти на інших зображеннях. Ключові точки, це щось, що має унікальні характеристики, наприклад, кут, сильний градієнт, лінія і таке інше. Можна виділити чотири основні етапи:

1. Вилучення особливостей: У кожному кадрі зображення виявляються і відстежуються ключові точки. Зазвичай використовують такі алгоритми як SIFT, SURF, ORB.

2. Відповідність особливостей: Особливості між послідовними кадрами порівнюються, щоб визначити, як вони зміщуються щодо один одного.

3. Визначення руху: на основі цих відповідностей обчислюється відносний рух камери між кадрами.

4. Оновлення траєкторії: Інтегрування послідовних відносних рухів для оцінки повної траєкторії.

Інерційна одометрія – це алгоритм оцінки положення та орієнтації об'єкта (наприклад, камери, робота або транспортного засобу) з використанням даних від інерційних вимірювальних пристроїв (IMU), таких як акселерометри та гіроскопи. Основна ідея – інтеграція показань IMU для відстеження переміщень та поворотів об'єкта з часом.

Алгоритм роботи інерційної одометрії:

1. Збір даних з IMU: акселерометри вимірюють лінійне прискорення  $a$ , гіроскопи вимірюють кутову швидкість  $\omega$ .

2. Калібрування IMU: коригування зсувів та шумів у показаннях IMU.

3. Обробка даних: видалення гравітаційного компонента з показань акселерометра, інтеграція кутової швидкості визначення орієнтації, інтеграція прискорень визначення швидкості і становища.

4. Корекція помилок.

Поєднавши дані від інерційної та візуальної одометрії, ми отримуємо інформацію про рух, яку зможемо використовувати у подальшій роботі. Для поєднання даних зазвичай використовують алгоритми двох груп. Це фільтр Калмана та група алгоритмів VINS-MONO (монокулярна візуальна інерційна система) [10, 11].

### Алгоритм отримання хмари ключових точок засобами ARCore

Хмару ключових точок засобами ARCore можна отримати за допомогою декількох методів з різною точністю. В рамках задачі побудови навігаційної системи найбільш цікавими є хіт тест та прямий запит хмари ключових точок. Характерною особливістю ключових точок отриманих засобами ARCore є те, що ці точки представлені у світових координатах. Програмно ми можемо отримати поточну позу камери, і визначити відстань від камери до точки [12], рис 2.

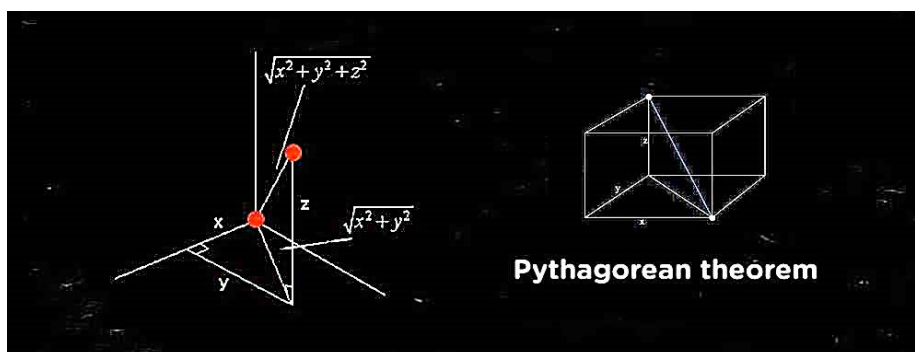


Рис. 2. Принцип визначення відстані

Метод `Frame.acquirePointCloud()` в ARCore надає хмару точок [13], які є набором 3D-координат, що визначають структуру оточення. Ці точки генеруються на основі аналізу зображень, сигналів з камери пристрою та алгоритмів виявлення зовнішніх подій. Кожна точка містить координати у світовій системі ( $x$ ,  $y$ ,  $z$ ) та рівень впевненості (довірливість), який показує, що ARCore дійсно точно визначив цю точку.

Хмара точок оновлюється миттєво, що дозволяє отримувати актуальні дані під час руху камери. Цей метод застосовується для розуміння просторової структури сцени, але має обмеження пов'язані з довірливістю. Кількість залежить від багатьох факторів, таких як освітлення та текстурність кадру. На практиці ми отримували близько 10 – 20 у приміщенні зі штучним освітленням та 100–250 для вуличних кадрів днем, у похмуру погоду.

Хіт-тести (`Frame.hitTest(x, y)`) надають можливість визначати 3D-координати точок на об'єктах сцени, використовуючи віртуальний промінь, спрямований з екрана пристрою. Цей метод особливо корисний для взаємодії з користувачем, наприклад, щоб визначити місце торкання поверхні в доповненій реальності. Результатом хіт-тесту є точки перетину променя з розпізнаними площинами, відстежуваними точками або віртуальними об'єктами. Кожна така точка включає позицію (Pose) і додаткову інформацію про тип поверхні. Хіт-тести підходять для завдань точної взаємодії з віртуальним середовищем, але працюють тільки в зонах, де ARCore виявив фізичні поверхні чи об'єкти. Зазвичай, їх кількість у кадрі невелика, менше 13, тому їх застосування для задач навігації ускладнено та дуже ситуативно.

Оскільки засоби ARCore дозволяють отримати матриці проєкції та виду, ми можемо перетворити знайдені ключові точки зі світових координат в нормалізовані координати. Це знадобиться для візуалізації даних та в подальшому буде використовуватися нейронною мережею для розпізнавання об'єктів. Алгоритм наведено на рис. 3.

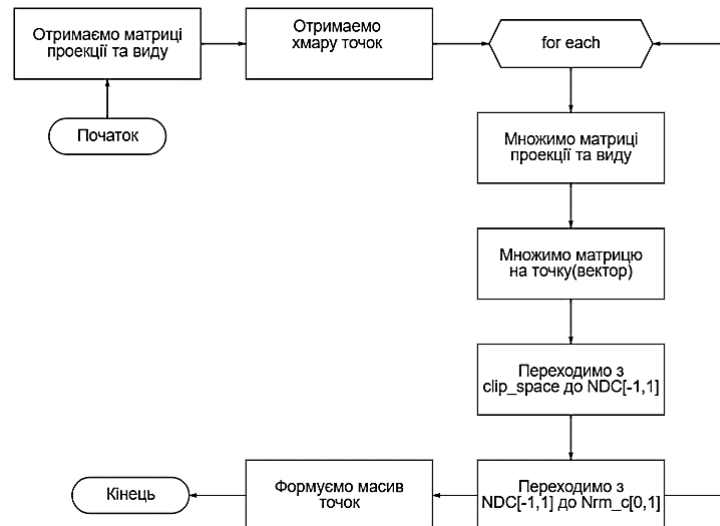


Рис. 3. Алгоритм отримання хмари ключових точок

Розглянемо необхідні перетворення. За допомогою методів `getProjectionMatrix` та `getViewMatrix` отримуємо програмними засобами матриці проєкції та виду. Виконавши множення, переходимо у `clip space`(1). Це проміжний координатний простір у графічному пайплайні рендерингу[11], який використовується у комп'ютерній графіці та обробці 3D-сцен.

$$ClipPosition = projMatrix \times ViewMatrix \times (PointPose, 1)^T, \quad (1)$$

де  $PointPose^T$  – координати точки у світових координатах;  $ViewMatrix$  – матриця виду для поточного кадру;  $projMatrix$  – матриця виду для поточного кадру;  $ClipPosition$  – координати точки у `clip space`.

Наступним кроком переходимо до NDC (нормалізованих координат пристрою). Координати у просторі відсікання (`clip space`) виражені у вигляді 4D-вектора. Щоб отримати нормалізовані координати пристрою, необхідно виконати перспективне ділення (2). В результаті цього кроку `ndcX` та `ndcY` будуть у діапазоні від -1 до 1:

$$ndcX = \frac{clipX}{clipW}, ndcY = \frac{clipY}{clipW}, \quad (2)$$

Далі залишилося лише перетворити нормалізовані координати пристрою на нормовані координати. Це можна зробити наступним чином(3).

$$Norm_x = \frac{NDC_x + 1}{2}, Norm_y = \frac{NDC_y + 1}{2}, \quad (3)$$

Тепер отримані дані можливо візуалізувати. Для цього збережемо отримані дані у форматі текстового файлу з координатами і відстанями у метрах та відповідного зображення.

### Візуалізація та аналіз отриманих даних

Для візуалізації отриманих даних ми використаємо мову пайтон, і також застосуємо YOLOv11n для детекції об'єктів у кадрі [14]. Для відрисовки рамок знайдених об'єктів застосуємо OpenCV [15]. Таким чином, ми зможемо оцінити, як дані будуть розмішуватись та скільки ключових точок потрапить на знайдений об'єкт. Проаналізуємо отримані результати, що зображені на (рис. 4). При цьому використовувалася стандартна модель, та у подальшому буде необхідно за допомогою переносу навчання адаптувати модель до задач навігації. Аналізуючи дані, ми бачимо багато викидів та не дуже високу точність.



Рис. 4. Приклади вуличних кадрів

Тому для отриманих результатів має сенс у подальшій роботі застосувати алгоритми для зменшення впливу викидів та покращення точності даних. Одним із ефективних підходів є використання кластеризаційних алгоритмів, таких як DBSCAN (Density-Based Spatial Clustering of Applications with Noise). Цей алгоритм дозволяє знаходити щільно згруповані ключові точки та відсікати точки, які є шумом або не належать до жодного кластеру. Його перевагою є здатність працювати без попереднього визначення кількості кластерів і ефективно виявляти вибірки навіть у складних умовах.

Іншим перспективним підходом є використання фільтрації даних на основі локального фактора вибивання (LOF – Local Outlier Factor), який дозволяє оцінити, наскільки точка є вибиванням відносно своїх найближчих сусідів. Цей метод особливо ефективний у випадках, коли дані мають нерівномірний розподіл.

Окрім того, для покращення стабільності та зменшення шуму в даних можна використувати фільтр Калмана, який добре підходить для роботи з послідовними даними. Цей підхід дозволяє прогнозувати положення точки, згладжуючи її траєкторію, та відсікати точки, які сильно відхиляються від очікуваного значення.

У рамках подальших досліджень планується комбінувати зазначені підходи для оптимального оброблення даних. Це дозволить зменшити похибки, підвищити якість отриманих даних та покращити загальну ефективність навігаційної системи. Реалізувати такий підхід можна на сучасній мікроконтролерних стендах [16, 17] для вивчення принципів роботи оптичних навігаційних систем різного призначення [18, 19].

## Висновки

Запропоновано концепцію навігаційної системи для незрячих людей, яка базується на використанні сучасних технологій доповненої реальності, зокрема ARCore. Система поєднує алгоритми SLAM, нейронну мережу та смартфон як основний обчислювальний модуль, що забезпечує її доступність та практичність.

Експериментальні результати показали, що кількість ключових точок значно залежить від умов освітлення та текстурності сцени. Отримані результати потребують подальшої обробки для зменшення викидів і підвищення точності.

Подальші дослідження будуть спрямовані на оптимізацію алгоритмів обробки хмари точок для зменшення кількості викидів, розробку та навчання адаптованих нейронних мереж, здатних точніше класифікувати перешкоди та важливі об'єкти, інтеграцію з тактильними модулями для покращення зворотного зв'язку з користувачем, експериментальну перевірку системи в реальних умовах експлуатації, зокрема на вулиці та у складних середовищах, для підвищення її ефективності та надійності.

#### Список літератури:

1. World Health Organization: WHO. Blindness and vision impairment. Who. int, Oct. 11, 2018. <https://www.who.int/en/news-room/fact-sheets/detail/blindness-and-visual-impairment>
2. Microsoft, "Seeing AI," Microsoft Garage. <https://www.microsoft.com/en-us/garage/wall-of-fame/seeing-ai/>
3. "Design principles – Wayfindr – Open Standard," Wayfindr, 2015. <https://www.wayfindr.net/open-standard/designing-for-vision-impaired-people/design-principles> (accessed Jan. 24, 2025).
4. Sokolov A., Avrunin O., Selivanova K. and Shushliapina N. Application of Augmented Reality Technologies for Determining Distances in Navigation System for the Blind // 2022 IEEE 16th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), Oct. 2024, pp. 530–533. doi: <https://doi.org/10.1109/tcset64720.2024.10755743>.
5. Sokolov A., Avrunin O. Evaluation of ARCORE library capabilities for determining the distance to objects in the frame // Optoelectronic Information-Power Technologies. 2024. Vol. 47, no. 1. P. 58–65. doi: <https://doi.org/10.31649/1681-7893-2024-47-1-58-65>.
6. Pastukh V., Andrushchak V., Beshley M., Klymash M., and Vdovychenko V. DEVELOPMENT OF AN INTELLIGENT NAVIGATION AND SPATIAL ORIENTATION SYSTEM FOR PEOPLE WITH VISUAL IMPAIRMENTS // Information and communication technologies electronic engineering. 2023. Vol. 3, no. 1. P. 53–63. doi: <https://doi.org/10.23939/ict2023.01.053>.
7. Tymkovych M. et al. 3D scanning technologies by optical RealSense cameras for SIREN-based 3D hand representation. Dec. 2023, doi: <https://doi.org/10.1117/12.3022737>.
8. Tymkovych M., Avrunin O., Selivanova K., Kolomiets A., Bednarchyk T., and Saule Smailova. Correspondence matching in 3d models for 3d hand fitting // Informatyka, Automatyka, Pomiar w Gospodarce i Ochronie Środowiska. 2024. Vol. 14, no. 1. P. 78–82. doi: <https://doi.org/10.35784/iapgos.5498>.
9. Use the ARCore Depth API for immersive augmented reality experiences // Google Codelabs. <https://codelabs.developers.google.com/codelabs/arcore-depth/#0>
10. How Visual Inertial Odometry (VIO) Works // Welcome to The Library!, Apr. 03, 2024. <https://www.thinkautonomous.ai/blog/visual-inertial-odometry/>.
11. Computer Vision: Algorithms and Applications. 2nd ed. szeliski.org. <https://szeliski.org/Book/>
12. 47billion, Distance Measurement on Mobile App using ARCore," 47Billion, May 19, 2021. <https://47billion.com/blog/distance-measurement-on-mobile-app-using-arcore/>.
13. Frame. Google for Developers, 2024. <https://developers.google.com/ar/reference/java/com/google/ar/core/Frame#acquirePointCloud>.
14. Ultralytics, "Predict," docs.ultralytics.com. <https://docs.ultralytics.com/modes/predict/#inference-sources>
15. OpenCV: Drawing Functions in OpenCV. docs.opencv.org. [https://docs.opencv.org/4.x/dc/da5/tutorial\\_py\\_drawing\\_functions.html](https://docs.opencv.org/4.x/dc/da5/tutorial_py_drawing_functions.html)
16. Avrunin O.G. Experience of Developing a Laboratory Base for the Study of Modern Microprocessor Systems / O.G. Avrunin, T.V. Nosova, V.V. Semenets // Proceedings of I International Scientific and Practical Conference «Theoretical and Applied Aspects of Device Development on Microcontrollers and FPGAs» MC&FPGA-2019, Kharkiv, Ukraine, 2019. P. 6–8.
17. Avrunin O., Sakalo S., Semenets V. Development of up-to-date laboratory base for microprocessor systems investigation // 2009 19th International Crimean Conference Microwave & Telecommunication Technology. Sevastopol, 2009. P. 301–302.
18. Оцінка можливостей бібліотеки ARCORE для визначення дистанції до об'єктів у кадрі / А. А. Соколов, О. Г. Аврунін // Опт-ел. інф-енерг. техн. 2004. Вип. 47, № 1. С. 58–65. DOI: 10.31649/1681-7893-2024-47-1-58-65.
19. Оцінка показників стереоендоскопічних систем в ринології / А.Соколов, О. Аврунін, Л. Коваль, О. Кадук // Опт-ел. інф-енерг. техн. Вип. 47, №. 1. С. 128–139. DOI: 10.31649/1681-7893-2024-47-1-128-139.

*Надійшла до редколегії 23.10.2024*

#### *Відомості про авторів:*

**Соколов Андрій Андрійович** – Харківський національний університет радіоелектроніки, аспірант кафедри біомедичної інженерії; Україна; e-mail: [andrii.sokolov@nure.ua](mailto:andrii.sokolov@nure.ua); ORCID: <https://orcid.org/0000-0002-8087-698X>

**Аврунін Олег Григорович** – д-р техн. наук, професор, Харківський національний університет радіоелектроніки, завідуючий кафедрою біомедичної інженерії; Україна; e-mail: [oleh.avrunin@nure.ua](mailto:oleh.avrunin@nure.ua), ORCID: <https://orcid.org/0000-0002-6312-687X>