

<https://doi.org/10.31891/2219-9365-2023-74-5>

УДК 378

САЗОНОВ Артем

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

<https://orcid.org/0000-0001-7124-5863>

ЗГУРСЬКИЙ Данило

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

КУЧКІН Олексій

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

## ВИБІР МЕТОДУ КЛАСТЕРИЗАЦІЇ ДЛЯ ЗАДАЧІ НАВІГАЦІЇ КРОКУЮЧОГО РОБОТА У НЕСТРУКТУРОВАНИХ СЕРЕДОВИЩАХ

*Для нестандартних умов роботи роботів в більшості випадків можуть не підійти прості методи, що реалізуються в пристроях для побутових потреб та тих, які використовуються в структурованому середовищі з дорогами, помітними перешкодами чи з певними мітками на які спиралась би алгоритми виявлення перешкод та побудови маршруту. Якщо необхідно провести пошукові чи евакуаційні заходи, використовуючи крокуючих роботів, то це вже складні умови з певними особливостями. Головне завдання статті полягає у знаходженні такого методу кластеризації, який би показував гарні результати в неструктурованому середовищі з необхідною точністю.*

*В роботі розглянуто загальновідомі методи кластеризації даних, що поділяються на чотири основні групи, які основані на різних шляхах обробки даних та розподілення їх на кластери. Наводяться основні поняття, на яких ґрунтуються принципи та алгоритми роботи методів. На аналізі переваги та недоліки обирається група методів, яка є найоптимальнішою для заданої задачі.*

*Експериментальне порівняння обраних методів проводиться на хмарах точок неструктурованих середовищ. Результати аналізуються на структуру виділених кластерів, виявлення шуму та швидкодію з урахуванням параметрів, що необхідні для роботи методу. Після аналізу якості кластеризації обирається метод з найкращими результатами для подальших досліджень. Для випробувань методу використовуються навісні, від'ємні та додатні перешкоди, такі як крони дерев, обрив та пагорби.*

*Ключові слова: кластеризація, крокуючий робот, неструктуроване середовище.*

SAZONOV Artem, ZGHURSKYI Danylo, KUCHKIN Aleksey

National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"

## CHOICE OF A CLUSTERIZATION METHOD FOR THE PROBLEM OF NAVIGATION OF A WALKING ROBOT IN UNSTRUCTURED ENVIRONMENTS

*In most cases, the simple methods implemented in devices for household needs and those used in a structured environment with roads, visible obstacles or with certain labels, on which the algorithms of obstacle detection and route construction would be based, may not be suitable for non-standard operating conditions of robots. If it is necessary to carry out search or evacuation measures using walking robots, then these are already difficult conditions with certain features. The main task of the article is to find such a clustering method that would show good results in an unstructured environment with the required accuracy.*

*The work considers well-known methods of data clustering, which are divided into four main groups based on different ways of data processing and their distribution into clusters. The main concepts on which the principles and algorithms of the methods are based are presented. Based on the analysis of advantages and disadvantages, a group of methods is selected, which is the most optimal for the given task.*

*An experimental comparison of the selected methods will be carried out on point clouds of unstructured environments. The results are analyzed for the structure of selected clusters, noise detection and speed, taking into account the parameters necessary for the method to work. After analyzing the clustering quality, the method with the best results is selected for further research. Hanging, negative, and positive obstacles such as tree crowns, cliffs, and hills are used for testing the method.*

*Keywords: clustering, walking robot, unstructured environment.*

### Постановка проблеми у загальному вигляді

#### та її зв'язок із важливими науковими чи практичними завданнями

Навігація роботів, як колісних так і крокуючих, у міських умовах із дорожньої інфраструктурою де об'єкти, такі як будинки, дерева, ліхтарні стовпи мають чітку структуру та досить легко ідентифікуються, що не викликає суттєвих ускладнень. Такий навколишній простір у якому елементи мають чітку організацію та орієнтацію і положення на зображенні називають структурованим. Проте, використання роботів у нетипових, на даний час, умовах, наприклад, в евакуаційних заходах на завалах, навігація у гірській та лісовій місцевості тощо, де середовище не є структурованим, оскільки містить багато перешкод, що наперед не мають чітко визначеного положення або ж їх форми не можуть бути описані простими геометричними примітивами [1] вносить певні обмеження. По-перше, неможливість використання колісних платформ, по-друге крокуючі роботи, задля забезпечення стійкості та надійності переміщення повинні мати

щонайменше 4 кінцівки. Саме використання останніх є ключовим фактором при виконанні евакуаційних заходів у неструктурованих середовищах. Водночас, самі середовища містять значну кількість невидимих перешкод таких як кущі, валуни, ями довільної форми, що у свою чергу визначає необхідність вибору універсального методу кластеризації, який працював би із будь-якими хмарами точок.

### Аналіз існуючих методів кластеризації

Задача групування даних в змістовні кластери є актуальною в будь-якій галузі. Процес кластерного аналізу умовно можна описати наступним чином: обирається вибірка у якій виділяються змінні для оцінки; обирається міра схожості, що є передумовою успішного застосування алгоритму кластеризації.

Дані, які необхідно кластеризувати для навігації, є точками у тримірному просторі, що представлений у вигляді хмари (point cloud) і їх можна віднести до інтервально-масштабованих змінних [2]. Це означає, що  $n$  об'єктів буде характеризуватися  $p$  безперервними вимірюваннями, таким чином формується матриця  $n \times p$ , для точок в системі  $0xyz$ , які визначаються координатами за трьома осями, у такому випадку  $p = 3$ .

У якості міри схожості зазвичай обирають відстань між двома об'єктами (точками). Функцію відстані між об'єктами  $x$  та  $y$  позначено  $d(x,y)$ , а метрика обирається із наступних варіантів:

- Евклідова відстань, є найпростішою з усіх тому і використовується найчастіше.

$$d(x,y) = \sqrt{\sum_i^p (x_i - y_i)^2};$$

- Квадрат евклідової відстані, для надання віддаленим об'єктам більшої ваги

$$d(x,y) = \sum_i^p (x_i - y_i)^2;$$

- Манхеттенська відстань, це середня різниця за координатами, однак вплив окремих великих різниць зменшується.

$$d(x,y) = \sum_i^p |x_i - y_i|$$

- Відстань Чебишева, коли об'єкти відрізняються лише за одною координатою.

$$d(x,y) = \max(|x_i - y_i|)$$

При аналізі даних обирається оптимальний підхід до їх кластеризації та група методів, яка його реалізує. Відомі на сьогодні методи кластеризації умовно поділяють на наступні групи:

- Методи розбиття;
- Ієрархічні методи;
- Методи на основі сітки;
- Методи, що базуються на щільності даних [3].

В основі кожного алгоритму кластеризації лежить поняття розбиття даних. Нехай існує множина  $\Lambda$ , яка складається з  $m \geq 2$  елементів. Розбиття цієї множини на  $1 \leq k \leq m$  непересічних непорожніх множин  $\pi_1, \dots, \pi_k$  називається  $k$ -розбиттям множини  $\Lambda$  і буде позначатися як  $\Pi = \{\pi_1, \dots, \pi_k\}$ . Елементи цього розбиття називаються кластерами [4].

Саме поняття «кластер» має досить мало формальних визначень. На сьогодні існують два підходи до розуміння даного поняття: внутрішня згуртованість (гомогенність) і зовнішня ізоляція [5].

Якщо подати  $k$  як параметр і ділити  $n$  об'єктів на  $k$  кластерів так, щоб кластери мали високу внутрішньокластерну подібність і низьку міжкластерну подібність, можна навести один з найпростіших алгоритмів –  $k$ -means.

*K-means (K-середніх)*

Алгоритмічна частина складається з наступних кроків:

1. Обрати  $k$  точок центроїда і ініціалізувати їх як центроїди.
2. Формувати  $K$  кластерів призначенням кожної точки найближчому центроїду і перераховувати центроїди кожного кластера доти, доки не буде виконаний критерій збіжності [6].

Координати центроїда розраховуються як:

$$m_i = \frac{1}{n_i} \cdot \sum_{x_j \in C_i} x_j$$

де  $n_i$  – кількість об'єктів, що належать  $i$ -му кластеру;

$x_j$  – точки, що належать кластеру  $C_i$ .

Таким чином, у наступній ітерації алгоритму отримується нове призначення з подальшим новим розташуванням центроїда. Цей процес повторюється до збіжності.

Загалом використовується критерій квадратичної похибки, який визначається наступним чином:

$$E = \sum_{i=1}^k \sum_{p \in C_i} \|p - m_i\|^2$$

де  $E$  – сума квадратичних похибок всіх об'єктів в базі даних,

$p$  – точка у просторі,

$m_i$  – середнє кластеру  $C_i$ .

Цільова функція робить згенеровані кластери максимально компактними та незалежними, використовуючи евклідову міру відстані. Також можна звернутися до інших видів відстані.

Переваги: простий і ефективний як для малих так і для великих наборів даних, з низькою складністю часу та простору. Недоліки: найважливішим є те, що результат легко локалізується, коли набір даних великий; необхідно заздалегідь встановити значення  $K$ , а алгоритм чутливий до першого вибору точки  $k$ .

Однак, однозначна кластеризація є достатньо грубим підходом для мало структурованих задач. За такої причини Енрік Г. Руспіні запропонував представлення кластерів як нечіткі множини [7]. Якщо  $\tilde{\chi}_j$  позначає функцію приналежності  $j$ -тому кластеру, то нечіткий розподіл є множиною нечітких підмножин  $\tilde{F}_1, \dots, \tilde{F}_k$ . Їхні функції приналежності визначаються таким чином:  $\sum_{j=1}^k \tilde{\chi}_j(p_i) = 1, i = 1, \dots, m$ .

Це визначення призначене для врахування викидів та інших нерівностей у викликаному розподілі. Крім того, щоб мати можливість визначити представників окремих кластерів, Данн ввів концепцію компактних і добре розділених (CWS) кластерів [8]. Щоби дозволити нечіткі розбиття множини об'єктів необхідно послабити критерій взятий з алгоритму K-means.

Нехай  $U = [u_{ij}]_{m \times k}$  – матриця, яка задовольняє наступні умови:

$$0 \leq u_{ij} \leq 1, \quad \sum_{j=1}^k u_{ij} = 1, \quad 0 < \sum_{i=1}^m u_{ij} < m, \quad i = 1, \dots, m; j = 1, \dots, k$$

Елемент  $u_{ij}$  визначає ступінь належності об'єкта  $p_i$  до класу  $C_j$ . Перша умова означає часткову належність об'єктів до класів, друга умова забезпечує повну належність об'єктів до виділеного набору класів, третя умова не дозволяє створювати порожні класи. Множина всіх матриць, що задовольняють ці умови, називається множиною нечітких розбиттів і  $U_{fk}$ . Також важливо, що розмірність простору нечітких кластеризацій значно перевищує розмірність простору чітких розподілів.

Прототипи класів представляють собою рядки матриці  $M = (\mu_1, \dots, \mu_k)^T$  розмірності  $k \times n$ .

Пошук нечіткого розбиття, що відповідає зазначеним умовам, здійснюється шляхом мінімізації показника якості форми:

$$E = \sum_{i=1}^m \sum_{j=1}^k u_{ij}^\alpha \|p_i - \mu_j\|^2$$

де  $\alpha > 1$  – показник нечіткості. Було показано [9], що коли  $\alpha \rightarrow 1$ , то алгоритм генерує кластери, ідентичні тим, що отримані з алгоритму ISODATA [10], а коли  $\alpha \rightarrow \infty$ , то значення  $u_{ij} \rightarrow 1/k$ , тобто отримуємо повністю нечітке розбиття. Типові значення параметра  $\alpha$  1,25 і 2.

Алгоритм *fuzzy C-means* (нечітких *C-середніх*) наступний [11]:

1. Ініціалізація початкового нечіткого розбиття:  $U = [u_{ij}]_{m \times k}$
2. Обчислення координат центроїдів кластерів:

$$\mu_{jl} = \frac{\sum_{i=1}^m u_{ij}^\alpha p_{il}}{\sum_{i=1}^m u_{ij}^\alpha}, \quad j = 1, \dots, k, \quad l = 1, \dots, n$$

3. Обчислення нових значень функцій приналежності:  $u_{ij} = \frac{d^{\frac{2}{\alpha-1}}(p_i, \mu_j)}{\sum_{l=1}^k d^{\frac{2}{\alpha-1}}(p_i, \mu_l)}$
4. Повторення кроків 2 та 3 до досягнення заданої точності.

Даний алгоритм реалізує базовий нечіткий розподіл, але існують також інші варіації нечіткого розподілу, наприклад алгоритм Густафсона-Кесселя [12] чи алгоритм Гаса-Геви [13].

В ієрархічні методи закладено підхід створення дерева з вкладених кластерів і, в залежності від того, поділяють вони більші кластери на менші, чи навпаки об'єднують менші кластери у більші, їх існує два типи: дивізійні та агломеративні відповідно.

Ієрархічний метод пропонує таке поняття як вкладене розбиття: розбиття  $\pi(v)$  вважається вкладеним в розбиття  $\pi(w)$ , якщо  $w < v$  і кожен кластер  $\pi(v)$  є підмножиною певних кластерів  $\pi(w)$ . Ідея найпопулярнішого, агломеративного, алгоритму полягає у тому, що з кожним кроком за певними ознаками будуть об'єднуватися два кластери з поточного  $k$ -розбиття. Ознаками можуть бути: відстань між найближчими об'єктами різних кластерів, середня відстань між всіма парами об'єктів в різних кластерах, відстань між центрами кластерів, та інші.

Робота агломеративного алгоритму наступна [14]:

1. Визначення вихідного розподілу одноелементними кластерами  $\Pi^{(m)} = \{\pi_1, \dots, \pi_k\}$
2. Побудова матриці схожості за обраною ознакою.
3. Об'єднання двох кластерів за обраною ознакою.
4. Кроки 2 та 3 повторювати доти, доки у множині  $\Pi$  буде більше одного елемента.

Оскільки в основі алгоритму дерево, а результатом кластеризації не має бути один кластер, є можливість зупинки при досягненні певної кількості кластерів. Коли ієрархічний метод об'єднав два об'єкти, їх більше неможливо розділити. Ця жорсткість є особливістю, що одночасно визначає перевагу у вигляді швидкодії та недолік у неможливості виправлення помилкових рішень.

Алгоритми на основі сітки вводять сітку в простір ознак. Їх реалізація часто схожа на комбінації щільнісного та сіткового підходів. Для аналізу просторових даних було створено метод на основі сітки статистичної інформації – STING (Statistical INformation Grid). Просторова зона поділяється на прямокутні комірки. Кілька різних рівнів таких прямокутних комірок різних розмірів утворюють ієрархічну структуру. Кожна комірка на високому рівні поділяється на декілька комірок наступного нижчого рівня.

Статистична інформація кожної комірки обчислюється та зберігається заздалегідь і використовується для відповідей на запити. Корінь ієрархії знаходиться на рівні 1, а з кожним наступним рівнем у комірок з'являються задана(за замовчуванням 4) дочірні, розмір яких базується на щільності об'єктів. Кожна комірка має свої параметри: кількість об'єктів в комірці, середнє значення всіх змінних у комірці, стандартне відхилення всіх наявних у комірці значень, максимальне і мінімальне значення та мітка статистичного розподілу, яка приймає наступні значення: «нормальний», «рівномірний», «відсутній».

Алгоритм STING [15]:

1. Визначити початковий шар.
2. Для кожної клітини цього шару обчислити довірчий інтервал ймовірності того, що ця комірка має відношення до запиту.
3. З інтервалу, обчисленого вище, позначити комірку як актуальну чи недоречну.
4. Якщо цей шар є нижчим шаром, перейти до кроку 6; В іншому випадку перейти до кроку 5.
5. Спуститися нижче по структурі ієрархії на один рівень. Перейти до кроку 2 для тих клітин, які утворюють відповідні клітини шару вищого рівня.
6. Якщо специфікація запиту виконується, перейти до кроку 8; В іншому випадку перейти до кроку 7.
7. Отримати ці дані, що потрапляють у відповідні клітини та зробити подальшу обробку. Повернути результат, який відповідає вимогам запиту. Перейти до кроку 9.
8. Знайти області відповідних клітин. Повернути ті регіони, які відповідають вимогам запиту. Перейти до кроку 9.
9. Кінець алгоритму.

Оскільки кожна комірка містить незалежний від запиту звіт за розташованими у ній даними, то сам алгоритм вважається незалежним від запиту. Формування сітки дозволяє паралельну обробку інформації та поступове оновлення. Структура сітки у вигляді прямокутних комірок призвела до обмеження та основного недоліку методу – неможливість виявлення діагональних меж кластерів.

Якщо припустити, що кластери можуть бути розподілені за допомогою вибіркового розподілу рішення про близькість. Тобто на невеликій відстані від будь-якого зразка в цьому кластері повинні бути зразки одного кластеру. В такому випадку необхідні алгоритми основані на щільності даних, одним з яких є DBSCAN (Density-Based Spatial Clustering of Applications with Noise). Категорію кластеризації отримують класифікуючи тісно пов'язані зразки в одну категорію. Класифікуючи всі групи тісно пов'язаних зразків у різні категорії, можна отримати кінцевий результат усіх кластеризованих категорій.

Для розуміння ідеї алгоритму визначається наступне:

Нехай існує дві точки  $x, y \in X$  та нехай  $\epsilon$  буде параметром.  $\epsilon$ -окіл об'єкту  $x \in$

$$N_\epsilon(x) = \{x' \in X: d(x, x') \leq \epsilon\}$$

Об'єкт  $x \in X$  називається внутрішньою точкою кластера, якщо її  $\epsilon$ -окіл містить хоча би  $minPts$  об'єктів, тобто коли  $|N_\epsilon(x)| \geq minPts$ , де  $minPts$  є параметром.

Об'єкт  $x \in X$  називається граничною точкою кластера, коли  $|N_\epsilon(x)| < minPts$ , але його окіл має хоча б одну внутрішню точку.

Якщо  $x \in X$  не є внутрішньою та не є граничною, тоді це вважається викидом [16].

При побудові кластеру даним методом між кожною точкою необхідно виявити її щільнісну досяжність: точка  $y \in X$  є безпосередньо щільно-досяжною з точки  $x \in X$ , якщо  $y \in N_\epsilon(x)$ , і, крім того,  $x$  є внутрішньою точкою, тобто оточена достатньо великою кількістю інших точок. Зв'язок прямої щільної досяжності є асиметричним. Тоді  $y$  називається щільно-досяжною з точки  $x$ , якщо існує послідовність  $x_1, \dots, x_n$  точок, така що  $x_1 = x, x_n = y$ , і кожна точка  $x_{i+1}$  безпосередньо щільно-досяжна з  $x_i, i = 1, \dots, n - 1$ . Визначене відношення асиметричне, а  $y$  може бути граничною точкою. Тому вводиться поняття щільної зв'язності: точки  $x, y \in X$  щільно-зв'язні, якщо існує така точка  $z \in X$ , що і  $x$ , і  $y$  щільно-досяжні з  $z$ .

Кластери, згенеровані алгоритмом DBSCAN, мають такі властивості:

- Усі точки, що належать кластеру, взаємно щільно-зв'язані.
- Якщо внутрішня точка щільно-зв'язана з іншою точкою кластера, то вона також є елементом цього кластера.

Алгоритм DBSCAN [14]:

Вхідні дані: Дані  $X = (x_1, \dots, x_m)^T$ , максимальна відстань між внутрішніми точками.

Результат: Розбиття  $C = \{C_1, \dots, C_k\}$ , кількість кластерів  $k$ .

1. Ініціалізація. Позначення точок з множини  $X$  як внутрішні, граничні або шумові точки.
2. Видалення зашумлених точок.
3. З'єднання ребром сусідніх внутрішніх точок (тобто внутрішні точки, розташовані на відстані не більше  $\epsilon$ ).

4. Формування кластеру із сусідніх внутрішніх точок.

5. Призначення граничних точок одному з кластерів, з якими вони знаходяться по сусідству.

На підставі аналізу роботи алгоритму DBSCAN доцільно підкреслити такі його особливості:

- метод стійкий до шуму, який розташований далеко від ядра щільності;
- відсутня необхідність задавати кількість кластерів.
- можуть бути знайдені кластери довільної форми.

Проте він має і ряд несуттєвих недоліків:

- час збіжності кластеризації є відносно тривалим при великому наборі даних;
- порівняно із традиційним алгоритмом кластеризації K-середніх, він вимагає спільного налаштування двох параметрів: порогового значення відстані  $\epsilon$  та порогового значення  $minPts$  кількості зразків сусідства.

Задля забезпечення виявлення перешкод у неструктурованому середовищі ієрархічні методи та ті, для роботи яких необхідно задати кількість кластерів, недоречно використовувати. Методи на основі сітки не ефективно реалізуються для крокуючого робота, тому найпростішим і більш вдалим вибором будуть методи основані на щільності даних.

Основним з таких методів є вже описаний DBSCAN, у якості вхідних даних приймається хмара точок із зображенням навколишнього середовища та два параметри  $minPts$  та  $\epsilon$ . Вибір мінімальної кількості точок  $minPts$  залежить від налаштувань сенсору (LIDAR, RGB-D камери, камери глибини, тощо), завдяки якому буде отримано хмару точок, та певних припущень щодо зашумленості середовища, яке аналізується. Параметр  $\epsilon$  вимагає налаштувань емпіричним методом і може бути обраним у відсотках, по відношенню до загального розміру хмари точок. Існують також математичні методи для визначення цих параметрів, наприклад автори зазначеного вище методу запропонували створення відсортованого графу  $k$ -відстаней і інтерактивного визначення необхідних параметрів [16].

На основі DBSCAN створено багато методів, як наприклад OPTICS (Ordering Points To Identify the Clustering Structure) [17], який будує граф досяжності і використовуючи його знаходить кластери. Однак метод потребує більших обчислень, а невірно підібрані параметри можуть сильно спотворити кластери на виході. Ще одним методом, заснованим на щільності є MeanShift [18], який потребує лише один вхідний параметр – розмір вікна, в якому проходить пошук локальних максимумів у щільності даних. Параметр визначає наскільки далеко точки можуть бути розташовані одна від одної, знаходячись в одному кластері. Додатковими налаштуваннями може бути специфічна умова зупинки алгоритму, кількість ітерацій, або вказання наявного шуму у відсотках. Однак обидва алгоритми дуже чутливі до форми кластерів.

### Вибір методу та експеримент

Підготовка першої хмари точок (рис. 1), ділянка лісу з деревами та нерівною поверхнею [19], та усіх інших до кластеризації буде включати у себе відділення поверхні методом RANdom SAmple Consensus (RANSAC) [20].

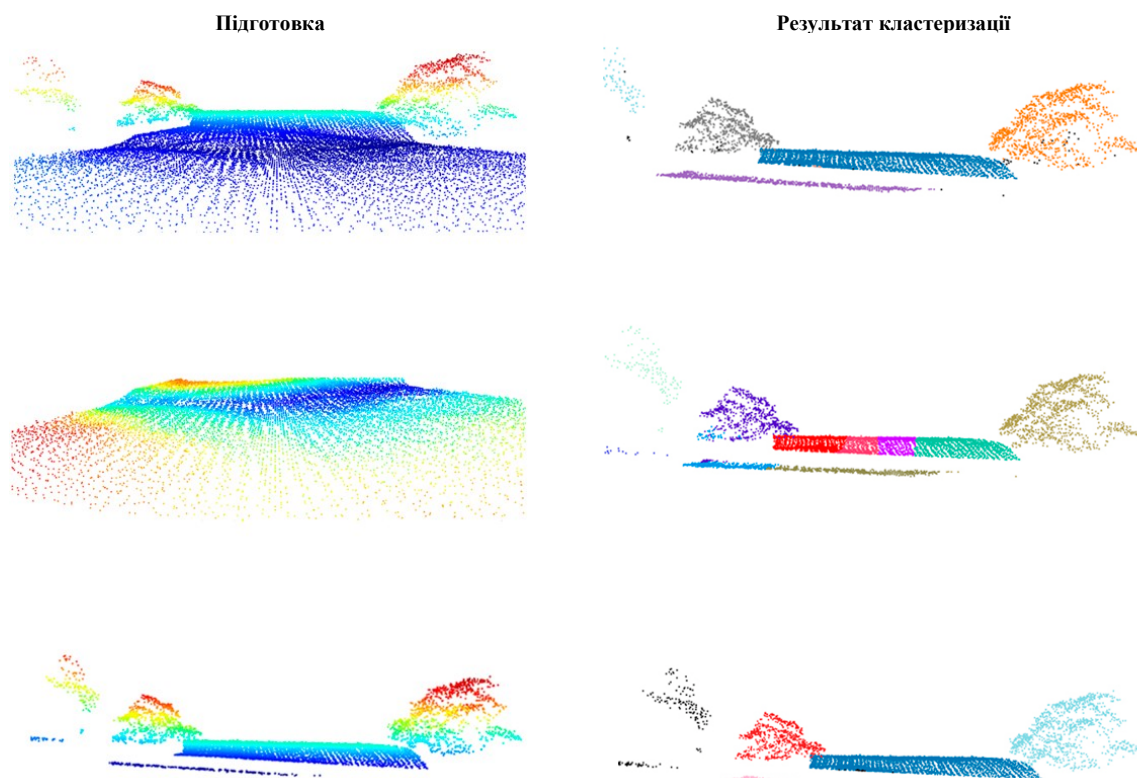


Рис. 1. Етапи підготовки даних та результати кластеризації

До масиву точок, що залишився, застосовано три обраних методи кластеризації: *DBSCAN* з параметрами  $minPts = 10$  та  $\epsilon = 2$ ; *MeanShift* з параметрами: розмір вікна дорівнює 7 та наявністю шуму; *OPTICS* на основі *DBSCAN* та із відношеннями такими, що розмір кластеру повинен бути не менш ніж 5% точок всієї хмари і відстань між точками не більше 5% розміру хмари. Шум буде позначено «-1» кластером та виділено чорним кольором.

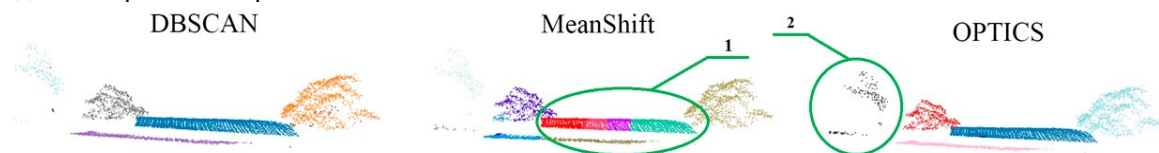


Рис. 2. Візуальний аналіз результату кластеризації різними методами

За візуальним аналізом результатів (рис. 2), методом *DBSCAN* визначено усі кластери та знайдено шумові точки протягом 1886 мс. Метод *MeanShift*, маючи в налаштуваннях наявність шуму його не знайшов, а цілий кластер «пагорб», позначений як «1» на рисунку 2, розбив на 4 частини. Час виконання однієї ітерації 9583 мс, якщо підвищити кількість ітерацій до 10, то – 37474 мс. Гарні результати за 6505 мс показав *OPTICS*, але він відкинув цілий кластер 2 (рис. 2) на передньому фоні. Обидва методи, які показали гарний результат, реалізують у собі алгоритм кластеризації *DBSCAN*. З цієї причини він може стати гарним варіантом для навігації крокуючого робота і буде використаний у наступних випробуваннях.

У наступній хмарі точок буде суттєва природна перешкода «обрив». Перед роботом знаходиться близько метра площини на якій він розташований, а за нею обрив, тому зібрані точки у хмарі: крони дерев і площини до обриву та за ним. Через переважну кількість точок у далекій площині RANSAC виділив саме її (рис. 3).

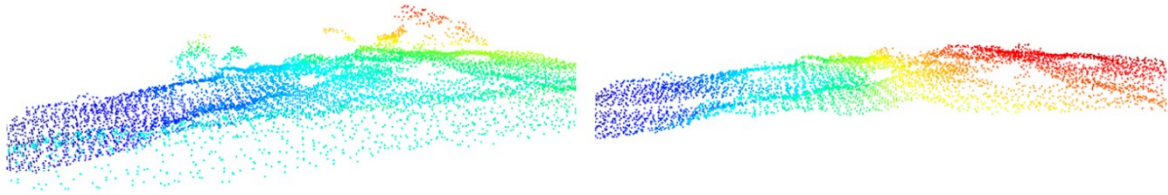


Рис. 3. Підготовчі етапи формування хмар точок для подальшої кластеризації

На кластеризацію поступили такі дані, як площина на якій стоїть робот, заглиблення у площині за обривом і крони дерев (рис. 4). Результат містить у собі гарно сформовані кластери і відсіяні шумові точки. На його аналізі можна побудувати маршрут робота на обхід цього рову. З хмари точок можна отримати такі дані як висота, ширина кластеру, знаходиться він вище або нижче площини переміщення робота, і відповідний масив даних використати при подальшій обробці для навігації.

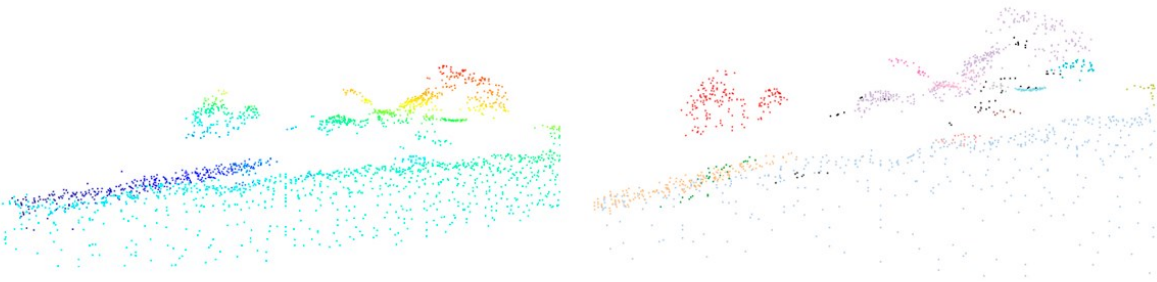


Рис. 4. Хмара точок до та після кластеризації

Остаточним експериментом буде хмара точок ярмарку [21], зібраною більш потужним LIDAR-ом (лазерним далекоміром), тому поверхня визначилась точніше (Рис. 5).

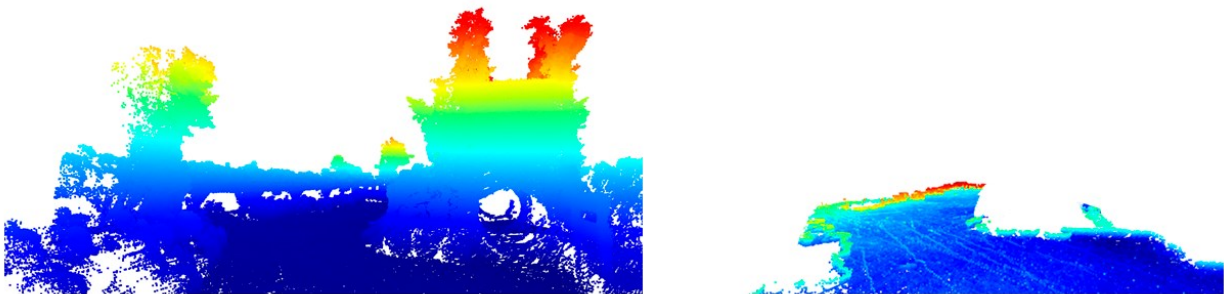


Рис. 5. Підготовка хмара точок для кластеризації

Досліджуваним методом отримані гарні результати (Рис. 6), де визначено кластери та виявлено шумові точки.

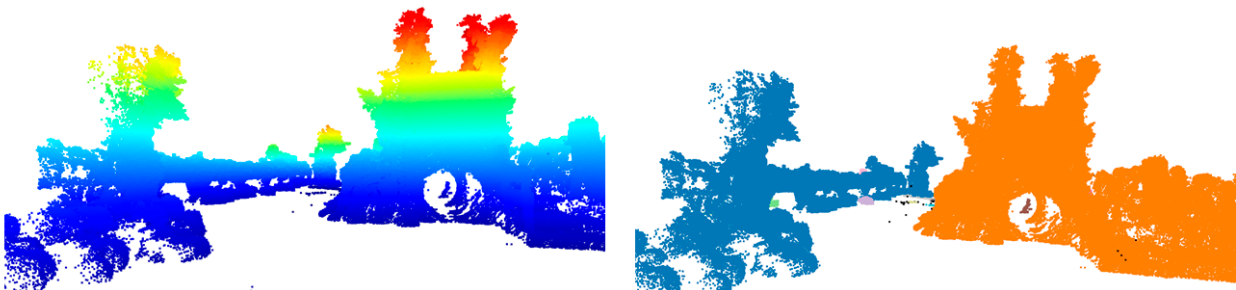


Рис. 6. Хмара точок до та після кластеризації

Примітно, що налаштування методу *DBSCAN* залишались незмінними і в усіх трьох випадках показали задовільне виконання поставленої задачі, але звісно для кожної ситуації існують свої оптимальні налаштування параметрів *minPts* та  $\epsilon$ .

### Висновки з даного дослідження і перспективи подальших розвідок у даному напрямі

Враховуючи сказане вище, методи які базуються на ієрархії та розбитті не відповідають вимогам, що висувуються до навігації крокуючих роботів в неструктурованих середовищах, оскільки кількість кластерів невідома заздалегідь, а методи на основі сітки розраховані на дані більшої розмірності і їх реалізація для крокуючого робота була би не доцільна. Водночас методи, що базуються на щільності даних можуть бути використані для даних умов, проте результати аналізу їх працездатності вказують, що метод *MeanShift* недостатньо якісно проводить кластеризацію хмар точок на заданих наборах даних, а метод *OPTICS* значну частину змістовних даних відсіює у шум, що є суттєвим недоліком порівняно із методом *DBSCAN*. Отже, для подальших досліджень та розв'язання поставлених задач навігації крокуючих роботів у неструктурованих середовищах буде застосовано саме метод *DBSCAN*.

### References

1. IGI Global. What is unstructured environment. <https://www.igi-global.com/dictionary/unstructured-environment/46619>
2. Kaufman, L., Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis* .. John Wiley. ISBN: 978-0-47031680-1
3. Hennig, C., Meila, M., Murtagh, F., & Rocci, R. (Eds.). (2015). *Handbook of Cluster Analysis* (1st ed.). Chapman and Hall/CRC. <https://doi.org/10.1201/b19706>
4. Rudolf Scitovski, Kristian Sabo, Francisco Martínez-Álvarez, Šime U. (2021) *Cluster Analysis and Applications* .. Springer. ISBN: 978-3030745516
5. Cormack, R. M. (1971) A review of classification. *Journal of the Royal Statistical Society A*, 134, 321–367.
6. Aggarwal, C.C., & Reddy, C.K. (Eds.). (2014). *Data Clustering: Algorithms and Applications* (1st ed.). Chapman and Hall/CRC. <https://doi.org/10.1201/9781315373515>
7. E.H. Ruspini, A new approach to clustering. *Inf. Control* 15, 22–32 (1969). [https://doi.org/10.1016/S0019-9958\(69\)90591-9](https://doi.org/10.1016/S0019-9958(69)90591-9)
8. J.C. Dunn, A fuzzy relative of the Isodata process and its use in detecting compact wellseparated clusters. *J. Cyber.* 3(3), 32–57 (1974)
9. J.C. Bezdek, S.K. Pal, *Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data* (IEEE, New York, 1992)
10. G. II. Ball and D. J. Hall, "ISODATA, an iterative method of multivariate analysis and pattern classification," *Behavioral Science*, vol. 12, pp. 153-155, 1967.
11. Höppner, Frank & Klawonn, Frank & Kruse, Rudolf & Runkler, Thomas. (2000). *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*. The Journal of the Operational Research Society. 51. 10.2307/254022.
12. Gustafson E. E., Kessel W. C. Fuzzy clustering with a fuzzy covariance matrix // In: Proc. IEEE CDC. – San Diego, California, 1979. – p. 761–766.
13. Gath I., Geva A. B. Unsupervised optimal fuzzy clustering // IEEE Trans. on Pattern Analysis and Machine Intelligence. – 1989. – 11. – p. 773–781.
14. Wierzchon, Slawomir & Kłopotek, Mieczyslaw. (2018). *Modern Algorithms of Cluster Analysis*. 34. 10.1007/978-3-319-69308-8.
15. Wang, W., Yang, J., Muntz, R., & Shanahan, J. (1997). STING: A statistical information grid approach to spatial data mining. *Proceedings of the 23rd International Conference on Very Large Data Bases (VLDB)*, 186-195.
16. Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD)*, 226-231.
17. Ankerst, M., Breunig, M. M., Kriegel, H.-P., & Sander, J. (1999). OPTICS: Ordering points to identify the clustering structure. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 49-60.
18. Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5), 603-619.
19. The National Ecological Observatory Network LIDAR Data Sample [Електронний ресурс] URL: <https://data.neonscience.org/data-products/explore> .
20. Martin A. Fischler and Robert C. Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 6 (June 1981), 381–395. <https://doi.org/10.1145/358669.358692>
21. Nall's pumpkin hill, USA, GeoSLAM Sample Point Cloud Data. [Електронний ресурс] URL: <https://geoslam.com/sample-data/>