SÉRIE “Technické vědy”

[https://doi.org/10.52058/2695-1592-2021-3\(3\)-103-113](https://doi.org/10.52058/2695-1592-2021-3(3)-103-113)

Наталія Бойко

*кандидат економічних наук, доцент кафедри Системи штучного інтелекту,
Національний університет «Львівська політехніка, м. Львів, Україна, e-mail:
nataliya.i.boyko@lpnu.ua,
<https://orcid.org/0000-0002-6962-9363>*

ПОРІВНЯННЯ ЕФЕКТИВНОСТІ МЕТОДІВ КЛАСТЕРИЗАЦІЇ МУЛЬТИМОДАЛЬНИХ ДАНИХ З ВІДДАЛЕНИХ ОБ'ЄКТІВ ЗА ДОПОМОГОЮ АКТИВНИХ ОПТИЧНИХ СИСТЕМ

Анотація. З кожним роком робототехніка розвивається все більше і більше. Створюються нові підходи до вирішення проблем руху, локалізації, автоматизації роботів. Багато моделей досягають чималих успіхів у вирішенні різних завдань. Чимало технічних комплексів створюється для військових цілей: виявлення цілі, її ліквідації. Створюються роботи пожежники; роботи рятувальники, здатні діставати людей з води, з уламків повалених будівель. Однією з багатьох тенденцій в робототехніці є перехід від телекерованих систем, які вимагають постійної участі людини для виконання всіх дій робота до автономних систем, в яких оператор лише вказує кінцеві і проміжні цілі. Це зручно для проведення інопланетних досліджень, де велика затримка сигналу не дозволяє здійснювати телеуправління.

На сьогоднішній день проблема автономності роботів дуже актуальна. Автономія вимагає від робота сприйняття навколишнього середовища. Знаючи карту місцевості, робот без проблем зможе визначити місцезнаходження об'єктів в просторі. Одна з труднощів виникає, коли робот не має уявлення про місцевість і не знає своїх координат. У цьому випадку йому необхідно здійснювати рухи і при цьому створювати карту місцевості, використовуючи різні сенсорні пристрої і алгоритми.

Ключові слова: алгоритм, лідар, кластеризація, мультимодальні дані, сенсор.

Nataliya Boyko

*Ph.D., Associated Professor at the Department of Artificial Intelligence, Lviv
Polytechnic National University,
Lviv, Ukraine, e-mail: nataliya.i.boyko@lpnu.ua,
<https://orcid.org/0000-0002-6962-9363>*

COMPARISON OF EFFICIENCY OF METHODS OF CLUSTERIZATION OF MULTIMODAL DATA FROM REMOTE OBJECTS USING ACTIVE OPTICAL SYSTEMS

Abstract. Every year robotics develops more and more. New approaches to solving problems of movement, localization, automation of robots are created. Many models achieve considerable success in solving various problems. Many technical complexes are created for military purposes: target detection, its elimination. Firefighters are working; rescuers work, able to get people out of the water, from the rubble of fallen buildings. One of the many trends in robotics is the transition from telecontrolled systems, which require constant human participation to perform all the actions of the robot to autonomous systems, in which the operator only specifies the ultimate and intermediate goals. This is convenient for alien research, where a large signal delay does not allow for remote control.

To date, the problem of robot autonomy is very relevant. Autonomy requires the robot to perceive the environment. Knowing the map of the area, the robot will be able to easily determine the location of objects in space. One of the difficulties arises when the robot has no idea of the terrain and does not know its coordinates. In this case, he needs to make movements and at the same time create a map of the area using various sensor devices and algorithms.

Keywords: algorithm, leader, clustering, multimodal data, sensor.

Постановка проблеми. За останні декілька років робототехніка досягла великих результатів, але тим не менше відкритих задач у цій сфері є достатня кількість. Однією з таких задач є задача кластеризації даних з 2д лідару. Лідар - це технологія отримання та обробки мультимодальних даних про віддалені об'єкти за допомогою активних оптичних систем, що використовують явища відбиття світла і його розсіювання в прозорих і напівпрозорих середовищах.

Аналіз останніх досліджень і публікацій. У роботі «LiDAR Clustering and Shape Extraction for Automotive Applications Master's thesis in Systems, Control and Mechatronics» [7, 14], автор описує різні системи координат, закінчуючи методами кластеризації точок зібраних з даних лідару. Результати обробки даних були гарно проаналізовані а також візуалізовані на різноманітних типах графіків.

У статті «Algorithmic clustering of LiDAR point cloud data for textural damage identifications of structural elements» [4, 9] автор порівнює два методи класифікації для конкретної проблеми для визначення пошкоджень на текстурі. У статті запропонований алгоритм аналізу для порівняння методів для пошуку прикладних задач, які неможливо вирішити за допомогою лідару.

В роботі «A Clustering Method Based on K-Means Algorithm» [10, 15] автор пояснює роботу алгоритму к-середніх. Пропонує свої рекомендації щодо його застосування, а також візуалізує результати виконання алгоритму за допомогою стислого подання даних у таблицях.

У публікації «A Density-Based Algorithm for Discovering Clusters in Large

Spatial Databases with Noise» [8, 12] автор пояснює роботу алгоритму кластеризації DBSCAN. Ця стаття зосереджена на великих датасетах, які містять шуми. Це ідеально підійшло під дані з лідар датчика, оскільки він вимірює з похибкою і часто в даних можуть з'явитись шуми. Крім того високоточні лідар датчики генерують нам дуже багато даних, які потрібно опрацьовувати. В даній статті проілюстровано скільки часу триває алгоритм при різній кількості вхідних мультимодальних даних. Вона допомогла мені з вибором кількості вхідних даних, оскільки їх можна контролювати.

У дослідженні «Алгоритм кластеризації OPTICS» [9, 13] пояснюється алгоритм OPTICS. У ній наводиться його відмінність від алгоритму DBSCAN та представлений псевдокод, що допомагає реалізувати цей алгоритм власноруч.

Метою статті є використання відомих методів кластеризації для обробки мультимодальних даних з 2д лідару, порівняти їх ефективність. Також однією з задач є оптимізація методів для конкретних підзадач, їх параметрів.

Для досягнення зазначеної мети вирішуються наступні завдання:

1. Вивчення алгоритмів кластеризації мультимодальних даних з віддалених об'єктів за допомогою активних оптичних систем.
2. Реалізація алгоритмів локалізації мобільного робота з використанням мови програмування Python.
3. Вивчення впливу розрахунку коефіцієнта силуету для кластеризації мультимодальних даних.
4. Дослідження та аналіз роботи алгоритмів кластеризації для двовимірного простору.

Теоретичною основою для написання даної роботи послужили праці зарубіжних авторів з проблем автоматизації і локалізації в робототехніці.

Виклад основного матеріалу. Для проведення експериментів було обрано наступні алгоритми:

1. Алгоритм OPTICS - це алгоритм знаходження щільності на основі кластерів у просторових даних. Він був представлений Міхаелем Анкерстом, Маркусом Брюінгом, Хансом Крігелем і Йоргом Сандером [1, 11].
2. Алгоритм DBSCAN - це алгоритм кластеризації, заснований на щільності, тобто він групує разом точки, які тісно розташовані позначаючи як викиди точки, які знаходяться самотньо в областях з малою щільністю (найближчі сусіди яких лежать далеко) [3, 6].
3. Алгоритм k-середніх - теж один з найбільш використовуваних методів кластеризації. Він розбиває безліч елементів векторного простору на задане число кластерів k [2, 5].

Отож, розглянемо детальніше роботу кожного алгоритму.

Для алгоритму DBSCAN потрібні два параметри: ϵ (*eps*) і мінімальна кількість точок необхідних для утворення щільної області (*minPts*). Побудова починається з довільної точки, яка раніше не використовувалась. Виконується запит щодо точок з ϵ -околу, і, якщо там буде достатньо точок, то утворюється

кластер. Інакше, точка маркується як шумова. Зауважимо, що ця точка може бути знайдена пізніше в достатньо великому ε -околі іншої точки і належати кластеру, а не бути шумовою.

Якщо точка знаходиться у щільній частині кластера, то і її ε -окіл також буде частиною цього кластера. Отже, всі точки ε -околу також будуть додані у кластер, так само і ε -околи цих точок, якщо вони також будуть щільними. Цей процес продовжується доти, поки щільно-зв'язний кластер не буде знайдено повністю. Після цього виконується запит на нову необроблену точку, яка опрацьовується і, або знаходиться новий кластер, або вона позначається як шумова.

Для роботи алгоритму OPTICS (як і DBSCAN) потрібно два параметри - ε та $MinPts$. Працює він за схожим принципом, проте як більш розширений алгоритм DBSCAN для нескінченної кількості відстаней, які менші, ніж основний "радіус сусідства". Різниця полягає в тому, що членство кластерів не задається, замість цього зберігається порядок, в якому об'єкти обробляються і інформація, яка буде використовуватися розширеним алгоритмом DBSCAN для призначення членства в кластері (якщо це було взагалі можливо для нескінченного числа параметрів). Ця інформація складається лише з двох значень для кожного об'єкта: основна відстань та відстань доступності.

У алгоритмі k -середніх визначається кількість кластерів, які необхідно утворити. Випадковим чином обирається k спостережень, які на цьому кроці вважаються центрами кластерів. Кожне спостереження «приписується» до одного з n кластерів — відстань до якого найкоротша. Розраховується новий центр кожного кластера як елемент, ознаки якого розраховуються як середнє арифметичне ознак об'єктів, що входять у цей кластер. Відбувається така кількість ітерацій (повторюються кроки 3-4), поки кластерні центри стануть стійкими (тобто при кожній ітерації в кожному кластері опиняться одні й ті самі об'єкти), дисперсія всередині кластера буде мінімізована, а між кластерами — максимізована.

Для початку експерименту опишемо, що нам відома точка $C(x, y)$, де x та y належать полю дійсних чисел. Точка C задає початкове положення датчика. Крім цього нам відомий кут α , це кут, з яким датчик випускає промені щоб отримати відстань до найближчої перешкоди. І останнє це параметр d_i це відстань яку знайшов i -тий промінь до найближчої перешкоди.

З цих трьох параметрів можна порахувати $M_i(x, y)$, де x та y належать полю дійсних чисел. Точка M_i задає положення перешкоди об яку відбився промінь. І вектор $vec(x, y)$, де x та y належать полю дійсних чисел. vec буде здавати вектор руху робота при зміні даних з датчика. Основним параметром для нас є точки M_i , оскільки саме їх ми будемо класифікувати. Решта параметрів потрібні, для підтримки положення робота у просторі а також для знаходження наших основних параметрів. Отже тип даних для роботи є `float`. Всі операції, що проводились у експериментальній є визначені над полем дійсних чисел.

Розглянемо аналіз складності алгоритмів в залежності від типу даних. Отож, алгоритм k -середніх: для нього аналітично виводиться часова складність кластеризації n d -мірних векторів на k кластерів за i ітерацій $O(i*k*d*n)$.

Адаптуючи цю складність для поставленої задачі, проаналізувавши мультимодальні дані можна зробити висновок, що вона еквівалентна $O(n^2)$. Вимірність вектора у нас 2, тому цим можна знехтувати, кількість кластерів обиралась невелика і цим теж можна знехтувати при виведенні асимптотичної складності.

У загальному випадку алгоритм DBSCAN має квадратичну обчислювальну складність через пошук *Eps*-сусідів $O(N^2)$. Однак автори [13] алгоритму використовували для цієї мети спеціальну структуру даних - R-дерева, в результаті пошук *Eps*-сусідства для однієї точки становить - $O(\log(n))$. Загальна обчислювальна складність DBSCAN - $O(n \cdot \log(n))$.

Подібно DBSCAN алгоритм OPTICS обробляє кожну точку тільки один раз і виконує один запит під час цієї обробки. Якщо заданий *eps*, гарантує роботу запиту сусідства за час $O(\log(n))$, то отримаємо загальний час роботи $O(n \cdot \log(n))$. Автори статті [11] по OPTICS повідомляють про константне сповільнення в 1,6 рази в порівнянні з DBSCAN. Зауважимо, що значення *eps* може сильно впливати на витрати алгоритму, оскільки занадто велике значення може збільшувати складність запиту до лінійної. І загальна складність збільшиться до квадратичної.

Наступним кроком є вимірювання часу на виконання алгоритмів під час оброблення мультимодальних даних. Для цього дані беруться з власної реалізації програмного забезпечення. При проведенні експерименту було використано бібліотеку *sklearn* на мові програмування *python*. Параметри алгоритмів підбиралися найбільш ефективними для об'єктивного порівняння їх між собою. Тести проводились на робочій машині з наступними характеристиками:

- CPU: Intel core i5 3.8GHz (4 cores and 4 virtual cores);
- RAM: 4094 MB (1200 MHz);

В Таблиці 1 наводяться результати тестувань.

Таблиця 1

Заміри часу виконання алгоритмів

Назва алгоритмів	n=400	n=4000	n=10000	n=50000
k-середніх	0,03	0,08	0,06	0,6
DBSCAN	0,06	0,35	1,3	25
OPTICS	0,03	0,4	1,8	55

Заміри часу виконання, що наведені в Табл. 1 показали нам, що алгоритм k-середніх є найшвидшим серед своїх конкурентів. Це зумовлюється тим, що обчислювальна складність алгоритм є лінійною функцією. Найдовше відпрацьовував алгоритм OPTICS, при великій кількості точок може не справлятися з поставленою задачею, особливо коли динаміка руху робота є великою. Також візуально це представлено на Рис.1.

k-середніх, DBSCAN і OPTICS

Рис. 1. Візуалізація росту часу виконання в залежності від n .

На Рис. 2 за допомогою методу кластеризації k-means можна побачити кластеризовані точки. Зміст який ми можемо надати даному методу кластеризації – це поділ перешкод на певні сектори навколо себе. Цей алгоритм хоча і є простим в обчисленні, але він може бути недостатньо змістовний для нашого типу задачі.

Рис 2. Візуалізація кластерів для алгоритму k-середніх

На Рис. 3 представлено кластеризовані точки за допомогою методу

кластеризації DBSCAN. Зміст який ми можемо надати даному методу кластеризації – це виділення реальних об'єктів. Тут можна чітко відрізнити кожен лінію, адже вона розфарбована в унікальний колір. Цей алгоритм показує себе одним з найкращим для задач уникнення перешкоди, оскільки після кластеризації, кожен кластер представляє собою окрему перешкоду, яку можна уникнути знаючи де вона починається і де закінчується.

Рис 3. Візуалізація кластерів для алгоритму DBSCAN

На Рис. 4 бачимо кластеризовані точки за допомогою методу кластеризації OPTICS. Подібно до алгоритму DBSCAN він кластером робить окрему перешкоду. Але є такі лінії які містять в собі дані з різних кластерів, оскільки цей алгоритм менш залежний від коефіцієнта eps . Цей метод показав себе краще у візуальній частині кластеризації, а ніж k-середніх, та гірше за DBSCAN.

Рис 4. Візуалізація кластерів для алгоритму OPTICS

Розрахуємо коефіцієнти силуету, для дослідження роботи алгоритмів кластеризації. Коефіцієнт силуету - це одна з основних оцінок алгоритмів кластеризації. Вона показує наскільки схожими є дані одного кластеру, та наскільки відрізняються дані між різними кластерами. Формула обчислення даного коефіцієнта для окремого випадку це (Формула 1):

$$S = \frac{b - a}{\max(a, b)} \quad (1)$$

Значення варіюються в проміжку [-1, 1], де від'ємні значення свідчать про некоректну кластеризацію а близькі до одиниці – про якісну. В Таблиця 2 наведені розраховані коефіцієнти силуету по кожному алгоритму.

Таблиця 2

Розрахунок коефіцієнту силуету

Назва алгоритмів	Коефіцієнт силуету
k-середніх	0,4948
DBSCAN	0,8129
OPTICS	0,6098

Найкращий коефіцієнт силуету має алгоритм кластеризації DBSCAN. Що підтверджується попереднім експериментом з візуалізацією. Найгірший коефіцієнт силуету є у метода k-середніх. Це досить логічно оскільки багато точок одного кластеру межують з точками іншого.

Отож, алгоритм k-середніх показав себе найшвидшим, серед тих що ми розглядали, його час виконання залежить лінійно від кількості даних, які ми передаємо йому на вхід. Візуальний зміст методу не є дуже ефективним для уникнення перешкод. Також коефіцієнт силуету як міра ефективності методу кластеризації у нього вийшов найгіршим.

Алгоритм DBSCAN показав себе достатньо хорошим хоча не найшвидшим, серед тих що ми розглядали, його час виконання залежить в середньому квадратично від кількості даних які йому передаються на вхід. Візуальний зміст методу є чітким. Він чітко розділяє 2-х вимірні об'єкти на координатній площині. Це дуже добре, оскільки кластер і є по суті перешкодою. Даний метод показав найкращий коефіцієнт силуету.

Алгоритм OPTICS показав себе найгірше по швидкості. Він не підходить для задач пов'язаних з 2д лідаром, оскільки такі задачі потребують менш затратного алгоритму по часу. Візуальний зміст дещо схожий на DBSCAN але з певними кластерами “шумами”. Коефіцієнт силуету в нього кращий за k-середніх, але гірший від DBSCAN.

Отже, коротко підсумовуючи, найкращим методом по більшій кількості параметрів себе показав DBSCAN. У задачах пов'язаних з даними лідару, він буде встигати відпрацьовувати по часу, також він чітко ділить об'єкти на площині та має найкращий показник коефіцієнту силуету.

Висновки. Проводячи цю роботу на меті було дослідити декілька основних питань, окрім безпосередньо реалізації алгоритму. По-перше, хотілося повидити себе краще, а який гірше. Саме ці два показники і впливають на вибір науковців

для здійснення кластеризації мультимодальних даних, які передаються на відстані. У проведених експериментах були представлені та візуалізовані результати завдань. Також проводилась оцінка складності використаних алгоритмів кластеризації мультимодальних даних. Вони працюють з сильно дискретизованими та багатовимірними просторами. На практиці дані алгоритми потрібно застосовувати хоча б в дво-вимірному просторі (де до уваги беруться дві координати).

Також у роботі досліджувався вплив сенсорів для посилення та отримання мультимодальних даних. Для цього розраховувався коефіцієнт силуету та затрата часу. Дослідження показало, що залежність лінійна. Тобто кількість сенсорів алгоритмів лінійно впливає на час передавання мультимодальних даних.

Реалізовані алгоритми, а саме їхня візуалізація показує, як впливають розраховані коефіцієнти на кластеризацію мультимодальних даних, які передаються на відстані. Саме коефіцієнт силуету конкретизує позицію сенсора. Якщо сенсор бачить тільки дані одного типу, він вважає, що робот знаходиться у всіх місцях де дані цього типу одночасно. Але з виконанням операції пересування, позиція об'єкта весь час конкретизується і об'єкт більш точно себе локалізує.

Також важлива перевага реалізованих алгоритмів – це здатність легко візуалізувати процес (звісно в одновимірному або двовимірному просторі). Це відбувається за рахунок того, що на кожній ітерації розраховується тензор ваг.

Підсумовуючи переваги розрахованих коефіцієнтів, можна сказати що для дослідження вибрані досить точні алгоритми кластеризації мультимодальних даних. Їхній спектр використання не такий широкий, як у інших фільтрах частинок, але вони не сильно ресурсозатратні для проектів, в яких робиться акцент на точність.

В даній роботі було показано можливість застосування засобів кластерного аналізу для досліджень бази просторових мультимодальних даних згенерованих симулятором 2-д лідару. Підсумовуючи зроблену роботу, зазначимо, що засоби кластерного аналізу чудово проявили себе для аналізу мультимодальних даних взятих з 2д лідару. Результати роботи кожного з алгоритмів були наведені у вигляді рисунків з кластерами, які були розфарбовані в окремі кольори. Результат порівняння алгоритмів кластеризації засвідчив, що не всі обрані алгоритми добре зарекомендували себе на вхідному наборі даних.

Найкращим алгоритмом себе зарекомендував DBSCAN. Застосування як алгоритмів, так і програмних засобів аналізу просторових мультимодальних даних у сфері робототехніки представляє широке поле для подальших досліджень та вдосконалень.

Література:

1. Мінін А. А. Навігація і управління мобільним роботом, оснащеним лазерним далекомірором [Електронний ресурс]. - Режим доступу: <https://www.dissercat.com/content/navigatsiya-i-upravlenie-mobilnym-robotom-osnashchennym-lazernym-dalnomerom>.

2. Thrun S. Google's driverless car [Електронний ресурс]. - Режим доступу: http://www.ted.com/talks/sebastian_thrun_google_s_driverless_car.
3. Team page Stanford Racing [Електронний ресурс]. - Режим доступу: <http://cs.stanford.edu/group/roadrunner/stanley.html>.
4. Большаков А.А. Управління рухом мобільного робота [Електронний ресурс] / А.А. Большаков, Д.Л. Лисицький. - Режим доступу: <https://cyberleninka.ru/article/n/upravlenie-dvizheniem-mobilnogo-robota>.
5. Dunn M. Mobile robotics (Robot). Module 1 [Електронний ресурс]. - Режим доступу: <https://learn.open2study.com/mod/youtube/view.php?id=79916>.
6. Borenstein J. Where am I? Sensors and Methods for Mobile Robot Positioning [Електронний ресурс] / J. Borenstein, H.R. Everett, L. Feng. - Режим доступу: <https://www.yumpu.com/en/document/read/33598354/where-am-i-sensors-and-methods-for-mobile-robot-positioning>.
7. Siegward R. Introduction to Autonomous Mobile Robots [Електронний ресурс] / R. Siegward, I.R. Nourbakhsh, D. Scaramuzza. - Режим доступу: <https://mitpress.mit.edu/books/introduction-autonomous-mobile-robots-second-edition>.
8. Choset H. Principles of Robot Motion: Theory, Algorithms, and Implementations [Електронний ресурс] / H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, S. Thrun. - Режим доступу: <http://mathdep.ifmo.ru/wp-content/uploads/2018/10/Intelligent-Robotics-and-Autonomous-Agents-series-Choset>.
9. Arbib M. Depth and Detours: An Essay on Visually Guided Behavior [Електронний ресурс] / M. Arbib, A. Hanson. - Режим доступу: <https://journals.sagepub.com/doi/10.1177/105971239500300404>.
10. Koren Y. Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation [Електронний ресурс] / Y. Koren, J. Borenstein. - Режим доступу: https://www.cs.cmu.edu/~motionplanning/papers/sbp_papers/integrated1/borenstein_potential_field_1imitations.pdf.
11. Borenstein J. Navigating Mobile Robots: Sensors and Techniques [Електронний ресурс] / J. Borenstein, H. Everett, L. Feng. - Режим доступу: <https://dl.acm.org/doi/10.5555/546829>.
12. Puttkamer E. Autonome Mobile Roboter [Електронний ресурс] / E. Puttkamer, E. Von. - Режим доступу: <https://dl.acm.org/profile/81100649191>.
13. Lumelsky V. Dynamic Path Planning for a Mobile Automation with Limited Information on the Environment [Електронний ресурс] / V. Lumelsky, A. Stepanov. - Режим доступу: <https://dl.acm.org/profile/81100649191>.
14. Kamon I. Sensory-Based Motion Planning with Global Proofs [Електронний ресурс] / I. Kamon, E. Rivlin. - Режим доступу: <http://www.cs.technion.ac.il/~ehudr/publications/pdf/KamonR95i.pdf>.

References:

1. Minin, A. A. (2008). Navigachiya i upravliniya mobilnym robotom, osnashchenym lazernym dalecomirom [Navigation and control of a mobile robot equipped with a laser rangefinder]. *www.dissercat.com*. Retrieved from <https://www.dissercat.com/content/navigatsiya-i-upravlenie-mobilnym-robotom-osnashchennym-lazernym-dalnomerom>. [in Russian].
2. Thrun, S. (2017). Google's driverless car. *www.ted.com*. Retrieved from http://www.ted.com/talks/sebastian_thrun_google_s_driverless_car [in English].
3. Team page Stanford Racing. (2006). *cs.stanford.edu*. Retrieved from <http://cs.stanford.edu/group/roadrunner/stanley.html> [in English].
4. Bolshakov, A.A., & Lysichky, D.L. (2011). Upravliniya ruhom mobilnogo robota [Mobile robot motion control]. *cyberleninka.ru*. Retrieved from <https://cyberleninka.ru/article/n/upravlenie-dvizheniem-mobilnogo-robota> [in Russian].
5. Dunn, M. (2015). Mobile robotics (Robot). Module 1. *learn.open2study.com*. Retrieved from <https://learn.open2study.com/mod/youtube/view.php?id=79916> [in English].

6. Borenstein, J., Everett, H.R., & Feng, L. (1996). Where am I? Sensors and Methods for Mobile Robot Positioning. *www.yumpu.com*. Retrieved from <https://www.yumpu.com/en/document/read/33598354/where-am-i-sensors-and-methods-for-mobile-robot-positioning> [in English].
7. Siegward, R., Nourbakhsh, I.R., & Scaramuzza, D. (2004). Introduction to Autonomous. *mitpress.mit.edu*. Retrieved from <https://mitpress.mit.edu/books/introduction-autonomous-mobile-robots-second-edition> [in English].
8. Choset, H., Lynch, K., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., & Thrun, S. (2005). Principles of Robot Motion: Theory, Algorithms, and Implementations. *mathdep.ifmo.ru*. Retrieved from <http://mathdep.ifmo.ru/wp-content/uploads/2018/10/Intelligent-Robotics-and-Autonomous-Agents-series-Choset> [in English].
9. Arbib, M., & Hanson, A. (2010). Depth and Detours: An Essay on Visually Guided Behavior. *journals.sagepub.com*. Retrieved from <https://journals.sagepub.com/doi/10.1177/105971239500300404> [in English].
10. Koren, Y., & Borenstein, J. (1991). Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation. *www.cs.cmu.edu*. Retrieved from https://www.cs.cmu.edu/~motionplanning/papers/sbp_papers/integrated1/borenstein_potential_field_1imitations.pdf [in English].
11. Borenstein, J., Everett, H., & Feng, L. (1997). Navigating Mobile Robots: Senfors and Techniques. *dl.acm.org*. Retrieved from <https://dl.acm.org/doi/10.5555/546829> [in English].
12. Puttkamer, E., & Von, E. (2000). Autonome Mobile Roboter. *dl.acm.org*. Retrieved from <https://dl.acm.org/profile/81100649191> [in English].
13. Lumelsky, V., & Stepanov, A. (1986). Dynamic Path Planning for a Mobile Automation with Limited Information on the Environment. *dl.acm.org*. Retrieved from <https://dl.acm.org/profile/81100649191> [in English].
14. Kamon, I., & Rivlin, E. (1997). Sensory-Based Motion Planning with Global Proofs. *www.cs.technion.ac.il*. Retrieved from <http://www.cs.technion.ac.il/~ehudr/publications/pdf/KamonR95i.pdf> [in English].