

О.А. Петрова¹, аспірант
Г.В. Табунщик, канд. техн. наук, доцент,
Запорізький національний технічний університет, м. Запоріжжя, Україна
petrovaaoa353@gmail.com

Дослідження ефективності пошукових алгоритмів для систем навігації всередині приміщення

В статті досліджено та проведено аналіз алгоритмів для пошуку оптимального маршруту. Для порівняння алгоритмів використовувались наступні параметри: вартість проходу по коміркам, складність алгоритму та використання пам'яті. Після аналізу використаних факторів було обрано три алгоритми для подальшого дослідження використання для оцінки розміщення всередині приміщення.

Ключові слова: *iBeacon, пошуковий алгоритм, маршрут, картографічне представлення, системи навігації.*

DOI: 10.31474/1996-1588-2018-1-26-80-85

Вступ

Ідея " Smart Campus " для університетів полягає в тому, що кампус спілкується з користувачем. Доступні індивідуальні дані для студентів, викладачів та відвідувачів залежно від профілю та часу доби. Розроблена система складається з трьох основних підсистем: мобільний додаток для різних операційних систем iOS, Android; CMS для оновлення рекламної інформації; система управління, яка складається з різних компонентів, спрямованих на налагодження апаратних характеристик. Smart Campus - це мобільна програма, яка надає користувачам різноманітну функціональність, що дозволяє працювати як в режимі онлайн режиму, так і в режимі офлайн, що виявляє сигнал від маяків [1].

В рамках програми Smart Campus для відображення положення на карті необхідно мати картографічне представлення приміщення з зазначенням положення кожного iBeacon-маячка і мати можливість прокласти маршрут. Для визначення поточного положення на карті приміщення було отримано модифікований інтегрований метод заснований на аналізі сигналу від iBeacon і акселерометра, модифікований методом Калмана, що дозволяє оперативнo коригувати значення поточного положення. Однак для ефективної реалізації програми необхідно вибрати алгоритм для пошуку найкоротшого шляху.

Одним із прикладів впровадження цих технологій стали навігаційні системи всередині приміщення, які використовують електронні карти.

Навігаційні системи з електронними картами з'явилися на початку 1980-років. Але їх інформаційні ресурси були обмеженими. Однак і з такими можливостями ці системи мали перевагу перед урахуванням руху на паперових картах.

З ходом часу навігаційні системи досить швидко удосконалювалися. Процес зростання функціональних можливостей систем та збільшення кількості інформації можна представити наступними етапами [1]:

- застосування спрощених карт-схем;
- підключення навігаційних засобів для позиціонування;
- використання растрових карт;
- відображення векторних карт;
- пошук даних і відображення карт в різних масштабах;
- виконання планування шляху;
- видача довідок про картографічні об'єкти;
- автоматичне коректування карт;
- інтеграція радіолокаційної інформації.

В області картографії для самохідних систем з'явилися такі особливості [2]:

- картографічне зображення вже не служить безпосереднім джерелом інформації, а забезпечує візуалізацію інформації, зібраної і збереженої в базах просторових даних;
- в інформаційних процесах комп'ютерної обробки просторових даних карта здійснює функції специфічного інтерфейсу між людиною та комп'ютером;

- картографічна продукція орієнтована не тільки на вузьке коло фахівців і спеціалізованих організацій, а на найрізноманітніші потреби різних верств населення;

- з'явилася велика кількість картографічних сервісів і служб, в тому числі в мережі Інтернет, які істотно розширили перелік послуг за допомогою картографічних зображень;

- почали розвиватися нові технологічні напрямки - мобільна картографія, тривимірна, мультимедійна і анімаційна картографія.

Задачі картографічного представлення розглядалися в роботах: Берлянт А.М. [2], Р. Уоттерс [3], Гапанюка Ю.Е. [4].

На даному етапі розвитку навігаційних систем важливою проблемою є коректне відображення розміщення всередині приміщення і відповідно визначення поточних координат, так як робота систем позиціонування та навігації всередині приміщення неможлива без заздалегідь відомого плану приміщення. Електронні карти грають велику роль при навчанні самохідних систем з обмеженими ресурсами, тому задача побудови маршруту в реальному часі є одною із розповсюджених та актуальних проблем при створенні систем навігації, моделюванні переміщення по пересувних об'єктах.

Також всі системи навігації мають загальний недолік – при певних умовах сигнал може не доходити до приймача, приходиться зі значними спотвореннями, затримками або зникати при ввімкненні режиму тиші. Тому розробка програмного забезпечення для формування маршруту на основі заздалегідь підготовленої карти з урахуванням всіх особливостей розміщення є актуальною.

Постановка задачі

Для переміщення пристроїв всередині приміщення досліджують данні з бездротових та інерційних пристроїв. Однак дослідження картографічного представлення приміщення дозволяє поперше оптимізувати маршрут, що дозволить зменшити час необхідний для досягнення точки призначення та по-друге підвищити точність оцінки визначення поточних координат.

Традиційно картографічні данні представляються в істинному і у відносному русі [2].

У режимі істинного руху карта нерухома, а умовний знак користувача переміщається по ній відповідно до кута і шляхової швидкості. При підході символу користувача до краю екрану автоматично здійснюється зрушення зображення карти, щоб крапка не зникла з екрану. Правила такого зміщення карти неоднакові. В одних системах зрушення карти здійснюється по лінії кута, в інших вона зміщується так, щоб умовний знак користувача потрапив в зазначену оператором область.

У режимі відносного руху відображуваний символ користувача не має лінійного переміщення і знаходиться в центрі екрану. Зображення карти «пливе» щодо позначки користувача зі швидкістю, що дорівнює і протилежно спрямованій вектору шляхової швидкості користувача.

Авторами був розроблений модифікований інтегрований метод, який дозволяє підвищити точність визначення поточного положення на карті приміщення [5]. В даному методі одним з входних параметрів є точне місцезнаходження відповідно до карти.

Незважаючи на велику теоретичну базу, на жаль представлено не так багато додатків, що до-

зволяють людям, далеким від математики і програмування, використовувати існуючі розробки для вирішення практичних завдань. Використовуючи такі популярні картографічні сервіси для навігації всередині приміщення як «IndoorAtlas» [7] і Indoor Maps [8], не завжди можна вирішити поставлене завдання, через недостатню швидкість. Тому виникає необхідність вибору алгоритму для побудови оптимального маршруту.

Так як навігаційна система це система з обмеженими ресурсами, то важливим параметрами є швидкість визначення поточного положення та мінімізація використання пам'яті.

Тому автори поставили наступні завдання:

- провести аналіз алгоритмів для побудови оптимального маршруту;
- провести експериментальне дослідження застосування алгоритмів для побудови оптимального маршруту.

Вибір алгоритму для реалізації пошуку шляху на карті приміщення

Існує велика кількість алгоритмів [9]:

- алгоритм Дейкстри [10];
- пошук в ширину (BFS, Breadth-First Search) [11];
- алгоритм пошуку A* [12];
- алгоритм пошуку з ітеративним поглибленням IDA* (Iterative-Deepening A*) [13];
- пошук точки стрибка (Jump Point Search) [14].

Для проведення аналізу розглядався граф G (V, E), де V - кількість вершин, E - кількість ребер.

Вибір алгоритму для реалізації можна здійснити, проаналізувавши таблицю, в якій розглядаються ключові фактори.

Вартість проходу по коміркам визначає чи підходить даний алгоритм для задачі пошуку оптимального маршруту по карті, чи ні [15]. Деякі алгоритми призначені для пошуку шляху в лабіринті і вартість проходів може бути 1 або ∞ , наприклад, в Breadth-First Search.

Також розглядається поняття складності алгоритму - це спосіб виміряти, наскільки швидко програма або алгоритм працюють. Аналіз складності дозволяє пояснити, як буде вести себе алгоритм при зростанні вхідного потоку даних.

Існує кілька методів аналізу складності алгоритмів за часом, найпоширенішими з них є:

- амортизаційний аналіз;
- аналіз паралельних алгоритмів;
- асимптотичний аналіз складності алгоритмів.

Амортизаційний аналіз - метод аналізу алгоритмів, що дозволяє здійснювати оцінку часу виконання послідовності з n операцій над деякою структурою даних. Час виконання вираховується

за всіма n операціями і оцінюється середній час виконання однієї операції в гіршому випадку. [16]

Аналіз паралельних алгоритмів - при паралельному виконанні алгоритму різні частини програми виконуються на різних обчислювальних пристроях, і часто виходить, що результат обчислення в одному пристрої є входом для іншого пристрою. Час, необхідний для обміну даними між обчислювальними пристроями, залежить від топології комунікаційної підсистеми і може виявитися порівнянними (або навіть перевищувати) з часом, необхідним для виконання операцій алгоритму. Тому для паралельних алгоритмів точний аналіз складності вимагає врахування як кількості кроків алгоритму (з урахуванням можливості паралельного виконання), так і кількість необхідних пересилань [17].

Асимптотичний аналіз складності алгоритмів - метод аналізу алгоритмів, в якому розглядається поведінка алгоритму при $n \rightarrow \infty$. При асимптотичному аналізі обчислювальної складності алгоритму оцінюють кількість операцій для таких випадків:

– найгірший випадок (worst case) - максимальна кількість операцій, необхідних для обробки набору вхідних даних;

– середній випадок (average case) - середня кількість операцій, необхідних для обробки набору вхідних даних;

– найкраща нагода (best case) - мінімальна кількість операцій, необхідних для обробки набору вхідних даних.

Для запису асимптотичної складності алгоритмів використовуються різні асимптотичні позначення, наприклад, O (Big O).

Існує поняття тимчасової складності і воно особливо важливо для задач, які передбачають інтерактивний режим роботи програми, або для завдань управління в режимі реального часу. Як правило, підвищення точності обчислень веде до збільшення часу [18].

Об'ємна складність програми стає критичною, коли обсяг оброблюваних даних виявляється на межі обсягу оперативної пам'яті [18].

Існує три можливих випадки:

кращий випадок (best case) - це екземпляр завдання (набір вхідних даних), на якому алгоритм виконує найменше число операцій;

найгірший випадок (worst case) - це екземпляр завдання, на якому алгоритм виконує найбільшу кількість операцій.

середній випадок (average case) - це "середній" екземпляр завдання, набір "усереднених" вхідних даних. В середньому випадку оцінюється математичне очікування кількості операцій, що виконуються алгоритмом. Не завжди очевидно, які вхідні дані вважати «усередненими» для завдання.

У таблиці 1. Наведена порівняльна характеристика алгоритмів пошуку оптимального мар-

шруту за такими критеріями: вартість переходу по комірках, найгірший час, найбільший об'єм пам'яті, складність алгоритму. Також в таблиці 1 розглядалися наступні позначення:

$O(n)$ - кількість операцій для пошуку вершини;

m - релаксація;

d - глибина пошуку;

b - коефіцієнт розгалуження;

$|V|$ - вершини;

$|E|$ - ребра.

h^* - оптимальна евристика,

Таблиця 1 - Таблиця порівняння алгоритмів

Алгоритм	Вартість переходу по комірках	Worst time Найгірший час	Worst space (найбільший об'єм пам'яті)	Складність алгоритму
Дейкстри	Будьяка	$O(n^2)$	$O(V)$	$O(n^2+m)$
BFS,)	1 або ∞	$O(b^{d+1})$	$O(b^d)$	$O(V + E)$
A*	Будьяка	$O(b^{d+1})$	$O(b^d)$	$O(\log h^*(x))$
IDA*	1 або ∞	$O(b^d)$	$O(bd)$	$O(b^d)$
JPS	1 або ∞	$O(n^2)$	$O(n)$	$O(n^2)$

Аналіз показав, що тільки три алгоритми з розглянутих можна використовувати: алгоритм Дейкстри, A *, стрибкові точки. З цих алгоритмів найоптимальнішим за часом є стрибкові точки, тому що алгоритм не вимагає додаткових витрат пам'яті, таким чином подальші дослідження будуть присвячені його оптимізації.

Оскільки частіше за все ми маємо справу з двомірними картами, нам буде потрібна одна або кілька систем координат, що відповідають різним проекціям. Такі системи координат на площині називаються картографічними прямокутними системами координат, вони дозволяють точно вказувати положення об'єктів на плоских картах. Карта поточного місцезнаходження отримує вибірку від iBeacon-маячка та задається шлях, який будеється відповідно до обраного алгоритму для побудови оптимального маршруту. Для оцінки необхідно порівняти час досягнення заданої точки та кількість переходів при цьому.

Тому представимо карту у вигляді матриці $[M, N]$, де M - кількість точок по осі X, N - кількість точок по осі Y.

Для порівняння були розглянуті алгоритми Дейкстри, A *, стрибкові точки.

Модифікація методу визначення поточного розміщення для Smart Campus

Нехай карта представлена у вигляді матриці $[M, N]$, де M - кількість точок по осі X , N - кількість точок по осі Y .

Будемо розглядати три види матриць:

1. $[1,10]$, $[2,10]$, $[4,10]$
2. $[1,30]$, $[2,30]$, $[4,30]$;
3. $[1,40]$, $[2,40]$, $[4,40]$.

Детальні результати досліджень наведені в таблицях 2, 3 і 4, t - час роботи алгоритму в мілісекундах, N - кількість проходів.

В дослідженнях розглядаємо рух між двома точками. Для кожної матриці:

1. в матриці $[1,10]$ рух з точки $[1,0]$ до $[1,10]$;

2. в матриці $[2,10]$ рух з точки $[2,0]$ до $[2,10]$;
3. в матриці $[4,10]$ рух з точки $[4,0]$ до $[4,10]$;
4. в матриці $[1,30]$, рух з точки $[1,0]$ до $[1,30]$;
5. в матриці $[2,30]$ рух з точки $[2,0]$ до $[2,30]$;
6. в матриці $[4,30]$ рух з точки $[4,0]$ до $[4,30]$;
7. в матриці $[1,40]$ рух з точки $[1,0]$ до $[1,40]$;
8. в матриці $[2,40]$, рух з точки $[2,0]$ до $[2,40]$;
9. в матриці $[4,40]$ рух з точки $[4,0]$ до $[4,40]$;

Таблиця 2 - Результати для першого виду матриць

Алгоритм	$[1,10]$		$[2,10]$		$[4,10]$		Σt , мс	ΣN
	t, мс	N	t, мс	N	t, мс	N		
Дейкстри	6,43	420	27,34	1811	119,33	7286	153,1	9517
A*	11,17	334	16,57	953	67,53	4203	95,217	5490
Стрибкові точки	2,37	80	13,07	446	48,74	2408	64,18	2934

Таблиця 3 - Результати для другого виду матриць

Алгоритм	$[1,30]$		$[2,30]$		$[4,30]$		Σt , мс	ΣN
	t, мс	N	t, мс	N	t, мс	N		
Дейкстри	26,88	1636	97,31	6133	411,8	25088	535,99	32857
A*	22,69	1286	54,57	3397	247,74	13946	325	18629
Стрибкові точки	9,64	208	28,2	1356	120,78	6152	158,62	7716

Таблиця 4 - Результати для третього виду матриць

Алгоритм	$[1,40]$		$[2,40]$		$[4,40]$		Σt , мс	ΣN
	t, мс	N	t, мс	N	t, мс	N		
Дейкстри	50,72	3224	218,38	11415	838,62	48448	1107,72	63087
A*	41,63	2540	128,93	7142	456,03	25282	626,59	34964
Стрибкові точки	7,59	320	48,96	2104	236,27	9836	292,82	12260

Після проведення практичного порівняння для роботи був обраний алгоритм стрибкові точки. Як видно з таблиці порівнянь даний алгоритм має пріоритет в швидкості виконання від 103 до 248%, а в кількості проходів пріоритет від 157 до 360%.

Висновки

У даній роботі були досліджені алгоритми для побудови оптимального маршруту при навігації всередині приміщення.

Наукова новизна роботи полягає у модифікації методу визначення поточного розташування за рахунок використання алгоритму визначення найоптимальнішого маршруту стрибкові точки.

Після порівняння складності алгоритму і гіршого часу виконання, було вибрано три найоптимальніші алгоритми, які використовувалися для

експериментального порівняння. Після проведення експерименту Для подальшої роботи було обрано алгоритм стрибкові точки, так як він має найвищу швидкість виконання. Його пріоритет в швидкості виконання від 103 до 248%, а в кількості проходів пріоритет від 157 до 360%.

Практична цінність роботи полягає в застосуванні обраного алгоритму для картографічного уявлення приміщення в рамках системи навігації інтерактивного університету.

Дана робота виконана на кафедрі програмних засобів Запорізького національного технічного університету за пріоритетним напрямом «Перспективні засоби переробки інформації та кібернетичні системи», в рамках теми № 0117U000615 «Інформаційна система діагностування розподілених міні-комп'ютерних систем в багатокомпонентному зовнішньому середовищі».

Список літератури

1. Tabunshchik G. (2015), Smart-campus Infrastructure Development Based on Ble 4.0 / G. Tabunshchik, Merode D. Van, Y. Goncharov, K. Patrahalko, [Rozrobka infrastruktury Smart-Campus na bazi Ble 4.0] // Электротехнические и компьютерные системы. - 2015. - № 18. - С. 17-20.
2. Ананьев, Ю. С. (2003), Геоинформационные системы. : Учеб. Пособие / Ю. С. Ананьев – Томск: Изд. ТПУ, - 70 с.
3. Берлянт, А. М. (2002), Картография / А. М. Берлянт – М.: Аспект Пресс. – 336 с.
4. R. Watters (2016) Teaching Map and Compass: Navigating from the Classroom to the Outdoors .[Prepodavaniye karty i kompassa: perekhod s klassa na ulitsu]– available at: — <http://www2.isu.edu/outdoor/mapshort.htm>
5. Гапанюк Ю.Е (2015) Методы оценки положения объекта в пространстве \ Жуков Р.В, Гапанюк Ю.Е Молодежный научно-технический вестник ФС77-51038
6. Петрова, О. А. (2017), Метод определения текущего расположения в системах позиционирования и навигации внутри помещения / Петрова О. А., Табунщик Г. В. Дирк Ван Мероде –Електротехнічні та комп'ютерні системи. – № 25. – С. 270-278.
7. IndoorAtlas . – Access Mode: <http://www.indooratlas.com/>
8. Indoor Maps . – Access Mode: <https://www.mapspeople.com>
9. Савельева, Е. В. (2016) Определение оптимального маршрута прокладки трубопровода available at: – 2016 – http://www.tnlib.ru/jirbis/files/upload/ooks/VKR/2016/IMiKN/Sevalneva_VKR.pdf
10. Новиков, Ф. А. (2002) Дискретная математика для программистов [Текст] / Ф. А. Новиков – СПб.: Питер,– 38 с.
11. Поиск в ширину (BFS, Breadth-First Search) available at: – <https://brestprog.neocities.org/lections/bfs.html>
12. Hart, P. E. (1968), A formal basis for the heuristic determination of minimum cost paths / Hart P. E., Nils J. Nilsson, Bertram Raphael [Formal'na osnova dlya evrystychnoho vyznachennya minimal'nykh vytrat] – Systems Science and Cybernetics, IEEE Transactions on 4.2, 1968: – P. 100-107.
13. Рассел, С. (2006), Составление допустимых эвристических функций / С. Рассел, П. Норвиг. – Искусственный интеллект: современный подход — 2-е изд.. — М.: Вильямс.— С. 170 — 173. —ISBN 5-8459-0887-6.
14. Harabor, D. D. (2011) Online Graph Pruning for Pathfinding On Grid Maps. / Harabor Daniel Damirand Alban Grastien – Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence.- P. 1114-1119
15. Структуры данных и алгоритмы. Теория. Графы и их основные сферы применения в информатике [Электронный ресурс]. – Режим доступа: http://study-and-dev.com/blog/sda_theory_graphs/
16. Курносов, М. Г. Амортизационный анализ (amortized analysis) available at: – <http://www.mkurnosov.net/teaching/index.php/DSA/Fall2015>.
17. Адигеев, М. Г. Анализ сложности параллельных алгоритмов. available at: – <https://open-edu.sfedu.ru/pub/2416>.
18. Понятие сложности алгоритма available at: – <https://habrahabr.ru/post/104219/>
19. Сложность алгоритмов available at: – <http://kufas.ru/programming174.htm>

Надійшла до редакції 15.05.2018

O. Petrova¹, G. Tabunshchuk¹¹Zaporizhzhya National Technical University**INVESTIGATION OF THE PATH SEARCHING ALGORITHMS EFFECTIVENESS FOR THE INDOOR NAVIGATION SYSTEMS**

The article provides the analysis of the shortest path detection algorithms and possibility of their implementation to the indoor navigation systems. As wireless networks are not always available and noise can influence the level of the signal, it is good solution for indoor self-navigation systems to combine different approaches which include also maps, as they can improve the correctness of the object location detection. The developed by the authors method for the location detection for the indoor-navigation systems consider the case when the cartographic rectangular coordinate systems of the buildings are provided and for each floor markers by iBeacon are defined and cartographic rectangular coordinate systems are associated with them. As indoor navigation systems mainly based on architecture with limited resources it is very important to use optimal algorithms with regards to the time and memory usage. The authors also provided analysis of the different approaches for algorithm complexity analysis, such as depreciation analysis; analysis of parallel algorithms; asymptotic analysis of complexity of algorithms. Selected best three solutions, Dijkstras' algorithm, A* algorithm and jump point search algorithm were investigated as part of the integrated method of location detection for different prerequisites, such as different maps, starting and ending point. Provided experiments confirmed that the best solution regarding selected metrics is jump point search algorithm. Its priority is in the speed of execution from 103 to 248%, and in the number of passes the priority is from 157 to 360%. Developed method was implemented into the navigation systems for smart-campus infrastructure, which was developed in Zaporizhzhya National Technical University.

Keywords: iBeacon, search algorithm, route, cartographic representation, navigation system.**О. А. Петрова¹, Г. В. Табунщик¹**¹Запорожский национальный технический университет**ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ ПОИСКОВЫХ АЛГОРИТМОВ ДЛЯ СИСТЕМ НАВИГАЦИИ ВНУТРИ ПОМЕЩЕНИЯ**

В статье проведен анализ алгоритмов для поиска оптимального маршрута. Для сравнения алгоритмов использовались следующие параметры: стоимость прохода по ячейкам, сложность алгоритма и использования памяти. После анализа использованных факторов были выбраны три алгоритма для дальнейшего исследования использования для оценки размещения внутри помещения.

Ключевые слова: iBeacon, поисковый алгоритм, маршрут, картографическое представление, системы навигации.