

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Багатоагентний алгоритм розв'язання задачі
орієнтації на місцевості «S-bots» - systems

(тема)

Виконав:

студент II курсу, групи СПМ-21-2
Кравець В.Є.
(прізвище, ініціали)

Спеціальність 123 «Комп'ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва освітньої програми)

Керівник: доц. Токарев В.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ

(підпис)

Коваленко А.А.

(прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 123 «Комп'ютерна інженерія» _____
(код і повна назва)

Тип програми _____ освітньо-наукова _____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Системне програмування _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту _____ Кравцю Владиславу Євгеновичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Багатоагентний алгоритм розв'язання задачі орієнтації на місцевості «S-bots» - systems _____

затверджена наказом по університету від “ 03 ” квітня 2023 р. № 318 Ст _____

2. Термін подання студентом роботи до екзаменаційної комісії 17 травня 2023 р. _____

3. Вхідні дані до роботи 1) дослідити багатоагентний горизонтально масштабований алгоритм розв'язання задачі simultaneous localization and mapping; 2) провести огляд задачі simultaneous localization and mapping; 3) дослідити метод Демпстера - Шафера; 4) представлення вихідних даних: згідно нормативних документів. _____

4. Перелік питань, що потрібно опрацювати у роботі _____

1) огляд літератури за темою роботи; _____

2) аналіз предметної області; _____

3) вибір та обґрунтування методики дослідження; _____

4) проведення експериментальних досліджень; _____

5) висновки. _____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) _____

Слайд-презентація – 18 слайдів _____

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Огляд літератури за темою роботи	04.04.23 - 07.04.23	
2	Вибір та обґрунтування методики дослідження	08.04.23 - 13.04.23	
3	Вибір інструментальних засобів	14.04.23 - 18.04.23	
4	Проведення експериментів	19.04.23 - 03.05.23	
5	Оформлення матеріалів кваліфікаційної роботи	04.05.23 - 09.05.23	
6	Подання кваліфікаційної роботи керівникові та її попередній захист	10.05.23 - 11.05.22	
7	Подання кваліфікаційної роботи на рецензування	12.05.23 - 16.05.23	

Дата видачі завдання 03 квітня 2023 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

доц. Токарев В.В.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 64 с., 20 рис., 1 табл., 1 дод., 10 джерел.

ТЕОРІЯ ДЕМПІСТЕРА-ШАФЕРА, EMBEDDED, «S-BOTS» - SYSTEMS, WI-FI.

Метою кваліфікаційної роботи є дослідження багатоагентного алгоритму розв'язання задачі simultaneous localization and mapping для «S-bots» - systems на базі методу Демпстера-Шафера.

У ході виконання кваліфікаційної роботи розглядаються наукові завдання, які потрібно вирішувати під час проектування «S-bots» - systems – визначення своєї локації, тобто орієнтація «s-bots» на місцевості. Від того, наскільки точна карта місцевості, яка знаходиться в пам'яті embedded комп'ютера автопілотованих «S-bots» - systems і згідно з якою рухаються «s-bots», залежить наскільки безпечно для людей вони будуть рухатися.

Застосування цієї теорії дозволить збільшити точність побудови карти місцевості, а також зменшити вплив шумів у порівнянні з класичною Байєсівською теорією.

ABSTRACT

Master's thesis: 64 pages, 20 figures, 1 tables, 1 appendices, 10 sources.

DEMPSTER-SCHAFFER THEORY, EMBEDDED, «S-BOTS» - SYSTEMS, WI-FI.

The purpose of the qualification work is to study a multi-agent horizontally scalable algorithm for solving the problem of simultaneous localization and mapping for «S-bots» - systems based on the Dempster-Schafer theory.

In the course of the qualification work, scientific tasks are considered that need to be solved when designing «S-bots» - systems – determining their location, that is, the orientation of «s-bots» on the ground. How accurate the map of the area, which is in the memory of the embedded computer of the autopilot «S-bots» - systems, and according to which the «s-bots» move, depends on how safe they will move for people.

The application of this theory will increase the accuracy of building a map of the area, as well as reduce the effect of noise compared to the classical Bayesian theory.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП	9
1 ОГЛЯД РІШЕНЬ ЗАДАЧІ SIMULTANEOUS LOCALIZATION AND MAPPING.....	11
1.1 Огляд підходів до вирішення задачі simultaneous localization and mapping.....	13
1.2 Огляд алгоритму виділення особливих точок для вирішення задачі simultaneous localization and mapping.....	113
1.3 Огляд алгоритму із застосуванням розширеного фільтра Калмана для вирішення задачі simultaneous localization and mapping.....	21
1.4 Огляд алгоритму «швидкого simultaneous localization and mapping».....	22
1.5 Огляд алгоритму, побудованого на Байєсівській теорії для вирішення задачі simultaneous localization and mapping	24
1.6 Огляд алгоритму, побудованого на методі Фільтру частинок для розв'язання задачі simultaneous localization and mapping	25
1.7 Огляд алгоритму з графовим представленням карти для розв'язання задачі simultaneous localization and mapping	27
2 БАГАТОАГЕНТНІ АЛГОРИТМИ РІШЕННЯ ЗАДАЧІ SIMULTANEOUS LOCALIZATION AND MAPPING, ЇХ СТРУКТУРА І АРХІТЕКТУРА.....	31
2.1 Огляд алгоритмів розв'язання задачі багатоагентного simultaneous localization and mapping	32
2.2 Ролі «s-bots» у багатоагентному simultaneous localization and mapping алгоритмі.....	33

2.3 Аналіз багатоагентних алгоритмів розв'язання задачі localization and mapping	35
2.4 Огляд багатоагентного алгоритму з графовим поданням розв'язання задачі simultaneous localization and mapping	37
2.5 Огляд багатоагентного алгоритму, з неграфовим поданням розв'язання задачі simultaneous localization and mapping	38
3 БАГАТОАГЕНТНИЙ МАСШТАБУЄМИЙ АЛГОРИТМ РІШЕННЯ ЗАДАЧІ SIMULTANEOUS LOCALIZATION AND MAPPING ДЛЯ ЗГРАЇ «S-BOTS» - SYSTEMS	41
3.1 Опис багатоагентного алгоритму simultaneous localization and mapping	43
3.2 Застосування теорії Демпстера-Шафера для збільшення точності алгоритму	46
ВИСНОВКИ.....	52
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	53
ДОДАТОК А ГРАФІЧНИЙ МАТЕРІАЛ КВАЛІФІКАЦІЙНОЇ РОБОТИ	55

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ
І ТЕРМІНІВ

БПЛА – безпілотний літальний апарат

БСМ – бездротова сенсорна мережа

ІТС – інтелектуальна транспортна система

МСО – мережа, яка самоорганізується

ЕКФ – розширений фільтр Калмана (англ. Extended Kalman Filter)

SLAM – одночасне визначення власного положення та побудови карти
(англ. Simultaneous Localization And Mapping)

«S-BOTS» – рій ботів (англ. Swarm-Bot)

ВСТУП

Під час проведення огляду літератури на тему кваліфікаційної роботи було встановлено, що у 2021 році вироблено та введено в експлуатацію понад 450 тис. «s-bots». За прогнозами аналітиків дослідницьких компаній, до 2023 року ця кількість зросте до 750 тис. одиниць. Така статистика говорить про стрімке зростання індустрії автопілотованих «S-bots» - systems. Виручка одного з гігантів у цій галузі – компанії TeslaMotors – у 2020 році склала 25 млрд. доларів.

Таким чином, виробництво автопілотованих «S-bots» - systems набирає стрімкі обороти. Отже, будь-який науковий чи інженерний винахід у цій галузі викликає інтерес і стає перспективним напрямом для компаній, що працюють у цьому бізнесі.

Одним із глобальних завдань, які необхідно вирішувати при проектуванні «s-bots» – визначення своєї локації, тобто орієнтація на місцевості. Від того, наскільки точна карта місцевості, яка знаходиться в пам'яті embedded комп'ютера автопілотованих «S-bots» - systems і згідно з якою рухаються «s-bots», залежить, наскільки безпечно для людей вони будуть рухатися. Тому завдання орієнтації на місцевості формулюється таким чином, щоб забезпечити точність свого положення до сантиметрів. На сьогоднішній день жодна супутникова система локалізації на це не здатна. Для забезпечення такої точності визначення локації «s-bots», дослідники запропонували застосовувати алгоритми, що вирішують задачу SLAM (simultaneous localization and mapping) – одночасного визначення власного становища та побудова карти оточення у фізичному неорганізованому середовищі. Будувати карту оточення у фізичному неорганізованому середовищі відповідно до роботи алгоритму, замість використання заготовленої заздалегідь, має сенс у тому випадку, якщо відомо, що дані, які описують місцевість, можуть динамічно змінюватися за короткий проміжок

часу порівняно з тими даними, що були отримані раніше. Наприклад, «s-bot», який рухається проїжджою частиною, повинен уникати перешкоди, які можуть виникнути на шляху його прямування або об'їжджати тимчасово перекриті ділянки (на час ремонту) проїжджої частини. Забезпечити «s-bot» такою інформацією наперед є дуже складним завданням. Тому використовувати заготовлену карту, безумовно, корисно, але необхідно також оновлювати цю карту в міру руху «s-bot» у фізичному неорганізованому середовищі. Застосування кількох «s-bots» однієї «S-bots» - systems, працюючих спільно та разом будуєть карту оточення, дозволяє прискорити побудову карти місцевості та підвищити точність локалізації кожного «s-bot» з цієї «S-bots» - systems окремо.

Тому розробка масштабованих алгоритмів для «S-bots» - systems з обмеженими обчислювальними ресурсами при вирішенні задачі simultaneous localization and mapping, на сьогоднішній день є актуальною проблемою.

1 ОГЛЯД РІШЕНЬ ЗАДАЧІ SIMULTANEOUS LOCALIZATION AND MAPPING

Перш ніж говорити про багатоагентні «S-bots» - systems необхідно класифікувати існуючі одноагентні «s-bots» - підходи та визначити, чи можливе масштабування цих підходів та їх вартість. Задача simultaneous localization and mapping – це задача, яка ставиться перед «s-bots», що входять до складу однієї «S-bots» - systems, що перебувають у русі та яким необхідно дослідити невідому місцевість та побудувати карту цієї місцевості (рисунки 1.1 - 1.3).

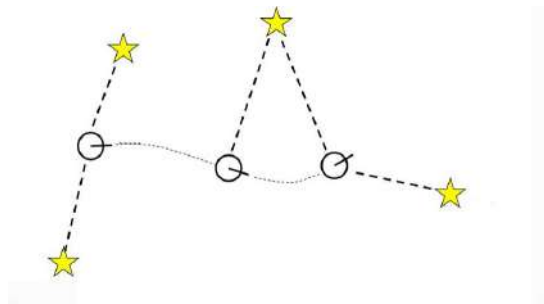


Рисунок 1.1 – Виявлення позицій «s-bot» у фізичному неорганізованому середовищі за орієнтирами

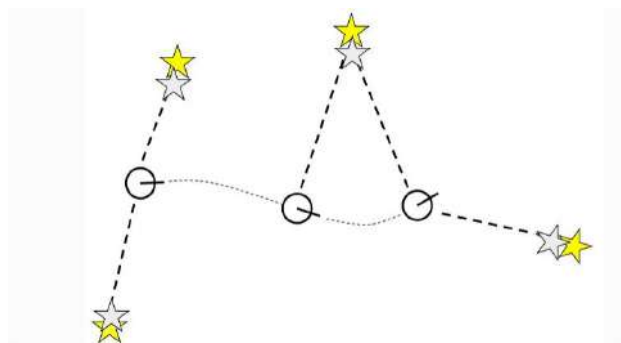


Рисунок 1.2 – Запам'ятовування позицій орієнтирів щодо позицій «s-bot» у фізичному неорганізованому середовищі

- мобільні «s-bots» рятувальники;
- марсоходи.

В даний час автономні мобільні «s-bots» не володіють поки що видатними обчислювальними потужностями. Це накладає обмеження на рівень складності алгоритмів розв'язання поставленого завдання. Також існує клас складних алгоритмів, які мають виконуватися на серверах, у той час як мобільні «s-bots» виконують лише роль спостерігачів за оточенням.

В останньому випадку необхідно стежити за якістю зв'язку між усіма вузлами «S-bots» - systems, що призводить до зниження стійкості та автономності системи. Додаткову складність у вирішення розглянутої задачі вносить той факт, що немає можливості скористатися сервісами, які надає GPS. Наприклад, це неможливо у приміщенні, де похибка GPS порівнянна з шириною коридору. Мобільний «s-bots» може покладатися лише на:

- датчики відеокамери;
- лазерний далекомір;
- гіроскоп;
- сонар.

1.1 Огляд підходів до вирішення задачі simultaneous localization and mapping

На даний момент, виходячи з різних початкових умов, існують різні підходи до вирішення задачі simultaneous localization and mapping. Можна провести класифікацію рішень за різними критеріями. Справедливо помітити, що немає прямого зв'язку між якимись певними типами з однієї та іншої класифікації.

Однак серед найбільш популярних алгоритмів простежується закономірність у реалізації певних комбінацій типів з різних класифікацій.

Перший тип класифікації – за розмірністю спостережень. Якщо оснастити мобільний «s-bot» відеокамерою, то він повністю спостерігає

навколишній тривимірний простір, і побудована ним карта місцевості також матиме розмірність, що дорівнює трьом. Такі алгоритми зазвичай є досить вимогливими до обчислювальних ресурсів, оскільки їх операційні вхідні дані є хмарами точок, отриманих за допомогою камери або комбінації камер, або за допомогою тривимірного лідара (рисунок 1.4).

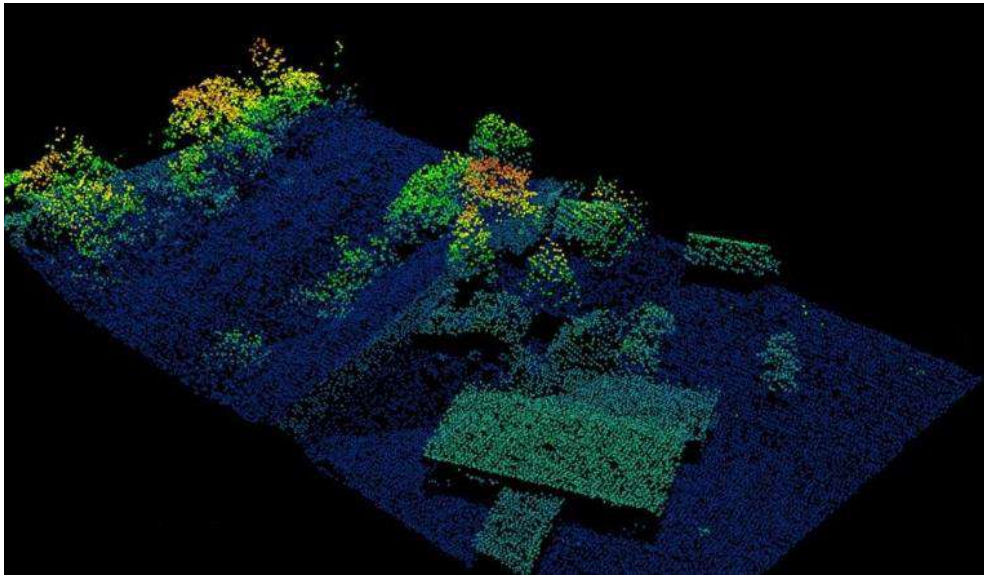


Рисунок 1.4 – Приклад хмари точок, отриманої з тривимірного лідара

Часто для спрощення обробки вхідних даних або з інших причин використовуються лазерні далекоміри або сонари, які проводять вимірювання у площині. Побудована при використанні таких датчиків карта являє собою план оточення. Двовимірні алгоритми є більш швидкими, ніж тривимірні, через меншу кількість вхідних даних. Їх перевагою є той факт, що вони можуть обробляти дані лідарів, які мають дуже високу роздільну здатність і, отже, будувати точний план карти, маючи низьку похибку при побудові траєкторії руху (рисунок 1.5).

Другий тип класифікації – за типом застосовуваних сенсорів. Зазвичай цей спосіб класифікації безпосередньо пов'язаний з попереднім, однак є й деякі винятки. В якості сенсорів використовується відеокамера (рисунок 1.6).

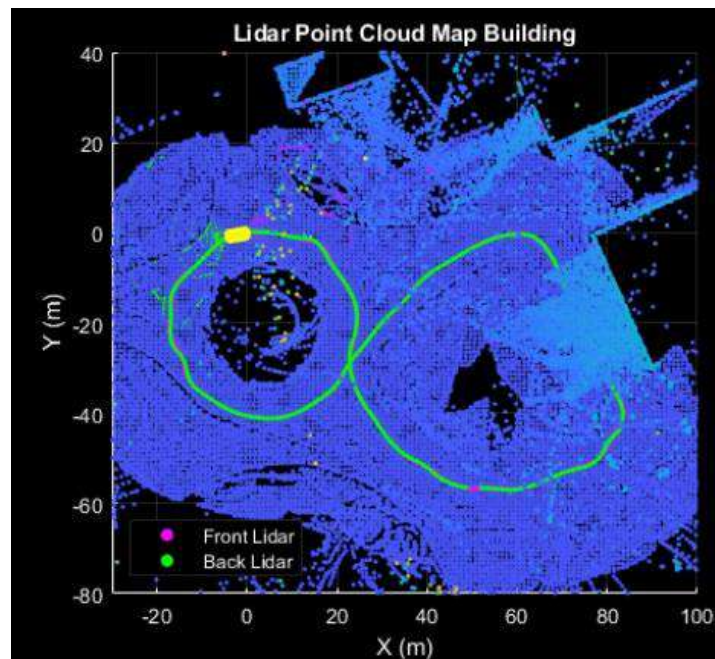


Рисунок 1.5 – Приклад хмари точок, отриманої з двовимірного лідара

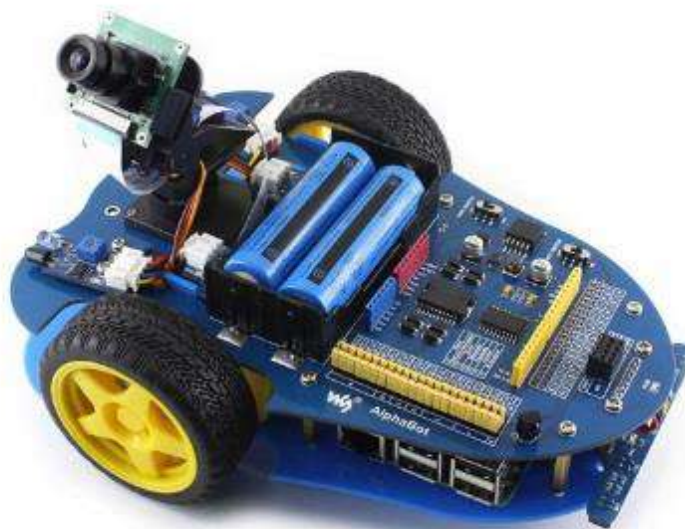


Рисунок 1.6 – В якості сенсора використовується відеокамера

Це може бути як одна камера, так і набір з декількох, а також «всеспрямована» камера, яка знімає за один раз оточення на 360 градусів. У якості сенсорів використовується лазерний далекомір – пристрій, який вимірює відстань у всіх напрямках з кроком дискретизації (рисунок 1.7).

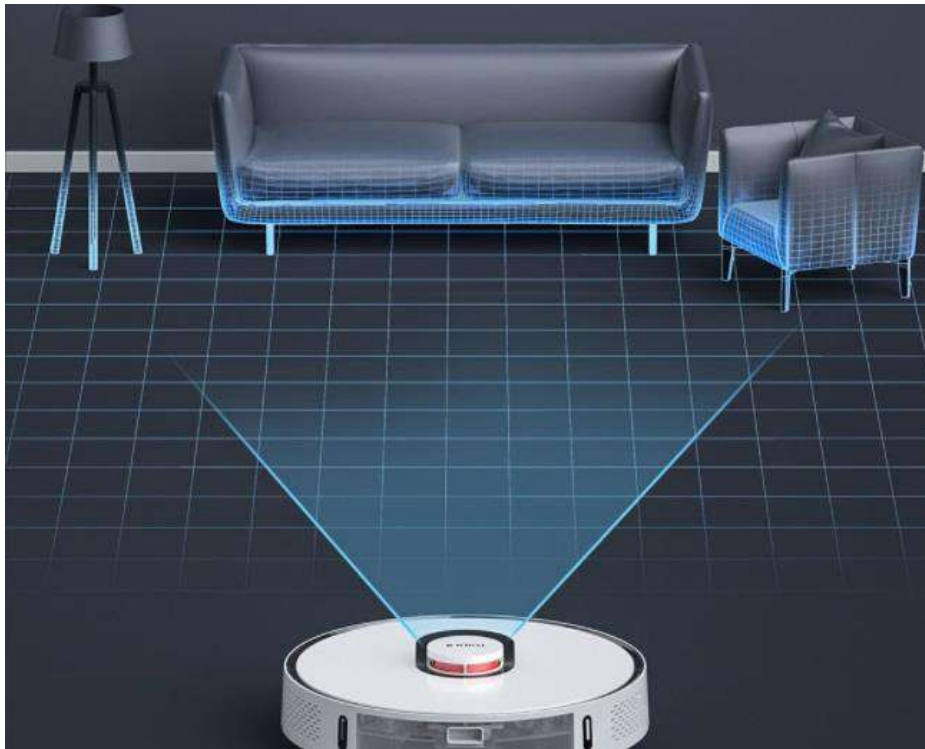


Рисунок 1.7 – В якості сенсора використовується лазерний далекомір

Зазвичай візуальний алгоритм будує тривимірну карту, а лазерний – двомірну, але ця залежність не є обов'язковою. Так, наприклад, можливо будувати тривимірну карту за допомогою лазерного лідара або двовимірну карту, використовуючи відеокамеру. Класичні алгоритми, що стоять на початку розвитку задачі *simultaneous localization and mapping*, працювали саме так. За набором відеозображень будувалася двовимірна карта перешкод.

Третій тип класифікації – за способом обробки вхідних вимірювань. Часто не потрібно будувати повну карту місцевості, досить обмежитися деяким набором орієнтирів. Такі орієнтири мають бути виділені з даних сенсорів та використані для побудови карти. Історично саме такий тип алгоритмів було розроблено першим, використовуючи «сирі» виміри. Набагато повнішу карту можна отримати, якщо використовувати всі дані, отримані від сенсорів «s-bot». Якщо відеокадр або лазерний скан використовувати повністю, то інформації для побудови карти стає більше, і карта стає набагато точнішою (рисунок 1.8).

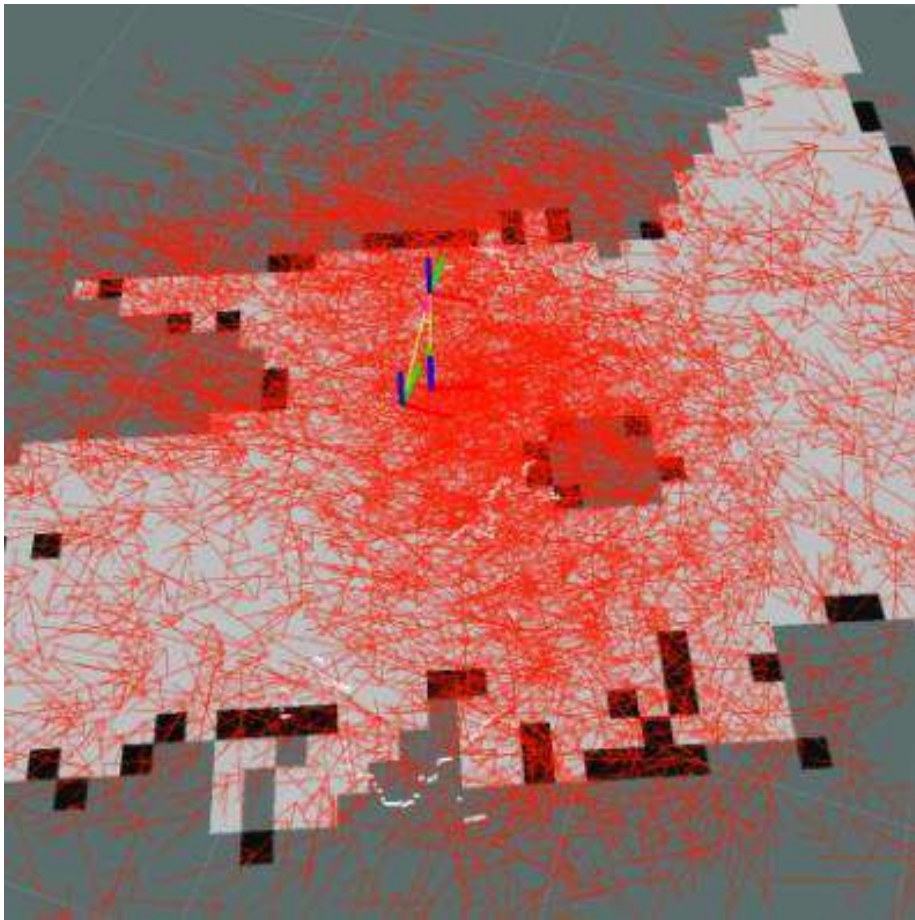


Рисунок 1.8 – Приклад побудови повної карти місцевості

Четвертий тип класифікації – за способом представлення карти. Карта в алгоритмі *simultaneous localization and mapping* – це один із результатів роботи алгоритму, тому питання про її структуру стоїть найбільш гостро. Очевидним виглядає рішення представити карту у вигляді масиву комірок (двовимірних або тривимірних), де кожна комірка містить ймовірність бути зайнятою (рисунок 1.9). Тоді весь простір можна розділити на невеликі області, кожна з яких зайнята якоюсь перешкодою, або вільна. У сучасних підходах все частіше використовують інший спосіб представлення карти.

Графовий підхід передбачає, що карту розподілено по вузлах графа. У кожному вузлі є одне або кілька вимірювань. Ребра графа містять у собі інформацію про переміщення «s-bot», яке необхідно здійснити, щоб почати спостерігати вимірювання з відповідного вузла. Перевагою такого алгоритму

є можливість визначити, чи відвідував «s-bot» місце, з якого знімається поточне спостереження, чи ні. Якщо відвідував, то поточне спостереження обов'язково корелюватиме зі спостереженням із вже побудованих вершин.

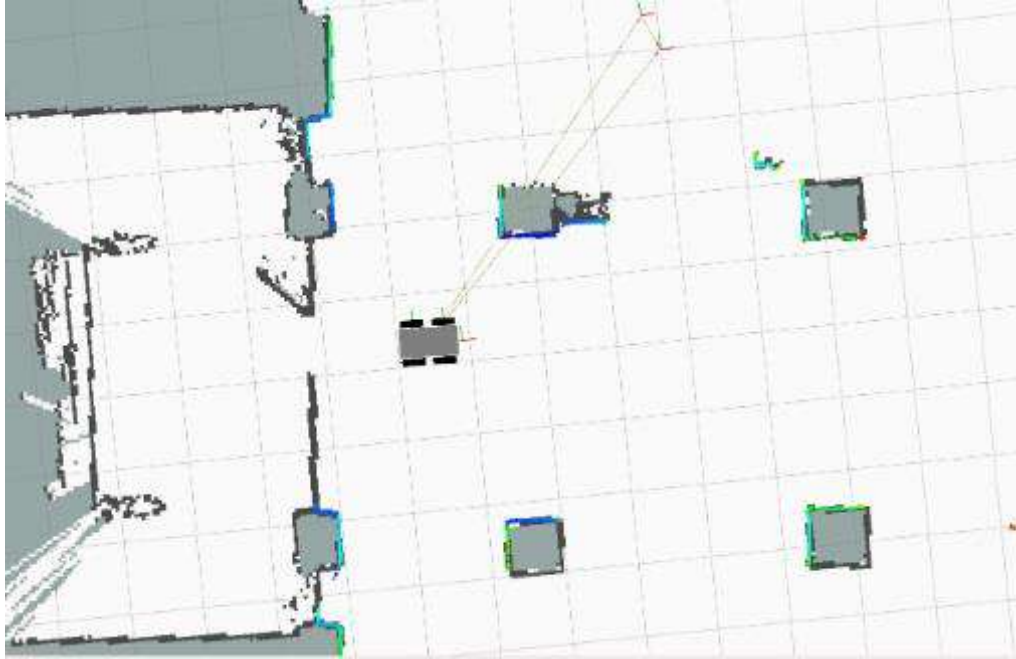


Рисунок 1.9 – Приклад подання карти місцевості у вигляді масиву комірок

Тоді у графі утворюється цикл, інваріантом якого є сума векторів-трансформацій між вузлами графа, що дорівнює нулю (зазвичай на практиці цей інваріант може не виконуватися через помилки та похибки обчислень). У цей момент включається механізм зміни ваг ребер графа з досягнення нуля у векторній сумі ваг. Це призводить до збільшення точності побудованої карти (рисунок 1.10).

П'ятий тип класифікації – характером зміни навколишнього середовища. Такий тип алгоритмів передбачає, що оточення не змінюється з часом. Якщо агент повністю вивчив якусь кімнату і залишив її, а через деякий час повернувся назад, то він виявить, що нічого в цій кімнаті не змінилося. Такий тип алгоритмів може бути використаний в офісних приміщеннях або складах.

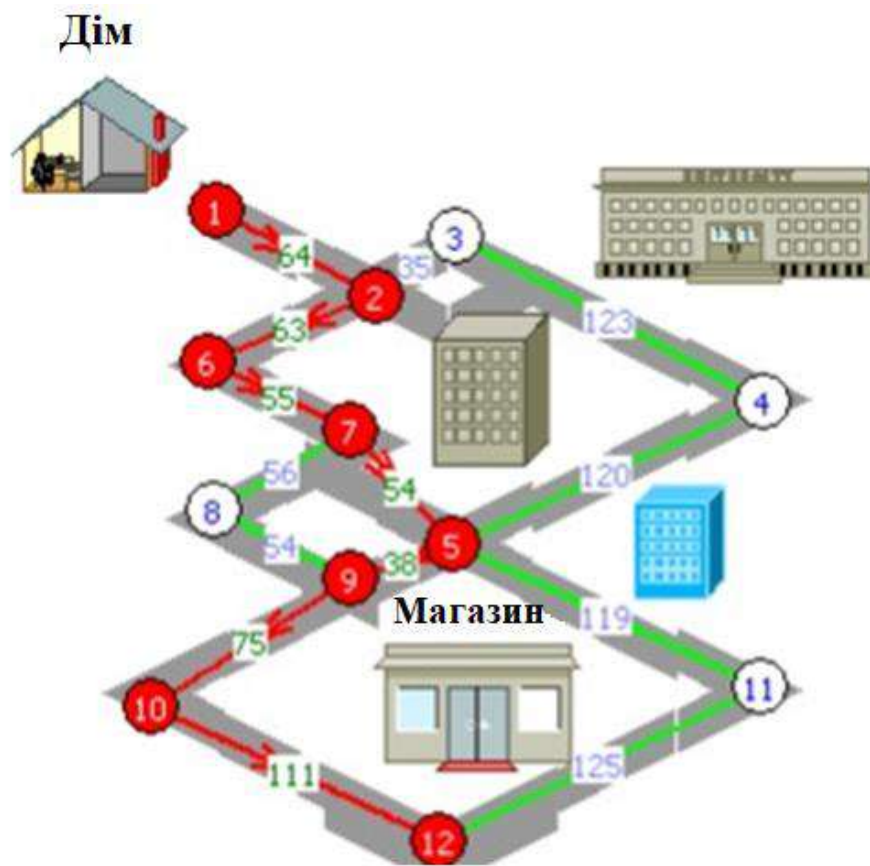


Рисунок 1.10 – Приклад представлення карти місцевості у вигляді графа

У випадку, якщо *simultaneous localization and mapping* алгоритм застосовується за межами будівель (наприклад, у місті, де є інтенсивний рух), з'являється необхідність вміти коригувати карту з урахуванням перешкод, що переміщуються.

1.2 Огляд алгоритму виділення особливих точок для вирішення задачі *simultaneous localization and mapping*

Історично першими з'явилися алгоритми, які виділяли особливі точки на оточенні, що спостерігається, і будували карту, яка складається з таких орієнтирів. Ідея дуже близька до людського сприйняття світу. Орієнтуючись у незнайомому місті, людина відзначає для себе, наприклад, високі вежі та

визначає приблизну відстань між ними, а далі використовує їх як орієнтир. Для автономних обчислювальних «s-bot» можна реалізувати подібний алгоритм. Необхідно виділити на знятих вимірах особливі точки, з яких в майбутньому і буде складатися карта. Чим більше таких точок буде визначено в процесі роботи, тим точніше буде результуюча карта та позиція «s-bot».

Також на точність впливає кількість позицій, із яких вдалося розрізнити особливу точку. Тому для таких алгоритмів питання про робасте виділення особливих точок стоїть дуже гостро. Робота алгоритму поділена на кілька етапів, серед яких можна виділити зняття вимірювань (скан, кадр та ін.), скан-матчинг (порівняння вимірювання та карти) та оновлення карти. На черговому етапі алгоритму вхідними даними є:

- побудована на даний момент карта;
- гіпотеза про становище «s-bot»;
- чергове спостереження.

Насамперед необхідно виділити особливі точки чи орієнтири. Потім після такої передобробки запускається компонент, який зветься скан-матчер (scan matcher) – співставник сканів, завдання якого – зіставити поточне спостереження із вже побудованою картою, щоб визначити поточне положення «s-bot». Після того, як позиція оцінена, з'являється можливість оновити карту, якщо це необхідно.

У алгоритмах, що розглядаються, карта складається з набору орієнтирів (їх координат щодо точки, де алгоритм розпочав свою роботу). Побудовані координати орієнтирів залежать від точки початку, проте відстань та взаємне розташування між ними залишається незмінним. Саме цей принцип є основою такого роду алгоритмів. Якщо скласти матрицю коваріацій орієнтирів, то з'явиться можливість застосувати алгоритм фільтрації Калмана для оновлення карти.

1.3 Огляд алгоритму із застосуванням розширеного фільтра Калмана для вирішення задачі simultaneous localization and mapping

Перша ідея розв'язання задачі simultaneous localization and mapping, запропонована 1990 року, спиралася на розширений фільтр Калмана. Ідея в тому, що всі перешкоди мають корелювати між собою, тобто (якщо «s-bot» працює в нерухомому оточенні) взаємне розташування перешкод не повинно змінюватись. А це означає, що, спостерігаючи пару орієнтирів під різними кутами, можна підвищити точність оцінки їхнього розташування. Докладніше алгоритм виглядає так. Алгоритм складається з багатьох ітерацій, кожна з яких включає кілька кроків:

- крок № 1. «S-bot» здійснює чергове спостереження, виділяє особливі точки. Відповідно до поточної апріорної оцінки позиції «s-bot» (виходячи з розташування орієнтирів щодо «s-bot») обчислюється розташування орієнтирів у глобальних координатах;

- крок № 2. Координати знайдених орієнтирів порівнюються з координатами орієнтирів, які були включені в карту на попередніх кроках. Цей крок дозволить уточнити позицію орієнтирів, які вже спостерігалися раніше, або змінити дисперсію похибки розташування орієнтирів. Якщо знайдений орієнтир відсутній у попередній карті, він додається в поточну карту;

- крок № 3. Виходячи з уточненої карти (у тому числі уточнених позицій орієнтирів, що спостерігаються) будується апостеріорна оцінка позиції «s-bot».

Необхідно зауважити, що апріорна оцінка на кроці №1 будується виходячи з апостеріорної оцінки, побудованої на кроці №3 на попередній ітерації, а також виходячи з даних одометрії – відомостей про переміщення «s-bot», виміряних датчиками переміщення (наприклад, гіроскопом). Однак слід зауважити, що цей алгоритм непридатний для масштабування. Кожен новий орієнтир, будучи включеним до карти, збільшує розмірність матриці

коваріацій орієнтирів. Слід пам'ятати, що карта з погляду алгоритму, що розглядається, – це вектор орієнтирів, що містить їх координати. Матриця коваріацій орієнтирів «пов'язує» орієнтири між собою. Це потрібно для того, щоб у разі уточнення координат однієї з перешкод можна було також уточнити координати інших орієнтирів.

Такий підхід, однак, починає вимагати великих обчислювальних витрат, щойно орієнтирів набирається значна кількість. Крім того що зростає вектор орієнтирів, при додаванні нового елемента у вектор необхідно обчислювати коваріацію нової перешкоди з усіма попередніми.

Слід зазначити, що специфіка алгоритму, заснованого на фільтрі Калмана, вимагає обчислювати матрицю зворотну до матриці коваріацій, що спричиняє збільшення часу роботи алгоритму з квадратичною швидкістю.

Сучасні сенсори дозволяють знімати спостереження довкілля із частотою щонайменше 30 Гц, отже, алгоритм має встигати обробляти щонайменше 30 ітерацій на секунду. Саме тому застосування розширеного фільтра Калмана для вирішення задачі *simultaneous localization and mapping* в сучасних системах практично зникло, а йому на заміну прийшли алгоритми, засновані на складніших ідеях або припущеннях, але менш вимогливі до обчислювальних ресурсів.

1.4 Огляд алгоритму «швидкого *simultaneous localization and mapping*»

Алгоритм «швидкого *simultaneous localization and mapping*» – це алгоритм, який продовжує концепцію алгоритму із застосуванням розширеного фільтра Калмана для вирішення задачі *simultaneous localization and mapping*, але збільшує продуктивність з допомогою відсутності кореляції між перешкодами. Іншими словами, загальний підхід до алгоритму залишається практично таким самим, як у алгоритмі із застосуванням розширеного фільтра Калмана, але вважається, що кожен орієнтир не залежить від усіх інших. Така постановка неминуче породжує

неоднозначність карти. Розв'язати її можна єдиним чином – знайти точку, з якою можна буде спостерігати два-три об'єкти одночасно. Якщо ж така точка не знайдена, залишається сподіватися, що спостерігач не помилився, коли будував траєкторію між точками, у яких він проводив спостереження. Графічно послідовність спостережень та можлива результуюча карта представлені на рисунку 1.11.

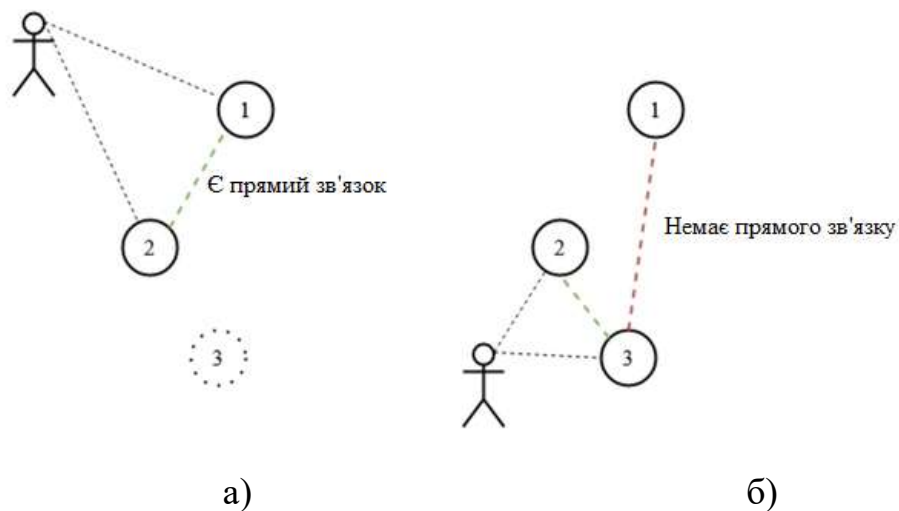


Рисунок 1.11 – Процес визначення перешкод спостерігачем: а) спостерігач бачить перешкоди 1 і 2 та запам'ятовує їхнє взаємне розташування; б) спостерігач бачить перешкоди 2 і 3, але зв'язок між 1 і 3 безпосередньо не будується

Щоб вирішити невизначеність, часто вдаються до апарату, що називається фільтром частинок. На етапі визначення положення вибирається не одна найбільш ймовірна гіпотеза, а декілька, наприклад, три. У цьому випадку будуть існувати три можливі карти, і кожне нове спостереження буде потрібно зіставляти з трьома картами. Процес відбуватиметься до тих пір, поки якась із гіпотез не буде відкинута, оскільки нові спостереження перестануть будь-яким чином співвідноситися з картою.

При проведенні експериментальних досліджень отримані результати

показали, що точність підходу алгоритму «швидкого simultaneous localization and mapping» (незважаючи на суттєве місце для невизначеності) залишається високою. Тому, алгоритм «швидкого simultaneous localization and mapping» залишається на сьогодні одним із базових алгоритмів.

1.5 Огляд алгоритму, побудованого на Байєсівській теорії для вирішення задачі simultaneous localization and mapping

Як було показано вище, ідея виділення особливих точок на кадрі накладає значні обмеження на структуру карти та алгоритм в цілому. Існує альтернативний метод представлення карти, який дозволяє зберігати модель карти в масиві, що динамічно змінюється. У простому випадку йдеться про двовимірний масив, кожен елемент якого – це «клітина», що відповідає області простору. Іншими словами, карта є двовимірним планом навколишнього середовища, розділеним на комірки. Кожна комірка такої карти містить число – можливість бути зайнятою. Приклад карти з коміркою розміром 10×10 см показано на рисунку 1.12.

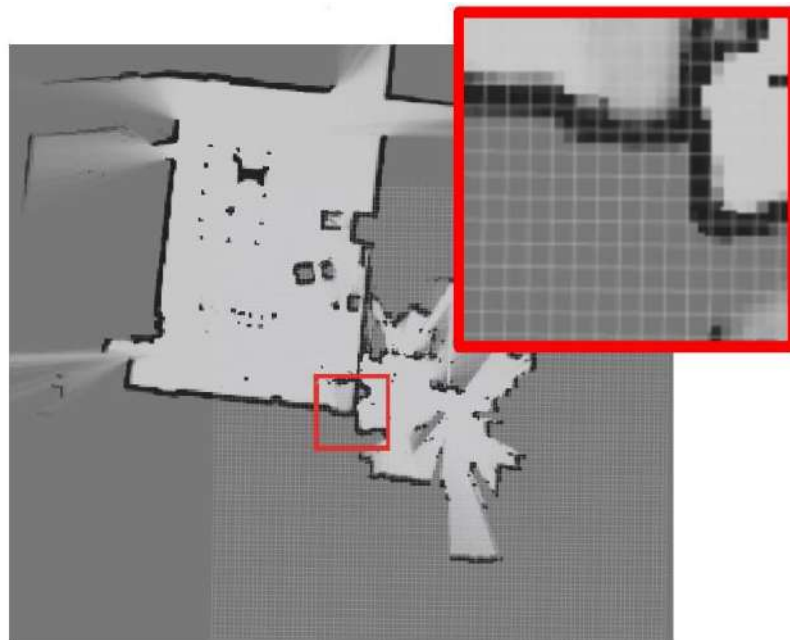


Рисунок 1.12 – Приклад картки зайнятості

З рисунка 1.12 ясно, що при достатньо невеликому розмірі комірок, така карта не відрізняється від звичайного плану оточення, і навіть містить більше інформації, ніж набір орієнтирів. Найкраще такий формат карти підходить для алгоритмів, що працюють з даними, отриманими за допомогою лазерних ледарів.

1.6 Огляд алгоритму, побудованого на методі Фільтру частинок для розв'язання задачі *simultaneous localization and mapping*

Основним недоліком підходу, описаного в підрозділі 1.5, є неможливість простої перевірки консистентності вихідних даних. Якщо в процесі роботи алгоритму щось пішло не так, сам алгоритм не зможе розпізнати цю ситуацію та згодом отримає неправильні дані. Для вирішення цієї проблеми застосовується метод Фільтру частинок. Введемо позначення. Вважатимемо часткою стан системи в певний момент (позиція «s-bot» і побудована ним карта).

Припустимо, що мобільний «s-bot» прибув в одну з двох присутніх на карті вертикальних колон (рисунок 1.13). Також припустимо, що інших орієнтирів у мобільного «s-bot» нема. Отже, мобільному «s-bot» необхідно виконати локалізацію, використовуючи лише дані про перебування перед однією з колон і невідомо, перед якою саме.

У цей момент можна зробити дві рівноймовірні гіпотези про поточне положення мобільного «s-bot». Саме такий алгоритм використовує метод Фільтру частинок. Під час виконання алгоритму можуть виникати ситуації, коли доведеться додавати гіпотези, і тоді може виявитися більше 10 різних гіпотез щодо становища мобільного «s-bot» і, відповідно, про карту. Карти в частках відрізняються, оскільки після поділу можливих позицій усі частинки «приймають» однакові подальші спостереження та вбудовують їх у поточну картку частки.

Таким чином, вже через кілька кроків після поділу на дві частинки, карти можуть значно відрізнятись. Якщо під час роботи алгоритму відбувається ситуація, коли виміри не можуть бути зіставлені з картою єдиним чином, необхідно створити нову частинку та відстежувати відразу обидві, обчислюючи правдоподібність кожної з них.

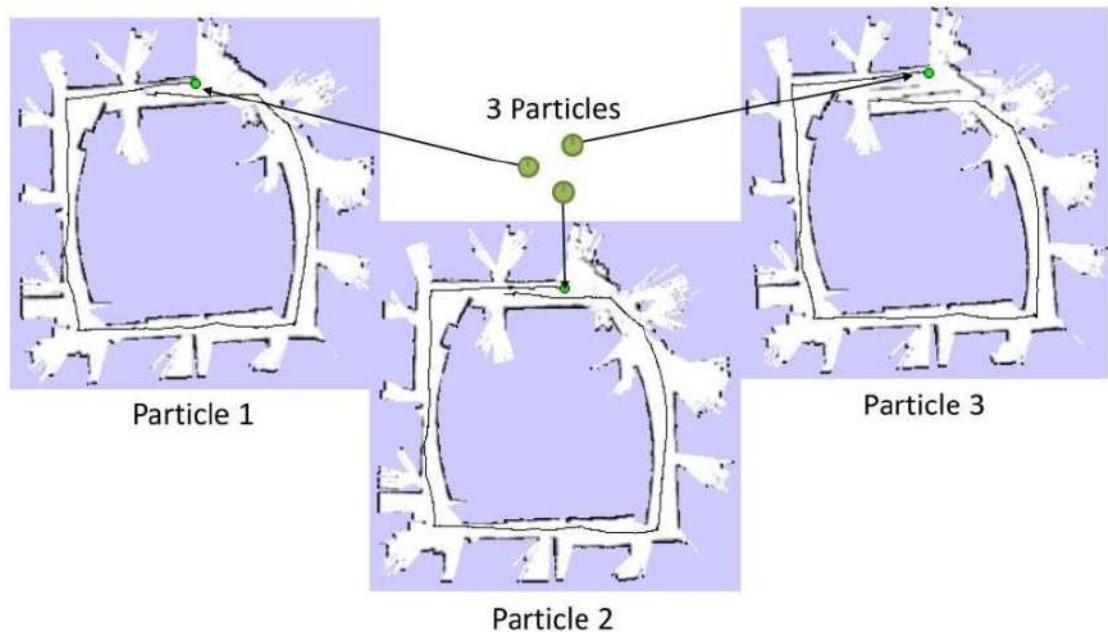


Рисунок 1.13 – Три частинки в алгоритмі, побудованому на методі Фільтру частинок

Правдоподібність гіпотези ґрунтується на консистентності карти. Так, якщо нові спостереження не суперечать побудованій карті (це можна з'ясувати на етапі скан-матчингу при обчисленні вартості скана), то карту можна вважати консистентною і гіпотезу правдоподібною. Якщо ж нові спостереження регулярно суперечать карті, можна зробити висновок про те, що гіпотеза про позицію мобільного «s-bot» була зроблена невірно, і її правдоподібність – низька. Фільтр частинок (гіпотез) повинен обчислювати правдоподібність всіх гіпотез і стежити за тим, щоб у наборі частинок не було неймовірних.

Одночасно з цим він повинен доповнювати набір частинок, якщо на етапі побудови карти було знайдено кілька високоймовірних гіпотез. В обох випадках фільтр має інструментарій, щоб додавати або видаляти частинки. Цей процес називається ресамплінг. В даний час алгоритм, який використовує метод Фільтру частинок, вважається стандартним алгоритмом розв'язання задачі *simultaneous localization and mapping*.

Алгоритм, який використовує метод Фільтру частинок, представляє послідовність наступних кроків:

- крок № 1. Накласти поточне спостереження на карту в кожній частинці.
- крок № 2. Обчислити вагу кожної частинки.
- крок № 3. Якщо є частинки із гранично низькою вагою – видалити їх;
- крок № 4. Якщо на кроці № 1 є кілька способів накласти спостереження на карту – додати частинки;
- крок № 5. Перейти до кроку № 1.

Недоліком такого алгоритму є його обчислювальна складність, оскільки кожна нова частинка додає додаткове завдання *simultaneous localization and mapping*. Крім того, рішення про видалення частинки зі списку розглянутих приймається на основі різних евристик.

1.7 Огляд алгоритму з графовим представленням карти для розв'язання задачі *simultaneous localization and mapping*

Найбільш популярним на даний момент є графовий підхід для побудови карти. При такому підході початкові виміри поєднуються у вузли графа, а ребрами служить трансформація між позиціями, з яких можна ці виміри зняти. Докладніше простий графовий алгоритм можна описати так. Коли мобільний «s-bot» отримує новий вимір, він додає до графа нову вершину, в яку записує цей вимір. Нова вершина з'єднується одним ребром із попередньою доданою вершиною. Вага цього ребра – це сукупність

координат: переміщення та поворот між позиціями мобільного «s-bot» на поточному та попередньому кроці. Цю вагу можна отримати, використовуючи одометрію або скан карти, можлива комбінація цих методів. Через деякий час після початку роботи алгоритму буде побудовано граф з певної кількості вершин, кожна з яких з'єднана тільки з сусідніми вершинами. Такий граф являє собою карту. Для того, щоб побудувати карту у звичайному її розумінні, необхідно пройти по всіх вузлах графа та, використовуючи позицію мобільного «s-bot» та позиції точок у вимірах, нанести їх на карту. В результаті вийде хмара точок, що описують карту. Також замість такої хмари можна використовувати сітку зайнятості. Коли мобільний «s-bot» відвідує місце, в якому він вже побував, тобто знімає вимір, що збігається з однією з існуючих вершин графа, то підключається алгоритм замикання циклів і оновлення ребер в циклі. Може виявитися, що сума переміщень у ребрах, що утворюють цикл, не дорівнює нулю (тобто помилка, що накопичилася, змушує дані переміщення мобільного «s-bot» суперечити даним сенсора). В цьому випадку необхідно оновити ваги ребер так, щоб сума переміщень мобільного «s-bot» всередині циклу справді дорівнювала нулю. Процес перебалансування ребер неможливо стандартизувати, оскільки незрозуміло, яке з ребер зробило найбільшу помилку. Зазвичай для вирішення такого завдання використовується фільтрація, заснована на старінні даних: старіші вимірювання змінюють значніше, ніж новіші. Існує підхід, де кожне ребро графа змінюється однаково, тобто накопичена неточність вимірів ділиться на всі ребра.

Для того, щоб мінімізувати описані вище недоліки, існують різні модифікації підходу. Так, наприклад, замість створення нової вершини графа, що містить кожен новий вимір, кілька спостережень групують в одній вершині. Тоді граф стає менше і процес пошуку вершин, що повторюються, значно прискорюється. Однак у цьому випадку виникає питання про достатню кількість спостережень у вершині. У деяких випадках використовують поділ на відстані: як тільки мобільний «s-bot» відійшов від

початкової точки на деяку порогову відстань, спостереження записуватимуться в іншу вершину.

Наріжним каменем графового алгоритму є обчислювальна складність. Адже для замикання циклів необхідно порівняти новий вимір із кожною вершиною графа. Існує підхід, що дозволяє прискорити цей компонент алгоритму. Згідно з цим підходом перевірка унікальності нового виміру відбувається не відразу в момент появи цього виміру. Процес пошуку циклів у графі запускається окремо від процесу зняття вимірів. Наприклад, вимірювання можуть надходити з частотою 30 спостережень за секунду, і запускати з такою частотою перебір усіх вершин у графі – дуже затратна задача.

У цьому випадку пошук циклів відбувається набагато рідше: наприклад, раз на 2 секунди. Якщо об'єднати дві розглянуті модифікації для графового алгоритму (об'єднання кількох вимірів в одну вершину графа і винесення замикання циклів в окремий потік), то отриманий алгоритм буде близьким до алгоритму, представленого компанією Google і що має назву Cartographer. На високопродуктивних настільних комп'ютерах Cartographer працює практично в режимі реального часу: алгоритм встигає обробити чергове спостереження до моменту, коли буде знято наступне. Якщо запустити такий алгоритм на низькопродуктивних комп'ютерах, таких як Raspberry Pi, можна побачити, що замикання циклів (найдорожча операція) не встигає завершитися до появи чергового спостереження. Для вирішення цієї проблеми в алгоритмі Cartographer замикання циклів складається із двох етапів:

- передобробки;
- постобробки.

На етапі передобробки оновлюються ваги ребер графа, а постобробка запускається один раз наприкінці роботи алгоритму, щоб продемонструвати користувачеві кінцеву карту. Подібний підхід, коли новий вимір необхідно вбудувати у вже побудовану карту, відмінно підходить для масштабування

алгоритму. У багатоагентному алгоритмі такий інструмент є корисним, оскільки мобільний «s-bot» обчислює положення іншого мобільного «s-bot», використовуючи його спостереження. Проте недоліком багатоагентного графового підходу є його обчислювальна складність.

Серед рішень одноагентної задачі *simultaneous localization and mapping* на даний момент найточнішими є графові алгоритми. Така назва класу алгоритмів забезпечується графовою структурою карти. У вершинах графа розміщуються вимірювання, зняті мобільним «s-bot» з різних позицій. Ребра графа є зваженими та описують переміщення, яке має здійснити мобільний «s-bot» з попередньої вершини, щоб отримати змогу зняти спостереження з наступної вершини. Проте висока точність обумовлена підвищеними вимогами до обчислювальних ресурсів. Для зменшення цих вимог можна використовувати алгоритм на основі методу фільтру частинок. Використання такого фільтра дозволяє зберігати та здійснювати рейтинг кількох гіпотез про стан карти та позиції на випадок, якщо найімовірніша гіпотеза містить помилку. Але й застосування фільтра частинок також потребує підвищеної кількості обчислювальних ресурсів.

Для того, щоб вирішити задачу на низькопродуктивних пристроях, необхідно базуватись на двовимірних неграфових, однопотезних та лазерних алгоритмах. Саме такий клас алгоритмів є найменш вимогливим до ресурсів, але програє точно іншим класам.

2 БАГАТОАГЕНТНІ АЛГОРИТМИ РІШЕННЯ ЗАДАЧІ SIMULTANEOUS LOCALIZATION AND MAPPING, ЇХ СТРУКТУРА І АРХІТЕКТУРА

Цей розділ присвячений аналізу багатоагентних алгоритмів розв'язання задачі simultaneous localization and mapping, особливостям їхньої архітектури. У ній описані характеристики, якими повинен володіти багатоагентний алгоритм, що підходить для пристроїв з малою обчислювальною потужністю. Класифікація багатоагентних алгоритмів багато в чому співпадає з класифікацією одноагентних, тому в цьому розділі будуть розглянуті лише нові проблеми, з якими стикаються «s-bots», що входять до складу однієї «S-bots» - systems, і вирішують задачу simultaneous localization and mapping. На відміну від одноагентного алгоритму розв'язання задачі simultaneous localization and mapping в «S-bots» - systems, яка містить n - кількість «s-bots», виникають додаткові підзадачі:

- підзадача № 1. Визначити ролі мобільних «s-bots». «S-bots» можуть бути рівноправними і виконувати однакові завдання, а можуть реалізовувати, наприклад, клієнт-серверну архітектуру.

- підзадача № 2. Визначити спосіб обчислення взаємного розташування «s-bots». Це необхідно, щоб об'єднувати збудовані карти. Існують підходи, коли взаємне розташування «s-bots» відомо у початковий момент часу. Також існують алгоритми, які передбачають обчислення взаємного розташування у момент, коли «s-bots» знаходяться в безпосередній близькості щодо один одного;

- підзадача № 3. Визначити спосіб об'єднання карт. Можливий підхід, коли нові виміри «s-bots» вбудовуються у всі карти відразу. Існують алгоритми, у яких, навпаки, карти об'єднуються малу кількість разів за всю роботу алгоритму. У будь-якому випадку необхідно визначити методику вирішення конфлікту нових даних із накопиченими;

- підзадача № 4. Визначити, що є вихідними даними. Якщо в системі є кілька «s-bots», кожен з яких зберігає власний відмінний від інших примірник карти, то виникає питання, дані якого з примірників карти вважати вихідними?

2.1 Огляд алгоритмів розв'язання задачі багатоагентного simultaneous localization and mapping

Один з перших алгоритмів, що вирішують задачу багатоагентної simultaneous localization and mapping за умови наперед невідомого взаємного розташування «s-bots» заснований на інформаційних фільтрах. Такий підхід дозволяє позбавитися властивого simultaneous localization and mapping алгоритму недоліку, заснованого на розширеному фільтрі Калмана, обчислення матриці коваріацій, знайдених на карті перешкод. Натомість пропонується обчислювати інформаційну матрицю, при побудові якої допускається, що перешкоди можуть бути незалежними. Основним недоліком матриці коваріацій в алгоритмі на основі розширеного фільтра Калмана є обчислювальна складність роботи, а також складність додавання до неї нових перешкод. Крім того, при великій кількості перешкод робота з матрицею великої розмірності неминуче тягне за собою похибки обчислень.

Застосування ж інформаційної матриці, навпаки, дозволяє враховувати ті перешкоди, які мобільний «s-bots» спостерігає у конкретний час. Такий підхід одночасно збільшує швидкість роботи алгоритму, заснованого на виділенні перешкод, і дозволяє створити децентралізовану «S-bots» - systems.

У запропонованому алгоритмі процес поєднання карт є прямолінійним. Якщо є кілька карт, які слід об'єднати, то спочатку об'єднується пара карт, потім отримана карта об'єднується з третьою і так далі. Для того щоб виконати злиття карт, взаємне розташування «s-bots» пропонується обчислювати за допомогою накладання карт, побудованих різними «s-bots».

Перешкоди на одній карті розбиваються на трійки, і для кожної трійки (незалежно від інших) є відповідність на іншій карті (інформаційній матриці).

Після знаходження найбільш ймовірної трансформації між картами, позиція кожної перешкоди в обох картах перераховується за Байєсовськими правилами об'єднання випадкових величин. Якість такого підходу безпосередньо залежить від способу обчислення дескрипторів перешкод.

Передбачається, що коли «s-bots» розпізнають перешкоду, можна буде визначити, чи є вона новою, чи така перешкода вже присутня на карті. Більше того, якщо перешкода існує, має бути спосіб визначення, яка саме ця перешкода. Вимога до наявності такого механізму – основний недолік роботи цього алгоритму, оскільки він залежить від оточення.

2.2 Ролі «s-bots» у багатоагентному simultaneous localization and mapping алгоритмі

Архітектура багатоагентного simultaneous localization and mapping алгоритма впливає на алгоритми, які виконують кожні «s-bots», що входять до складу однієї «S-bots» - systems. Тому насамперед необхідно розібратися, які ролі можуть виконувати «s-bots», що входять до складу однієї «S-bots» - systems. Найпростішою є архітектура «централізованого єдиного початку» з одним головним «s-bot» і якоюсь кількістю підпорядкованих «s-bots» (рисунок 2.1).

Дотримуючись запропонованого алгоритму, карта в єдиному екземплярі будується на головному «s-bot», який приймає початкові спостереження від підпорядкованих «s-bots». Деякі підпорядковані «s-bots» не виконують жодних спеціальних дій, крім зняття вимірювань та відправлення їх головному «s-bot».

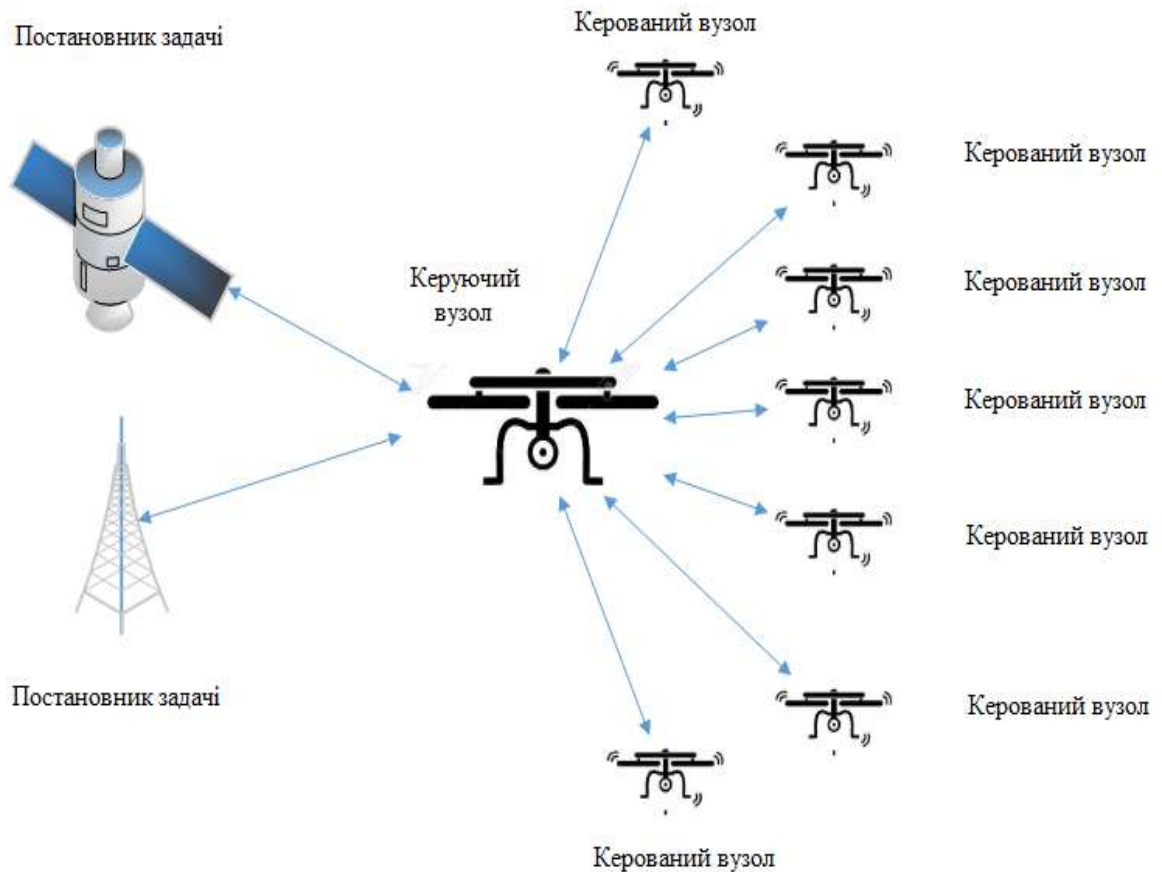


Рисунок 2.1 – Приклад архітектури «централізованого єдиного початку» з одним головним «s-bot»

Перед початком роботи алгоритму вказується початкова позиція всіх «s-bots», відповідно головний «s-bot» має можливість будувати одну карту, в яку послідовно вбудовуються спостереження кожного підпорядкованого «s-bot».

Одночасно із цим, головний «s-bot» вирішує завдання локалізації для кожного підпорядкованого «s-bots». Бувають випадки, коли початкова позиція всіх «s-bots» може бути не задана з абсолютною точністю. Для того, щоб у цьому випадку будувати одну карту, пропонується використовувати алгоритм на базі методу Фільтру частинок.

До переваг архітектури «централізованого єдиного початку» можна віднести такі параметри:

- параметр № 1. Рольова вага підпорядкованого «s-bot» може бути дуже низькою, оскільки такі «s-bots» виконують роль лише рухомих сенсорів без будь-яких обчислювальних операцій;

- параметр № 2. «S-bots» - systems стійка до втрати одного чи кількох підпорядкованих «s-bot» до тих пір, поки в «S-bots» - systems залишається хоча б один підпорядкований «s-bot».

До недоліків архітектури «централізованого єдиного початку» можна віднести такі параметри:

- параметр № 1. Залежність живучості всієї «S-bots» - systems від головного «s-bot»;

- параметр № 2. Головний «s-bot» однієї «S-bots» - systems повинен мати достатні потужності, щоб обробити всі дані, отримані від підпорядкованих «s-bots», що потенційно ускладнює масштабування всієї «S-bots» - systems;

- параметр № 3. Збій або вихід з ладу головного «s-bot» тягне за собою безповоротний вихід із ладу всієї «S-bots» - systems.

Ці недоліки призвели до меншої популярності алгоритмів, що використовують архітектуру «централізованого єдиного початку» та стали затребуваними алгоритми, що використовують архітектуру «децентралізованого управління».

2.3 Аналіз багатоагентних алгоритмів розв'язання задачі localization and mapping

Останнім часом для «S-bots» - systems стали замість терміна «множина» «s-bots» застосовувати термін «згряя» «s-bots», яка є виконавцем багатоагентного алгоритму localization and mapping. Згряя «s-bots» – набір автономних мобільних «s-bots», що входять до складу однієї «S-bots» -

systems і мають порівнянне технічне оснащення для спостереження навколишнього середовища в обмеженій околиці, що не мають ієрархії та обмінюються інформацією один з одним для спільної розмітки фізичного неорганізованого середовища (рисунок 2.2).

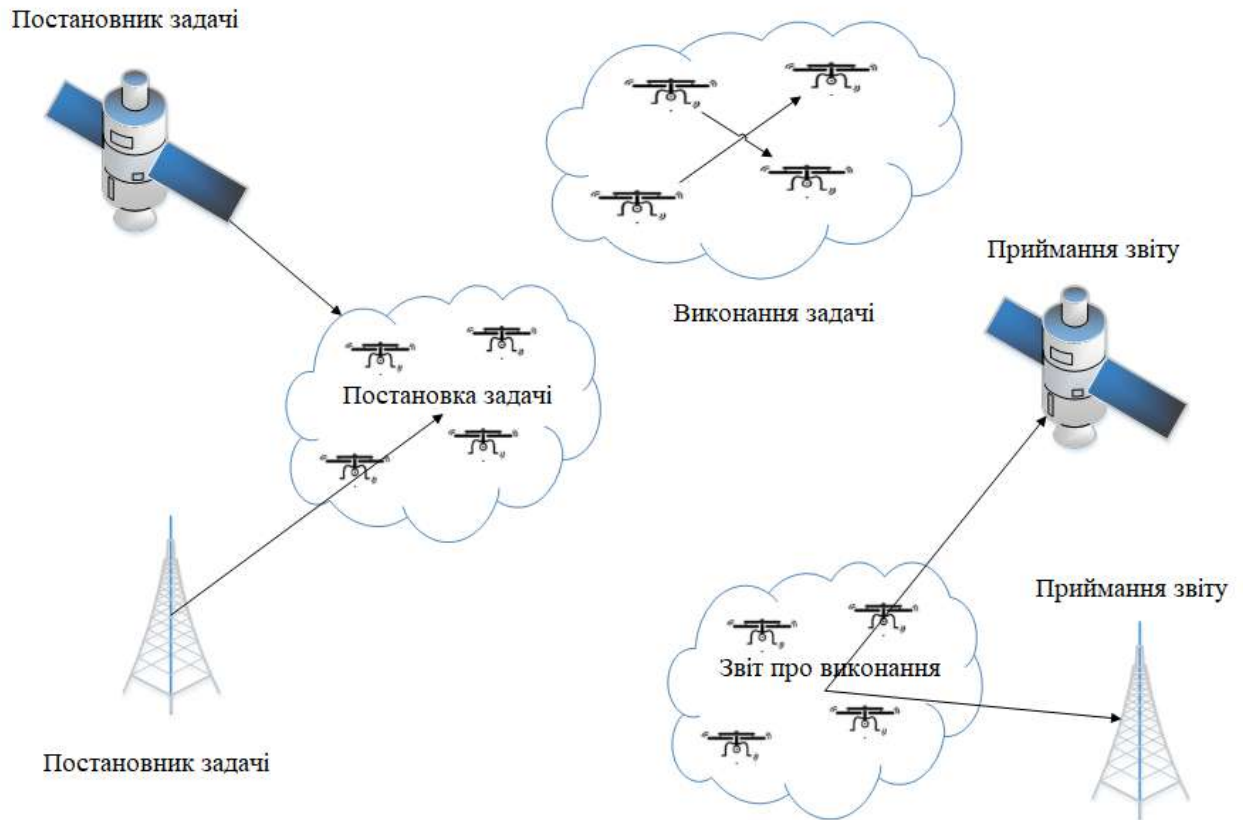


Рисунок 2.2 – Приклад архітектури «децентралізованого управління» із застосуванням терміну «зграя» «s-bots»

З цього випливає, що кожен «s-bot» в зграї виконує одні й ті ж функції. Іншими словами, у зграї відсутня вертикальна ієрархія, у тому числі відсутній головний «s-bot», відсутність якого може спричинити зупинку роботи всієї «S-bots» - systems. Кожний «s-bot» у зграї самостійно визначає, коли необхідно почати обмін інформацією з іншими «s-bot». Звичайно, підстави для прийняття такого рішення у всіх членів у зграї однакові.

2.4 Огляд багатоагентного алгоритму з графовим поданням розв'язання задачі simultaneous localization and mapping

Серед одноагентних алгоритмів, що вирішують задачу simultaneous localization and mapping, найбільш популярними на цей момент є графові. Популярність цих алгоритмів обумовлена масштабованістю та можливістю виправляти помилки у вже побудованій карті. Остання особливість таких алгоритмів дуже затребувана в багатоагентних підходах. В рамках графового підходу для кожного нового виміру підшукується вузол у графі. Для виконання цієї дії існує алгоритм замикання циклів, який може запускатись у кілька разів рідше, ніж зняття вимірювань.

Така ідея може бути ефективно використана у графовому алгоритмі для обчислення взаємного розташування «s-bots». Структуру графового алгоритму можна без змін застосувати для цього завдання. Якщо перший і другий «s-bot» відвідували одне й те саме місце, то після отримання виміру з цього місця від другого «s-bot» можна буде виконати замикання циклів і визначити місце, з якого були зняті ці вимірювання.

Таким чином, для графового підходу можна виділити дві основні переваги:

- простота у масштабуванні;
- можливість глобальної зміни карти у процесі роботи алгоритму.

Графовий підхід має той же недолік, яким володіли одноагентні алгоритми з графовим поданням розв'язання задачі simultaneous localization and mapping алгоритми: зниження швидкості роботи алгоритму з часом, коли граф накопичує велику кількість вершин.

Розглянемо один із прикладів реалізації графового алгоритму. В якості датчика вимірювань тут виступає лазерний далекомір. Архітектура цього рішення – розподілена мережа, в якій «s-bot» можуть зв'язатися один з одним тільки перебуваючи на невеликій відстані один від одного. Під час обміну інформацією кожен «s-bot» передає другому свої текущие измерения. Потім

відбувається оновлення карт всіх «s-bots» та обмін оновленими графами для виконання злиття карт.

На етапі обробки поточного спостереження другим агентом відбувається вибір вузла власного графа, з яким можна зіставити поточне спостереження і обчислити взаємне розташування першого і другого «s-bots» (навіть якщо вони знаходяться не в одній точці простору, а на деякій відстані один від одного).

2.5 Огляд багатоагентного алгоритму, з неграфовим поданням розв'язання задачі simultaneous localization and mapping

Головним недоліком графових алгоритмів є потреба у великих обчислювальних ресурсах. Точність обчислень повинна залежати від часу, що дано на обробку. Якщо алгоритм розроблений згідно з цією парадигмою, то його точність безпосередньо залежить від обчислювальних ресурсів мобільного «s-bot» (чим продуктивніше «s-bot», тим більше операцій за той самий час він зможе виконати, і тим точніше буде відповідь). Графовий підхід не вкладається в цю парадигму, йому потрібна фіксована кількість часу для отримання точної відповіді. У цьому підрозділі показані альтернативні підходи, їх переваги та недоліки.

Вже згадувалося, що однією з перших модифікацій базового алгоритму із застосуванням розширеного фільтру Калмана для вирішення задачі simultaneous localization and mapping був алгоритм «швидкого» simultaneous localization and mapping. Він працював швидше, похибка його роботи була мала порівняно з виграними ресурсами. Таким чином, серед багатоагентних алгоритмів для вирішення задачі simultaneous localization and mapping закономірно з'явився багатоагентний алгоритм «швидкого» simultaneous localization and mapping. Ідея цього алгоритму полягає в тому, що передбачається запуснути одноагентний алгоритм «швидкого»

simultaneous localization and mapping на кожному окремому «s-bot».

В кваліфікаційній роботі запропонована архітектура мережі з router, однак алгоритм може добре працювати і в архітектурі з розподіленою мережею. Аналогічно одноагентному підходу тут використовується метод Фільтра частинок, тобто одночасно враховуються і вираховуються ймовірності одразу кількох можливих наявних карт і позицій «s-bot». Як і в будь-якому алгоритмі на базі методу Фільтра частинок виникає питання, як створити два набори частинок, побудованих різними «s-bot». Прямолінійний варіант - порівнювати кожен частку один з одним - займає дуже багато обчислювального часу. Тому зазвичай йдуть шляхом відомого погіршення точності в обмін на швидкість роботи алгоритму. Дослідниками розроблено методику, яка пропонує об'єднати всі частинки з урахуванням їхньої ваги та побудувати усереднену карту на кожному «s-bot», а потім об'єднати усереднені карти. Безумовно, такий підхід вносить додатковий шум у побудовану карту, що негативно впливає на всі побудовані під час роботи частинки, але дозволяє виконати злиття карт максимально швидко.

Серед багатоагентних алгоритмів, як і раніше, великою популярністю користуються алгоритми, ролі «s-bots» у яких розподіляються нерівномірно. Ключовий недолік таких алгоритмів полягає в тому, що вихід з ладу різних «s-bots» по-різному відбивається на роботі всієї «S-bots» - systems. З огляду на популярність графових підходів в одноагентних алгоритмах вирішення задачі simultaneous localization and mapping, існує велика кількість графових багатоагентних алгоритмів. Додатковою перевагою таких алгоритмів над неграфовими є те, що вони, за визначенням, містять компонент, що реалізує замикання циклів, який можна використовувати для визначення взаємного розташування «s-bots». Замість цього запропоновано поняття «зграї» «s-bots», що вирішують задачу одночасної локалізації та побудови карти. Приналежність алгоритму, що розробляється, до класу двовимірних

неграфових одноагентних алгоритмів додає вимоги про архітектуру «s-bots». Для підвищення живучості всієї «S-bots» - systems необхідно, щоб усі «s-bots» виконували однакові ролі: і знімали спостереження, обробляли їх, та вибирали час для синхронізації.

З БАГАТОАГЕНТНИЙ МАСШТАБУЄМИЙ АЛГОРИТМ РІШЕННЯ ЗАДАЧІ SIMULTANEOUS LOCALIZATION AND MAPPING ДЛЯ ЗГРАЇ «S-BOTS» - SYSTEMS

У попередніх розділах наведено вимоги, які необхідно застосувати до алгоритму для досягнення умов високої продуктивності. Крім цього, необхідно визначити умови використання алгоритму, вимоги до обладнання мобільних «s-bots», що входять до складу однієї «S-bots» - systems та технологіям, що використовуються «s-bots». Обмеження, що накладаються на фізичне навколишнє середовище та «s-bots»:

- обмеження № 1. Досліджуваний алгоритм використовується «s-bots», що знаходяться в приміщенні. Це означає, що розмір досліджуваного простору не перевищує кількох сотен квадратних метрів. В цьому випадку неможливо використовувати GPS, оскільки похибка визначення місця розташування за допомогою GPS становить кілька метрів, що недопустимо;

- обмеження № 2. В якості датчика можна використовувати лазерний далекомір, ефективний радіус якого вимірюється десятками метрів. Вибір лазерного далекоміра в якості сенсора дозволяє будувати карту у вигляді сітки зайнятості. Таким чином, карта, яка побудована в результаті роботи алгоритму, буде являти собою план будівлі, по якому переміщаються «s-bots»;

- обмеження № 3. Алгоритм не вирішує задачу керування мобільним «s-bots» для дослідження оточення. У задачу досліджуваного алгоритму входять лише обробка вимірювань, отриманих від лазерного далекоміра та датчика одометрії, та побудова карти;

- обмеження № 4. Приміщення, в якому переміщуються «s-bots» не повинно містити областей, що динамічно змінюються. Поява рідкісних і невеликих шумів можлива (наприклад, переміщення людей чи інших

мобільних «s-bots»). Однак не допускається переміщення меблів або колон у приміщенні, особливо в моменти, коли «s-bots» не спостерігає за відповідною частиною оточення;

- обмеження № 5. Відсутність перепаду висоти. Необхідно, щоб усі спостереження проводилися в одній площині. Крім того, для досягнення цієї вимоги потрібно щільно зафіксувати лазерний далекомір на «s-bots»;

- обмеження № 6. Всі «s-bots», що входять до складу однієї «S-bots» - systems і виконують алгоритм simultaneous localization and mapping, повинні бути однаково обладнані апаратно. Тобто не тільки мати однакові обчислювальні можливості, а й однакову мобільну платформу та однакові лідари. Ця вимога обумовлена тим, що кожен «s-bots» виконує однаковий алгоритм, тому структура вихідних даних у «s-bots» повинна співпадати. Якщо, наприклад, обчислювальні потужності одного з «s-bots» виявляться більше, ніж у іншого, то якість побудованої карти у «s-bots» з меншою потужністю виявиться гірше внаслідок браку часу на обробку даних. У цьому випадку карта, яка отримана в результаті злиття, буде не тільки містити області з різним ступенем чіткості, але й у разі перетину деяких областей менш чітка карта може внести значну похибку в результат;

- обмеження № 7. Область застосування досліджуваного алгоритму – це малобюджетні і, отже, низькопродуктивні «s-bots». Тому алгоритм не повинен містити кроків, на виконання яких існує обмеження часу (наприклад, обробка нових вимірів). Чим більше часу алгоритм отримує на роботу, тим точнішим буде результат – ця умова повинна дотримуватися при запуску на будь-яких пристроях. За таким принципом працює, наприклад, стохастичний скан-матчер Монте-Карло. Ідея така, що він шукає найточнішу позицію «s-bots», вибираючи її випадковим чином із області пошуку. Зрозуміло, що чим більше часу отримає такий скан-матчер на роботу, тим більша ймовірність вибору позиції, яка є найбільш близькою до оптимальної.

3.1 Опис багатоагентного алгоритму simultaneous localization and mapping

При дослідженні було встановлено, що для того, щоб алгоритм працював стійко, необхідно в якості вхідних даних отримувати дані лазерного скана та дані одометрії. Вже говорилося, що одометрія не є обов'язковим параметром, проте вона служить для збільшення точності. В алгоритмі немає суворої вимоги на наявність одометрії, проте її присутність дозволяє кожному «s-bots» окремо здійснювати обчислення з більшою точністю. Одометрія (як апіорна оцінка позиції) разом із новим лазерним сканом та побудованою на даний момент картою передається на вхід скан-матчера.

Скан-матчер обчислює різницю між апіорною та справжньою оцінкою позиції «s-bots». Справжньою позицією називається та, з якої можна виконати лазерний скан. Лазерний скан може суперечити карті через помилки скан-матчера на попередніх кроках алгоритму або через неповноту карти.

Для обчислення найбільш ймовірної позиції можна скористатися Байєсовським підходом до обчислення ймовірності. Тоді ймовірність позиції обчислюється як середня сума ймовірностей усіх точок скану, накладеного на карту із заданої позиції. Кожна точка скана означає перешкоду, і в ідеальних умовах кожна точка повинна потрапити до зайнятої клітини карти. Однак недостатньо ввести бінарний поділ на зайняті та вільні клітини. Насправді клітина карти може бути занадто великою, і тоді точки скана можуть розташовуватися в ній, як показано на рисунку 3.1.

Логічно вважати, що клітина має певну можливість бути зайнятою. Ця ймовірність заснована не тільки на факті, що на клітину потрапляє хоч одна точка лазерного скана, але й на тому, як багато променів із лазерного далекоміра може пройти крізь клітину до зустрічі з перешкодою.

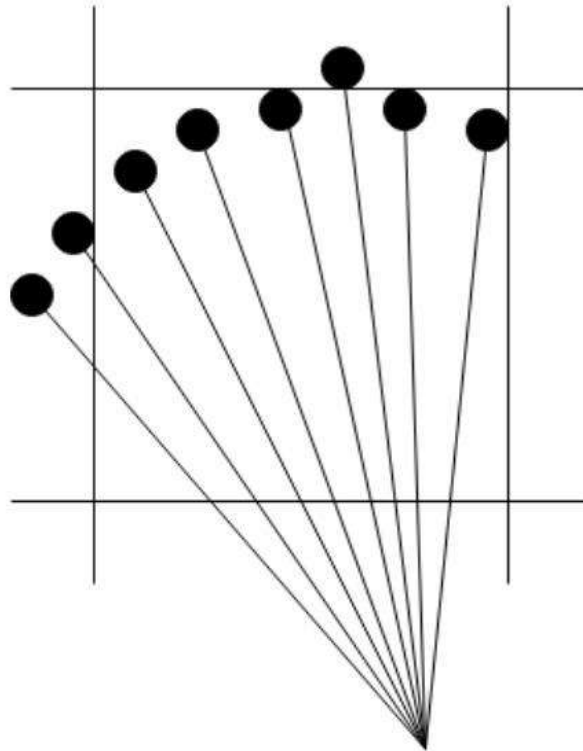


Рисунок 3.1 – Приклад частини лазерного скана. Багато точок скана потрапляє на клітину, проте таку клітину не можна вважати зайнятою

Така структура клітини відкриває можливість обчислювати ймовірність позиції. Ймовірність позиції може бути обчислена як середня сума ймовірностей клітин, в які потрапили точки лазерного скана:

$$p_{\text{pose}}(\text{scan}) = \sum_{p \in \text{points}} o_p \cdot \omega_p, \quad (3.1)$$

де p – точка скана,

o_p – зайнятість точки p на карті,

ω_p – вага точки скана.

Вага точки скана вводиться для того, щоб зробити роботу алгоритму стійкішою в умовах коридорів. У цьому випадку різні позиції «s-bots», що розташовані вздовж напрямку коридору, мають дуже близькі значення

ймовірностей. Відмінність величини ймовірностей вносять точки скана, відмінні від стін коридору. Зазвичай такі точки розташовуються протягом руху «s-bots». Отже, точки лазерного скана, що розташовані безпосередньо навпроти лазерного далекоміра, мають більшу вагу.

Задача скан-матчера: обчислити позицію з найбільшою ймовірністю (коли точки лазерного скана, випущені з цієї позиції, потрапили на клітини карти, що мають найбільшу ймовірність бути зайнятими).

Як тільки знайдено позицію з більшою ймовірністю, ніж усі попередні, пошук починає вестись у деякому радіусі навколо цієї позиції. Цей процес показаний на рисунку 3.2.

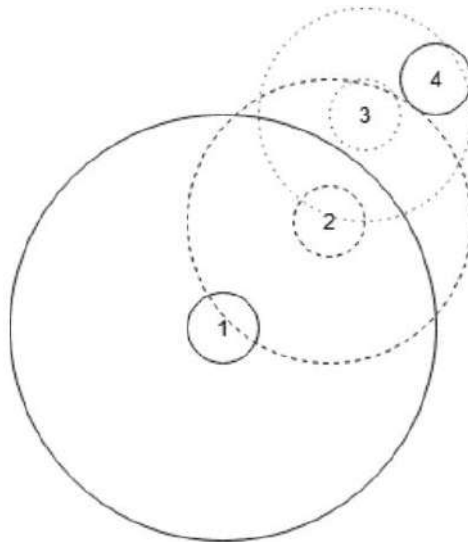


Рисунок 3.2 – Візуалізація ітераційного алгоритму стохастичного пошуку позиції з найбільшою ймовірністю

На рисунку 3.2 показано послідовність кроків. Позиція 1 є апріорною оцінкою позиції. З деяким радіусом навколо неї стохастично перебираються позиції та обчислюються їх ймовірності. Перебір ведеться до тих пір, поки не буде знайдена позиція 2 з більшою ймовірністю, ніж ймовірність позиції 1. Навколо позиції два будується менший радіус, ніж радіус навколо позиції 1. Потім операція повторюється кілька разів.

3.2 Застосування теорії Демпстера-Шафера для збільшення точності алгоритму

Згідно з Байєсовським підходом у кожній комірці карти міститься число від 0 до 1, що визначає ймовірність клітини бути зайнятою. Щоб обчислити цю ймовірність, необхідно оцінювати положення конкретних точок скана у клітині щодо точки, з якої спостерігався лазерний скан. Кожна клітинка карти має можливість p – бути зайнятою, і можливість q – бути вільною. Ці можливості пов'язані співвідношенням $p + q = 1$. Однак, у реальних умовах слід також запровадити змінну, яка описує стан «невідомо». Такий стан клітини може бути, коли вона знаходиться в ще не дослідженій області. Також клітина може бути в стані «невідомо», коли два послідовні скани суперечать один одному. Таким чином, клітина описується трьома ймовірностями:

- ймовірність бути зайнятою – p ;
- ймовірність бути вільною – q ;
- ймовірність бути невідомою – u .

Ці три ймовірності пов'язані між собою тотожністю $p + q + u = 1$. На відміну від попереднього підходу, де одна з ймовірностей виражається через іншу, цей підхід дозволяє висловлювати кожну вірогідність через пару інших. Постає запитання, як об'єднувати клітини, дотримуючись цих правил.

Для відповіді на це питання застосовується теорія Демпстера-Шафера.

Щоб суворо дотримуватися цієї теорії, необхідно також ввести стан конфлікту в клітині карти. З природного погляду конфлікт – це стан клітини не бути в жодній з можливих станів. Але насправді ніяка клітинка карти не може бути в стані конфлікту, а отже, цю ймовірність необхідно розподіляти між іншими станами.

Необхідно помітити, якщо клітина знаходиться в невідомому стані, вона знаходиться одночасно у вільному і зайнятому станах. У контексті розглянутого завдання теорія Демпстера-Шафера застосовується під час

роботи скан-матчера, коли необхідно з'ясувати ймовірність клітини картки бути зайнятою.

Відразу після роботи скан-матчера настає другий етап застосування цієї теорії. У порожні комірки карти, які потрапили в область спостереження лідара, необхідно помістити можливість бути вільними і зайнятими. Для цього застосовується правило об'єднання ймовірностей, що описується формулою:

$$m_{1,2}(A) = \frac{1}{1-K} \sum_{B \cap C = A \neq 0} m_1(B) \cdot m_2(C), \quad (3.2)$$

де m – це ймовірність стану комірки;

A, B, C – це різні стани: чи є комірка вільною, зайнятою або невідомою (вільною або зайнятою).

Значення K обчислюється за такою формулою:

$$K = \frac{1}{1-K} \sum_{B \cap C = 0} m_1(B) \cdot m_2(C), \quad (3.3)$$

де m – це ймовірність стану;

B, C – це різні стани.

Робота цієї формули наведена в таблиці 3.1, де показано злиття двох комірок із встановленими ймовірностями.

Таблиця 3.1 – Об'єднання ймовірностей клітин бути зайнятими згідно з теорією Демпстера-Шафера

Ймовірності	Клітина 1 (m_1)	Клітина 2 (m_2)	Об'єднана клітина (m_{12})
Зайнята (p)	0.35	0.11	0.33
Вільна (q)	0.25	0.31	0.40
Невідомо (u)	0.4	0.58	0.27

Більш детально пояснити величину $m_{12}(p)$ можна, розписавши докладніше

формулу (3.2):

$$m_{1,2}(p) = \frac{m_1(p) \cdot m_2(p) + m_1(p) \cdot m_2(u) + m_1(u) \cdot m_2(p)}{1 - (m_1(p) \cdot m_2(q) + m_1(q) \cdot m_2(p))}, \quad (3.4)$$

де m_1 , m_2 , m_{12} – це ймовірність стану клітини;

p – зайнятий стан;

q – вільний стан;

u – невідомий стан, коли комірка одночасно і зайнята, і вільна.

Вже згадувалося, що такий підхід поєднання клітин застосовується на етапі, коли необхідно зіставити клітину карти та точку лазерного скана. Для цього спочатку точку скана необхідно перетворити на клітину з певним розподілом ймовірності. Вид розподілу – це окрема дослідницька задача, яка виходить поза рамки даної кваліфікаційної роботи. У цій кваліфікаційній роботі ймовірність вільного стану комірки прямо пропорційна довжині відрізка, який долає промінь лазерного далекоміра, проходячи крізь клітину, до моменту зустрічі з перешкодою.

Таким чином, можна запровадити поняття зайнятості точки скана. На рисунку 3.3 показано одну точку скана, накладену на комірку карти. Зайнятість точки дорівнює відношенню довжини відрізка b до довжини $a+b$.

У випадку Байєсовської теорії комірка містить одне число – можливість бути зайнятою. Після завершення роботи скан-матчера, коли обчислена справжня позиція «s-bot», що спостерігає конкретний лазерний скан, необхідно оновити це число для кожної комірки. Тоді можна в одній системі координат, центр якої знаходиться в поточній позиції «s-bot», розмістити і поточну карту, і спостереження. У цей момент скан буде накладено на карту так само, як і на етапі роботи скан-матчера. Однак тепер немає необхідності обчислювати вартість такого накладення. Потрібно

просто оновити карту відповідно до такого накладання скана і перевірити, у яких клітинах карти знаходяться точки лазерного скана. Далі необхідно змінити значення зайнятості у кожній клітині відповідно до того, як саме точка потрапляє у клітину.

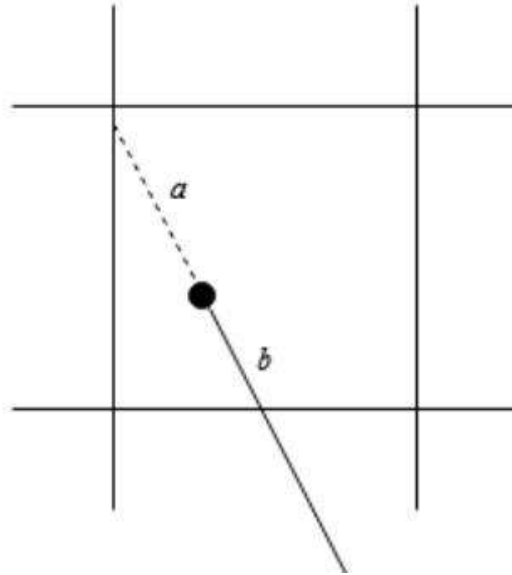


Рисунок 3.3 – Обчислення зайнятості точки лазерного скана, що потрапила в клітину карти, через відношення $b/(a+b)$

Найбільш очевидним є спосіб оновлення комірки обчисленням середнього значення між характеристиками зайнятості клітини та зайнятості точки. Очевидним недоліком такого підходу буде висока чутливість комірки карти до похибок лазерного далекоміра. На рисунку 3.4 наведено графік зміни характеристики зайнятості клітини в залежності від характеристик зайнятості точок скану, що сильно коливаються.

На графіку синім кольором відображається зайнятість точки скана, а коричневим кольором відображається зайнятість клітини. Аналізуючи графік, можна зробити висновок, що при сильному коливанні зміни зайнятості точки, зайнятість клітини сильно коливається. Таке коливання зайнятості точки може бути пов'язане з шумом лазерного далекоміра, коли точка виявляється то на ближньому, то на дальньому кордоні клітини.

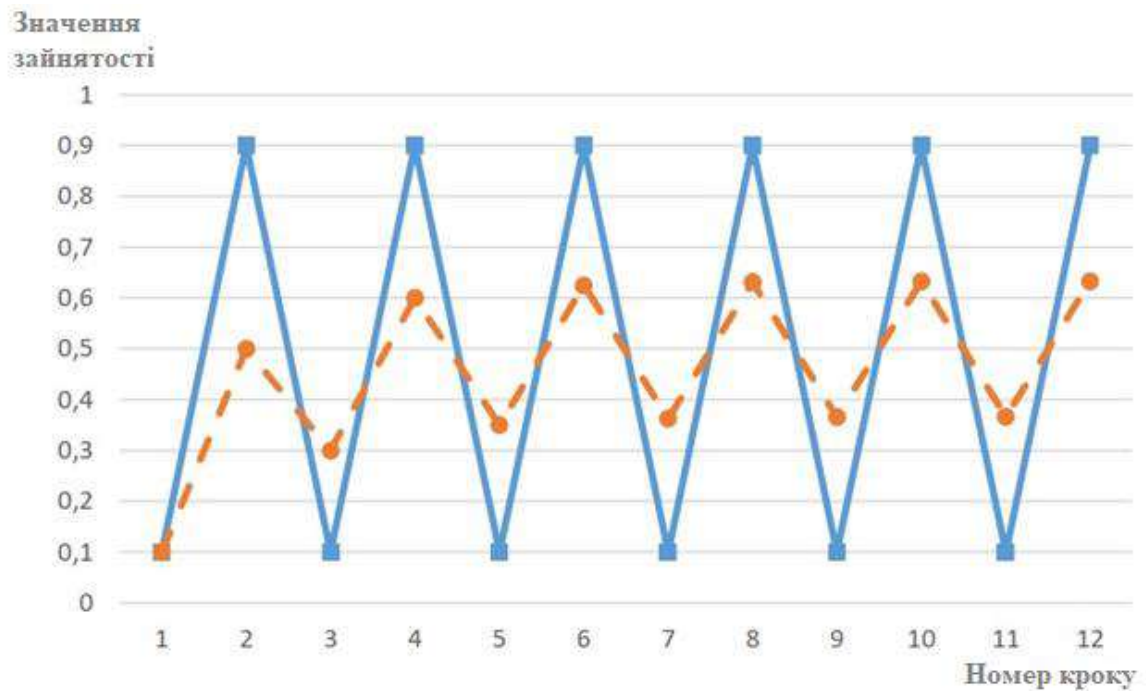


Рисунок 3.4 – Коливання зайнятості клітини в залежності від зайнятості точки при оновленні зайнятості відповідно до середнього значення

Для того, щоб позбавитися такої високої локальної чутливості, але зберегти адекватність моделі, застосовується оцінка зайнятості комірки відповідно до середнього арифметичного всіх значень, накопичених на попередніх кроках. Виходячи з послідовності значень зайнятості точки скана, зайнятість клітини за n кроків обчислюється за формулою:

$$p_C = \frac{1}{n} \sum_{i=1}^n p_i \quad (3.5)$$

де p_C – зайнятість комірки карти;

p – зайнятості точки скана;

n – кількість спостережень.

Такий підхід зменшує чутливість до промахів далекоміра, і графік залежності зайнятості клітини карти від зайнятості точки, за наявності промаху, згладжується сильніше (рисунок 3.5).

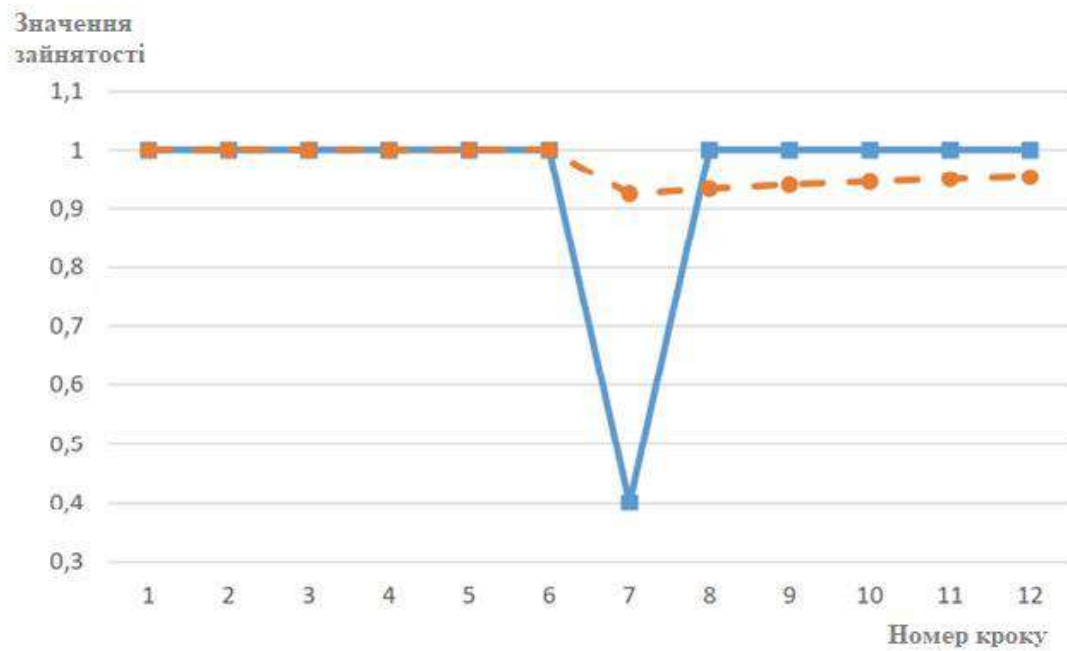


Рисунок 3.5 – Зайнятість клітини при наявності промаху при використанні середнього значення зайнятості всіх точок, що надійшли раніше

На графіку синім кольором відображається зайнятість точки скана, а коричневим кольором відображається зайнятість клітини.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було проведено огляд та аналіз існуючих одноагентних алгоритмів, що вирішують задачу simultaneous localization and mapping. Після проведення порівняльного аналізу та класифікації було побудовано класифікацію одноагентних алгоритмів за різними характеристиками.

Було проведено огляд багатоагентних алгоритмів, що вирішують задачу simultaneous localization and mapping. Після цього було досліджено масштабований алгоритм багатоагентного рішення задачі simultaneous localization and mapping. Продемонстровано, що він має низьку обчислювальну складність за рахунок застосування методу Монте-Карло до задачі скан-матчингу, а також відмови від графової структури алгоритму та застосування теорії Демпстера-Шафера. Горизонтальна масштабованість алгоритму досягається за рахунок відсутності поділу ролей «s-bots», що виконують його.

Було проведено імітаційне моделювання обчислення ймовірностей клітин бути зайнятими, згідно з теорією Демпстера-Шафера. Після проведення імітаційного моделювання, було збудовано графіки коливання зайнятості клітини, залежно від зайнятості точки при оновленні зайнятості відповідно до середнього значення.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Пат. 118921 Україна, МПК H04W 64/00. Спосіб передачі цифрових даних мультикоптерною системою між сегментами розподіленої сенсорної мережі та базовою станцією / В.М. Ткачов, В.В. Токарев - № u201704085; заявл. 24.04.2017; опубл. 28.08.2017. Бюл. № 16. 5с.

2. Створення науково-методичних основ забезпечення живучості мережевих систем обміну інформацією в умовах зовнішнього впливу потужного НВЧ випромінювання: звіт про НДР (заключ.) № держреєстрації 0117U003916.: Ф76/109-2017 / Харків. нац. ун-т радіоелектроніки; керівник Г. И. Чурюмов. – Харків, 2017. – 116 с.

3. Ruban I.V., Churyumov G.I., Tokarev V.V., Tkachov V.M. Provision of Survivability of Reconfigurable Mobile System on Exposure to High-Power Electromagnetic Radiation / I.V. Ruban, G.I. Churyumov, V.V. Tokarev, V.M. Tkachov // Selected Papers of the XVII International Scientific and Practical Conference on Information Technologies and Security: (ITS 2017). CEUR Workshop Processing., 30 nov. 2017 y. - Kyiv, 2017. - P. 105-111.

4. Serkov A., Kravets V., Yakovenko I., Churyumov G., Tokariev V., Nannan W. Ultra Wideband Signals in Control Systems of Unmanned Aerial Vehicles / A. Serkov, V. Kravets, I. Yakovenko, G. Churyumov, V. Tokariev, W. Nannan // The 10th IEEE International Conference on Dependable Systems, Services and Technologies: (DESSERT'2019)., 5-7 june, 2019 y. - Leeds, 2019. - P.26 - 29.

5. Серков О. А., Пустовойтов П. Є., Яковенко І. В., Лазуренко Б. О., Чурюмов Г. І., Токарев В. В., Наннан Ванг. Надширокосмугові технології в системах управління мобільними об'єктами. / О. А. Серков, П. Є. Пустовойтов, І. В. Яковенко, Б. О. Лазуренко, Г. І. Чурюмов, В. В. Токарев, Ванг Наннан // Сучасні інформаційні системи. - 2019. - Т.3, №2. - С.22-27.

6. Серков О.А., Князев В.В., Лазуренко Б.О., Яковенко І.В., Чурюмов

Г.І., Токарев В.В. Надширокосмугові технології в задачах забезпечення електромагнітної сумісності рухомих об'єктів / О.А. Серков, В.В. Князев, Б.О. Лазуренко, І.В. Яковенко, Г.І. Чурюмов, В.В. Токарев // Проблеми електромагнітної сумісності перспективних бездротових мереж зв'язку (EMC-2019):збірник наукових робіт четвертої міжн. наук.-техн. конф., 24 жовт. 2019 р. - Харків, 2019. - С. 55-57.

7. Krivoulya G., Tokariiev V., Iilina I, Shcherbak V. Mathematical Model for Finding Probability of Detecting Victims of Man-Made Disasters Using Distributed Computer System with Reconfigurable Structure and Programmable Logic / G. Krivoulya, V. Tokariiev, I. Iilina, V. Shcherbak // IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology: (PIC S&T), 06-09 oct. 2020y. - Kharkiv, 2020. - P.197 - 201.

8. Кривуля Г.Ф, Токарев В.В., Щербак В.К. Забезпечення живучості безпроводних сенсорних мереж з використанням якірних вузлів // Інформаційно-керуючі системи на залізничному транспорті: тези доповідей 35-ї міжн. наук.-практ. конф., 14 листопада. 2022р. - Харків. - С. 49- 50.

9. Krivoulya G., Tokariiev V., Iilina I., Lebediev O., Shcherbak V. Algorithm of Iterations of Distribution of Subtasks Between «S-Bot» in One «Swarm-Bot» System // Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Systems: (COLINS 2022). CEUR Workshop Proceedings., 12-13 may. 2022 y. - Gliwice, Poland. - P. 1531-1541.

10. Г. Кривуля, В. Токарев, І. Ільїна, В. Кравець. Взаємодія між «s-bots» однієї «Swarm-bot» system у фізичному неорганізованому середовищі. // Системи управління, навігації та зв'язку. - 2023. №1(71). - С.108-111. Doi: 10.26906/SUNZ.