

**ДОДАТОК А**  
**Апробація результатів роботи**

1

Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра комп'ютерно-інтегрованих технологій, автоматизації та робототехніки

**VIII Міжнародна Конференція  
ВИРОБНИЦТВО  
&  
МЕХАТРОННІ СИСТЕМИ 2024**



**VIII International Conference  
MANUFACTURING  
&  
MECHATRONIC SYSTEMS 2024**



2024

VII International Conference

25-26 October

Kharkiv

## ЗМІСТ

<i>Svitlana Alyokhina</i>	
System Approach to the Positive Energy District Analysis .....	12
<i>Dmytro Gurin</i>	
Розробка динамічного представлення параметрів моделі опису навколишнього середовища колаборативного робота .....	15
<i>Artem Hubar</i>	
Automation of Power Grid Element Management to Enhance Energy Efficiency .....	19
<i>Артем Бронніков, Стеценко Катерина</i>	
Автономний робот на Raspberry Pi з аналізом облич та емоцій в реальному часі .....	22
<i>Andrii Lvov, Svetlana Sotnik</i>	
Analysis of electronic locks existing systems .....	24
<i>Artem Tverdokhlib, Svetlana Sotnik</i>	
Intelligent tools for optimizing information and search engines .....	28
<i>Igor Zarubin, Svetlana Sotnik</i>	
Basic principles of building aerial robots .....	32
<i>Pavlo Sukhno, Svetlana Sotnik</i>	
Critical review of GSM network structure .....	37
<i>Oleksii Shevchenko, Nataliia Furmanova, Vadim Yakovenko, Yaroslav Lukash</i>	
Assessment of the quality of brushless DC motors .....	42
<i>Artem Zhulai, Nataliia Furmanova</i>	
System for monitoring and alerting in a coal mine .....	45
<i>Сніжана Вичужаніна, Олександр Малий</i>	
Огляд щодо використання радіоаматорами радіочастотного спектру в Україні .....	48

<i>Воронов Денис, Сезонова Ірина</i>	
Розробка методу визначення швидкості переміщення об'єктів на основі аналізу зображень .....	51
<i>Oleh Hurtovyi</i>	
Features of Functional Testing for Low-Power Consumption Devices with Built-In Batteries .....	55
<i>Варвара Карташова, Артем Бронніков</i>	
Роль експертних систем та голосового керування в сучасному виробництві .....	58
<i>Антон Паньков</i>	
Інноваційний підхід до візуалізації: розробка автоматизованого модуля для збору, обробки та збереження поточних даних .....	62
<i>Олег Посашков, Олександр Цимбал</i>	
Аналіз існуючих методів підтримки прийняття рішень у віддаленому управлінні виробництвом .....	65
<i>Дмитро Максимов, Дмитро Нікітін</i>	
Види зварювання для верстату точкового зварювання з ЧПУ .....	69
<i>Олексій Фарафонов, Наталія Фурманова, Олександр Малий</i>	
Розроблення технології паралельного керування за допомогою вебінтерфейсу мобільним роботом під керуванням ROS .....	71
<i>Дмитро Янушкевич, Леонід Іванов, Ігор Толкунов</i>	
Застосування методів вербального аналізу в інтелектуальних системах управління у сфері гуманітарного розмінування .....	75
<i>Данило Ясир</i>	
Вибір математичної моделі для управління якістю продукції в умовах безперервного виробництва .....	79
<i>Дмитро Дриньов</i>	
Використання елементів штучного інтелекту для вирішення задач моделювання динамічних процесів .....	83
<i>Ганна Самойленко</i>	

# Розробка методу визначення швидкості переміщення об'єктів на основі аналізу зображень

Воронов Денис<sup>1</sup>, Сезонова Ірина<sup>2</sup>

1. Кафедра КІТАР, Харківський національний університет радіоелектроніки, УКРАЇНА, Харків, пр. Науки. 14., email: denys.voronov@nure.ua

2. Кафедра КІТАР, Харківський національний університет радіоелектроніки, УКРАЇНА, Харків, пр. Науки. 14., email: iryna.sezonova@nure.ua

**Анотація:** В даному матеріалі наведено приклад використання бібліотеки OpenCV для розробки методу аналізу зображень.

**Ключові слова:** OpenCV, Алгоритм, інтеграція.

## I. ВСТУП

Сучасні системи моніторингу дорожнього руху, відеоспостереження, аналіз спортивних подій та системи безпеки значною мірою залежать від алгоритмів, що здатні точно вимірювати швидкість та тип об'єктів. Визначення швидкості може бути корисним для покращення безпеки на дорогах, запобігання нещасним випадкам, а також для автоматизації багатьох рутинних процесів, таких як контроль транспортних потоків. Одним із найбільш поширених і потужних інструментів для аналізу зображень є бібліотека OpenCV, вона надає безліч функцій для реалізації складних алгоритмів обробки відео. Її впровадження дозволить оптимізувати процес аналізу виробничих систем стеження та безпеки, а також надасть нові можливості для їх розвитку

## II. МЕТОД АНАЛІЗУ ЗОБРАЖЕНЬ

OpenCV (Open Source Computer Vision Library) — це відкрита бібліотека для комп'ютерного зору та обробки зображень, яка надає широкий набір інструментів для аналізу візуальних даних, таких як фото, відео та зображення з камер у реальному часі. Вона підтримує різні мови програмування (Python, C++, Java), працює на різних операційних системах (Windows, macOS, Linux, Android) і включає понад 2500 алгоритмів для таких задач, як розпізнавання об'єктів, трекінг руху, аналіз оптичного потоку тощо.

OpenCV дозволяє обробляти зображення в реальному часі завдяки підтримці апаратного прискорення (GPU), що робить її особливо корисною в системах відеомоніторингу, автоматизації, автомобільній промисловості, медицині та робототехніці. Вона має велику спільноту розробників, що забезпечує легку інтеграцію та широкі можливості для розробки в сфері комп'ютерного зору.

OpenCV надає широкий набір інструментів і функцій, що дозволяють значно спростити створення алгоритму шляхом застосування вже готових модулів для детекції, трекінгу об'єктів та аналізу їх переміщень. Основні етапи, які можна автоматизувати за допомогою OpenCV, включають попередню обробку зображень, виділення контурів об'єктів, трекінг їх руху між кадрами та розрахунок параметрів швидкості. Це дозволяє реалізувати систему в реальному часі, що є важливою перевагою в контексті сучасних виробничих процесів, де необхідна швидка і точна обробка великих обсягів даних.

Прикладом може виступати аналіз руху машин у площі дорожньої розв'язки розбитої на сектори рис.1.

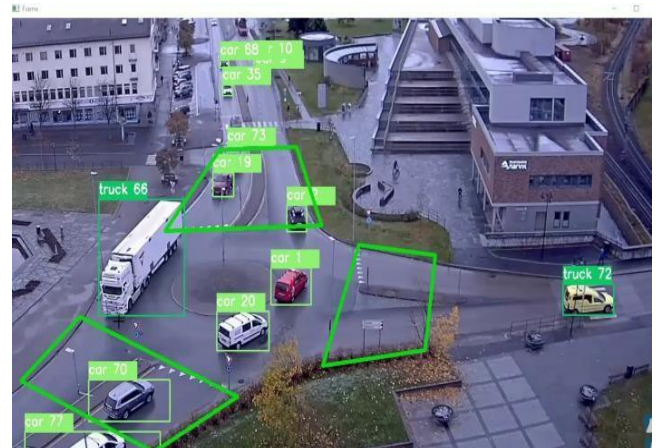


Рис.1. аналіз руху машин у площі дорожньої розв'язки

Алгоритм на основі OpenCV здатний забезпечити високу точність завдяки використанню методів оптичного потоку, кореляції та інших підходів для аналізу руху. Це дозволяє автоматично визначати швидкість переміщення об'єктів без втручання людини, що значно підвищує ефективність і зменшує ймовірність помилок на виробництві. Крім того, можливість інтеграції з іншими системами, такими як роботизовані комплекси чи автоматизовані лінії контролю якості, дозволяє використовувати ці дані для автоматичної корекції процесів або прогнозування можливих відхилень.

У разі невикористання OpenCV або аналогічних бібліотек розробка алгоритму потребуватиме створення власних рішень для кожного етапу, включно з обробкою зображень, виділенням об'єктів, аналізом руху та обчисленням швидкості. Це не тільки значно збільшить час розробки, але й підвищить складність оптимізації алгоритму для роботи в реальному часі. Крім того, такі алгоритми можуть виявитися менш точними та продуктивними, оскільки не використовуватимуть готових і перевірених рішень, які вже є у складі OpenCV. Відсутність готових засобів для паралельної обробки даних і використання апаратного прискорення також негативно вплине на продуктивність і масштабованість системи, що є важливим для великих виробничих процесів.

Розробка власного методу визначення швидкості переміщення об'єктів для виробництва на основі OpenCV може мати кілька важливих переваг порівняно з наявними аналогами. Перш за все, власний метод дозволяє гнучко адаптувати алгоритми під конкретні потреби

підприємства. Це особливо важливо для виробництв з унікальними технологічними процесами, де типові рішення можуть не повністю відповідати вимогам щодо точності, продуктивності або інтеграції. Власний метод дозволяє оптимізувати систему під специфічні умови експлуатації, такі як розмір об'єктів, швидкість їх руху та особливості виробничих процесів, що забезпечує підвищену ефективність. Прикладом може слугувати зчитування визначених заздалегідь об'єктів на виробництві, таких як коробки, навантажувачі, люди тощо рис.2.



Рис.2. зчитування визначених заздалегідь об'єктів. Ще однією перевагою є можливість глибокої інтеграції з наявними інформаційними системами підприємства, що дозволяє отримати повну автоматизацію, від збору даних до ухвалення рішень. Наприклад, метод може бути інтегрований з системами контролю якості, логістики або управління запасами, що створює єдиний інформаційний простір для управління виробництвом. Власний метод може бути також масштабованим, забезпечуючи можливість легкої адаптації до нових умов або розширення виробничих потужностей без необхідності суттєвих змін у програмному забезпеченні.

#### 1. ПРИНЦИП РОБОТИ АЛГОРИТМУ

Алгоритми OpenCV працюють за принципом аналізу зображень і відео з використанням методів комп'ютерного зору. Основний процес включає кілька етапів, починаючи від попередньої обробки вхідних даних і закінчуючи виведенням необхідних результатів, таких як координати об'єктів, їх траєкторії або швидкість переміщення.

Перший етап — завантаження та обробка зображень або відео. OpenCV може працювати з різними форматами файлів, зчитувати потоки з камер у реальному часі або завантажувати статичні зображення з файлової системи. На цьому етапі зазвичай проводиться попередня обробка: перетворення зображень у необхідний формат (наприклад, в сірі тони), зменшення шумів або інші фільтраційні операції для покращення якості даних рис.3.

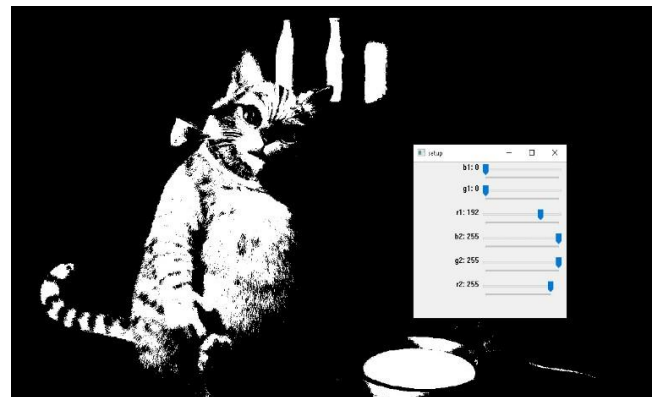


Рис.3. фільтраційні операції для покращення якості даних

Другий етап — виділення об'єктів. Це ключовий момент у роботі алгоритмів OpenCV. Для цього використовують різні методи, такі як порогова обробка (thresholding), детекція контурів (edge detection) або алгоритми сегментації зображень. Часто застосовують методи детекції країв, такі як алгоритм Канні (Canny) рис.4, який виділяє межі об'єктів у зображенні, або методи оптичного потоку, що дозволяють відслідковувати зміну положення пікселів у послідовності кадрів.

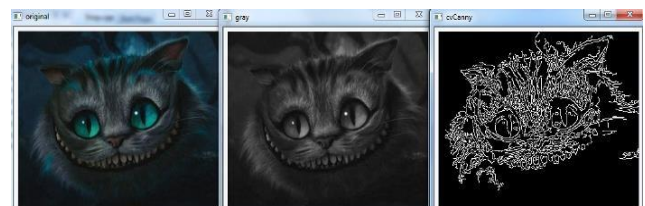


Рис.4. алгоритм Канні (Canny)

Третій етап — трекинг об'єктів. Після того, як об'єкти виділені, їх необхідно відстежувати у наступних кадрах відео. Це досягається шляхом порівняння позицій об'єктів між кадрами. Існує кілька підходів до трекингу, зокрема методи, засновані на кореляції, або використання більш складних алгоритмів, таких як Kalman Filter (рис.5) або Mean Shift. Такі алгоритми дозволяють враховувати зміну положення об'єктів навіть у складних умовах, таких як часткове перекриття або зміни освітлення.

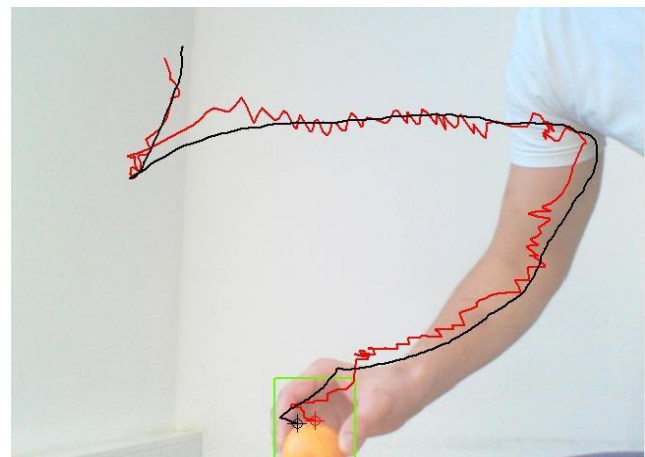


Рис.5. Kalman Filter

Четвертий етап — аналіз руху та розрахунок швидкості. Для визначення швидкості переміщення об'єктів OpenCV використовує методи оптичного потоку (optical flow), які дозволяють аналізувати зміну позицій пікселів між кадрами і визначати напрямок та швидкість руху об'єктів. Найпоширеніший алгоритм для цього — метод Лукаса-Канаде (Lucas-Kanade) рис.6, який дозволяє обчислювати вектор швидкості кожного пікселя, що переміщається між кадрами. На основі отриманих даних можна розраховувати середню швидкість об'єктів або їх траєкторії.

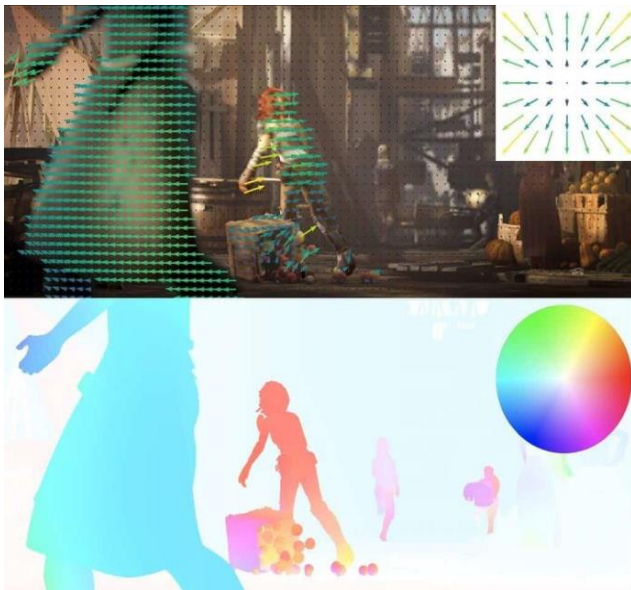


Рис.5. метод Лукаса-Канаде (Lucas-Kanade)

Останній етап — інтерпретація результатів та виведення. Після завершення аналізу руху алгоритм OpenCV виводить оброблені дані у зручному форматі — це можуть бути як числові дані (координати, швидкість), так і графічні зображення (відображення траєкторій руху на зображеннях). OpenCV також дозволяє інтегрувати результати в інші системи, такі як системи керування виробничими процесами або автоматизовані аналітичні системи.

Завдяки таким крокам OpenCV забезпечує високу точність і продуктивність під час аналізу зображень і відео, що робить його незамінним інструментом для реалізації алгоритмів визначення швидкості об'єктів у різних виробничих або наукових проектах.

Прикладом де представлений кожен етап може бути система контролю за рухом конвеєрних стрічок та автоматизації виробничого процесу на заводах, що виготовляють і пакують продукцію. Камери встановлюються вздовж виробничої лінії, де постійно рухаються вироби чи деталі по конвеєру. Камери фіксують відео або роблять серію зображень продукції під час її руху по стрічці. За допомогою OpenCV проводиться аналіз зображень для виділення об'єктів (виробів), які рухаються

по конвеєру. Застосовуються методи для трекінгу об'єктів між кадрами для оптичного потоку, щоб точно визначити, як швидко рухаються об'єкти. Алгоритми дозволяють виміряти швидкість кожного виробу або деталі та відстежувати їхню траєкторію. Якщо швидкість руху на конвеєрі виходить за межі допустимих параметрів, система автоматично коригує швидкість конвеєра або видає попередження оператору. Така система також здатна виявляти збої, коли продукція неправильно рухається або застрягає. Це дозволяє значно підвищити точність і швидкість контролю над виробничим процесом, знизити ризики простоїв або пошкодження продукції, а також оптимізувати роботу обладнання без втручання людини.

## I. ВИСНОВКИ

Впровадження власного методу визначення швидкості переміщення об'єктів на основі аналізу зображень в автоматизовані виробничі процеси приводить до підвищення точності моніторингу та контролю, оптимізації роботи обладнання, автоматизації збору даних і аналізу руху об'єктів у реальному часі. Це дозволяє підприємству ефективніше керувати ресурсами, зменшити кількість помилок і збоїв, а також забезпечити безперервність і якість виробництва. Така інтеграція сприяє підвищенню продуктивності, зниженню витрат на технічне обслуговування, мінімізації простоїв та покращенню конкурентоспроможності підприємства.

**ДОДАТОК Б**  
**Демонстраційний графічний матеріал**

Розроблення системи автоматичного керування мобільним роботом на основі аналізу просторово-геометричної інформації на виробничому підприємстві

Кваліфікаційна робота студента

Воронова Д.О гр. КІТПВм-23-1.

Спеціальність: 174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка

Керівник: проф. каф. КІТАР Сезонова І.К.

## Актуальність

У сучасному виробництві автоматизація відіграє ключову роль у підвищенні ефективності та продуктивності. Використання мобільних роботів із системою автоматичного керування дозволяє вирішувати завдання, пов'язані з логістикою, транспортуванням і контролем якості у реальному часі. Технологія LiDAR забезпечує точне просторове сканування та створення тривимірних карт, що робить її незамінною для роботи в умовах динамічного середовища.

## Об'єкт, предмет, мета роботи

- Об'єктом дослідження є процес автоматизованого керування мобільними роботами у виробничих умовах.
- Предмет дослідження охоплює методи, алгоритми та моделі навігації, засновані на використанні LiDAR.
- Основна мета роботи – підвищення ефективності автоматизованих систем шляхом розробки алгоритмів та програмної реалізації динамічного вибору маршруту з урахуванням змін середовища.

## Завдання роботи

У межах дослідження були поставлені наступні завдання:

1. Провести аналіз сучасних систем автоматизації мобільних роботів.
2. Розробити алгоритми пошуку оптимального шляху (A\*) та уникнення перешкод (AABB).
3. Вибрати апаратні компоненти для реалізації системи.
4. Реалізувати програмну модель системи.
5. Провести тестування в умовах змодельованого середовища.

## Аналіз сучасних технологій

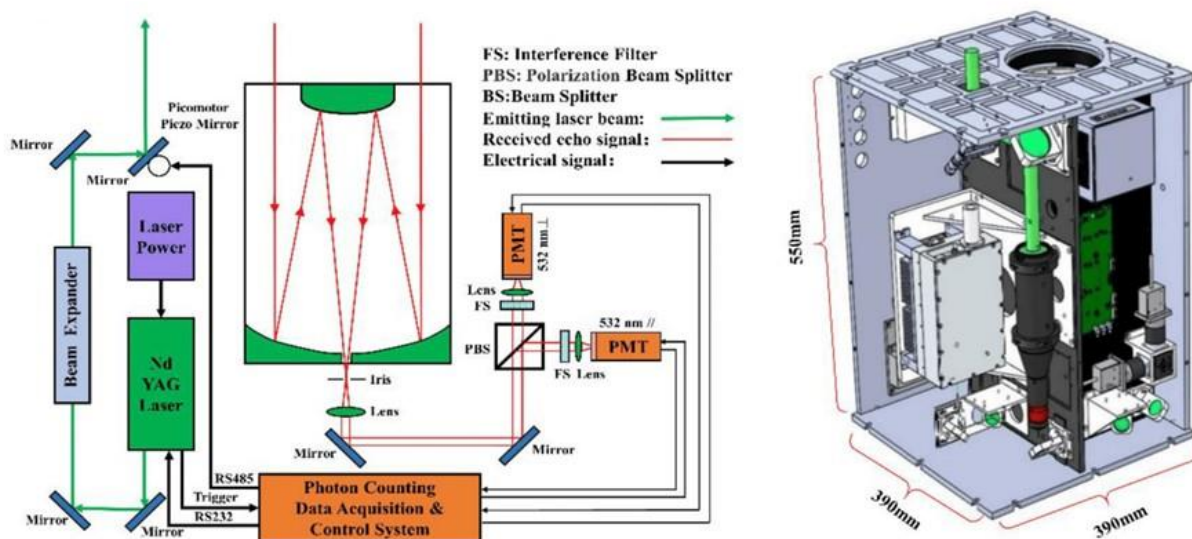
На основі аналізу сучасних технологій було виявлено, що LiDAR є одним із найефективніших сенсорів для створення тривимірних карт середовища. Він дозволяє визначати розташування об'єктів із високою точністю.

Алгоритми A\* та AABV широко застосовуються для навігації мобільних роботів, забезпечуючи оптимальний пошук маршруту та ефективно уникнення зіткнень із перешкодами.

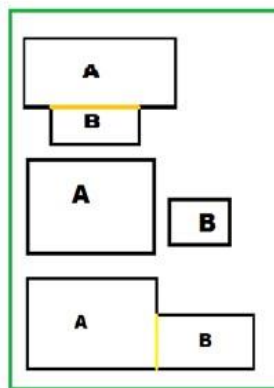
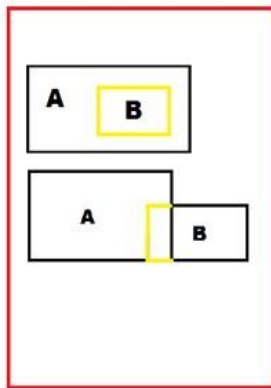
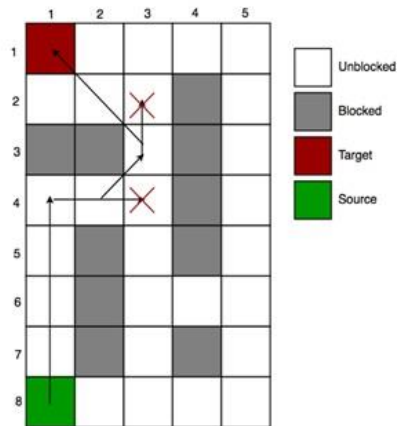


# Вибір апаратних компонентів

Для реалізації системи було обрано LiDAR як основний сенсор, що забезпечує збір просторово-геометричної інформації. Для обробки даних і управління системою були обрані алгоритми A\* та AABV. Програмне забезпечення було реалізовано на мові Kotlin, що дозволило досягти високої продуктивності та гнучкості системи.



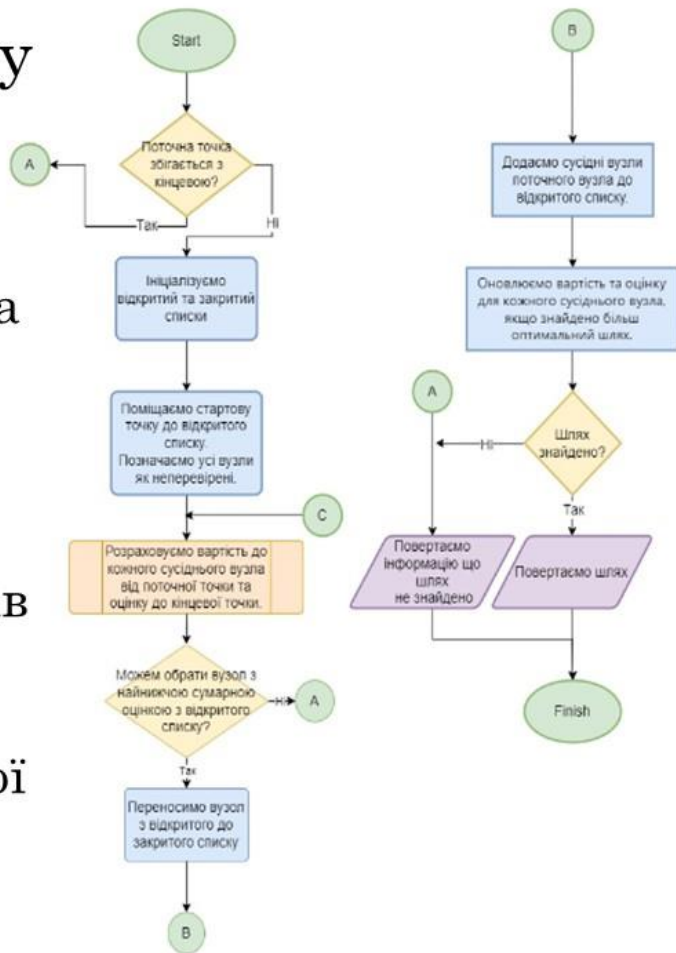
# Розробка алгоритмів



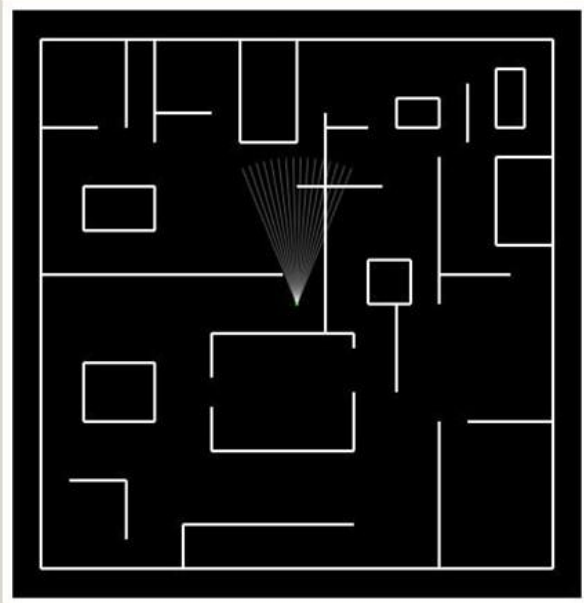
Алгоритм A\* використовується для пошуку оптимального маршруту з урахуванням розташування перешкод. AABV застосовується для визначення зіткнень із перешкодами, використовуючи мінімально обмежувальні рамки. Завдяки інтеграції цих алгоритмів забезпечується адаптивність системи до змін середовища в режимі реального часу. Програмне забезпечення дозволяє динамічно перераховувати маршрут у разі появи нових перешкод.

## Схема алгоритму пошуку шляху

Алгоритм пошуку шляху, базований на  $A^*$ , працює за принципом поступового оцінювання можливих маршрутів і вибору найоптимальнішого шляху від початкової точки до цільової.



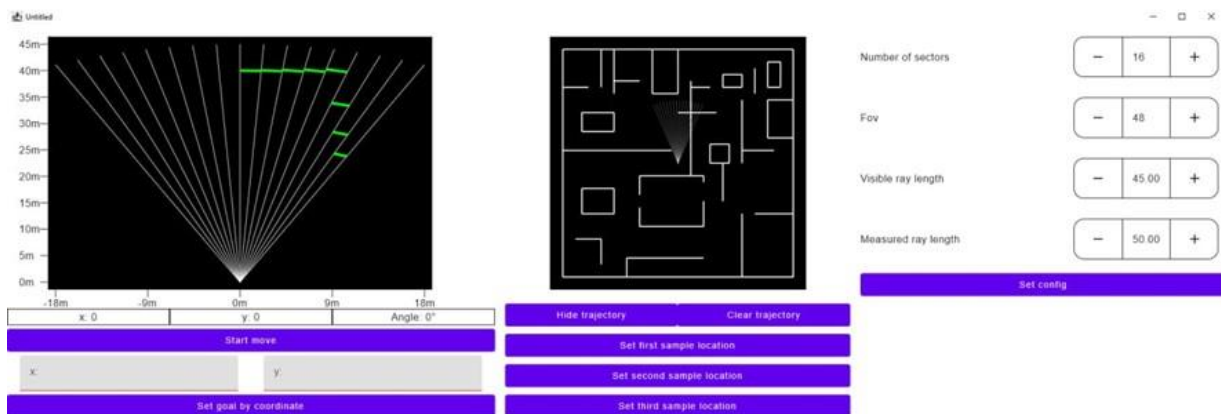
# Тестування системи



Для тестування розроблено змодельоване середовище з перешкодами різної складності. Алгоритми були перевірені на точність побудови маршруту та здатність обходити перешкоди. Результати показали, що система може адаптуватися до змін середовища та ефективно виконувати завдання.

# Результати роботи

У результаті роботи було створено систему, яка забезпечує точну та динамічну навігацію мобільного робота. Програмне забезпечення дозволяє візуалізувати маршрут та дії робота в реальному часі. Система показала високу ефективність під час тестування у змодельованому середовищі.



## Перспективи розвитку

Подальший розвиток системи може включати інтеграцію додаткових сенсорів, таких як камери або ультразвукові датчики, для підвищення точності. Використання елементів штучного інтелекту дозволить прогнозувати рух перешкод і створювати ще більш складні маршрути. Це сприятиме впровадженню системи в реальні виробничі процеси.

## Висновки

Робота продемонструвала можливість створення адаптивної системи навігації мобільного робота на основі LiDAR. Розроблені алгоритми забезпечують точність і ефективність, що відкриває перспективи для подальшого впровадження у виробництво. Система може бути використана для автоматизації складських, логістичних і промислових процесів.

**ДОДАТОК В**  
**Лістинг програми**

```
package factory

import model.GridSpace
import model.Node
import model.Point
import util.minus
import java.util.PriorityQueue
import kotlin.math.sqrt

internal class PathFactory {
    fun get(start: Point, goal: Point, space: GridSpace): List<Point> {
        val openList = PriorityQueue<Node>()
        val closedList = hashSetOf<Node>()
        val startNode = Node(Point(start.x.toDouble(), start.y.toDouble()), null)
        startNode.g = 0.0
        startNode.h = heuristic(Point(start.x.toDouble(), start.y.toDouble()), goal)
        startNode.f = startNode.g + startNode.h
        openList.add(startNode)
        while (openList.isNotEmpty()) {
            val current = openList.poll()
            if (current.point == goal) {
                return getPath(current)
            }
            closedList.add(current)
            for (neighbor in space.getNeighbors(current.point)) {
                val tentativeGScore = current.g + distance(current.point, neighbor)
                val neighborNode = Node(neighbor, current)
                neighborNode.g = tentativeGScore
                neighborNode.h = heuristic(neighbor, goal)
            }
        }
    }
}
```

```

    neighborNode.f = neighborNode.g + neighborNode.h
    if (closedList.any { it.point == neighborNode.point } && tentativeGScore >=
neighborNode.g) {
        continue
    }
    if (openList.any { it.point == neighborNode.point } && tentativeGScore >=
neighborNode.g) {
        continue
    }
    openList.add(neighborNode)
}
}
return emptyList()
}

```

```

private fun getPath(node: Node): List<Point> {
    val path = mutableListOf<Point>()
    var current = node
    while (current.parent != null) {
        path.add(current.point)
        current = current.parent!!
    }
    path.add(current.point)
    return path.reversed()
}

```

```

private fun distance(point1: Point, point2: Point): Double {
    val dx = (point1.x - point2.x).toDouble()
    val dy = (point1.y - point2.y).toDouble()

```

```
    return sqrt(dx * dx + dy * dy)
}

private fun heuristic(point: Point, goal: Point): Double {
    return distance(point, goal)
}
}
```

