

Факультет	Автоматики і комп'ютеризованих технологій
	(повна назва)
Кафедра	Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки
	(повна назва)

2025 p.

Я, Ярош-Іванов Микита Віталійович, як здобувач вищої освіти ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Я не використовував штучний інтелект для підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

18 грудня 2025 р.

Микита ЯРОШ-ІВАНОВ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет Автоматики і комп'ютеризованих технологій
Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки
Рівень вищої освіти другий (магістерський)
Спеціальність 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка
Тип програми освітньо-професійна
Освітня програма Комп'ютеризовані та робототехнічні системи

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

« __ » _____ 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачу Ярош-Іванову Микиті Віталійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка системи технічного зору для системи управління сортувальним роботом

Затверджена наказом по університету від 10.11.2025 р. No 1018 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 27.12.2025 р.

3. Вихідні дані до роботи _____

3.1 Система технічного зору;

3.2 Середовище розробки Microsoft Visual Studio;

3.3 Бібліотека OpenCV;

3.4 Мова програмування Python.

4. Перелік питань, що потрібно опрацювати в роботі _____

4.1 Вступ;

4.2 Аналіз предметної області;

4.3 Інтеграція технічного зору в систему управління промисловим роботом;

4.4 Аналіз бібліотеки OpenCV, вибор мови програмування;

4.6 Практична реалізація системи технічного зору;

4.6 Питання пов'язані з охороною праці;

4.7 Висновки.

5. Перелік графічного матеріалу із зазначенням креслень, схем, плакатів, комп'ютерних ілюстрацій Демонстраційний матеріал, представлений у форматі презентації PowerPoint (*.pptx). – с. формату А4.

6. Консультанти розділів роботи

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз технічного завдання	01.09.25-10.09.25	Виконано
2	Опрацювання літератури за темою	11.09.25-24.09.25	Виконано
3	Інтеграція технічного зору у систему управління промисловим роботом	25.09.25-4.10.25	Виконано
4	Аналіз бібліотеки OpenCV і вибор ПЗ	5.10.25-17.10.25	Виконано
5	Практична реалізація системи технічного зору	18.10.25-12.12.25	Виконано
6	Оформлення пояснювальної записки	14.12.25-16.12.25	Виконано
7	Подання роботи на нормоконтроль	17.12.25	Виконано
8	Подання роботи на перевірку Інтернет-сервісом StrikePlagiarism	24.12.25	Виконано
9	Подання роботи на рецензію		
10	Подання роботи на підпис зав. кафедри		
11	Подання атестаційної роботи в ЕК		

Дата видачі завдання 1.09.2025

Здобувач _____
(підпис)

Микита ЯРОШ-ІВАНОВ
(прізвище, ініціали)

Керівник роботи _____
(підпис)

доц. Андрій ФРОЛОВ
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 75 с., 1 табл., 17 рис., 2 дод., 28 джерел.

ТЕХНІЧНИЙ ЗІР, ПРОМИСЛОВИЙ МАНІПУЛЯТОР,
КАЛІБРУВАННЯ КАМЕРИ, РОЗПІЗНАВАННЯ, ПОШУК КОНТУРУ.

Об'єктом дослідження є визначення положення виробу в просторі та його належність до певного класу.

Предметом дослідження є технічний зір, що використовує алгоритм пошуку виробу за кольором.

Метою кваліфікаційної роботи є розробка системи управління сортувальним роботом із застосуванням технології комп'ютерного зору, здатної без участі людини здійснювати сортування виробів на конвеєрній лінії за заданими заздалегідь ознаками.

У процесі дослідження були вивчені основні функції та сфери застосування технічного зору. Розроблено алгоритм роботи системи керування сортувальним роботом та обрано програмне забезпечення для написання програмного коду роботи системи. Спроектовано технічну частину системи і проведено експерименти для визначення ефективності роботи системи (протестовано метод пошуку виробу за кольором).

ABSTRACT

Explanatory note: 75 p., 1 tabl., 17 fig., 2 app., 28 sources.

TECHNICAL VISION, INDUSTRIAL MANIPULATOR, CAMERA CALIBRATION, RECOGNITION, LOOP SEARCH.

The object of the study is to determine the position of the product in space and its belonging to a certain class.

The subject of the study is technical vision, using an algorithm for finding a product by color.

The purpose of the qualification work is to develop a control system for a sorting robot using computer vision technology, capable of sorting products on a conveyor line according to predetermined signs without human intervention.

During the study, the main functions and areas of application of technical vision were studied. The algorithm of the control system of the sorting robot is developed and the software for writing the program code of the system operation is chosen. The technical part of the system was designed and experiments were conducted to determine the efficiency of the system (test the method of searching for a product by color).

ЗМІСТ

Список скорочень	8
Вступ	9
1 Структура та особливості систем технічного зору роботів	12
1.1 Основні функціональні задачі систем технічного зору у робототехніці	12
1.2 Використання відеоінформації у системах управління роботами	15
1.3 Висновки до розділу	23
2 Інтеграція технічного зору у систему управління промисловим роботом	24
2.1 Розробка математичної моделі опису відео з камери	24
2.2 Постановка задачі розробки системи технічного зору для промислового робототехнічного комплексу	34
2.3 Вибір інструментів програмування	37
2.4 Висновки до розділу	43
3 Практична реалізація проекту системи технічного зору	44
3.1 Розміщення камер	46
3.2 Пошук об'єкта за кольором	47
3.3 Контурний аналіз	53
3.4 Алгоритм роботи програми та її реалізація	54
3.5 Охорона праці	58
3.6 Висновки до розділу	59
Висновки	60
Перелік джерел посилення	61
Додаток А Апробація результатів наукових досліджень	64
Додаток Б Демонстраційний матеріал	74

СПИСОК СКОРОЧЕНЬ

СТЗ – системи технічного зору;

ERP – Enterprise Resource Planning (планування ресурсів підприємства);

MES – Manufacturing Execution System (система управління виробничими процесами);

OpenCV – Open Source Computer Vision Library (бібліотека алгоритмів комп'ютерного зору);

ROS – Robot Operating System (операційна система для роботів);

USB – Universal Serial Bus (універсальна послідовна шина).

ВСТУП

У світі широко поширене використання обробки зображень для вирішення різних задач автоматичного контролю, управління та виміру.

Щоб промислові роботи та інші системи автоматизації виробництва цілеспрямовано взаємодіяли з об'єктами та спокійно рухалися у просторі, що оточує виробничі лінії, вони повинні мати можливість бачити та аналізувати довкілля.

Це саме те, із чим комп'ютерні системи зору можуть допомогти. Машинний зір включає аналіз візуальної інформації для подальшого прийняття рішень. Це допомагає зробити будь-яку дію щодо об'єкта, який знаходиться у фокусі. Найпростіший приклад використання технології: перевірка стану продукту на конвеєрній стрічці або перед відправкою поштою.

Історично алгоритми машинного зору впроваджувалися у невеликій кількості у виробництво, оскільки це було дуже затратно. Однак вартість, продуктивність та споживання енергії в електронних системах, які були досягнуті сьогодні, проклали шлях для поширення машинного зору в різних галузях. Звичайно, їх реалізація все ще складна, але це вже стало набагато простіше і дешевше, ніж будь-коли.

У цій роботі технологія машинного зору буде впроваджена в систему керування сортувальним роботом.

Задача сортування схожа на задачу розпізнавання образів і полягає у вимірі координат і орієнтації об'єктів. Об'єкти, які мають відносно просту форму, в даному випадку прямокутну, однозначно характеризуються контуром зображення, що реєструється. Для вирішення задач оцінки параметрів і розпізнавання, що характеризують орієнтацію та координати

таких об'єктів, використовують алгоритми, засновані на аналізі контурів зображень.

Нині на ринку представлені рішення на різних принципах визначення положення, наприклад, з урахуванням аналізу структурованого світла, з урахуванням часопростірних принципів, стереоскопічні рішення із виділенням особливих точок зображення. Але основною популярністю як і раніше користуються класичні рішення на основі цифрових відеокамер, так як вони дозволяють за невеликі вкладення отримати варіант системи визначення координат, що легко налаштовується, і класу об'єкта з високою надійністю і простотою установки [1,2].

Метою кваліфікаційної роботи є розробка системи управління сортувальним роботом із застосуванням технології комп'ютерного зору, здатної без участі людини здійснювати сортування виробів на конвеєрній лінії за заданими заздалегідь ознаками.

Об'єктом дослідження є визначення положення виробу в просторі та його належність до певного класу.

Предметом дослідження є технічний зір, що використовує алгоритм пошуку виробу за кольором.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- провести аналіз основних функцій та сфер застосування технічного зору;
- сформулювати задачу розробки системи технічного зору для промислового робототехнічного комплексу;
- провести аналіз та обрати програмне забезпечення для написання програмного коду роботи системи сортування;
- розробити алгоритм роботи системи керування сортувальним роботом;

– розробити технічну частину і провести експерименти для визначення ефективності роботи системи (протестувати метод пошуку виробу за кольором);

– оформити кваліфікаційну роботу згідно ДСТУ 3008:2015 [3], а також з методичними вказівками із підготовки та оформлення кваліфікаційних робіт здобувачів другого (магістерського) рівня спеціальності 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка [4].

Результати дослідження наведено у збірнику студентських наукових статей «Автоматизація та приладобудування» ADED-2025 [5].

Результати роботи відповідають Цілі сталого розвитку 9: «Промисловість, інновації та інфраструктура».

1 СТРУКТУРА ТА ОСОБЛИВОСТІ СИСТЕМ ТЕХНІЧНОГО ЗОРУ РОБОТІВ

1.1 Основні функціональні задачі систем технічного зору у робототехніці

Роботи, забезпечені СТЗ, досить широко і успішно застосовуються в багатьох галузях промисловості на таких операціях, як контроль і відбраковування різних деталей і виробів за зовнішнім виглядом, завантаження обробних верстатами деталей з конвеєрів або з бункерів, сортування і орієнтоване укладання в касети деталей і заготовок, у тому числі неупорядковано навалом, збирання та комплектація вузлів, зварювання, фарбування, пакування виробів, монтаж електронних схем та інших. СТЗ використовують також для забезпечення умов техніки безпеки на роботизованих робочих місцях, організації обліку та складування предметів виробництва, маршрутизації матеріалопотоків у гнучких виробничих системах, керування транспортними мобільними роботами.

Функціональне призначення СТЗ робота в залежності від галузі та конкретних умов його застосування досить різноманітне. Найбільш типовими функціями, що виконуються СТЗ у робототехніці, є наступні: реєстрація наявності об'єкта в поле зору датчика зовнішньої (стосовно роботи) відеоінформації; підрахунок числа об'єктів, що знаходяться в полі зору або пройшли перед відеосенсором; зчитування та розшифрування міток (колірних або штрихових кодів, буквено-цифрових символів, етикеток та інших маркерів); виявлення перешкод, що заважають рухам робота; вимірювання геометричних та фізичних параметрів об'єктів; контроль виробів на наявність видимих дефектів, повноту комплектації, відповідність стандартам; класифікація (розпізнавання) об'єктів; визначення місцезнаходження та орієнтації об'єктів у робочій зоні робота; пошук необхідних об'єктів робочої

сцени; вимірювання швидкості об'єктів, що рухаються; стеження за переміщеннями та змінами швидкості об'єктів; візуальне визначення змін у часі спостережуваних фізичних та технологічних процесів; забезпечення інформації для задачі та корекції траєкторій робочого органу робота в адаптивному режимі; навігація та наведення мобільних роботів; візуальний контроль правильності виконання операцій роботом; калібрування геометричних параметрів маніпулятора; прив'язка його системи координат до системи координат робочого простору; забезпечення вимог техніки безпеки.

Функціональні задачі СТЗ, характерні для робототехнічних додатків, можна умовно розділити за рівнем їх відносної складності. До елементарних задач зазвичай належать: виявлення наявності об'єкта; вимірювання відстані до об'єкта, його лінійних чи кутових переміщень, швидкості; вимірювання геометричних параметрів об'єкта (лінійних та кутових розмірів, площі тощо); визначення фізичних характеристик випромінювання від об'єкта (інтенсивності, спектрального складу та ін.); підрахунок числа об'єктів та ін.

Більш складне коло задач виконує, наприклад, СТЗ, яка забезпечує систему управління маніпуляційним роботом інформацією, необхідною для захоплення неупорядкованих об'єктів. До цих задач входять: огляд робочої сцени (плоскої чи просторової) для пошуку представляючого інтерес об'єкта – одиночного чи однієї з кількох, лежачого ізольовано від інших чи такого, що стикається (перекривається) із ними; визначення розташування та орієнтації цього об'єкта; його класифікація (якщо є кілька класів об'єктів). При цьому об'єкти можуть відрізнятися як розмірами і формою (силуетом), так і текстурою, кольором тощо, спочивати чи перебувати у русі. Цей приклад ілюструє той факт, що відносна складність функціональних задач СТЗ характеризується цілою низкою ознак, серед яких можна виділити такі.

Розмірність сцени, що розглядається. Просторові задачі, що вимагають аналізу тривимірних сцен, як правило, набагато складніші за «плоскі» задачі, в яких достатньо обробки зображення однієї двовимірної проекції сцени на картинну площину. Проміжне положення займають квазіплоскі задачі

(«розмірності 2,5»), де розглядається картинна площина з додаванням інформації про «глибину» (дальність) видимих точок зображення.

Ступінь ізолюваності об'єктів. За цією ознакою задачі СТЗ класифікуються так (у порядку зростання складності): задачі з одиночним об'єктом; задачі з декількома ізолюваними (тобто недоторканими) об'єктами на зображенні; задачі з зображеннями об'єктів, що стикаються (але не перекриваються); задачі із перекриттям одних об'єктів робочої сцени іншими.

Однорідність об'єктів. Ця ознака характеризує складність задач класифікації, що стоять перед СТЗ. Якщо всі об'єкти, які можуть бути пред'явлені СТЗ, ідентичні, ця проблема взагалі не виникає. За наявності ряду неоднакових об'єктів задачі їх класифікації тим складніше, що «схожі» одне одного їх зображення, т. е. що більше класифікуючих ознак необхідно однозначного описи відмінностей кожного об'єкта від інших.

Стаціонарність робочої сцени. Зображення об'єктів, що покояться, обробляти та аналізувати простіше, ніж зображення об'єктів, які переміщуються щодо відеосенсорів (наприклад, при русі деталей по конвеєру, зоровому очуєтвленні робота, роботі СТЗ типу «очей на руці»).

Число елементів зображення. Чим більше розміри поля зору і чим вище необхідна роздільна здатність СТЗ, тим складніше задачі, оскільки доводиться обробляти більше елементів зображення (пікселів).

Число градацій яскравості. Задачі, у яких досить аналізувати силуетні обриси об'єктів, т. е. працювати з двоградаційними (бінарними) зображеннями, зазвичай, вирішуються простіше задач, потребують аналізу багатоградаційних (напівтонових) зображень.

Спектральний склад відеосигналу. Наділити робот здатністю сприймати кольорові зображення, взагалі кажучи, складніше, ніж забезпечити його монохроматичним зором.

Крім того, на складність тих чи інших задач СТЗ істотно впливають такі фактори, як якість освітлення робочої сцени, його стабільність, рівень оптичних та електричних перешкод, контрастність об'єктів щодо фону, що

відображають властивості їх поверхні та ін.

Чим складніші задачі з урахуванням всієї сукупності описаних критеріїв здатна вирішувати СТЗ, тим вище рівень функціональної гнучкості («інтелекту») робота може бути забезпечена її допомогою.

Для виконання свого функціонального призначення СТЗ робота в загальному випадку повинна забезпечувати: сприйняття оптичного сигналу та формування зображення; попередню обробку зображення з метою ослаблення впливу шумів, поліпшення контрастності, корекції спотворень, стиснення інформації іт. д.; сегментацію зображення сцени на складові – виділення потрібних об'єктів, їх фрагментів або характерних особливостей; опис зображень – розрахунок їх геометричних та інших характеристик, обчислення класифікуючих ознак, визначення розташування та орієнтації; аналіз зображення з розпізнаванням образів чи класифікацією об'єктів та інтерпретацією сцен на основі моделі проблемного середовища; передачу одержаних результатів у систему управління робота.

Створення СТЗ робота, що відповідає її конкретному функціональному призначенню, включає такі етапи: 1) постановку задачі на змістовному рівні та складання технічного завдання на СТЗ; 2) вибір структурної та функціональної схем СТЗ; 3) математичне формулювання задач функціональних блоків СТЗ; 4) розробку алгоритмів розв'язання функціональних задач; 5) вибір засобів реалізації алгоритмів та конкретизації технічних характеристик апаратних, апаратно-програмних та програмних блоків СТЗ відповідно до технічного завдання; 6) розробку апаратного та програмного забезпечення СТЗ.

Перш ніж перейти до розгляду цих питань, зупинимось на особливостях роботи з технічним зором.

1.2 Використання відеоінформації у системах управління роботами

Сенсорна система робота повинна забезпечувати його систему

управління інформацією про поточну ситуацію у зовнішньому середовищі: наявність, тип, параметри, місцезнаходження та орієнтацію об'єктів маніпулювання (впливу); правильності та якості виконання роботом технологічних операцій та/або інших дій; існування перешкод і шляхи їх обходу тощо. У систему управління надходять також задачі (команди) роботи від людини-оператора, інших роботів, ЕОМ вищих рівнів, технологічного устаткування чи інших пристроїв.

У сучасних робототехнічних системах найбільш переважною є багаторівнева ієрархічна структура управління [6].

Інформація від СТЗ може надходити на різні рівні системи управління. Наприклад, дані про розташування перешкод бувають потрібні на *рівні* для побудови моделі робочого середовища з метою планування дій робота; результати класифікації об'єктів можуть знадобитися на *стратегічному* рівні для розчленування загального плану дій на конкретні маніпуляційні операції, задачі їхньої послідовності та параметрів; відомості про місцезнаходження та орієнтацію об'єктів необхідні для формування на тактичному рівні необхідних рухів робочого органу та ланок маніпулятора, за якими, у свою чергу, будуються програмні закони узгодженої зміни відповідних ступенів рухливості; інформація про відхилення фактичної траєкторії (скажімо, зварного шва) від запрограмованої може бути використана безпосередньо на виконавчому рівні для вироблення керуючих сигналів на приводи при відпрацюванні програми з метою корекції переміщення робочого органу робота.

У реальних робототехнічних системах часто складно розмежувати задачі описаних рівнів управління. Функції верхніх рівнів багато в чому перебирає людина-оператор. Нижній рівень керуючої системи фізично може бути реалізований безпосередньо в силових блоках виконавчих пристроїв, інші рівні зазвичай будуються на базі універсальних або спеціалізованих обчислювальних блоків.

Розглянемо способи включення СТЗ до робототехнічних комплексів.

Якщо СТЗ мобільних роботів зазвичай встановлюються на платформі, що рухається, і переміщаються разом з роботом, то при зоровому очуствленні маніпуляційних роботів використовуються два основних методи установки відеосенсорів.

В більшості промислових робототехнічних систем з технічним зором відеосенсиори встановлюються ізольовано від маніпулятора (зазвичай їх розташовують нерухомо зверху чи збоку робочої сцени). Можна використовувати кілька відеосенсорів для огляду сцени з різних точок, можливі керовані переміщення ліній зору сенсорів [6]. Незалежна від маніпуляційного робота установка відеосенсорів має ту перевагу, що геометричні співвідношення, що зв'язують їх системи координат із системою координат робочого простору, не змінюються в процесі роботи. Крім того, багато операцій СТЗ та керуючого пристрою маніпуляційного робота можна виконувати паралельно, коли маніпулятор виходить із поля зору відеосенсора. З іншого боку, саме при захопленні об'єктів, коли їх бачити найважливіше, рука робота загороджує частину робочої сцени.

У цьому сенсі певну перевагу мають СТЗ типу «очей на руці»: у них сам маніпулятор ніколи не потрапляє у поле зору відеосенсора. Важливо також, що вони дозволяють наводити робочий орган об'єкт без вимірювання абсолютних координат останнього. Проте роботи з «очима на руці» поки поширені порівняно нешироко, насамперед, внаслідок великої складності «прив'язки» кількісної інформації про геометрію об'єктів, що отримується, до нерухомої координатної системи. Найбільш перспективними є адаптивні робототехнічні комплекси, в яких поєднуються обидві розглянуті вище схеми установки відеосенсорів, а також використовуються комбінації СТЗ з сенсорами інших модальностей в єдиній мультисенсорній системі [6].

Однак незалежно від способу розміщення відеосенсорів, отримана з їх допомогою інформація використовується для управління адаптивним роботом відповідно до одного з двох наступних принципів. Перший базується на безперервному (або квазінеперервному) надходженні відеосигналу зворотного

зв'язку для корекції траєкторії руху робочого органу робота. За цим принципом діють сервосистеми, які мінімізують відхилення поточного стану робочого органу від бажаного, що визначається за даними візуального спостереження за цільовим об'єктом. Другий принцип заснований на введенні інформації від СТЗ в пристрій управління робота дискретними порціями. Кожна така порція служить для вироблення програмних рухів на черговий інтервал часу, протягом якого робот переміщається в наступне цільове положення «наосліп», тобто без безперервного візуального зворотного зв'язку.

Кожен із зазначених принципів використання відеоінформації для управління роботами має свою сферу застосування. Так, адаптивні зварювальні роботи, візуально визначаючи стик деталей, що зварюються, коригують рух головки вздовж шва при безперервному надходженні сигналу зворотного зв'язку. Такий спосіб знаходить застосування при взятті рухомих деталей з конвеєра, стеженні за смугою, що направляє рух мобільного робота, підтримці заданої відстані від фарбопульту до поверхні, що фарбується, і т.п. В рамках цього підходу в останні роки розробляються методи візуального сервоуправління безпосередньо за видом одержуваного зображення або за параметрами, що розраховуються в результаті його обробки. Так, у системі «очей на руці» сервопривід переміщає маніпулятор до тих пір, поки не будуть досягнуті потрібні значення таких параметрів, як положення будь-якої обраної точки об'єкта в системі координат «очей», площа зображення певної грані об'єкта, відношення площ сусідніх граней і т. п. Хоча системи з «візуальним сервоуправлінням», безсумнівно, перспективні, досі набагато ширше поширені СТЗ, засновані на другому з описаних вище принципів – «целевказанні». Розглянемо дещо докладніше методи використання зорової інформації у таких СТЗ роботів.

У найпростішому випадку за результатами роботи СТЗ на підставі апріорно зазначених умов приймається рішення тільки про запуск або зупинення (переривання) заздалегідь заданої програми дій робота, переході на ту чи іншу жорстку підпрограму, зміну послідовності виконання команд

керуючої програми, повна інформація про кожну з яких повинна бути внесена до керуючого пристрою ще до початку роботи робота.

Більш широкі можливості забезпечуються при адаптивній зміні самих керуючих програм робота відповідно до фактичної обстановки, що визначається за інформацією від СТЗ. Адаптивні роботи здатні автоматично формувати рухи у процесі роботи без необхідності апіорної вказівки людиною детальних законів зміни всіх керованих координат.

Розглянемо, наприклад, порівняно нескладну задачу адаптивного позиціонування маніпулятора під час захоплення попередньо неорієнтованої деталі, що лежить на робочій площині. Робот повинен взяти об'єкт, положення та орієнтація якого на робочій площині визначаються СТЗ за інформацією з телекамери, яка нерухомо укріплена над сценою (рис. 1.1). Введемо такі системи координат: $O_a X_a Y_a Z_a$ – абсолютна; $O_o X_o Y_o Z_o$ – пов'язана з об'єктом; $O_p X_p Y_p Z_p$ – пов'язана з основою робота; $O_z X_z Y_z Z_z$ – пов'язана із захватним пристроєм (робочим органом) робота; $O_k X_k Y_k Z_k$ – пов'язана з телекамерою, точніше – з її картинною площиною, на яку відображається зображення робочого поля.

При розрахунку переходів з однієї системи координат до іншої дуже зручний математичний апарат однорідних координат [6]. Однорідними координатами точки в тривимірному просторі називаються чотири числа h_1, h_2, h_3, h_4 не рівні всі одночасно нулю та пов'язані з її декартовими координатами x, y, z співвідношеннями $x = h_1/h_4, y = h_2/h_4, z = h_3/h_4$. Наприклад, точці (x, y, z) можна поставити у відповідність вектор однорідних координат $(h_1 \ h_2 \ h_3 \ 1)^T$. За допомогою однорідних координат можна компактно описати перехід із деякої координатної системи O_0XYZ в систему $O_1X'Y'Z'$, зсунуту і повернену щодо першої:

$$h' = T_0^1 \cdot h,$$

де h' – вектор однорідних координат в новій системі; h – вектор однорідних координат у старій системі, а матриця переходу має вигляд

$$T_0^I = \begin{pmatrix} x_{x''} & y_{x''} & z_{x''} & O_{0x''} \\ x_{y''} & y_{y''} & z_{y''} & O_{0y''} \\ x_{z''} & y_{z''} & z_{z''} & O_{0z''} \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

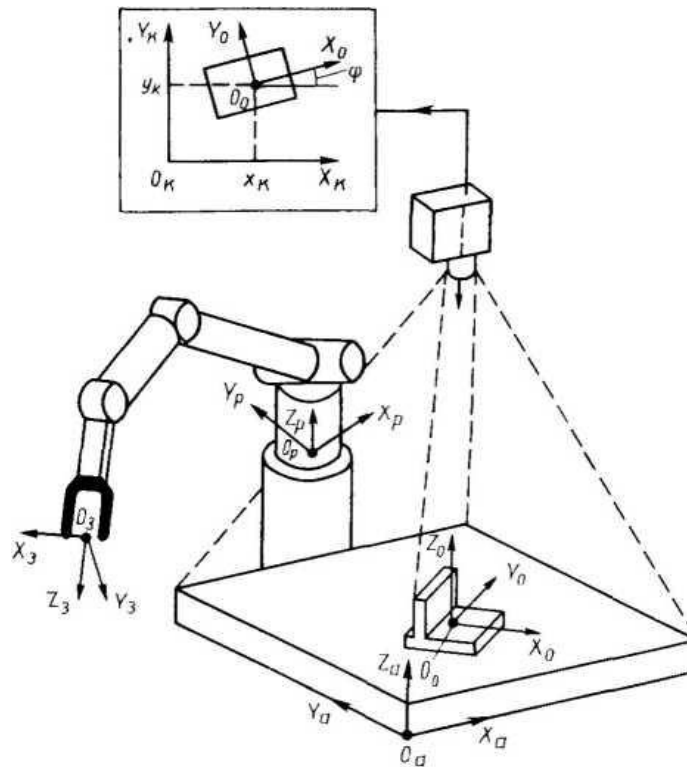


Рисунок 1.1 – Приклад задачі взяття роботом неорієнтованого об'єкта за інформацією від СТЗ

Тут останній стовпець являє собою вектор однорідних координат початку вихідної системи в новій координатній системі, а 3×3 -матриця, що стоїть у верхньому лівому кутку, описує поворот старої системи щодо нової (її елементами є координати одиничних векторів осей вихідної системи в новій після поєднання їх початків шляхом паралельного перенесення).

Щоб виконати завдання, в прикладі, що розглядається, робот повинен вивести захоплюючий пристрій в таке положення, при якому об'єкт виявився б між його губками, а їх орієнтація забезпечувала надійне утримання об'єкта. Положення та орієнтацію захватного пристрою можна описати у вигляді 4×4 -

матриці переходу T^R , що визначає поворот і паралельне перенесення системи $O_3 X_3 Y_3 Z_3$ щодо будь-якої обраної координатної системи, в якій найбільш зручно задавати необхідний стан захватного пристрою. Для кожного конкретного типу об'єкта маніпулювання (і для кожного стійкого положення цього об'єкта в робочій площині) можна заздалегідь вибрати найбільш зручне номінальне положення захоплення, тобто виписати матрицю переходу T_0^3 , що задає необхідне (цільове) положення системи координат захватного пристрою $O_3 X_3 Y_3 Z_3$ в системі координат $O_o X_o Y_o Z_o$, пов'язаній з об'єктом.

З рис. 1.1 можна зробити висновок, що положення системи $O_o X_o Y_o Z_o$ у картинній площині телекамери (т. е. у системі координат $O_k X_k Y_k Z_k$) в даному випадку вдається описати матрицею

$$T_o^k = \begin{pmatrix} \cos \varphi & -\sin \varphi & 0 & x_k \\ \sin \varphi & \cos \varphi & 0 & y_k \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

де φ – кут орієнтації об'єкта в робочій площині, який може бути знайдений СТЗ в реальному часі, як і координати x_k, y_k геометричного центру силуету об'єкта.

При установці та калібруванні телекамери за допомогою спеціальних методик можна визначити матрицю переходу T_k^a із системи $O_k X_k Y_k Z_k$ абсолютну систему координат $O_a X_a Y_a Z_a$. Аналогічно мають бути відомі місцезнаходження основи робота та його просторова орієнтація, тобто перетворення T_a^p з абсолютної системи $O_a X_a Y_a Z_a$ у систему $O_p X_p Y_p Z_p$.

Нагадаємо, що саме в системі координат $O_p X_p Y_p Z_p$, пов'язаній з основою робота, вирішується зворотна кінематична задача: за заданим станом (положенням та орієнтацією) захватного пристрою визначаються необхідні («програмні») значення всіх керованих координат робота. Знаючи матриці переходу $T_a^p, T_k^a, T_0^k, T_3^0$ шляхом їх послідовного перемноження зліва можна розрахувати необхідний стан захватного пристрою в системі $O_p X_p Y_p Z_p$ як суперпозицію перерахованих перетворень: $(T^R)_p = T_a^p T_k^a T_0^k T_3^0$.

Зауважимо, що можна відразу провести калібрування телекамери, так,

щоб, пройшовши проміжні переходи в абсолютну систему координат, отримати матрицю, яка зв'язує системи $O_K X_K Y_K Z_K$ та $O_P X_P Y_P Z_P$. З використанням цієї матриці T_K^P розрахунок програмного стану захватного пристрою по черговій порції інформації про координати x_K, y_K і орієнтації ϕ об'єкта, що надходить від СТЗ, дещо спрощується: $(T^R)_P = T_K^P T_0^K T_3^0$.

Отже, в даному прикладі СТЗ забезпечила цілевказівку системі управління робота, яка по матриці $(T^R)_P$ може вирішити задачу адаптивного позиціонування, тобто визначити програмні значення всіх керованих координат робота для поточного робочого інтервалу і виробити управляючі впливи на виконавчі приводи для відпрацювання оперативно сформованих програмних значень, що і дозволяє вивести захоплюючий пристрій (вже за відсутності постійного візуального зворотного зв'язку) в потрібну точку простору з необхідною орієнтацією.

Поперемінне виконання аналогічних описаних вище етапів цілевказівки від СТЗ, визначення наступного бажаного стану робочого органу та вирішення зворотної кінематичної задачі для впорядкованої послідовності «опорних» точок дає можливість здійснити адаптивне формування просторової траєкторії рухів робота (у тому числі з обходом перешкод). Якщо вибрати досить малий інтервал між сусідніми точками, то з використанням відповідних методів інтерполяції можна в адаптивному режимі забезпечити контурне управління роботом, що дозволяє переміщати його робочий орган за необхідним просторово-часовим законом (наприклад, вести зварювання із заданою швидкістю на підставі відеоінформації про фактичне розташування). Очевидно, що такий спосіб є альтернативою підходу з безпосередньою візуальною корекцією траєкторії в безперервному контурі візуального зворотного зв'язку без проміжного розрахунку програмних рухів.

Способи використання відеоінформації в системах управління мобільних роботів в принципі можна звести наступних варіантів:

- зміна послідовності виконання дій, кожна з яких повністю визначена

своєю жорсткою програмою, на підставі сигналів від СТЗ відповідно до заздалегідь зазначених умов;

- адаптивна корекція заданої траєкторії переміщення мобільного робота на підставі відеоінформації, що характеризує його відхилення від необхідного шляху;

- цільовказівка (визначення за допомогою СТЗ відносного положення робота та мети) для адаптивного наведення робота;

- адаптивне формування маршруту (прокладання траси) мобільного робота в реальному часі за даними поточного огляду;

- управління виконавчими механізмами апаратів (з тими самими варіантами використання даних від СТЗ).

На закінчення слід зазначити, що, використання відеоінформації в системах управління роботами дозволяє забезпечити їм широке коло додаткових функціональних можливостей.

1.3 Висновки до розділу

У першому розділі було проведено аналіз основних функцій та сфер застосування технічного зору. Також було обґрунтовано актуальність теми дослідження, мету і задачі роботи, а також визначено практичну значущість. Поставлені завдання є актуальними у контексті розробки і вдосконалення автоматизованих систем сортування, особливо в умовах їх впровадження у промисловості.

2 ІНТЕГРАЦІЯ ТЕХНІЧНОГО ЗОРУ У СИСТЕМУ УПРАВЛІННЯ ПРОМИСЛОВИМ РОБОТОМ

2.1 Розробка математичної моделі опису відео з камери

Відеопотік з камери розглядається як дискретизований у часі просторово-часовий процес інтенсивностей, що формується оптикою, сенсором та трактом стиснення, і зазнає впливу шумів, спотворень лінзи, розмиття руху та затримок передачі. Для подальшого керування цей потік має бути приведений до узгоджених систем координат через калібрування й проєкції, із врахуванням параметрів внутрішньої та зовнішньої геометрії камери. Ключовими є моделювання часової дискретизації, компенсування латентності, а також статистичний опис невизначеності вимірювань. Така постановка дозволяє побудувати узагальнену математичну схему перетворень від “сирого” кадру до метрично інтерпретованих ознак. У підсумку це створює підґрунтя для розроблення математичних моделей опису відопотоку, включно з моделями проєкції, шуму, згладжування, відстеження та компенсації затримок.

Формалізуємо дані отримані з робота, як дискретизацію безперервного відеопотоку камери з урахуванням пропуску кадрів (frame skipping), що напряду впливає на затримку, частоту оновлення оцінок та стабільність керування. Нехай безперервний потік інтенсивностей $I(x, y, t)$ дискретизується в часі з періодом $\Delta t = 1/FPS$:

$$X_k = I(\cdot, \cdot, t_k), \quad t_k = k\Delta t, \quad k \in \mathbb{N}, \quad (2.1)$$

де $I(\cdot, \cdot, t_k)$ – безперервна функція інтенсивності зображення у момент часу t , яка описує яскравість кожної точки просторових координат (пікселів) у потоці

відео, тобто фактично це тривимірне поле $I(x, y, t)$, де x, y – координати пікселя, а t – час;

X_k – дискретизоване зображення (кадр) відеопотоку, отримане у момент часу t_k , що є матрицею інтенсивностей або кольорових значень, що становить миттєвий стан відеопотоку;

t_k – дискретні часові відмітки, які визначають моменти зйомки кадрів, при умові що кожен наступний кадр формується через інтервал Δt ;

Δt – період дискретизації або часовий інтервал між двома послідовними кадрами, що обернено пропорційний частоті кадрів $\Delta t = 1/FPS$;

$k \in \mathbb{N}$ – натуральний індекс кадру, який нумерує послідовність дискретних моментів часу, тобто $k = 0, 1, 2, \dots$

Вираз (2.1) дає можливість описати перехід від безперервного потоку світлових інтенсивностей $I(x, y, t)$ до послідовності дискретних кадрів X_k , що є основою для математичного моделювання цифрового відеопотоку камери.

Нехай початковий кадр $X_k \in \mathbb{R}^{H \times W \times 3}$. Припустимо, що початковий кадр ресайзиться афінним перетворенням до $\tilde{X}_k \in \mathbb{R}^{h \times w \times 3}$, $(h, w) = (320, 320)$:

$$\tilde{X}_k(u, v) = X_k(\alpha u, \beta v), \alpha = H/h, \beta = W/w, \quad (2.2)$$

де $X_k(x, y)$ – оригінальний кадр відеопотоку з камери FPV, визначений у координатах вихідного зображення розміром $H \times W$;

$\tilde{X}_k(u, v)$ – масштабована версія кадру X_k , яка приведена до розміру $h \times w$, що відповідає вхідному формату нейронної мережі;

$\alpha = H/h$ – коефіцієнт масштабування по вертикалі, який показує, скільки пікселів оригінального зображення відповідає одному пікселю у зменшеному кадрі;

$\beta = W/w$ – коефіцієнт масштабування по горизонталі, що визначає співвідношення між шириною оригінального кадру та новим розміром;

(u, v) – координати пікселя у масштабованому зображенні, а $(\alpha u, \beta v)$ – відповідна точка в оригінальному кадрі.

Модель (2.2) знижує обчислювальне навантаження, зберігає пропорції об'єктів та забезпечує коректне позиціонування детектованих областей при подальшій реконструкції координат у вихідному масштабі. Далі відбувається квантування:

$$\mathbf{X}_k = \text{round}(\tilde{\mathbf{X}}_k) \in \{0, \dots, 255\}^{h \times w \times 3}, \quad (2.3)$$

де $\tilde{\mathbf{X}}_k$ – масштабований кадр відеопотоку (з моделі 2.2), значення якого можуть бути дійсними числами в діапазоні $[0, 255]$ або $[0, 1]$, залежно від методу інтерполяції. Містить тривимірну структуру піксельних значень розміром $h \times w \times 3$, де 3 – кількість кольорних каналів (RGB);

$\mathbf{X}_k$ – результат операції округлення (дискретизації) кожного елемента $\tilde{\mathbf{X}}_k$ до найближчого цілого значення в межах множини $\{0, 1, 2, 3, \dots, 255\}$ тобто це цифрове зображення у форматі, придатному для обчислень в фреймворках машинного зору;

$\text{round}(\cdot)$ – функція квантування (округлення), яка переводить неперервні або дробові інтенсивності в цілочисельний діапазон рівнів яскравості, та забезпечує узгодженість представлення з реальними сенсорними даними камери, де пікселі мають дискретну глибину кольору (8 біт);

$\{0, \dots, 255\}$ – множина можливих рівнів інтенсивності для кожного каналу кольору, що відповідає 8-бітному кодуванню RGB-компонентів; $h, w, 3$ – просторові розміри кадру після масштабування, відповідно h – висота, w – ширина, а 3 – це три канали кольору (Red, Green, Blue).

Модель (2.3) забезпечує уніфікацію числового формату піксельних даних, зменшує похибки, пов'язані з обчислювальною інтерполяцією, і гарантує сумісність з апаратними моделями, де кожен кадр повинен мати чітко визначену дискретну глибину кольору. Це дозволяє отримувати дані в оптимальному вигляді для точного розпізнавання об'єктів у режимі реального часу.

Для того щоб зв'язати піксельні координати з напрямками променів/просторовими координатами не обхідно описати Пінохольну модель камери, що необхідно для оцінки реальних відстаней та навігаційних рішень дрона. Класична Пінохольна модель камери (геометрія проєкції) має наступний вигляд:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K[R|t] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.4)$$

де K – внутрішні параметри;

$R|t$ – положення камери відносно світу;

X, Y, Z – 3D-точка;

u, v – піксель.

Тепер необхідно розробити стохастичну модель детектора об'єктів використовуючи MobileNet, це дає можливість зрозуміти що повертає модель розпізнавання: набори боксів, класів і ймовірностей – як випадкові величини.

Тобто для кожного кандидата i на виході маємо:

- клас $C_i \in \{1, \dots, K\}$;
- нормалізований бокс $b_i = [y_1, x_1, y_2, x_2] \in [0,1]^4$;
- оцінку достовірності $s_i = P(C_i|X_k) \in [0,1]$.

Що можна описати у векторній формі:

$$D_k = \{(C_i, b_i, s_i)\}_{i=1}^{N_k}, \quad (2.5)$$

де D_k – множина детекцій (результатів розпізнавання) у кадрі k що описує всі об'єкти, виявлені моделлю нейронної мережі у цей момент часу;

N_k – кількість об'єктів, які були виявлені на кадрі X_k , визначається як кількість трійок C_i, b_i, s_i ;

C_i – ідентифікатор (клас) i -го об'єкта, який відповідає певній категорії. У межах системи керування роботом цей параметр дозволяє розрізняти типи об'єктів для вибору відповідної стратегії реакції або траєкторії руху;

$b_i = [y_1, x_1, y_2, x_2]$ – вектор координат граничного прямокутника (bounding box) для i -го об'єкта, який визначає його положення в межах кадру;

s_i – оцінка впевненості (score), тобто ймовірність того, що виявлений об'єкт дійсно належить до класу C_i . Значення s_i визначається як результат застосування сигмоїдної або софтмакс-функції на виході нейронної мережі. У задачі комп'ютерного зору FPV-дрону цей параметр використовується для фільтрації хибних детекцій і визначення пріоритету об'єктів для подальшої обробки.

Модель (2.5) формалізується результат роботи нейронної мережі-детектора об'єктів у момент часу t_k , який містить набір знайдених об'єктів у кадрі X_k та описує структуру даних, що поєднує інформацію про класифікацію, локалізацію та впевненість детектора для кожного розпізнаного елемента сцени.

Введемо порогову фільтрацію для відкидання малої ймовірності детекції для зменшення помилкових спрацьовувань у реальному часі. Представимо це наступним виразом:

$$D_k^{(\tau)} = \{(C_i, b_i, s_i) \in D_k | s_i \geq \tau\}, \quad (2.6)$$

де τ – поріг упевненості, при умові більший $\tau \rightarrow$ менше FP, більше FN.

Проведемо масштабування боксів назад у координати вихідного кадру, тобто перевід нормалізованих координат $[0,1]$ до пікселів вихідного кадру для коректного відмалювання та подальших розрахунків:

$$\begin{aligned} y_1^{px} &= [y_1 \cdot H] & y_2^{px} &= [y_2 \cdot H] \\ x_1^{px} &= [x_1 \cdot W] & x_2^{px} &= [x_2 \cdot W] \end{aligned} \quad (2.7)$$

де H, W – розмір оригінального кадру.

Роздивимося питання накладання боксів Intersection over Union (IoU) та NMS, що дає злиття дубльованих детекцій одного об'єкта через метрику перетину-над-об'єднання та не-максимальне придушення. Площа перетину/об'єднання двох боксів b_i та b_j можна визначити з наступного виразу:

$$IoU(b_i, b_j) = \frac{area(b_i \cap b_j)}{area(b_i \cup b_j)}, \quad (2.8)$$

де $b_i = [y_1, x_1, y_2, x_2]$ – координати першого граничного прямокутника (bounding box), який може представляти або прогноз моделі, або поточне спостереження об'єкта. Його площа визначається $area(b_i) = (x_2 - x_1) \cdot (y_2 - y_1)$;

$b_j = [y_1', x_1', y_2', x_2']$ – координати другого прямокутника, який використовується для порівняння, він може бути еталонним (ground truth) або попередньою оцінкою положення того самого об'єкта;

$area(b_i \cap b_j)$ – площа спільної (перекритої) області між двома прямокутниками, та відображає ступінь фактичного збігу позицій об'єктів. Чим більший цей показник, тим точніше модель виявила об'єкт;

$area(b_i \cup b_j)$ – площа об'єднаної області двох прямокутників, тобто сумарна площа обох боксів мінус зона їхнього перетину, та показує повну область, охоплену обома оцінками;

$IoU(b_i, b_j) \in [0,1]$ – відношення площі перетину до площі об'єднання, тобто це числовий показник ступеня відповідності двох локалізацій. Значення 1 означає повний збіг, а 0 – повну відсутність перекриття.

Для вдосконалення розробляемого метода розпізнавання та ідентифікації введемо базовий трекінг через фільтр Калмана, що дасть можливість стабілізувати траєкторії боксів між кадрами, компенсувати пропуски та шум; критично для системи керування роботом, де керування залежить від плавної оцінки. Модель опишемо наступним чином:

– стан об’єкта (у пікселях) як центр і швидкість:

$$\mathbf{x}_k = [c_x, c_y, c_x', c_y', w, h]^T ; \quad (2.9)$$

– динаміка з рівномірною швидкістю:

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_{k-1}, \quad \mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k, \quad (2.10)$$

де $\mathbf{x}_k$ – вектор поточного стану об’єкта у момент часу t_k , у контексті даного дослідження, цей вектор може містити координати центру об’єкта в кадрі, його швидкість, розміри граничного прямокутника або навіть орієнтацію;

$\mathbf{x}_{k-1}$ – вектор стану на попередньому кроці часу t_{k-1} , з якого прогнозується нове положення об’єкта;

$\mathbf{F}$ – матриця переходу стану (state transition matrix), яка визначає динаміку зміни стану системи між моментами часу;

$\mathbf{w}_{k-1}$ – вектор процесного шуму, який моделює невизначеність або збурення динаміки руху;

$\mathbf{z}_k$ – вектор вимірювання (спостереження), отриманий із сенсорних або зорових даних (наприклад, координати центра об’єкта та його розмір з детектора нейронної мережі).

$\mathbf{H}$ – матриця спостереження (observation matrix), яка встановлює зв’язок між істинним станом об’єкта $\mathbf{x}_k$ та вимірюваними параметрами $\mathbf{z}_k$;

$\mathbf{v}_k$ – вектор шуму вимірювання, який описує похибки спостереження, зумовлені неточністю детекції, розмиттям або шумом сенсорів.

Для того щоб узгодити нові вимірювання з існуючими треками для коректної ідентичності об’єктів. Введемо асоціацію “детекція $\leftrightarrow$ трек”

використовуючи метричну функцію Intersection-over-Union cost (IoU-cost). IoUcost – це метрична функція вартості (cost function), яка використовується для порівняння двох граничних прямокутників (bounding boxes) у задачах асоціації об'єктів – тобто при зіставленні детекцій між кадрами у трекінгу або при об'єднанні результатів з різних моделей. Формуємо матрицю вартостей $\mathbf{C}$ між прогнозами треків $b^{(j)}$ та новими боксами $b^{(i)}$:

$$C_{ij} = 1 - IoU(b^{(j)}, b^{(i)}), \quad (2.11)$$

де C_{ij} – елемент матриці вартостей (cost matrix), який визначає “ціну” призначення i -го нового детектованого об'єкта $b^{(i)}$ до j -го прогнозованого треку $b^{(j)}$. Тобто C_{ij} належить діапазону: $C_{ij} \approx 0$ означає високу подібність між боксами (ймовірно той самий об'єкт) та відповідно $C_{ij} \approx 1$ означає низьку подібність або різні об'єкти;

$b^{(j)}$ – прогнозований bounding box j -го об'єкта, отриманий із моделі динаміки (наприклад, фільтра Калмана), тобто цей бокс є очікуваним положенням об'єкта у поточному кадрі на основі попередніх спостережень;

$b^{(i)}$ – виміряний bounding box, отриманий від детектора об'єктів (нейронної мережі) у поточному кадрі, це фактична спостережувана позиція можливого об'єкта.

$IoU(b^{(j)}, b^{(i)})$ – метрика перекриття (Intersection over Union), що показує ступінь збігу між прогнозованим і детектованим положенням об'єкта.

Вираз (2.11) математично задає інверсну міру подібності між прогнозованим і спостережуваним положенням об'єкта. Чим більший збіг (висока IoU), тим менша вартість C_{ij} , а отже, тим більша ймовірність, що це один і той самий об'єкт. Такий підхід лежить в основі алгоритмів трекінгу об'єктів, необхідних для стабільного розпізнавання в реальному часі.

Проведемо оцінку затримки інференсу та її компенсація, це час виконання Δt_{inf} прямо впливає на “застарівання” вимірювань, в наслідок чого потрібна модель для компенсації при керуванні. Представимо виміряну

затримку як: $\Delta t_{inf} = t_{end} - t_{start}$. Модель екстраполяція центру боксу на момент відображення опишемо наступним чином:

$$\begin{bmatrix} \tilde{c}_x \\ \tilde{c}_y \end{bmatrix} = \begin{bmatrix} c_x \\ c_y \end{bmatrix} + \Delta t_{inf} \begin{bmatrix} \dot{c}_x \\ \dot{c}_y \end{bmatrix}, \quad (2.12)$$

де $\tilde{c}_x$, $\tilde{c}_y$ – кориговані (екстрапольовані) координати центра об’єкта у кадрі після врахування затримки обробки. Це прогнозоване актуальне положення об’єкта у момент, коли робот отримує готовий результат розпізнавання;

c_x , c_y – поточні координати центра об’єкта, отримані від детектора у момент часу, коли зображення було захоплено камерою. Ці координати відображають положення об’єкта у “минулому” кадрі відносно центру зображення або системи координат відеопотоку;

$\dot{c}_x$, $\dot{c}_y$ – швидкості зміни координат центра об’єкта (похідні від часу), що показують, наскільки швидко об’єкт переміщується у горизонтальному (x) і вертикальному (y) напрямках у площині кадру. Ці значення отримуються або з фільтра Калмана, або чисельним диференціюванням траєкторії;

Δt_{inf} – часова затримка інференсу (inference time delay), тобто час між моментом отримання кадру з камери та моментом, коли модель нейронної мережі повертає результат розпізнавання. Це ключовий параметр, що визначає, наскільки “застарілими” є дані при прийнятті рішення.

Вираз (2.12) є лінійною моделлю прогнозування положення об’єкта з урахуванням латентності системи, яка підвищує точність наведення камери, стабілізації зображення та реакції дрона в режимі реального часу, особливо при високих швидкостях польоту або великому обчислювальному навантаженні нейронної мережі.

Також потрібно врахувати оцінку розташування об’єктів на земній (або іншій) площині, щоб оцінювати відстань/напрямок в реальних одиницях.

Допустимо що об'єкт лежить на площині Π , то існує гомографія H така що:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = H \begin{bmatrix} X_{\Pi} \\ Y_{\Pi} \\ 1 \end{bmatrix}, \quad (2.13)$$

де $\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$ – вектор однорідних координат пікселя у зображенні, u – горизонтальна координата (вздовж осі X кадру), а v – вертикальна координата (вздовж осі Y кадру). Ці координати визначають положення точки на площині зображення, отриманого камерою;

λ – масштабний (нормувальний) коефіцієнт, який переводить однорідні координати в афінні (реальні) шляхом нормалізації;

H – матриця гомографії (3×3), яка описує проєктивне перетворення між двома площинами – площиною зображення (image plane) та площиною сцени (ground plane або робоча поверхня);

$\begin{bmatrix} X_{\Pi} \\ Y_{\Pi} \\ 1 \end{bmatrix}$ – однорідні координати точки на площині Π (наприклад, земної 1 поверхні або стола, на якому розташовані об'єкти). Тут X_{Π} , Y_{Π} – метричні координати точки у площині світу, відносно системи координат сцени.

Враховуючи що данні ми отримуємо з камери потрібно врахувати зсув об'єкта в кадрі на керуючі команди управління. Похибка наведення в нормалізованих координатах:

$$e_u = \frac{u_c - u_0}{W}, e_v = \frac{v_c - v_0}{H}, \quad (2.14)$$

де (u_0, v_0) – центр кадру.

Для згладжування похибок які виникають при виконанні команд yaw/pitch пропонується для нормалізацій кадрів використовувати PID-керування:

$$\begin{aligned}w_{yaw} &= K_p e_u + K_i \int e_u dt + K_d \dot{e}_u, \\w_{pitch} &= \dot{K}_p e_v + \dot{K}_i \int e_v dt + \dot{K}_d \dot{e}_v,\end{aligned}\tag{2.15}$$

де K_p, K_i, K_d – коефіцієнти регулятора. Дає замикання контуру “бачу об’єкт → наводжу камеру”.

Перевага розроблених математичних моделей і виразів полягає в тому, що вони формують цілісну багаторівневу систему опису відеопотоку камери, здатну враховувати реальні фізичні, часові та стохастичні процеси, що впливають на якість і точність ідентифікації об’єктів. Розроблені моделі дозволяють відтворювати перехід від безперервного світлового потоку до дискретних кадрів із контролем параметрів дискретизації, масштабування, квантування та геометричних перетворень, що забезпечує адаптацію даних до нейронної мережі без втрати метричної узгодженості. Використання пінохольної моделі та гомографічного перетворення створює основу для просторової прив’язки зображень до реальних координат сцени, що підвищує точність оцінки положення об’єктів. Стохастичні рівняння фільтра Калмана і функції IoU-cost гарантують стабільність трекінгу навіть за наявності шумів і часткових перекриттів, а модель компенсації затримки інференсу забезпечує своєчасність реакції робота. У сукупності ці вирази створюють формалізовану основу для побудови інтелектуальної системи технічного зору робота, де ідентифікація та розпізнавання об’єктів відбуваються з високою точністю, стійкістю та узгодженням з динамікою реального руху.

2.2 Постановка задачі розробки системи технічного зору для промислового робототехнічного комплексу

На сьогоднішній день велику популярність набувають СТЗ як пристрої визначення положення координат корисного вантажу при роботі з маніпуляційними роботами. Прості представники подібних систем

дозволяють визначати декартові координати x і y , а також кут орієнтації R корисних вантажів, що знаходяться в одній площині робочого простору маніпуляційного робота, причому параметри цієї площини повинні бути заздалегідь відомі. Більш складні рішення дозволяють визначити три координати x , y і z у робочому просторі. Комплексні рішення дозволяють визначити всі шість координат x , y , z , φ , θ , ψ , але такі рішення вимагають особливих умов функціонування, тому їх сфера застосування сильно обмежена; крім того, у багатьох технологічних процесах відсутня необхідність визначення повного набору координат корисних вантажів, тому вартість і функціонал, що використовується, для таких систем будуть невиправданими.

СТЗ дозволяють без значних витрат переконфігурувати робоче місце, оскільки система визначення положення в значній мірі налаштовується на програмному рівні. Як правило, однієї камери, що входить до складу СТЗ, достатньо для отримання даних про робочий простір великої площі [9, 10]. У разі зміни технологічного процесу цю камеру можна використовувати для визначення положення корисних вантажів в нових умовах без необхідності втручання в її конструкцію або електричну схему.

Функціональні задачі СТЗ, характерні для роботехнічних комплексів, умовно поділяються за рівнем їх відносної складності. До простих задач можна віднести: виявлення наявності якогось об'єкта, вимірювання відстані до нього, обчислення його лінійних та кутових переміщень, швидкості; вимірювання геометричних параметрів об'єкта (лінійні та кутові розміри, площа), визначення фізичних характеристик випромінювання від об'єкта, підрахунок числа об'єктів у кадрі [11, 12].

Більш складні задачі виконує система, яка надає маніпулятору інформацію, необхідну для захвату невпорядкованих об'єктів. До цих задач відносять: огляд робочого простору для пошуку об'єкта, що цікавить, який

може бути одиночним або одним з декількох або його місцезнаходження може бути ізолювано, перекриватися іншими об'єктами. При цьому об'єкти, що спостерігаються, можуть відрізнятися не тільки формою і розміром, але і кольором, текстурою і т.д., знаходиться в русі або в спокої.

У цій роботі буде розроблено СТЗ у систему управління сортувальним роботом (рисунк 2.1). Сортування здійснюється за ознакою – кольором виробу.

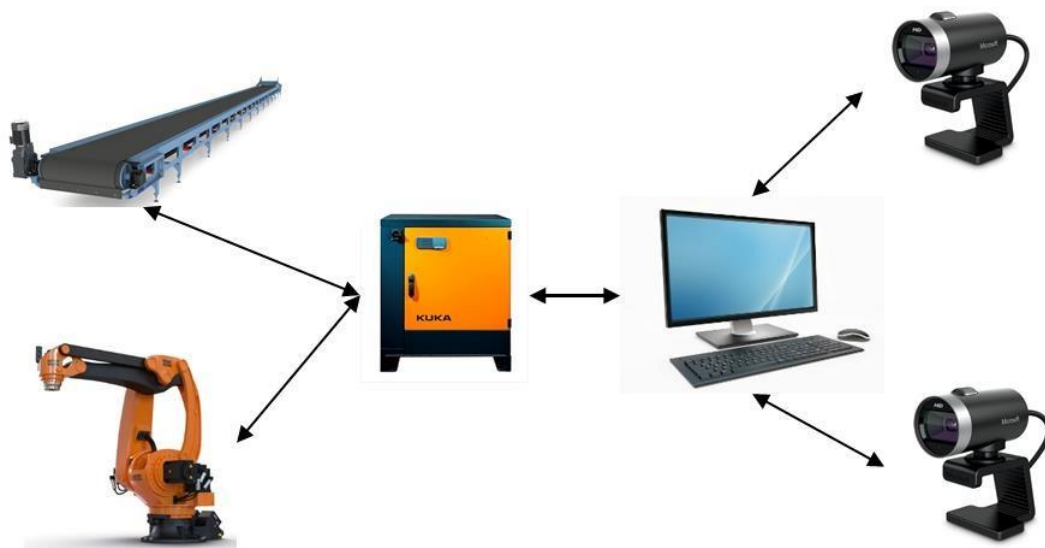


Рисунок 2.1 – Функціональна схема системи управління сортувальним роботом

Основними задачами технічного зору у цій системі є:

- визначення положення виробу на стрічці конвеєра. Необхідно визначення лише двох координат, тому що конвеєр знаходиться на постійній висоті;
- визначення орієнтації виробу на стрічці конвеєра. На який кут воно повернуто, щоб маніпулятор міг вирівняти її під час здійснення сортування;
- визначення кольору виробу. Для наочності їх буде три: червоний, синій та зелений.

Як очевидно з представлених вище цілей технічного зору, у цій роботі функціональні завдання технічного зору швидше можна віднести до простих.

Постановка завдання сортування майже сходиться з постановкою проблеми розпізнавання образів за умов попередньої невизначеності, а місцезнаходження та орієнтації предметів можна розцінювати як оцінки зображень предметів, фіксованих на тлі завад. Створення результативних алгоритмів та програм, які забезпечать автоматичний аналіз та обробку зображень для оцінки параметрів зображень та розпізнавання образів об'єктів, вважається значним напрямком покращення СТЗ у промислових роботехнічних комплексах.

До методів обробки зображень у СТЗ пред'являються такі вимоги [13,14]:

- забезпечення високої ефективності за заданими критеріями;
- мінімальні витрати часу для обробки інформації в режимі реального часу;
- мінімальні витрати обчислювальних ресурсів.

Тому для вирішення поставлених задач оцінки параметрів та розпізнавання будуть використані алгоритми, що базуються на аналізі реєстрованих контурів зображень. Щоб спростити метод опису зображень предметів, що фіксуються в СТЗ, було прийнято таке припущення: предмети мають порівняно елементарну форму, при якій контур заданого об'єкту легко зможе охарактеризувати цю форму.

2.3 Вибір інструментів програмування

Вибір мови програмування та середовища розробки є ключовим етапом при реалізації проекту. Серед існуючих високорівневих мов програмування,

таких як C++, Java, MATLAB та Python, кожна має свої переваги та обмеження. C++ забезпечує високу швидкість та можливість роботи з низькорівневими структурами, однак має складний синтаксис і значні витрати часу на розробку. Java вирізняється переносимістю та багатопотоковістю, але поступається у швидкості виконання та потребує більше ресурсів. MATLAB традиційно використовується для наукових розрахунків і моделювання, однак є закритою комерційною платформою, що обмежує масштабованість і вимагає високих ліцензійних витрат.

Python – високорівнева мова програмування загального призначення, орієнтована на підвищення продуктивності розробника та полегшення читання коду [18]. Синтаксис ядра Python мінімалістичний, в той же час, стандартна бібліотека включає великий обсяг корисних функцій та забезпечує високу гнучкість, що робить його оптимальним вибором для науково-технічних досліджень і чисельного моделювання. Для задач моделювання мультиагентних систем і керування роботами Python надає широкий набір інструментів: NumPy, SciPy для лінійної алгебри та оптимізації; Matplotlib, Plotly для візуалізації результатів; SymPy для аналітичних обчислень; а також бібліотеки штучного інтелекту (TensorFlow, PyTorch) для реалізації адаптивних алгоритмів. Додатковою перевагою Python є інтеграція з ROS (Robot Operating System) через пакети rospy, що забезпечує можливість створення програмного забезпечення для робототехнічних систем у реальному часі.

В таблиці 2.1 приведено порівняльний аналіз мов високого рівня програмування для розробки для написання програмного коду роботи системи сортування.

Таблиця 2.1 – Порівняльний аналіз мов високого рівня програмування

Критерій	Python	C++	Java
Простота синтаксису	Дуже проста, інтуїтивна, підходить для швидкої розробки	Складний синтаксис, вимагає більше часу на реалізацію	Простий, але більш громіздкий у порівнянні з Python
Швидкодія	Нижча, ніж у C++, але оптимізується бібліотеками (NumPy, Numba)	Висока продуктивність, особливо для обчислювально складних задач	Середня, залежить від JVM
Бібліотеки для робототехніки	Дуже велика кількість (ROS, NumPy, SciPy, TensorFlow, PyTorch)	Менша кількість, складніша інтеграція	Обмежені можливості у порівнянні з Python
Зручність для мультиагентних систем	Висока: підтримка симуляційних пакетів (Mesa, SimPy), інтеграція з AI	Складно реалізувати без спеціальних фреймворків	Потребує додаткових бібліотек, складніше ніж у Python
Інтеграція з ROS	Легка (rospy, rosbbridge)	Є підтримка (roscpp), але складніше налаштувати	Обмежена підтримка

Python виграє за простотою розробки, наявністю різноманітних бібліотек, AI та інтеграції з ROS. C++ підходить для високопродуктивних реальних систем, але вимагає більше часу та складної оптимізації. Java добре підходить для корпоративних рішень, але менш ефективна для швидкої розробки прототипів і наукового моделювання.

Таким чином, обґрунтуванням вибору даної мови програмування є її простота, низький поріг входження, велика кількість навчального матеріалу в інтернеті та можливість працювати з бібліотекою OpenCV

Одним з найпоширеніших програмних забезпечень для реалізації комп'ютерного зору є OpenCV.

OpenCV – бібліотека алгоритмів комп'ютерного зору, обробки зображень та чисельних алгоритмів загального призначення з відкритим кодом [19]. Реалізована на C/C++, також існують версії для Python, Java, Ruby, Matlab, Lua та інших мов. Вона розробляється з 1998 р., спочатку в компанії Intel, потім в Itseez за активної участі спільноти.

Бібліотека поширюється за відкритою ліцензією, що означає, що її можна вільно та безкоштовно використовувати як у відкритих проектах з відкритим додому, так і у закритих, комерційних проектах. Бібліотеку не обов'язково копіювати повністю у свій проект, можна використовувати шматки коду. Єдина вимога ліцензії – наявність у супровідних матеріалах копії ліцензії OpenCV.

Через ліберальну ліцензію бібліотека використовується багатьма компаніями, організаціями, університетами, наприклад, NVidia, Willow Garage, Intel, Google, Stanford University. Компанії NVidia та WillowGarage частково спонсорують її розробку.

Багатовимірна архітектура проекту представлена рисунку 2.2. Бібліотека складається із 16 модулів, реалізовано близько 1000 алгоритмів. Підтримуються основні операційні системи: MS Windows, Linux, Mac, Android, iOS. Є можливість використання сторонніх бібліотек, наприклад, для роботи з пристроєм Kinect (OpenNI), розробки паралельних програм (TBB) та ін.

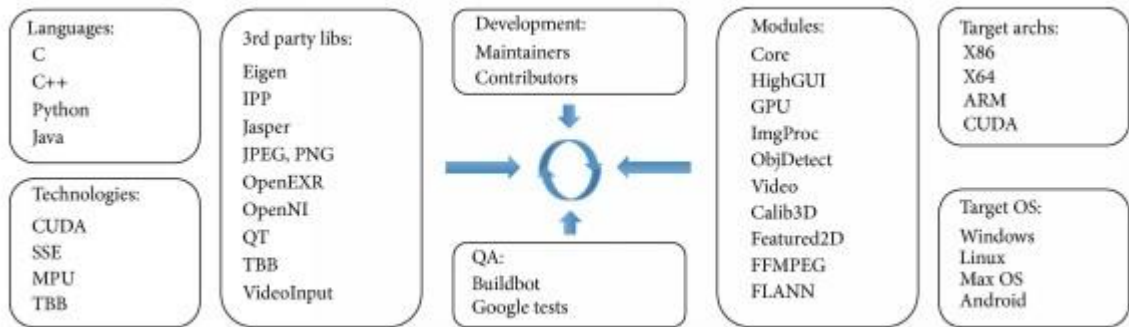


Рисунок 2.2 – Архітектура OpenCV

Основні модулі бібліотеки можна віднести до 4 груп (розділів):

- модулі Core, HighGUI, що реалізують базову функціональність (базові структури, математичні функції, генератори випадкових чисел, лінійна алгебра, швидке перетворення Фур'є, введення/виведення зображень і відео, введення/виведення у форматах XML, YAML та ін.);
- модулі ImgProc, Features2D для обробки зображень (фільтрація, геометричні перетворення, перетворення колірних просторів, сегментація, виявлення особливих точок та ребер, контурний аналіз та ін.);
- модулі Video, ObjDetect, Calib3D (калібрування камери, аналіз руху та відстеження об'єктів, обчислення положення в просторі, побудова карти глибини, детектування об'єктів, оптичний потік);
- модуль ML, що реалізує алгоритми машинного навчання (метод найближчих сусідів, наївний байєсівський класифікатор, дерева рішень, бустинг, градієнтний бустинг дерев рішень, випадковий ліс, машина опорних векторів, нейронні мережі та ін).

Все починається із захвату зображень (модуль HighGUI). Читається зображення з файлу або відео з мережевої камери через мережевий протокол.

Далі здійснюється попередня обробка (модуль ImgProc), така, як усунення шуму, вирівнювання яскравості, контрасту, виділення та видалення

відблисків, тіней. Наприклад, той самий об'єкт при різному освітленні виглядає по-різному. У яскравому світлі червона машина, рух якої необхідно відстежувати, буде яскраво-жовтогарячою. У похмуру погоду та сама машина виглядатиме червоно-рожевою. У цьому випадку на зображенні необхідно вирівняти кольори. Передобробка може бути як простою, так і може включати в собі цілу складну технологію.

Наступний етап – виділення особливостей (модулі `ImgProc`, `Features2D`). Наприклад, у задачі стеження за об'єктом це може бути пошук спеціальних точок на об'єкті, за якими легко спостерігати; для задач детектування (тобто виявлення на зображенні) – обчислення опису кожного пікселя.

Далі відбувається детектування об'єктів, що цікавлять нас, виділення значущих частин, сегментація зображення (модулі `ImgProc`, `ObjDetect`). Якщо, наприклад, камера нерухома, а рухоме зображення, можна використовувати алгоритми віднімання фону.

Після цього вирішується основна задача, така, як обчислення розташування об'єкта в 3D, реконструкція 3D структури, аналіз структури, реєстрація тощо (модулі `Calib3D`, `Contrib`, `Video`, `Stitching`, `Videostab`, `ML`). Наприклад, задача склеювання панорам зображень – це зіставлення частин різних кадрів, визначення необхідного перетворення. У задачі відеоспостереження це відновлення траєкторій об'єктів тощо.

Наприкінці відбувається розпізнавання та прийняття конкретних рішень (модуль `ML`). Наприклад, у системі відеоспостереження: виявлено небажаний об'єкт у кадрі чи ні. У задачі детектування тексту – чи детектовано текст, що саме за текст і т.д.

2.4 Висновки до розділу

У другому розділі проведено аналіз готових рішень від світових виробників промислової робототехніки. Особливу увагу приділено огляду існуючих досліджень і рішень, які використовуються у промислових системах. На основі проведеного аналізу сформульовано задачі розробки системи технічного зору промислового робототехнічного комплексу, обрано обладнання, що використовується, для побудови системи та програмне забезпечення для написання програмного коду роботи системи сортування.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОЕКТУ СИСТЕМИ ТЕХНІЧНОГО ЗОРУ

3.1 Розміщення камер

Камеру можна закріпити у різний спосіб. Найпростіший – це закріпити нерухомо на спеціальну основу поруч із можливим рухом корисних об'єктів у робочій області. На рисунку 3.1 показано приклад такого кріплення.



Рисунок 3.1 – Приклад кріплення камери на нерухому основу

Якщо камера знаходиться в робочій області руху маніпулятора, то в програмі управління роботом необхідно додатково накласти обмеження, щоб не допустити зіткнення рухомих елементів з датчиком. Крім того, бажано забезпечити зовнішній механічний захист. Не можна забувати, що під час роботи маніпуляційний робот своїми частинами закривати різні області робочої зони, тому для камери або необхідно забезпечити максимально можливий огляд поверхні, або програмно виключати з обробки кадри, коли видимість недостатня. При жорсткому кріпленні камери координати її установки повинні бути відомі і прив'язані до системи координат основи

маніпулятора для коректного визначення положення корисних об'єктів щодо робота та успішного виконання алгоритму управління.

Іншим варіантом встановлення СТЗ є кріплення на схват маніпулятора (рисунок 3.2). Конструкція багатьох камер передбачає встановлення на спеціальний фланець, який дозволяє схвату повноцінно функціонувати з камерою, що працює.



Рисунок 3.2 – Приклад кріплення камери на схват маніпулятора

В даному випадку, елементи маніпуляційного робота не заважатимуть під час огляду СТЗ. Для визначення положення корисних об'єктів у робочому просторі необхідно зв'язати координати, отримані від камери, з положенням схвату у просторі щодо основи маніпулятора. Для цього потрібно для кожної ітерації розраховувати пряме завдання кінематики. Крім того, якщо стоїть задача по камері визначити швидкість руху об'єктів, необхідно враховувати і швидкість руху схвата в тому числі. Для запобігання удару камери об частину маніпуляційного робота необхідно для останнього програмно накласти обмеження на можливі рухи.

Також важливою властивістю буде відсутність або можливість вимкнення системи автоматичного фокусування. Через дану систему утруднено використання багатьох сучасних камер, так як вони розраховані на зйомку нерухомого об'єкта, а у випадку з маніпуляційними роботами, що постійно переміщаються, і корисним вантажем буде часто втрачатися фокусування зображення. Об'єкти закріплюються над робочою поверхнею та

охоплює кутом огляду весь робочий простір. Точка закріплення камери має бути відома.

Виходячи з положення конвеєра, маніпулятора і відсутності відповідної камери для розташування її на маніпуляторі, самим практичним і зручним способом в даному випадку буде розміщення двох камер статично, збоку і в кінці конвеєра. Камера, яка знаходиться наприкінці конвеєра, надалі нумеруватиметься як №1, а та що збоку №2 (приклад наведено на рис. 3.3).



Рисунок 3.3 – Приклад розміщення камер на конвеєрі

Для більш швидкого виконання програми та відсікання непотрібної інформації кадри кожної камери обрізані до певної апріорної області, в якій відбувається пошук об'єкта (рис. 3.4 – рис. 3.5).



Рисунок 3.4 – Обрізаний кадр із камери №1

Дві камери потрібні для отримання тривимірних координат об'єкта. Одна камера отримує інформацію про об'єкт із зображення в координатах (X, Z) , друга (Y, Z) . При склеюванні двох отриманих зображень обчислюється тривимірна координата (X, Y, Z) положення центру об'єкта. В якості об'єктів були використані три види кубиків: червоний, зелений та синій.



Рисунок 3.5 – Обрізаний кадр із камери №2

3.2 Пошук об'єкта за кольором

Найпоширеніший спосіб виділити об'єкт – це колір.

Колір – це властивість тіл відбивати або випромінювати видиме випромінювання певного спектрального складу та інтенсивності [25-27]. Скрізь і всюди нас оточують колірні індикатори. Світлофори, білі та жовті лінії дорожньої розмітки, корпоративні кольори продуктів, дорожні покажчики та різні індикатори.

Однією з важливих проблем пошуку за кольором – це вплив багатьох факторів. Наприклад, освітленість. Не можна також забувати, що видимий колір – це результат взаємодії спектра світла, що випромінюється, і поверхні. Тобто, якщо білий лист освітлюватиме світлом червоної лампочки, то й лист здаватиметься червоним.

Вибір кольору для наступного пошуку об'єкта з таким кольором зводиться до виділення певної цільової області в колірному просторі. Передбачається, що об'єкт, що шукається, повинен володіти кольорами, що відповідають точкам з цієї цільової області. Тут можливі дві ситуації:

- цільова область заздалегідь виділяється у колірному просторі;

– цільова область визначається вказівками користувача, наприклад, описується колірними термінами природної мови людини: "синій", "червоний", "яскраво-жовтий", "темно-зелений" і т.п.

Перша ситуація реалізується у разі пошуку об'єктів заздалегідь відомих кольорів, наприклад, у задачах прямування за дорогою, стеження за жестами людей тощо. Друга ситуація виникає при взаємодії з людиною, коли вона, наприклад, дає команду роботу у вигляді фрази «принеси червоний кухоль». Інший приклад – це пошук за базою зображень із зазначенням кольору об'єкта або особливостей його композиції.

Задача побудови відповідностей між колірними термінами природної мови та областями колірного простору відома давно. Існує безліч методик вирішення цієї задачі.

Одна точка на шкалі H визначає двовимірну область з різними значеннями S і V (насиченість та значення). Але підходять не всі точки цієї області. Наприклад, якщо вибрано значення H для червоного тону, то при $S = 0$ і будь-якому V отримуємо відтінки, що відносяться до градацій сірого, а при малих V і будь-яких S отримуємо дуже темні відтінки, близькі до чорного кольору. Тому для заданого H цільову область на цій двовимірній ділянці визначаємо умовою: $V > V_{\min}$ та $S > S_{\min}$, де $V_{\min}$, $S_{\min}$ – деякі константи.

Якщо обмежуватися лише одним значенням H , цільова область буде двовимірною областю, і будь-які, навіть незначні зміни колірного тону виводять за межі цієї області. Тому в загальному випадку людина вибирає цільовий діапазон на шкалі H , вказуючи два значення: $H_{\min}$ та $H_{\max}$.

При цьому для кожного H з діапазону $[H_{\min}, H_{\max}]$ формується двовимірна область, подібна до вищезазначеної. Таким чином, як цільова область формується тривимірна область у просторі HSV. І ця область однозначно визначається вибором лише двох значень за шкалою H , а саме $H_{\min}$ та $H_{\max}$.

Було вирішено вибирати два значення $H_{\min}$ і $H_{\max}$. Дивлячись на шкалу тонів кольорів (H) моделі HSV (рисунок 3.6), стає зрозумілим, що величина

такого діапазону допустимих відхилень повинна бути змінною в залежності від значення H . Наприклад, діапазон тонів, що сприймаються як зелений колір, ширший за діапазон тонів, що сприймаються, як жовтий колір. Крім того, для різних людей довжини цих діапазонів відрізнятимуться через індивідуальні особливості розуміння колірних термінів.



Рисунок 3.6 – Шкала тонів кольору H моделі HSV

Для визначення даних параметрів було написано програму, що допомагає виділити певний колір у кадрі.

```
import cv2
import numpy as np
if __name__ == '__main__':
    def nothing(*arg): pass

cv2.namedWindow( "result" ) # створюємо головне вікно
cv2.namedWindow( "settings" ) # створюємо вікно налаштувань

cap = cv2.VideoCapture(1)
# створюємо 6 бігунків для налаштування початкового і кінцевого кольору фільтра
cv2.createTrackbar('h1', 'settings', 0, 255, nothing)
cv2.createTrackbar('s1', 'settings', 0, 255, nothing)
cv2.createTrackbar('v1', 'settings', 0, 255, nothing)
cv2.createTrackbar('h2', 'settings', 255, 255, nothing)
cv2.createTrackbar('s2', 'settings', 255, 255, nothing)
cv2.createTrackbar('v2', 'settings', 255, 255, nothing)
crange = [0,0,0, 0,0,0]

while True:
    flag, image = cap.read()
    img=image[320:450, 0:640]
    #боковая камера
    #img=image[120:290, 70:590]
    #img=image[0:480,0:640]
    hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV )

    # зчитуємо значення бігунків
    h1 = cv2.getTrackbarPos('h1', 'settings')
```

```

s1 = cv2.getTrackbarPos('s1', 'settings')
v1 = cv2.getTrackbarPos('v1', 'settings')
h2 = cv2.getTrackbarPos('h2', 'settings')
s2 = cv2.getTrackbarPos('s2', 'settings')
v2 = cv2.getTrackbarPos('v2', 'settings')

# формуємо початковий і кінцевий колір фільтра
h_min = np.array((h1, s1, v1), np.uint8)
h_max = np.array((h2, s2, v2), np.uint8)

# накладаємо фільтр на кадр у моделі HSV
thresh = cv2.inRange(hsv, h_min, h_max)
#edges = cv2.Canny (thresh, 100, 200)

cv2.imshow('result',thresh)
cv2.imshow('original',image)

cv2.imshow('result1', img)

ch = cv2.waitKey(5)
if ch == 27:
    break

```

Завдяки написаній програмі була знайдена маска для кожного кольору виробу з зображення з камери №1. Результати роботи даної програми представлені на рисунках 3.7-3.10.



Рисунок 3.7 – Початкове зображення сцени



Рисунок 3.8 – Отримана маска для синього кольору

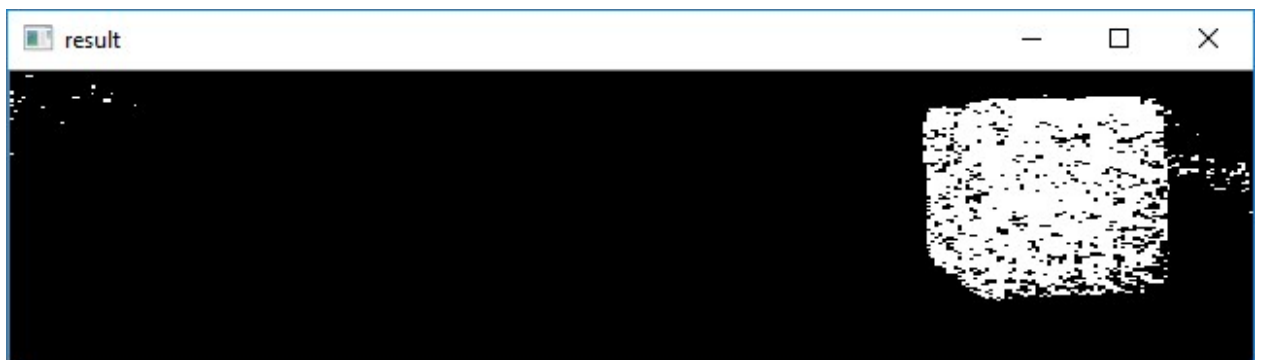


Рисунок 3.9 – Отримана маска для червоного кольору

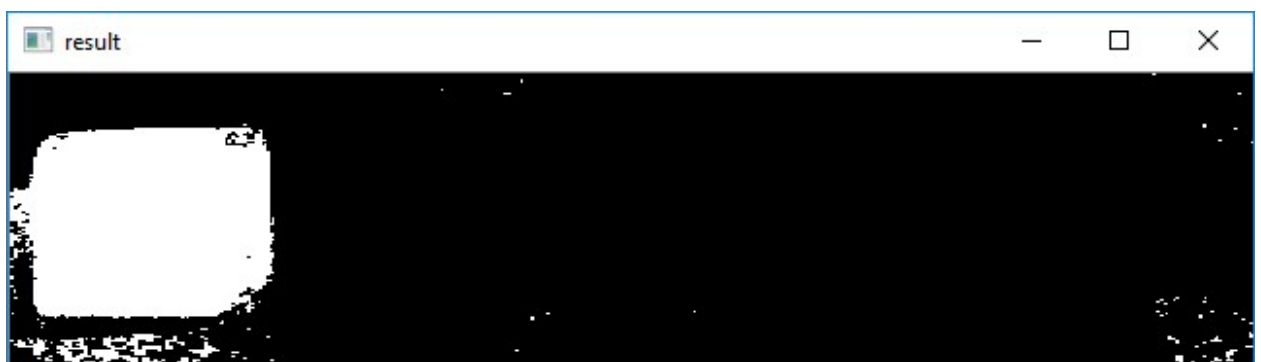


Рисунок 3.10 – Отримана маска для зеленого кольору

Камера №2 служить для визначення кольору виробу, що досягається за рахунок білого заднього фону тому для неї було обчислено маску всіх трьох кольорів (рис. 3.11).

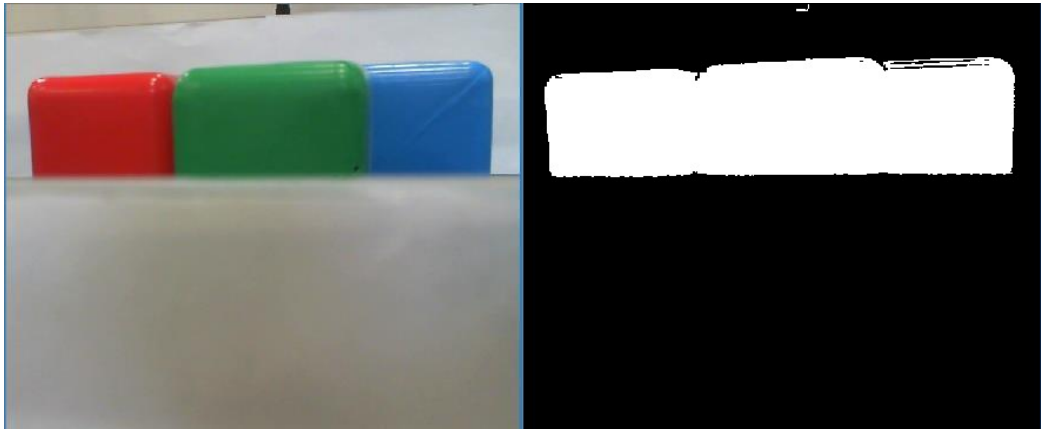


Рисунок 3.11 – Маска для вирахування білого фону із зображення камери №2

За отриманими значеннями можна програмно виділити об'єкт потрібного кольору, використовуючи функцію для вибірки тих пікселів зображення, які лежать в заданому інтервалі значень. Це функція `inRange (Mat src, Mat/Scalar lower, Mat/Scalar upper, Mat dst)`, вона проводить поелементну перевірку елементів масиву (пікселів) та перевіряє, чи лежать значення масиву між значеннями елементів двох інших масивів або скалярів.

За формулою:

$$dst(idx) = lower(idx) \leq src(idx) < upper(idx) ,$$

де `src` – вихідний масив;

`lower` – масив або скаляр з нижньою межею (включаючи), якщо це скаляр (`Scalar (x, y, z)`), то пікселі, значення яких нижче, ніж `x`, `y` та `z` для `HUE`, `SATURATION` і `VALUE` вважаються чорними пікселями в результуючому зображенні `dst`;

`upper` – масив або скаляр з верхньою межею (не включаючи), якщо це скаляр (`Scalar (x, y, z)`), то пікселі, значення яких вище або рівні `x`, `y` і `z` для `HUE`, `SATURATION` та `VALUE` вважаються чорними пікселями в результуючому зображенні `dst`;

`dst` – масив для зберігання результату.

Результат цього перетворення – чорно-біле зображення з виділеними об'єктами, які підходять під задані в аргументах умови.

3.3 Контурний аналіз

Контурний аналіз – це один із важливих та дуже корисних методів опису, розпізнавання, порівняння та пошуку графічних образів (об'єктів) [19]. Контур – це зовнішні обриси об'єкта. При проведенні контурного аналізу:

- вважається, що контур містить достатньо інформації про форму об'єкта;
- внутрішні точки об'єкта до уваги не беруться. Тому накладається ряд обмежень на сферу застосування контурного аналізу, які, в основному, пов'язані з проблемами виділення контуру на зображеннях;
- через однакову яскравість з фоном об'єкт може не мати чіткої межі, або може бути зашумлений завадами, що призводить до неможливості виділення контуру (успішне застосування лише при чітко вираженому об'єкті на контрастному тлі та відсутності завад);
- перекриття об'єктів або їхнє групування призводить до того, що контур виділяється неправильно і не відповідає межі об'єкта.

Серед методів отримання двійкового зображення можна виділити, наприклад, граничне перетворення або виділення об'єкта за кольором, які були описані раніше.

Після кожного з перетворень виходить бінарне зображення, яке однозначно задає межі об'єкта. Сукупність пікселів, що становлять межу об'єкта і є контуром об'єкта.

Щоб оперувати отриманим контуром, його необхідно якось представити (закодувати). Наприклад, вказувати вершини відрізків, що становлять контур.

Бібліотека OpenCV реалізує зручні методи для детектування та маніпуляції з контурами зображення. Для пошуку контурів на двійковому

зображенні використовується функція `findContours`. Ця функція може знаходити зовнішні та вкладені контури та визначати їхню ієрархію вкладення. Відобразити знайдені контури можна за допомогою функції `drawContours`.

Додаткові корисні функції:

- `boundingRect` – повертає прямокутник, яким можна обвести контур. Функція повертає прямокутник, у якого сторони строго вертикальні та горизонтальні;
- `minAreaRect2()`, яка повертає мінімально можливий прямокутник, яким можна обвести контур, але який може бути повернутий щодо системи координат зображення на певний кут.

3.4 Алгоритм роботи програми та її реалізація

Для визначення центру виробу, за допомогою двох камер, було розроблено алгоритм роботи програми. Суть полягає в наступному.

1. Кубик рухається конвеєром і при попаданні в кадр камери №2, завдяки відніманню білого фону знаходиться його контур. Як тільки центр виробу потрапляє до центру кадру, конвеєр зупиняється і визначається колір, що переважає в отриманому контурі.

2. Після цього починається робота із зображенням камери №1. Оскільки колір виробу визначений, то застосовується конкретна маска кольору для визначення контуру кубика та знаходження його центру.

Програмна реалізація алгоритму представлена нижче.

```
import numpy as np
import imutils
import cv2
from matplotlib import pyplot as plt
cap = cv2.VideoCapture(0)
# аркуш паперу
cap1 = cv2.VideoCapture(2)
```

```

color_blue = (255,0,0)
color_yellow = (0,255,255)
min_val, max_val = 232.75, 257.25
lower_paper = np.array([0,66,0])
upper_paper = np.array([255,255,229])
lower_conv = np.array([0,0,0])
upper_conv = np.array([0,0,0])

def contour(image,mincolor,maxcolor,title):
    hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    thresh= cv2.inRange(hsv, mincolor, maxcolor)
    blur = cv2.blur(thresh,(5,5))
    _, contours, hierarchy = cv2.findContours( blur.copy(),
                                              cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
    c = max(contours, key=cv2.contourArea)
    rect = cv2.minAreaRect(c) # намагаємося вписати прямокутник
    box = cv2.boxPoints(rect) # пошук чотирьох вершин прямокутника
    box = np.int0(box) # округлення координат
    cv2.drawContours(image,[box],0,(255,0,0),2)
    imgray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    mask = np.zeros(imgray.shape,np.uint8)
    cv2.drawContours(mask,[c],0,255,-1)
    mean_val = cv2.mean(image,mask = mask)
    M = cv2.moments(c)
    cx = int(M['m10']/M['m00'])
    cy = int(M['m01']/M['m00'])
    cv2.imshow(title, image)
    return cx,mean_val

# load the image, convert it to grayscale, and blur it slightly
while (1):
    ret,image = cap.read()
    img=image[120:290, 70:590]
    #cv2.imshow('test', img)
    X,mean_val=contour(img,lower_paper,upper_paper,'camera1')

    if min_val <= X <= max_val:
        cv2.putText(img, "Hello world!", (X,155),
                    cv2.FONT_HERSHEY_SIMPLEX, 1, color_yellow, 2)

    if mean_val[2]>120:
        lower_conv = np.array([0,93,45])
        upper_conv = np.array([5,255,255])
    elif mean_val[0]>120:

```

```

lower_conv = np.array([89,122,44])
upper_conv = np.array([140,255,255])

elif mean_val[1]>90:
    lower_conv = np.array([51,87,7])
    upper_conv = np.array([72,255,255])

_img1 = cap1.read()
frame=img1[320:450, 0:640]
Y,_=contour(frame,lower_conv,upper_conv,'camera2')
#print(X,Y)

k = cv2.waitKey(5) & 0xFF
if k == 27:
    break
cap.release()
cap1.release()
cv2.destroyAllWindows()

```

Результат роботи даної програми представлений на рисунках 3.12 - 3.14.

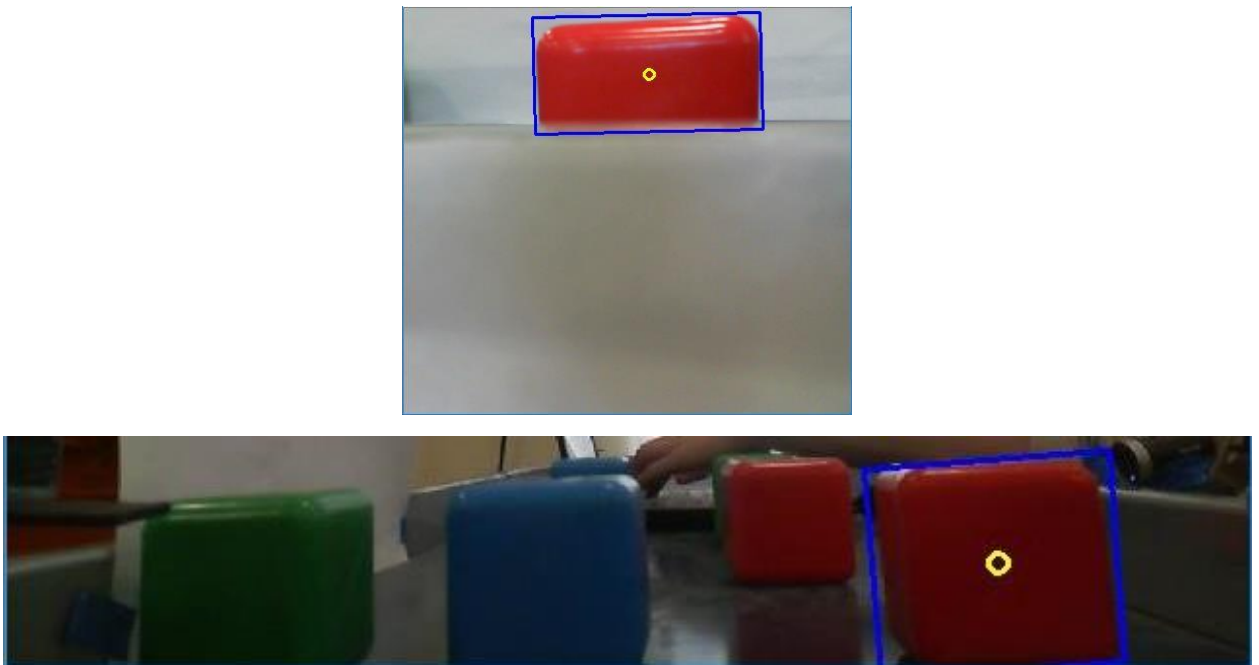


Рисунок 3.12 – Знаходження двох центрів червоного кубика

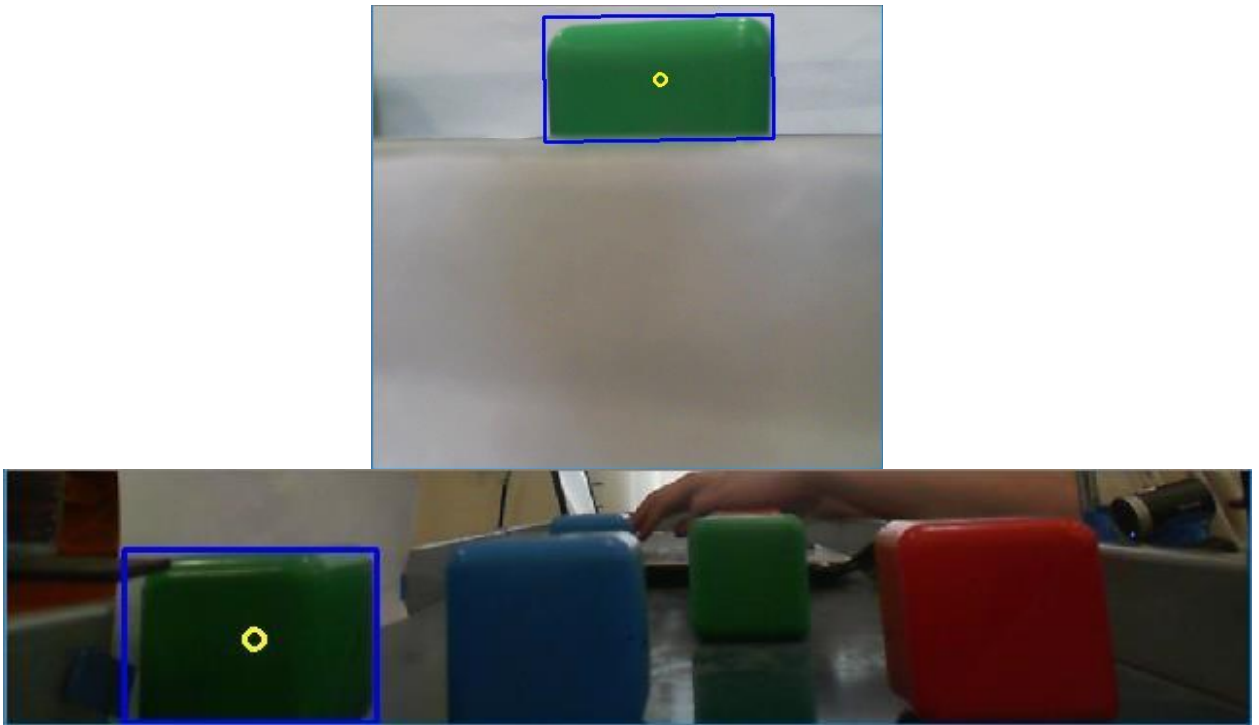


Рисунок 3.13 – Знаходження двох центрів зеленого кубика

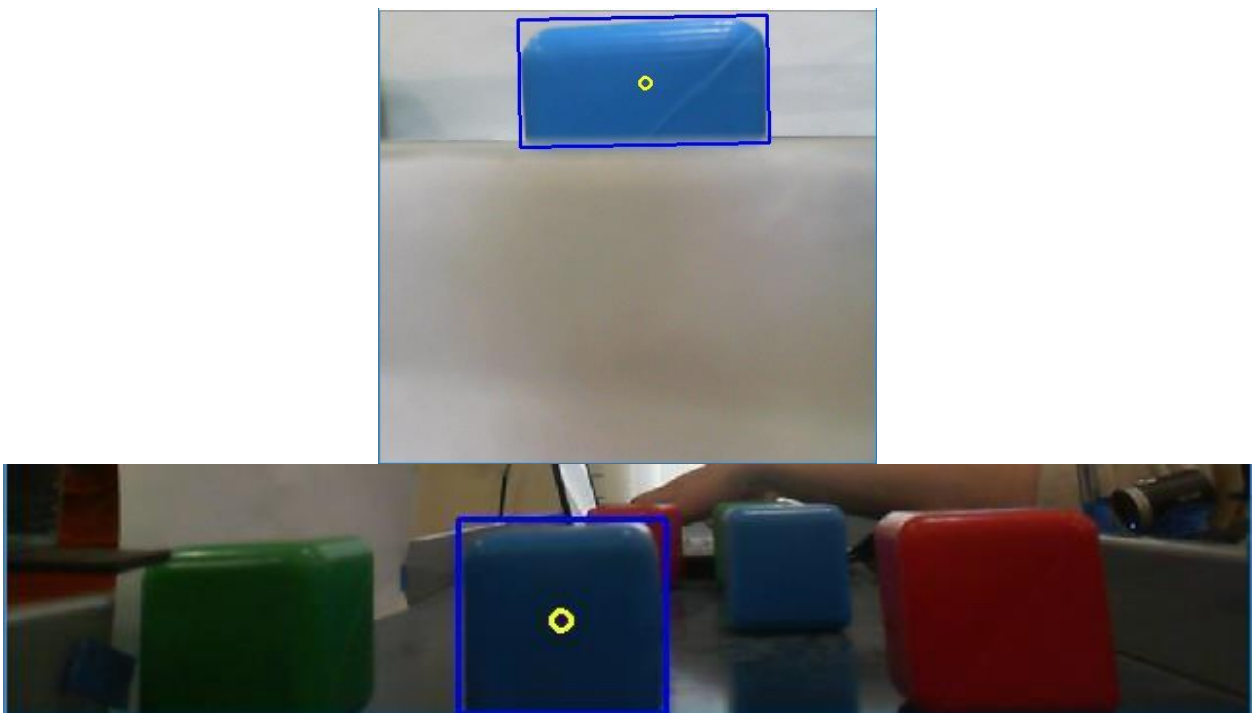


Рисунок 3.14 – Знаходження двох центрів синього кубика

3.5 Охорона праці

Охорона праці являє собою комплекс заходів, які спрямовані на попередження випадків, що є загрозою для безпеки і здоров'я людей під час виконання ними роботи. Охорона праці є комплексом заходів, які включають питання медицини, організації, техніки та інших заходів, які використовуються для запобігання нещасних, травматичних випадків, небезпечних впливів, вибухів, захисту від шкідливих речовин, завдання шкоди від удару струму, уникнення помилок, що мають небезпечні наслідки для здоров'я людини та безпеки підприємства.

Суттєвим та необхідним етапом охорони праці є навчання працівників дотриманню правил безпеки, надання спорядження індивідуального захисту і спеціального обладнання.

Тому необхідно виявити можливі небезпеки, виникнення яких є вірогідним під час роботи. При розробленні даної системи такими небезпеками можуть бути:

- небезпека впливу на зір розробника: за умови невиконання правил безпеки роботи за комп'ютером, зір розробника може бути пошкодженим;
- ризик удару струму: якщо електричні складові системи поєднані некоректно, система зазнала впливу рідини чи при інших недопустимих діях, користувач може отримати удар струму, що може викликати навіть смертельні наслідки;
- небезпека травм: компоненти системи можуть бути розташовані в небезпечних місцях;
- втрата інформації при її передаванні, помилки програмного забезпечення та інші подібні проблеми можуть призвести до несподіваних наслідків в роботі системи.

Наступним етапом після того, як було визначено потенційні загрози є вживання відповідних заходів безпеки для запобігання цих загроз і зменшення кількості ризиків. Такі заходи можуть складатись з:

- використання захисних окуляр для захисту очей під час розробки ПЗ;
- приховування всіх елементів системи, де проходить високий струм за допомогою захисних кожухів, для запобігання небезпеки удару струму;
- запровадження періодичного інструктажу, а також навчання працівників правилам безпечної розробки та роботи з автоматизованою системою, її компонентами та комп'ютерами. Працівники повинні бути інформовані щодо потенційних небезпек і методів роботи з системою.

3.6 Висновки до розділу

У третьому розділі описано розробку алгоритму роботи системи керування сортувальним роботом, створено технічну частину системи і проведено експерименти для визначення ефективності роботи системи сортування (протестовано метод пошуку виробу за кольором). Отримані результати підтверджують відповідність системи визначеним вимогам, що робить її придатною для подальшого впровадження у практичні системи автоматизації. Також в розділі розглянуто питання охорони праці.

ВИСНОВКИ

В кваліфікаційній роботі розглянуто основи технологій технічного зору та проведено обґрунтування актуальності та доцільності застосування даної технології в промисловій робототехніці.

Розроблено математичні моделі для ідентифікації та розпізнавання об'єктів, сформульовано задачу розробки системи технічного зору для промислового робототехнічного комплексу.

Розроблено алгоритм роботи СТЗ. Суть цього алгоритму полягає в тому, що положення виробу на конвеєрній стрічці визначається шляхом обробки зображення з відеокамери, що закріплена над робочою областю. Визначаються координати положення об'єкта на картинці, потім вони приводяться до системи координат, що з базою маніпуляційного робота. Принцип визначення координат об'єкта заснований на виділенні з відеопотоку, що надходить з камери, точок, що відповідають характерному кольору об'єкта; принцип сортування заснований на визначенні кольору об'єкта.

В результаті створено недорогу і просту систему сортування з використанням методів технічного зору, яка буде здатна знаходити виріб в просторі і проводити його класифікацію за заданими ознаками (кольором).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Поліщук М. М., Ткач М. М. Робототехнічні системи: проєктування і моделювання: навчальний посібник. Київ: КПІ ім. Ігоря Сікорського, 2021. 112 с.
2. Поліщук М. М., Ткач М. М. Робототехнічні системи та комплекси: мобільні роботи довільної орієнтації. Київ: КПІ ім. Ігоря Сікорського, 2023. 301 с.
3. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка, освітньо-професійних програм: «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи» / Упоряд. І. Ш. Невлюдов, Р. В. Артюх, В. В. Безкоровайний, Н. П. Демська, В.В. Євсєєв, О. І. Филипенко, О. М. Цимбал. Харків: ХНУРЕ, 2024. 57 с.
4. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. 29 с.
5. Ярош-Іванов М.В. Пошук об'єкта за кольором в системі технічного зору. «Автоматизація та приладобудування» ADED-2025, Випуск 2. С. 201-206.
6. Основи комп'ютерного зору. Режим доступу: https://learn.ztu.edu.ua/pluginfile.php/294778/mod_resource/content/1/%D0%9E%D0%A8%D0%86_%D0%9B-8_%D0%9A%D0%97%D1%96%D1%80.pdf (дата звернення 25.09.25).
7. KUKA.PerceptionTech. Режим доступу: <https://www.kuka.com/en-de/applications/other-robot-applications/sensor-systems-robot> (дата звернення 29.09.25).

8. *iRVision*. <https://www.fanuc.eu/eu-en/accessory/accessory/irvision> (дата звернення 21.09.25).
9. Borisov O.I., Gromov V.S., Pyrkin A.A., Bobtsov A.A., Nikolaev N.A., Robotic Boat Setup for Control Research and Education. *IFAC PapersOnLine*. 2016, V. 49, N. 6, P. 256-261.
10. Borisov O.I., Gromov V.S., Pyrkin A.A., Vedyakov A.A., Petranevsky I.V., Bobtsov A.A., Salikhov V.I., Manipulation Tasks in Robotics Education. *IFAC-PapersOnLine*. 2016, V. 49, N. 6, P. 22-27.
11. What is Machine Vision?. Retrieved May 1, 2018, from website: <https://machinevision.co.uk/machine-vision-products/vision-hardware/> (дата звернення 11.09.25).
12. Очима робота: що таке «машинний зір». Retrieved May 1, 2018, від website: <https://www.popmech.ru/technologies/238704-glazami-robota-hto-takoe-mashinnoe-zrenie/> (дата звернення 12.09.25).
13. Система технічного зору: особливості, завдання, принципи роботи, основні компоненти. URL: <http://bigbro.com.ua/sistematetchnogo-zoru-osoblivosti-zavdannya-printsipi-roboti-osnovni-komponenti/> (дата звернення: 5.10.2025).
14. Системи телебачення та технічного зору: навч. посібник, Вінниця: ВНТУ, 2014, 209 с.
15. KUKA KR 40 PA. Режим доступу: <https://svartech.com.ua/ua/p1393206997-kuka.html> (дата звернення 5.09.25).
16. KUKA KR C4. Режим доступу: <https://www.eurobots.com.ua/kr-c4-g64-uk.html> (дата звернення 5.09.25).
17. LifeCam Cinema. Режим доступу: <https://www.microsoft.com/accessories/ua/products/webcams/lifecam-cinema/h5d-00015> (дата звернення 6.09.25)
18. Python. Режим доступу: <https://ru.wikipedia.org/wiki/Python> (дата звернення 6.09.25).

19. OpenCV. Режим доступу: <https://docs.opencv.org/3.0-beta/doc/tutorials/tutorials.html> (дата звернення 23.09.25).
20. Hartley R., Zisserman A. Multiple View Geometry in Computer Vision. Cambridge University Press, 2004. 673 p.
21. Roger Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. IEEE Journal of Robotics and Automation, 3(4), 1987. P. 323-344.
22. Zhang B., Li Y. F. Automatic Calibration and Reconstruction for Active Vision Systems, Intelligent Systems, Control, and Automation. Science and Engineering. V. 57, Springer, 2012. P. 175.
23. Svoboda T., Martinec D., Pajdla T. A convenient multicamera self-calibration for virtual environments. PRESENCE: Teleoperators and Virtual Environments 14, 4, 2005. P. 407-422.
24. Camera Calibration. Режим доступу: https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_calib3d/py_calibration/py_calibration.html#calibration (дата звернення 16.11.25).
25. Іттен Йоганнес. Мистецтво кольору. ArtHuss, 2022. 96 с.
26. Основи роботи з кольоровими зображеннями в OpenCV. Режим доступу: <https://itmaster.biz.ua/programming/vision/opencv-basis.html> (дата звернення 6.10.25).
27. Ткачук А., Безвесільна О., Добржанський О., Павлюк Д. Ідентифікація об'єкта з використанням кольорового діапазону в схемі HSV. The scientific heritage. 2021, № 74. С. 50-56.
28. Аніщенко М. В. Промислові роботи: навч. посібник. Харків : НТУ "ХПІ", 2025. 464 с.