

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
Факультет комп'ютерних наук і технологій
(повне найменування інституту, назва факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБЛЕННЯ МОДЕЛЕЙ ДЛЯ ОБРОБЛЕННЯ ДАНИХ В
СЕРЕДОВИЩІ LABVIEW
DEVELOPMENT OF DATA PROCESSING MODELS IN THE LABVIEW
ENVIRONMENT

Виконав: студент(ка) 4 курсу, групи КНТ-128сп
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

Горб Д.Є.

(прізвище та ініціали)

Керівник Шитікова О.В.

(прізвище та ініціали)

Рецензент Морщавка С.В.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
“ ” 20 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Горба Данііла Євгеновича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Розроблення моделей для оброблення даних в середовищі Labview

Development of Data Processing Models in the LabVIEW Environment

керівник проєкту (роботи) Шитікова Олена Вікторівна, к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “30” березня 2021 року №103

2. Строк подання студентом проєкту (роботи) 17 травня 2021 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження. 2. Матеріали і методи. 3. Розробка архітектури програми. 4. Керівництво програміста. 5. Керівництво користувача 6. Додатки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	Шитікова О. В., доц.		
Нормоконтролер	Камінська Ж.К., ас.		

7. Дата видачі завдання “___” _____ року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	2–3 тижні	Розділ 1
3	Провести аналіз та класифікацію технічних засобів обробки даних;	4–5 тижні	Розділ 2
4	Провести аналіз методів для обробки зображень;	6 тиждень	Розділ 3
5	Провести аналіз алгоритмів пошуку розробити програмну реалізацію системи обробки даних з використанням середовища графічного програмування Labview	7–8 тижні	Розділ 3
6	Розробити керівництво програміста	9 тиждень	Розділ 4
7	Розробити керівництво користувача	10 тиждень	Розділ 5
8	Оформлення висновків	11 тиждень	Висновки
9	Оформлення пояснювальної записки та документів до неї.	12-13 тиждень	Додатки
10	Нормоконтроль та рецензування.	14–15 тижні	
11	Захист роботи.	16 тиждень	

Студент(ка)

(підпис) Горб Д.Є.
(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис) Шитікова О.В.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 71 с., 1 табл., 36 рис., 2 дод., 16 джерел.

LABVIEW, ГРАФІЧНЕ ПРОГРАМУВАННЯ, ОБРОБКА ДАНИХ, МОРФОЛОГІЧНІ МЕТОДИ

Об'єкт дослідження – методи та алгоритми для обробки даних.

Предмет дослідження – програмна реалізація методів та алгоритмів обробки даних.

Мета роботи – створення програмної реалізації системи для обробки даних, за рахунок використання середовища графічного програмування Labview.

Матеріали, методи та технічні засоби: графічна мова програмування «G», яка заснована на архітектурі потоків даних, персональний комп'ютер з процесором Intel Core 5 під управлінням операційної системи Microsoft Windows 10, середовище графічного програмування Labview.

Результати. Створено програмну реалізацію системи обробки даних

Висновки. Розроблено програмну реалізацію системи обробки даних, яка обробляє графічні дані отримані з відеозображення.

Галузь використання – відеоспостереження, робототехніка.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 71 pages, 36 figures, 1 tables, 2 appendixes, 16 sources.

LABVIEW, GRAPHIC PROGRAMMING, DATA PROCESSING, MORPHOLOGICAL METHODS.

Object of study is a methods and algorithms for data processing.

Subject of study is the software implementation of methods and algorithms for data processing.

The purpose of the work is to create a software implementation of the system for data processing, through the use of graphical programming environment Labview.

Materials, methods, and tools: the graphical programming language "G" is based on a data stream architecture, personal computer with Intel Core 5 processor running Microsoft Windows 10 operating system, Labview graphical programming environment.

Results. A software implementation of the data processing system has been created.

Conclusions. A software implementation of a data processing system that processes graphic data obtained from video images has been developed.

The field of use is video surveillance, robotics.

ЗМІСТ

С.

РЕФЕРАТ	4
ABSTRACT	5
ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК	8
ВСТУП	9
1. АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ..	11
1.1 Режими обробки даних	11
1.2 Способи обробки даних.....	13
1.3 Обробка зображень	14
2 МАТЕРІАЛИ І МЕТОДИ	16
2.1 Метод виявлення об'єктів на зображенні – кореляційне виявлення.	16
2.2 Морфологічні методи виявлення.....	18
2.3 Узгоджена фільтрація	19
2.4 Узагальнене перетворення Хафа	20
2.5 Обрання системи моделювання.....	20
2.6 Методи виявлення, реалізовані в середовищі Labview	27
2.7. Кольорові моделі, використовувані в машинному зорі	28
2.8. Інструменти пошуку збігів з еталоном в Labview	31
2.9. Інструменти реалізації стеження за об'єктом в середовищі Labview	32
3 РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМИ	35
3.1 Можливості середовища моделювання Labview	35
3.2 Обробка, аналіз зображень та відео в середовищі Labview.....	37
3.3 Функціональна схема досліджуваного алгоритму	38
3.4 Вимірювання відстані до об'єктів.....	38
4. КЕРІВНИЦТВО ПРОГРАМІСТА	43
4.1 Розробка програмної реалізації системи обробки даних в Labview ..	43
4.2 Створення шаблону	46
4.3 Вибір алгоритму пошуку.....	47

4.4 Створення масиву для пошуку за шаблоном	48
4.5. Вилучення зображення з маркером стеження.....	49
5 КЕРІВНИЦТВО КОРИСТУВАЧА	51
5.1 Запуск програми Labview	51
5.2 Створення шаблону	52
ВИСНОВКИ.....	56
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	57
ДОДАТОК А. Текст програми.....	59
ДОДАТОК Б. Слайди презентації.....	62

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

БД	база даних;
ПЗ	програмні засоби;
ПК	персональний комп'ютер;
HSI	Hue, Saturation, Intensity;
HSL	Hue, Saturation, Lightness;
HSV	Hue, Saturation, Value;
Labview	Laboratory Virtual Instrumentation Engineering Workbench.

ВСТУП

На сьогоднішній день у зв'язку з високим рівнем автоматизації різних сфер людської діяльності виникає потреба не тільки швидко та точно виконувати обчислення, але також ефективно розпізнавати візуальну інформацію. Та на основі проведеного аналізу вирішувати складні завдання управління та контролю.

Дані системи прийнято називати системами машинного зору. Які представляють собою сукупність інженерних технологій, методів і алгоритмів, пов'язаних із завданням інтерпретації сцени спостереження за її двовимірним проєкція (зображень), а також як практичне використання результатів цієї інтерпретації. Такі системи мають взаємопов'язану технологічну послідовність, яка включає наступні складові [1, 2]:

- отримання зображення від відеокамери;
- обробку зображення;
- логічний аналіз цифрового зображення і виділення потрібної інформації;
- переміщення камери в просторі.

Відеокамера та пристрій обробки зображення є головними складовими систем машинного зору, їх також називають «технічний зір» [1].

Системи машинного зору, призначені для використання на виробництві, в медицині, в обороні та ін. Дані системи повинні відрізнятися високою продуктивністю, надійністю та гнучкістю. Дані вимоги можуть бути виконані в системах, які побудовані на базі промислових персональних комп'ютерів (ПК) та мати стандартні шини PCI і PCIe. Для вирішення складних завдань, застосувань реального часу найбільшою мірою відповідають системи, побудовані на базі стандартів ComracPCI та PXI [2,3].

При побудові систем для обробки даних виникає необхідність також ефективно розпізнавати візуальну інформацію для її подальшого

використання, тому розробка системи для обробки даних з використанням середовища графічного програмування є досить актуальним завданням.

Об'єкт дослідження – методи та алгоритми для обробки даних.

Предмет дослідження – програмна реалізація методів та алгоритмів обробки даних.

Мета роботи – побудова системи для обробки даних, за рахунок використання середовища графічного програмування Labview.

Відповідно до мети були поставлені наступні завдання:

- провести аналіз та класифікацію технічних засобів обробки даних;
- провести аналіз методів для обробки зображень;
- провести аналіз алгоритмів пошуку;
- розробити програмну реалізацію системи обробки даних з використанням середовища графічного програмування Labview.

1. АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Режими обробки даних

При проєктуванні технологічних процесів необхідно враховувати режими їх реалізації. Режим реалізації залежить від об'ємно-часових особливостей вирішуваних завдань: періодичності і терміновості, вимог до швидкості обробки повідомлень, а також від режимних можливостей технічних засобів та технічних характеристик ПК. Існують: пакетний режим; режим реального масштабу часу; режим поділу часу; регламентний режим; запитальний; діалоговий; телеобробки; інтерактивний; одно програмний; багатопрограмний або мультиобробка [1-3].

Пакетний режим. Збір та реєстрація інформації, введення та обробка не збігаються за часом. Спочатку користувач збирає інформацію та формує її в пакети відповідно до обраних ознак, наприклад вид завдань. Після завершення прийому інформації проводиться її введення та обробка, в цей час відбувається затримка обробки. Цей режим використовується, як правило, при централізованому способі обробки інформації [4].

Діалоговий режим. Існує можливість користувача безпосередньо взаємодіяти з системою обробки в процесі роботи користувача. Програми обробки даних знаходяться в пам'яті ПК постійно, якщо ПК доступний в будь-який час, або протягом певного проміжку часу, коли ПК доступний користувачеві. Взаємодія користувача з обчислювальною системою у вигляді діалогу може бути багатоаспектною і визначатися різними факторами: мовою спілкування, активною або пасивною роллю користувача; хто є ініціатором діалогу - користувач або ПК; часом відповіді; структурою діалогу. Якщо ініціатором діалогу є користувач, то він повинен мати знання для роботи з процедурами, форматами даних. Якщо ініціатор - ПК, то машина сама повідомляє на кожному кроці, що потрібно робити з різноманітними можливостями вибору. Даний метод роботи називається "вибором меню".

Використання даного методу забезпечує підтримку дій користувача та формує їх послідовність [3, 4].

Діалоговий режим вимагає певного рівня технічної оснащеності користувача, тобто наявності терміналу або ПК, пов'язаних з центральним сервером. Цей режим використовується для доступу до інформації, обчислювальним або програмних ресурсів. Можливість роботи в діалоговому режимі може бути обмежена в часі та початку/ кінці роботи. Іноді розрізняють в діалоговому режимі ще запитальний, в такому випадку розуміється одноразове звернення до системи, після якого буде отримана відповідь та система вимкнеться, а у випадку діалогового режиму система після відповіді чекає подальших дій користувача.

Режим реального масштабу часу. Означає здатність обчислювальної системи взаємодіяти з контрольованими або керованими процесами в темпі протікання даних процесів. Час реакції ПК повинен задовольняти темпу контрольованого процесу або вимогам користувачів та мати мінімальну затримку. Як правило, цей режим використовується при децентралізованій та розподіленій обробці даних [4].

Режим телеобробки дає можливість користувачеві віддалено взаємодіяти з системою обробки.

Інтерактивний режим передбачає можливість двосторонньої взаємодії користувача з системою, таким чином, користувач має можливість впливати на процес обробки даних [4].

Режим поділу часу дає можливість системи виділяти свої ресурси групі користувачів по черзі. Система обробки обслуговує кожного користувача з такою швидкістю, що створюється враження одночасної роботи декількох користувачів. Даний ефект досягається за рахунок використання відповідного програмного забезпечення [3].

Однопрограмний та багатопрограмний режими характеризують здатність системи працювати одночасно з однією або декількома програмами.

Регламентний режим характеризується обмеженістю часі окремих завдань користувача. Наприклад, отримання результатів обробки по закінченню місяця, розрахунок відомостей нарахування зарплати до певних дат. Терміни рішення встановлюються заздалегідь відповідно до регламенту [5].

1.2 Способи обробки даних

Для обробки даних виділяють наступні способи обробки даних: централізований, децентралізований, розподілений та інтегрований.

Централізований спосіб передбачає наявність даних. При використанні даного способу користувач передає на ПК вихідну інформацію та отримує результати обробки у вигляді відповідних документів. Особливостями даного способу обробки є складність та трудомісткість налагодження швидкості, безперебійного зв'язку, великої завантаженості ПК інформацією, регламентацією термінів виконання операцій, організації безпеки системи від можливого несанкціонованого доступу [5].

Децентралізована обробка. Використання даного способу дає можливість автоматизувати певне робоче місце.

Розподілений спосіб обробки даних заснований на розподілі функцій обробки між різними ПК, наявними в мережі. Даний спосіб може бути реалізований двома шляхами: перший передбачає встановлення ПК на кожному рівні системи, при цьому обробка даних здійснюється одним або кількома ПК в залежності від реальних можливостей системи та потреб на поточний момент часу. Також даний спосіб можна реалізувати шляхом розміщення великої кількості різних процесорів всередині однієї системи. Даний підхід застосовується в системах обробки банківської та фінансової інформації. Переваги розподіленого способу: можливість обробляти в заданий термін дані будь-якого обсягу; високий рівень надійності, через те що при відмові одного технічного засобу є можливість миттєвої заміни його на

інший; скорочення часу та витрат на передачу даних; гнучкість системи, спрощення розробки та експлуатації ПЗ. Розподілений спосіб ґрунтується на комплексі спеціалізованих процесорів, тобто кожен ПК призначений для вирішення певних завдань, або завдань певного рівня [5, 6].

Інтегрований спосіб обробки інформації передбачає створення інформаційної моделі керованого об'єкта, тобто створення розподіленої бази даних. Такий спосіб забезпечує максимальну зручність для користувача. З одного боку, бази даних передбачають колективне користування та централізоване управління. З іншого боку, обсяг інформації, різноманітність вирішуваних завдань вимагають розподілу БД [5, 6]. Технологія інтегрованої обробки інформації дозволяє поліпшити достовірність, якість та швидкість обробки, тому що обробка проводиться на основі єдиного інформаційного масиву, одноразово введеного в ПК. Особливістю цього способу є розподілення процедур обробки від процедур збору, підготовки та введення даних.

1.3 Обробка зображень

Останнім часом все більшого застосування набувають цифрова обробка і цифровий аналіз зображень та використовуються в різних областях науки і техніки, таких як інтелектуальні роботи технічні комплекси, системи промислового контролю, системи управління рухомими апаратами, обробка даних дистанційного зондування, біомедичні дослідження, нові технології обробки документів. Дані системи обробки зображень мають узагальнену назву – машинний зір [2, 6].

Апаратні можливості, які використовуються в даній області досягли досить високого рівня та в багатьох аспектах наближаються до можливостей людини, що робить проблематику машинного зору досить актуальним напрямком. Розширення багатьох сенсорів для отримання відеоінформації практично відповідає числу елементів сітківки ока людини, а можливості ПК

та спеціальних процесорів близькі до характеристик, які використовуються в мозку для обробки зображень [6]. В біомеханіці активно ведуться розробки складних механічних маніпуляторів, достатніх для імітації моторної діяльності людини для управління різними технічними системами.

Таким чином для створення складних автономних та робототехнічних комплексів, «інтелектуальних машин», які функціонують в реальному масштабі часу, виникає необхідність вирішення такого завдання як розробка методів і алгоритмів «розуміння» зображень. Тому досить актуальним є завдання обробки графічних даних, а саме зображень.

2 МАТЕРІАЛИ І МЕТОДИ

Машинне зір – це напрямок науки, який динамічно розвиваються та широко використовується в різних галузях, починаючи з систем автоматичного контролю виробництва та закінчуючи інтелектуальними людино-машинними інтерфейсами. Однією з основних частин машинного зору є розпізнавання образів, що забезпечує завдання визначення належності отриманого зображення до одного зі збережених в пам'яті еталонів [4]. В даний час існують різні методи виявлення об'єктів на зображенні, заданих еталоном.

2.1 Метод виявлення об'єктів на зображенні – кореляційне виявлення

Одним з основних методів виявлення об'єктів на зображенні є кореляційне виявлення.

Даний метод реалізується за допомогою порівняння зображення з еталоном, в основі методу є розгляд зображень як двовимірних функцій яскравості. Процедура виявлення проводиться таким чином: еталон послідовно переміщається по зображенню, і в кожному положенні досліджується подібність між еталоном і реальною функцією яскравості на фрагменті. При цьому вимірюється міра близькості між зображеннями або відстань між ними [7].

Для обчислення відстаней між зображеннями використовують наступну формулу [8]:

$$p(f, d) = (\sum_{(x, y) \in X} (f(x, y) - d(x, y))^a)^{\frac{1}{a}}, \quad (2.1)$$

де $f(x, y)$, $d(x, y)$ – функції інтенсивності X -апертури зони пошуку;

$\alpha \in [1, \infty]$ визначає характеристики використовуваної метрики [6].

Нехай дано n еталонних зображень $\{d_i\}$, $i = 1, \dots, n$, кожне з яких відповідає i -му класу. Віднесення знову висунутого фрагмента зображення f до деякого класу j може здійснюватися, наприклад, за методом мінімального відстані до відповідного еталона [9].

$$j = \arg \min_i p(f, d_i), \quad (2.2)$$

Основний недолік даного методу – критерій виявлення залежить від лінійних розмірів зображення та еталона [9]. Тому частіше використовують, нормований коефіцієнт кореляції.

$$k(f, d) = \frac{\sum_{(x,y) \in X} (f(x,y) - f^0)(d(x,y) - d^0)}{\sqrt{\sum_{(x,y) \in X} (f(x,y) - f^0)^2} \sqrt{\sum_{(x,y) \in X} (d(x,y) - d^0)^2}}, \quad (2.3)$$

де f^0, d^0 - середні значення для зображень f і d .

Виявлення фрагмента зображення за допомогою методу максимальної кореляційної зв'язку [8]:

$$j = \arg \min_i k(f, d_i), \quad (2.4)$$

Перевірка достовірності детектування здійснюється на підставі отриманого значення максимальної кореляції. Якщо $k(f, g_i) \geq k_{min}$, то виявлення достовірне. В іншому випадку об'єкт вважається нерозпізнаним. Основні недоліки кореляційних схем:

- виявлення проявляються при яскравості і геометричних спотвореннях вхідного зображення в порівнянні з еталонним;
- при ракурсних викривленнях відбувається швидке зменшення кореляційної зв'язку [6].

2.2 Морфологічні методи виявлення

Подальшим розвитком кореляційного підходу є морфологічні методи аналізу сцени. Визначення даних методів дає Ю.П. Питьєв в своїй книзі: «Зображення однієї і тієї ж сцени можуть значно відрізнятися між собою при варіації умов їх реєстрації – таких, як освітлення, оптичні властивості поверхні об'єктів, властивості середовища, що впливає на оптичні властивості об'єкта, тощо. Цей факт ускладнює завдання аналізу сцен за їхніми зображеннями, оскільки зв'язок між розташуванням об'єктів і розподілом яскравості на полі зору неоднозначна. Проте, якщо всілякі зображення сцени можуть бути описані певним класом перетворень, які виконуються над деякими зображеннями цієї сцени, то характеристикою форми об'єктів на зображенні природно вважати максимальний інваріант даного класу перетворень. Оскільки цей інваріант, як правило, не дозволяє відновити форму об'єктів, він названий формою зображення, а методи аналізу зображень, засновані на даній ідеї, – морфологічними» [7]. Морфологічні методи орієнтовані на пошук вирішення завдань, які можна розділити на два класи. Перший клас включає в себе завдання виділення невідомих об'єктів на зображеннях відомої сцени, одержуваних при невідомих умовах. Вирішення даного завдання показано на рис. 2.1.

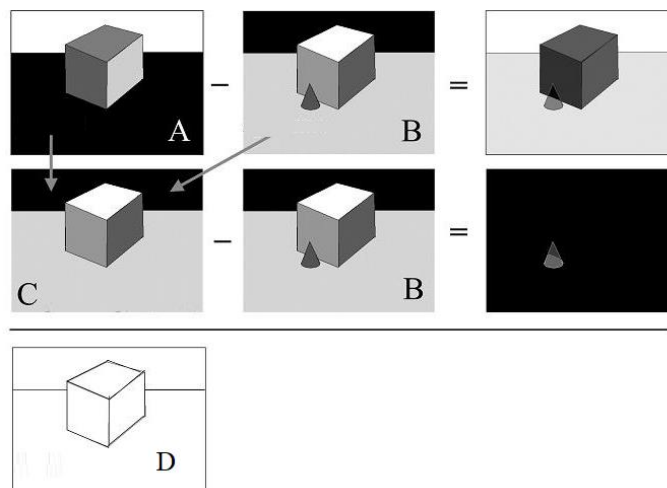


Рисунок 2.1 – Морфологічне виділення невідомого об'єкта

Для морфологічного виділення невідомого об'єкта розглядають два зображення однієї і тієї ж сцени – А та В, які відрізняються тим, що отримані в різних умовах освітленості, в результаті чого мають різні характеристики яскравості видимого поля, на зображенні В з'явився об'єкт, відсутній на зображенні А. Якщо порівняти зображення А та В, то отримане зображення не дозволить відокремити зміни в складі сцени від змін яскравості. Але, якщо сформувати вторинне зображення С, яке має форму зображення А, а яскравість елементів цієї форми оцінювалася б по зображенню В. Дане зображення називається проєкцією В на форму А, по морфології Питьєва [8].

Другий клас заснований на пошуку відомих об'єктів на невідомій сцені. Однак і морфологічні методи мають свої недоліки, наприклад не забезпечують необхідну стійкість при геометричних викривленнях, необхідність паралельного аналізу інформативності зображення. Незважаючи на свої недоліки морфологічні алгоритми мають високу стійкість до радіометричних перетворень і зашумлення [8].

2.3 Узгоджена фільтрація

У випадку, коли зображення розглядається як бінарне, еталонне виявлення зводиться до процедури узгодженої фільтрації. Узгоджена фільтрація є різновидом віконної фільтрації. Сенс узгодженої фільтрації полягає в тому, що створюється вікно, яке збігається за формою з шуканим об'єктом, призначається центральний піксель вікна, потім відбувається прохід вікна по вхідному зображенню. У кожному з можливих положень вікна підраховується число ненульових елементів вікна, якщо дане число більше деякого порога, то приймається рішення, відповідне виявлення об'єкта в цій точці, і в центральному пікселі поточного вікна на вихідному зображенні виставляється значення 1 [5]

2.4 Узагальнене перетворення Хафа

Для виявлення графічних примітивів, описаних аналітичними рівняннями, використовують перетворення Хафа. Розвитком якого є узагальнений метод Хафа, що дозволяє виявляти криві, що не описуються аналітичними рівняннями. Дані методи мають високий ступінь перешкодозахищеності, а також високою точністю визначення орієнтації і розташування. Однак існує обмеження на використання цієї групи методів - представлені алгоритми виявляють не напівтонові об'єкти, а їх контури, тому об'єкти з нечіткими контурами не можуть бути детектировані за допомогою методу Хафа [10].

2.5 Обрання системи моделювання

Одним з найбільш поширених програмних рішень для реалізації систем машинного зору є OpenCV.

OpenCV – це бібліотека алгоритмів комп'ютерного зору, обробки зображень та чисельних алгоритмів загального призначення з відкритим кодом. [11] та реалізована на C/C ++, також існують версії для Python та Java, Ruby, Matlab, Lua та інших мов. Бібліотека поширюється по відкритій ліцензії, отже, її можна вільно і безкоштовно використовувати як у відкритих проєктах з відкритим кодом, так і в закритих, комерційних проєктах. Бібліотеку не обов'язково копіювати цілком в свій проєкт, можна використовувати фрагменти коду. Єдина вимога ліцензії - наявність в супроводжуючих матеріалах копії ліцензії OpenCV.

Дана бібліотека використовується великою кількістю компаній, організацій, університетів, (NVidia, Willow Garage, Intel, Google, Stanford University). Компанії NVidia і WillowGarage частково спонсорують її розробку.

Багатовимірна архітектура проєкту представлена на рис. 2.2.

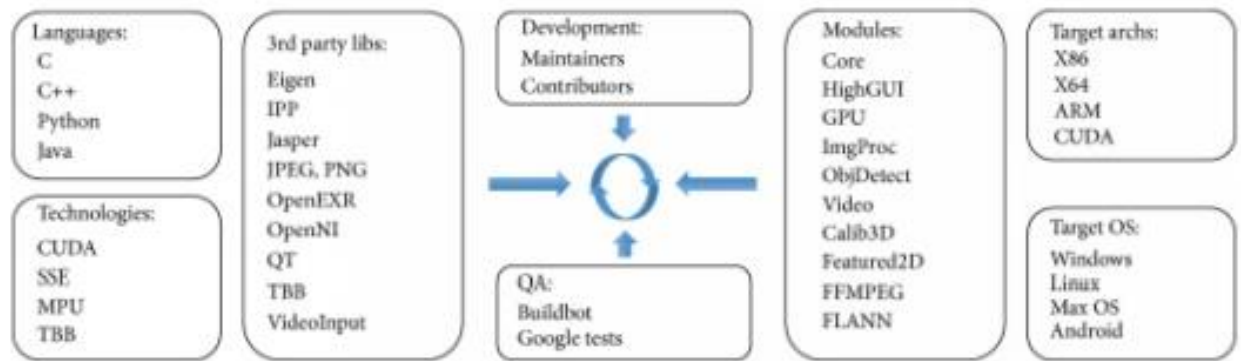


Рисунок 2.2 – Архітектура OpenCV

Бібліотека складається з 16 модулів, має реалізацію близько 1000 алгоритмів.

Підтримуються основні операційні системи: MS Windows, Linux, Mac, Android, iOS та має функцію підтримки сторонніх бібліотек.

Основні модулі бібліотеки можна віднести до наступних груп:

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

LabVIEW () - це середовище розробки і платформа для виконання програм, створених на графічній мові програмування «G» фірми National

Instruments (США). Дане середовище розробки містить LabView IMAQ бібліотеки для роботи з зображеннями.

LabVIEW використовується в системах збору та обробки даних, а також для управління технічними об'єктами і технологічними процесами. Ідеологічно LabVIEW дуже близька до SCADA-системам.

Графічна мова програмування «G», який використовується в LabVIEW, заснована на архітектурі потоків даних. Послідовність виконання операторів в таких мовах визначається не порядком їх слідування входах цих операторів. Оператори, не пов'язані з даними, виконуються паралельно в довільному порядку.

Програма LabVIEW називається і є віртуальним приладом та складається з двох частин:

- блокової діаграми, яка описує логіку роботи віртуального приладу;
- фронт панелі, яка описує зовнішній інтерфейс віртуального приладу.

LabVIEW підтримує величезний спектр обладнання різних виробників та має в своєму складі (або дозволяє додавати до

- базового пакету) численні бібліотеки компонентів;
- для підключення зовнішнього обладнання по найбільш поширеним інтерфейсів і протоколів (RS-232, GPIB-488, TCP / IP);
- для віддаленого управління ходом експерименту;
- для керування роботами і системами машинного зору;
- для генерації і цифрової обробки сигналів;
- для застосування різноманітних математичних методів обробки даних;
- для візуалізації даних і результатів їх обробки (включаючи 3D-моделі);
- для моделювання складних систем;
- для зберігання інформації в базах даних і генерації звітів;
- для взаємодії з іншими додатками в рамках концепції COM / DCOM / OLE.

Спеціальний компонент LabVIEW IMAQ - елемент керування відображенням зображень. який дозволяє користувачеві відображати тип даних зображення IMAQ. Даний елемент керування встановлюється з NI-IMAQ 2.6 та новіших версій, і його можна використовувати з LabVIEW 7.0 пізніше. Модуль розвитку зору потрібен для використання елемента керування відображенням зображень. Контроль відображення зображення дозволяє користувачеві відображати інформацію про зображення, змінювати поточний вигляд зображення та створювати область інтересів.

Бібліотеки IMAQ Vision можуть обробляти три типи зображень: відтінки сірого, кольорові та складні зображення.

IMAQ Vision може обробляти зображення з 8-бітовим, 10-бітовим, 12-бітовим, 14-бітним, 16-бітовим, кодуєчим кодом чи кольором. Спосіб кодування зображення залежить від природи пристрою збору зображення, типу обробки зображення, який потрібно використовувати, та типу аналізу, який потрібно виконати.

IMAQ Vision не підтримує інші типи кодування зображень, зокрема зображення, кодовані як 1-бітові, 2-бітові або 4-бітові зображення. У цих випадках IMAQ Vision автоматично перетворює зображення на 8-бітове зображення (мінімальна глибина бітів для IMAQ Vision) під час відкриття файлу зображення.

Кількість площин на зображенні відповідає кількості масивів пікселів, що складають зображення. Зображення у відтінках сірого або псевдокольорове зображення складається з однієї площини, тоді як справжнє кольорове зображення складається з трьох площини - по одній для червоної, синьої та зеленої складових компонент.

На кольорових зображеннях інтенсивність кольорової складової пікселя кодується на три різні значення. Кольорове зображення - це поєднання трьох масиви пікселів, що відповідають червоному, зеленому та синьому компонентам у RGB зображення. Зображення HSL визначаються їх відтінком, насиченістю та значення яскравості.

З метою обрання інструментальних засобів реалізації та подальшої роботи були проведені заміри часу виконання базових функцій систем обробки відео LabView IMAQ [13] та OpenCV [14]. Для даного дослідження були обрані класи функцій: читання файлів .avi, фільтрація, малювання, витяг каналів, висновок на екран. З кожного класу було обрано по одній функції, та були проведені заміри часу виконання даних функцій. Кожна функція виконувалася 100 разів в циклі. Для дотримання рівних умов функції бібліотеки IMAQ та бібліотеки OpenCV запускалися з середовища LabView. Порівняння показало більш високу (від 3 до 10 разів) ефективність бібліотеки OpenCV для функцій читання файлів, фільтрації, малювання примітивів, вилучення каналів. IMAQ виявився продуктивніше при копіюванні одного зображення в інше (на 10%) і при виведенні на екран (в 5 разів). На підставі того, що в задачі є необхідність відображення оброблюваного відео на екрані, було прийнято рішення використовувати LabView в якості базового інструменту.

Таблиця 2.1 – Порівняння бібліотек

Функція	Labview, мсек	OpenCV, мсек	Відношення часу Labview та OpenCV
1	2	3	4
Читання кадру з .avi	75197	9866	7,62
Вивід зображення на екран	2635 0,19	14207	0,19
Триколірне розмиття	166364	40455	4,11
Перетворення в сіру шкалу	18755	6038	3,10
Перетворення в сіру шкалу з подальшим розмиванням	62145	19495	3,18

Продовження таблиці 2.1

1	2	3	4
---	---	---	---

Копіювання зображення	3933	4413	0,89
Малювання відрізка (100 разів)	40318	4053	9,95
Витяг зеленого каналу	8098	2591	3,13
Перетворення з RGB в HLS	396111	124105	3,19
Копіювання кольорів пікселів з одного відрізка до іншого	2872	15454	5,38

Таким чином, після проведення дослідження та порівняння вибір було зупинено на програмно-апаратному комплексі Labview. Вибір обумовлено можливістю роботи з зображенням в реальному часі та необхідністю відображення оброблюваного зображення на екрані, в чому даний програмний засіб має перевагу.

2.6 Методи виявлення, реалізовані в середовищі Labview

NI Vision реалізує два методи для порівняння з шаблоном – pyramidal matching (пірамідального збіги) і low discrepancy sampling (низького розбіжності вибірки). Дані методи використовують нормовану взаємну кореляцію в якості основного методу [9].

Процес пошуку збігів складається з двох етапів: навчання та пошуку. На стадії навчання, алгоритм витягує значення сірого і / або контуру з шаблону зображення. Алгоритм організовує і зберігає інформацію таким чином, щоб полегшити швидкий пошук на вхідному зображенні. В NI Vision інформація, отримана на цьому етапі, зберігається як частина шаблону зображення.

На стадії пошуку збігу, схема узгодження алгоритму витягує значення сірого і/або краю градієнта інформацію з вхідного зображення. Потім алгоритм знаходить збіги шляхом розміщення областей на вхідному зображенні, де спостерігається високий рівень кореляції. Кореляція є

найбільш поширеним методом для пошуку шаблону в зображенні. Так як основний механізм для кореляції заснований на серії операцій множення, процес кореляції забирає багато часу. Технології, такі як MMX дозволяють паралельне множення, що скорочує загальний час обчислень. Для того щоб збільшити швидкість процесу пошуку збігу, зменшується розмір зображення і обмежується область зображення, в якому відбувається пошук. Pyramidal matching (пірамідального збіги) і image understanding (розуміння зображення) два способи збільшити швидкість процесу пошуку.

Pyramidal Matching дозволяє зменшити час обчислення збіги з шаблоном, за рахунок зменшення розміру зображення і шаблону. У Pyramidal Matching, як вхідне зображення, так і шаблон опитуються меншим просторовим дозволом з використанням гаусом піраміди [11].

Даний метод опитує кожен піксель і таким чином, зображення і шаблон обидва можуть бути зведені до однієї чверті своїх початкових розмірів для кожного послідовного рівня піраміди. На етапі навчання, алгоритм автоматично обчислює максимальний рівень піраміди, яка може бути використана для даного шаблону і дізнається дані, необхідні для подання шаблону [11]. Алгоритм намагається знайти оптимальний рівень піраміди (на основі аналізу даних шаблону), який дасть найшвидше і точний збіг. Два види даних можуть бути використані – значення сірого (в залежності від інтенсивності пікселів) і градієнти (наведені на основі інформації про контурах).

Pyramidal Matching розділяється на Gray Value Method (метод значень сірого) і Gradient Method (метод градієнтів). Gray Value Method використовує нормовані піксельні значення сірого, як функції. Це гарантує, що ніяка інформація не буде втрачена, що корисно, якщо шаблон не містить структуровану інформацію, але має складні текстури або різкі контури. Проте, цей метод має недолік, що пошук збігу страждає при зіткненні з затінюванням і нерівномірним зміною освітлення. Незважаючи на ці обмеження, метод

працює в широкому розмаїтті сценаріїв і підходить для загального використання [12].

Gradient Method використовує відфільтровані контурні пікселі в якості ознак. Контур зображення обчислюється з доданого півтонування, і пороговий градієнт інтенсивності обчислюється на основі аналізу зображення шаблону. Всі контурні вектори, значення яких вище порогових, зберігаються в якості ознак [12]. Збіг ґрунтується на векторній кореляції, а не на нормованій взаємної кореляції. Цей метод більш стійкий до затінення і змін освітленості в порівнянні з Gray Value Method, і часто швидше, так як менше даних має бути обчислено. Проте, чіткість контурів знижується при дуже низькій роздільній здатності, цей метод вимагає, використання більш високих дозволів ніж Gray Value Method.

NI Vision використовує неоднорідний метод створення шаблонів, який включає розуміння зображення для ретельного та ефективного опису шаблону. Ця інтелектуальна технологія вибірки спеціально включає в себе комбінацію контурних пікселів і область пікселів, що оточують контури [13].

NI Vision використовує неоднорідний метод створення шаблонів, який включає розуміння зображення для ретельного та ефективного опису шаблону. Ця інтелектуальна технологія вибірки спеціально включає в себе комбінацію контурних пікселів і область пікселів, що оточують контури. NI Vision використовує подібну техніку, коли користувач вказує на те, що об'єкт може бути повернутий на зображенні [13]. Даний метод використовує спеціально вибраний шаблон пікселів, чиї значення-або відносні зміни в значеннях, відображають обертання малюнка. Інтелектуальна вибірка шаблону зменшує надлишкову інформацію і підкреслює функцію, щоб реалізувати ефективну кореляцію.

2.7 Кольорові моделі, використовувані в машинному зорі

Для розробки системи обробки графічних зображень необхідно розглянути найбільш часто використовувані моделі кольору: HSL, HSV, HSI.

HSL (Hue, Saturation, Lightness) – модель, колірними координатами якої є тон, насиченість і світлість. Дана модель заснована на 3х-координатному способі запису кольору. Використовується циліндрична система координат [10].

HSV (Hue, Saturation, Value) – модель, в якій координатами кольору є тон, насиченість і значення [11]. Під значенням розуміється яскравість. Дана модель є нелінійним перетворенням моделі RGB.

Модель HSI (Hue, Saturation, Intensity) представляє собою модель, координатами кольору якої є тон, насиченість, інтенсивність. В системі HSI кожний колірний піксель кодується одним значенням загальної інтенсивності світла і двома значеннями тоном та насиченістю [12].

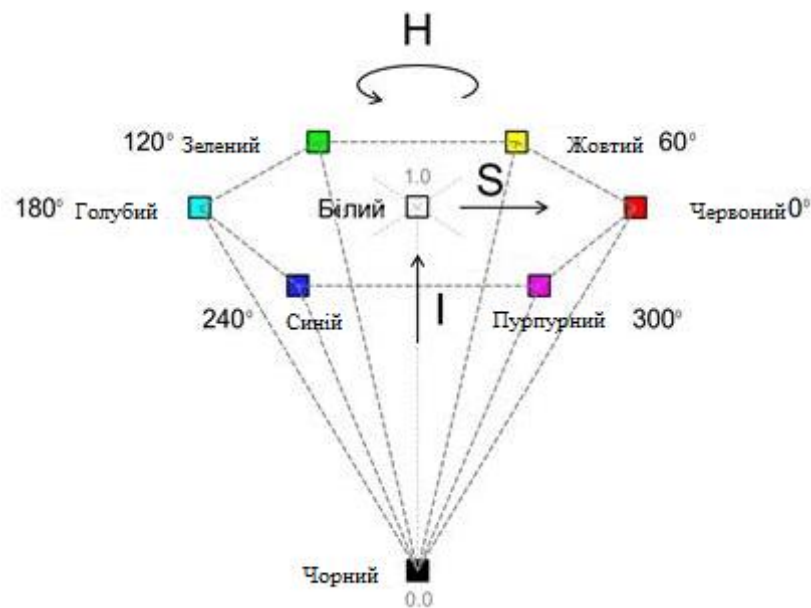


Рисунок 2.3 – Модель HSI

HSL, HSV, HIS часто використовуються в машинному зорі і аналізі зображень.

2.8. Інструменти пошуку збігів з еталоном в Labview

Зіставлення з еталоном це процес, при якому відбувається визначення еталонного зображення і збереження його в пам'яті. Техніка пошуку за шаблоном використовується для вимірювань, контролю виробництва, визначення положення об'єкта.

Середовище Labview надає широкі можливості для напрямку машинного зору, зокрема пошук збігів з еталоном. В Labview існують вбудовані функції пошуку збігів. Однією з них є «IMAQ Match Pattern 4» (рис.2.4). Дана функція шукає збіги з шаблоном на вхідному зображенні. Відмінною особливістю даної функції є необхідність попереднього завдання шаблону [14].

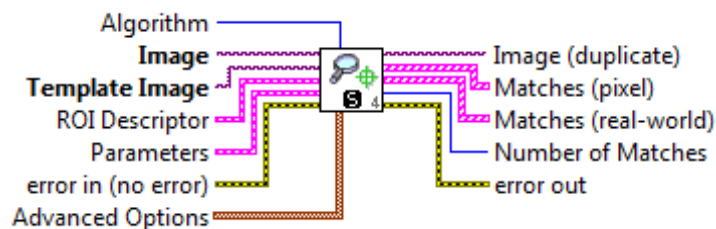


Рисунок 2.4 – IMAQ Match Pattern 4

У блок функції IMAQ Match Pattern входять наступні дані:

- Algorithm - вибір алгоритму пошуку збігів;
- Image - вхідне зображення;
- Template image - шаблонне зображення, на підставі якого буде проводиться пошук;
- ROI Descriptor - вибір області пошуку на вхідному зображенні;
- Parameters - параметри для контролю пошуку збігів;
- Error in- опис статусу помилки перед початком даної функції;
- Advanced Options - додаткові опції;
- Image (duplicate) - вихідне зображення;
- Matches (pixels) - масив збіглих кластерів, в піксельних координатах;
- Matches (real-world) - масив збіглих кластерів, в декартових координатах;

- Number of matches- число збігів;
- Error out - містить інформацію про помилки.

IMAQ Match Color Pattern – функція, що знаходить збіги з колірною шаблоном на вхідному зображенні (рис. 2.5) [15].

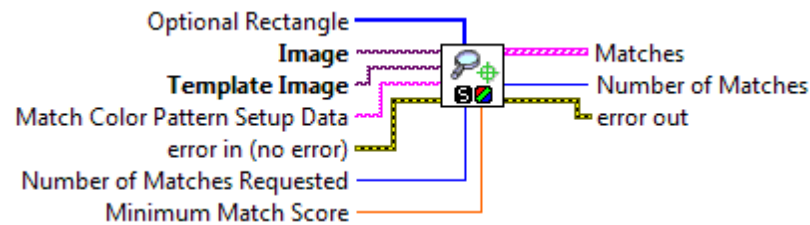


Рисунок 2.5 – IMAQ Match Color Pattern

Функція IMAQ Match Color Pattern включає в себе наступні дані:

- Optional Rectangle - визначає чотирьох-елементний масив, який складається з лівих, верхніх, правих і нижніх координат області процесингу;
- Image - вхідне зображення;
- Template Image - колірної шаблон;
- Match Color Pattern Setup Data - рядок містить інформацію, отриману з IMAQ Match Color Pattern;
- error in- містить інформацію про помилки;
- Number of Matches Requested - кількість можливих збігів;
- Minimum Match Score - мінімальне значення, при якому буде фіксуватися збіг;
- Matches - масив збіглих кластерів;
- Number of matches- число збігів;
- Error out - містить інформацію про помилки.

IMAQ Shape Match Tool – інструмент допомагає знаходити предмети на зображенні по заданих контурах (рис.2.6). Головною особливістю даного інструменту є те, що він здатний знаходити на зображенні об'єкти, схожі за обрисами на еталонні, але при цьому відрізняються від них за розміром [16].

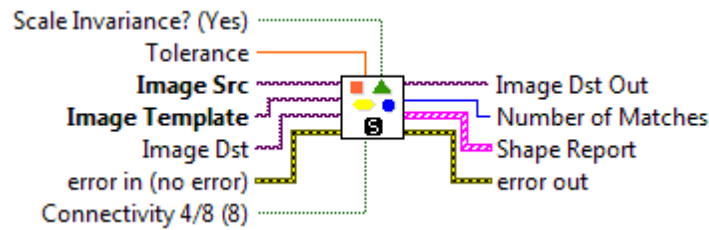


Рисунок 2.6 – Функція IMAQ Shape Match Tool

Вхідними даними функції IMAQ Shape Match Tool є:

- Scale Invariance? (Yes) - вказує, чи слід використовувати розмір або масштабну інваріантність при зіставленні об'єктів у вихідному зображенні для об'єкта шаблону;
- Tolerance - вказує на максимально допустиме відхилення між формою шаблону і подібних форм в зображенні;
- Image Src - посилання на вихідне зображення;
- Image Template - посилання на шаблон;
- Image Dst - посилання на бінарне зображення, яке містить об'єкти з Image Src які збігаються з Image Template;
- error in-містить інформацію про помилки;
- Connectivity 4/8 (8) - вказує тип з'єднання, що використовується алгоритмом для визначення часток;
- Image Dst Out - посилання на вихідне зображення;
- Number of Matches- число збігів;
- Shape Report - масив кластерів, в якому кожен кластер містить інформацію про яке співпало об'єкті;
- Error out - містить інформацію про помилки.

2.9. Інструменти реалізації стеження за об'єктом в середовищі Labview

Стеження за об'єктом має широке застосування в різних сферах науки, виробництва та життя. Так відстеження об'єктів застосовується в системах

оптичної локації, управління дорожнім і повітряним рухом, управління вбудованою побутовою технікою.

У середовищі Labview основним інструментом реалізації стеження за об'єктом (найбільш зрозумілим і простим) є Vision Assistant (рис.2.7).

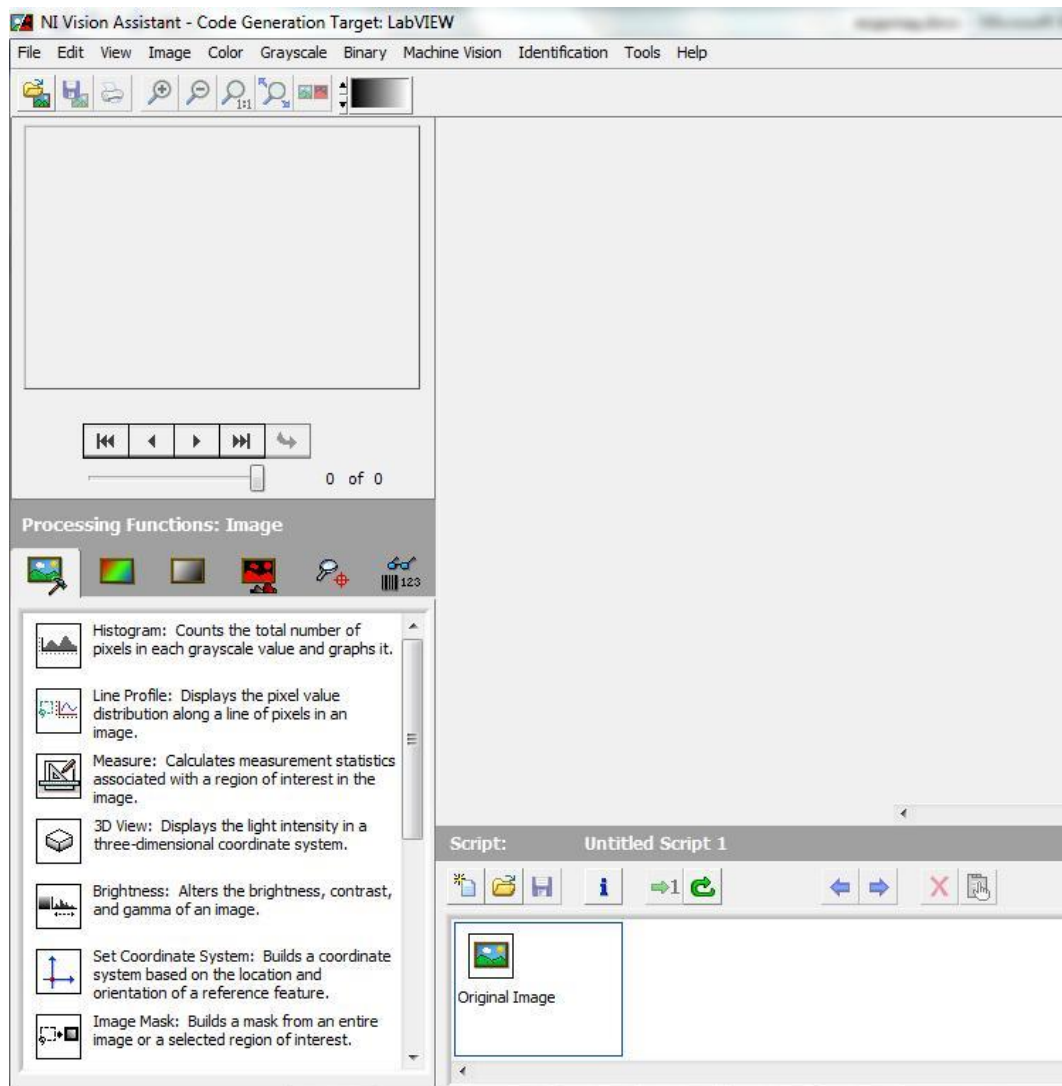


Рисунок 2.7 – Інтерфейс Vision Assistant

З його допомогою можна реалізувати різні варіації відстеження об'єкта. Так за допомогою інструментів Edge detector, Find Straight Edge, Adv.Straight Edge, Find Circular Edge можливо визначати межі об'єкта на чорно-білому зображенні. За допомогою інструментів Max Clamp, Clamp (Rake) можна визначати відстань між кордонами об'єкта (рис.2.8).

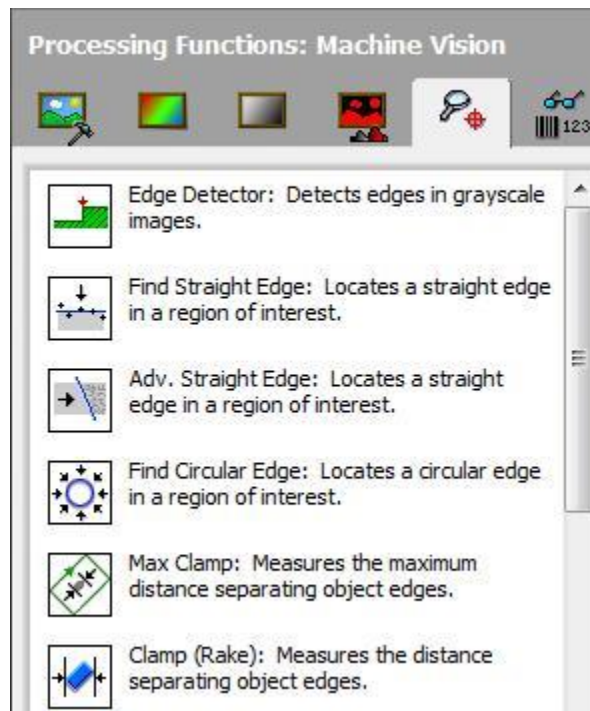


Рисунок 2.8 – Функції машинного зору

Крім того в Vision assistant реалізовані функції пошуку збігів (рис. 2.9). Інструмент Pattern matching дозволяє відстежувати об'єкт на відеопослідовності, який буде визначатися відповідністю із заздалегідь створеним шаблоном. Інший інструмент, який має такі ж можливості Geometric Matching, але його принцип роботи ґрунтується на пошуку збігів з геометричними фігурами. Object tracking дозволяє відстежувати об'єкти на групи послідовних зображень [16].

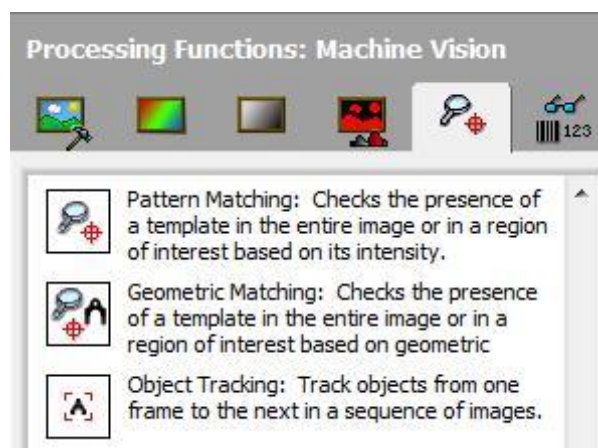


Рисунок 2.9 – Функції пошуку збігів

Також Vision Assistant дає можливість зчитування символної інформації (рис. 2.10).

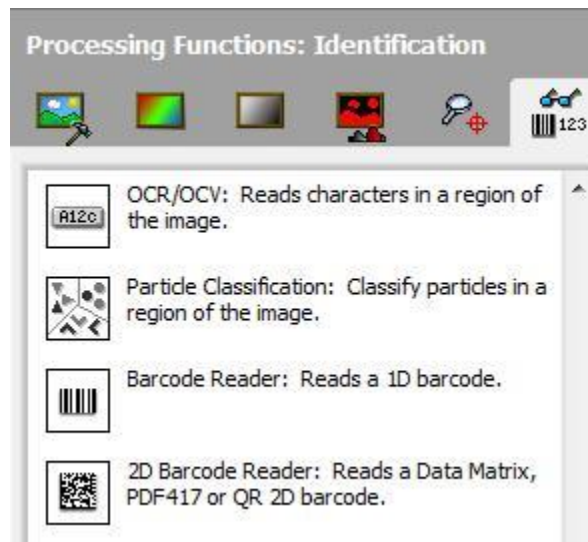


Рисунок 2.10 - Функції зчитування символної інформації

Так наприклад інструмент OCR дозволяє оцифровувати (розпізнавати) рукописний і друкований текст. Barcode Reader і 2D Barcode Reader дозволяють зчитувати штрих і QR коди, відповідно.

3 РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМИ

3.1 Можливості середовища моделювання Labview

Labview – програма компанії National Instruments, яка дозволяє імітувати середовище графічного моделювання та розробки додатків. Labview розшифровується як Laboratory Virtual Instrumentation Engineering Workbench. Середовище орієнтовано на вимірювання, збір даних і лабораторні дослідження [13]. Головна панель Labview зображена на рис. 3.1.

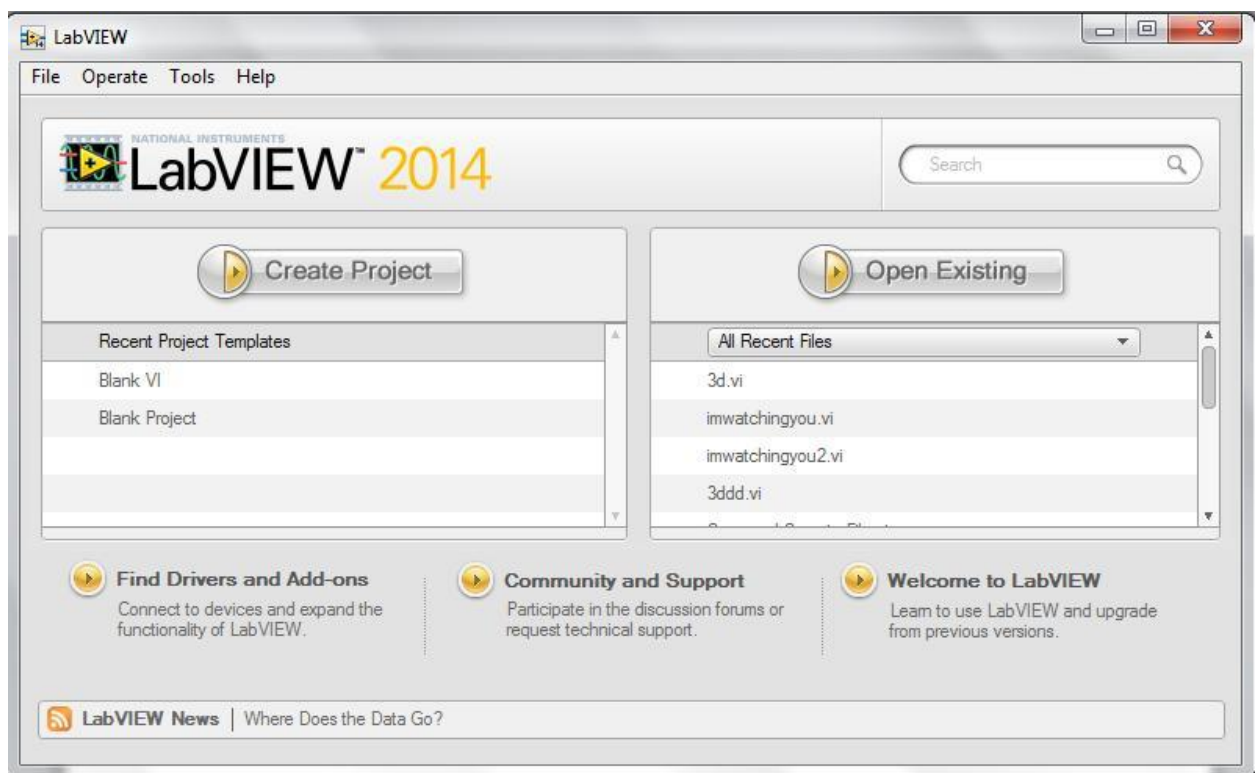


Рисунок 3.1 – Головна панель Labview

Програма LabVIEW є віртуальним приладом (Virtual Instrument) та складається з блок-діаграми (рис. 3.2), яка описує роботу віртуального приладу та фронт панелі (рис. 3. 3), яка описує інтерфейс віртуального приладу. При цьому віртуальні прилади можуть використовувати в якості складових частин для побудови інших віртуальних приладів [14].

Фронт панель віртуального приладу містить засоби введення-виведення: кнопки, перемикачі, світлодіоди, шкали, інформаційні табло та ін.

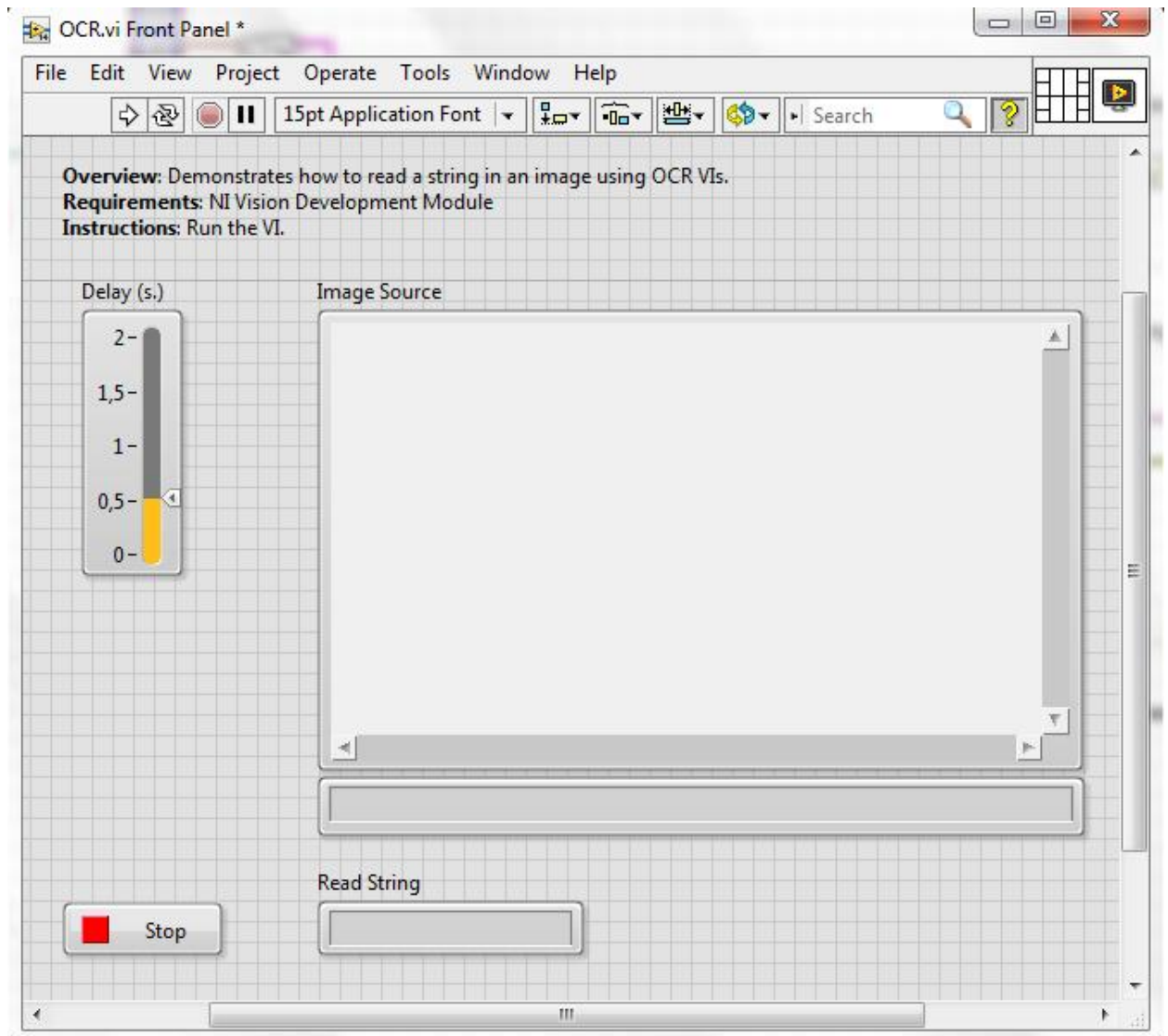


Рисунок 3.2 – Фронт панель LabVIEW

Програма побудована в середовищі LabVIEW представляє собою набір функціональних вузлів, які з'єднані провідниками для передачі даних. Джерелом даних програми можуть бути вузли (nodes) підпрограм або термінали (terminals) елементів управління на фронт панелі, а приймачами - термінали індикаторів, Sub VI виведення даних або їх дані збереженні в файлах [15]. Дана структура програми дозволяє реалізувати модель потоку даних, які потрапляють в програму через вказані вузли, обробляються та відправляються на вузли індикації. При цьому потоки, не пов'язані між собою загальними даними та обробляються паралельно [2].

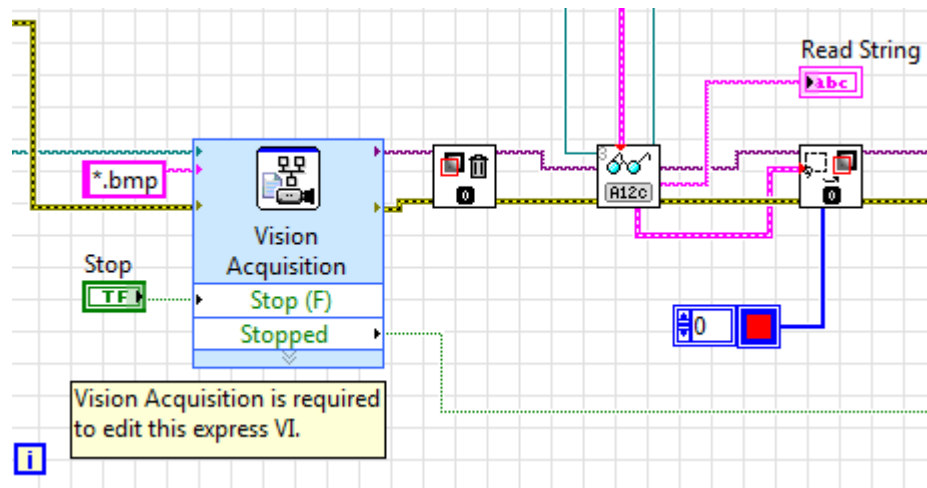


Рисунок 3.3 – Фрагмент блок-діаграми

3.2 Обробка, аналіз зображень та відео в середовищі Labview

Крім універсальних засобів розробки комп'ютерних контрольно-вимірювальних систем, National Instrument надає широкий спектр спеціалізованих програмних бібліотек і апаратних модулів. Однією з таких апаратно-програмних технологій є платформа машинного зору NI VISION, що складається з технології захоплення зображення Image Acquisition (IMAQ) і технології програмної обробки та аналізу зображення [16].

Модуль IMAQ складається з модулів захоплення зображення з великої кількості поширених джерел відеосигналів, а саме аналогових та цифрових камер різних конфігурацій і стандартів. Дані модулі спроектовані так, що їх функціями можна керувати програмно, що полегшує збір зображень з різних джерел [13].

Це дозволяє здійснювати роботу з зображеннями різного спектрального діапазону, від рентгенівського до інфрачервоного, з різною глибиною відцифрування, від 8 до 32 розрядів, з різною швидкістю імпортування - від одиничного зображення до десятків тисяч кадрів в секунду. Також для поліпшення роботи з відео і зображеннями створений модуль NI Vision Assistant. Його використання дозволяє проводити різний

набір операцій над зображеннями. Починаючи від роботи з кольором, закінчуючи розпізнаванням символів і об'єктів.

3.3 Функціональна схема досліджуваного алгоритму

Функціональна схема досліджуваного алгоритму пошуку збігів на зображенні представлена на рис. 3.4. Розглянемо блоки, що входять до неї та їх принцип роботи.



Рисунок 3.4 – Функціональна схема алгоритму пошуку збігів

Блок отримання зображення. Процес отримання зображення полягає в підключенні USB-камери, а також налаштування необхідних параметрів.

Блок перетворення зображення в чорно-біле. Перетворення зображення в чорно-біле (grayscale) відбувається з використанням моделі HSI, шляхом вилучення компоненти інтенсивності.

Блок створення шаблону. Виробляється виділення на чорно-білому зображенні області з об'єктом, який цікавить; виконується його виділення і збереження.

Блок вибору алгоритму пошуку. Даний блок реалізує вибір одного з алгоритмів пошуку об'єкта на зображенні. Вибір може варіюватися в залежності від умов спостереження.

Блок створення масиву для пошуку за шаблоном. Процес створення масиву для пошуку за шаблоном реалізується за допомогою перетворення вхідного зображення в масив значень яскравості і порівняння його з масивом, створеним за таким же принципом з шаблонного зображення.

Блок виведення зображення з маркером стеження.

3.4 Вимірювання відстані до об'єктів

Вимірювання відстані до об'єктів знаходить застосування в різних сферах життя, починаючи від астрометрії, систем наведення і закінчуючи забезпеченням безпеки дорожнього руху. На даний момент існує кілька способів визначення відстані до об'єкта, що розрізняються між собою середовищем застосування (статичні зображення і відеопослідовності) і процесом отримання зображення. Для вимірювання на статичному зображенні підходить використання однієї камери і мірного об'єкта. Описана система використовує зображення, що отримується з відеокамери, на якому крім об'єкта присутній мірна лінійка. Лінійка повинна розташовуватися строго посередині зображення. Для калібрування необхідно зіставити координати об'єкта на зображенні з відміткою на лінійці. Так само необхідно знати розмір лінійки в пікселях і в метрах, а також координату пікселя, що є фактичною серединою лінійки.

Замість лінійки можна використовувати маркувальний конус, розташований в центрі вимірюваної області.

Для розрахунку відстані використовується формула:

$$B = \frac{LK}{\frac{W}{X} - 1 + K}, \quad (3.1)$$

де B – шукана відстань;

L – довжина лінійки;

W – довжина лінійки в пікселях;

X – координата об'єкта на зображенні;

$K = (W/M) - 1$ - коефіцієнт, що відображає нахил камери;

M – координата середини лінійки.

Даний метод може бути реалізований на мові Python з використанням бібліотеки OpenCV [14].

Іншим варіантом вимірювання розмірів об'єкту та відстані до нього є використання оптичних систем. При використанні оптичних систем використовують закони оптики для отримання зображень різних масштабів для одного і того ж об'єкта. Один з методів вимірювання розмірів і положення об'єкта полягає в послідовному наближенні та видаленні лінзи або екрану, в заданому діапазоні Dx , відносно один одного. Де H – висота об'єкта, h – висота об'єкта на екрані.

В даному випадку відстань визначається за такою формулою:

$$B_i = \frac{d_i F}{d_i - F}, \quad (3.2)$$

де B_i – відстань від об'єкта до лінзи i -го зображення масиву;

F – фокусна відстань лінзи;

d_i - відстань від лінзи до зображення i -го кадру масиву.

Також можна використовувати більш складну систему, що складається з двох лінз. В такому випадку відстань визначається шляхом послідовного застосування формули 3.2 до кожної лінзи.

Крім описаних вище способів існує ще один, суть якого полягає у визначенні відстані до об'єкта в дискретні моменти часу за допомогою оптичного фотоприймача, що має переналаштовану оптичну систему з двома відомими граничними фокусними відстанями. Де a - відстань від передньої головної точки H , до осьової точки A предмета y . В результаті роботи даної системи отримують два зображення, різного масштабу і обчислюють відстань до об'єкта за такою формулою:

$$a = \frac{1 - \frac{y_1}{y_2}}{\frac{y_1}{y_2} \frac{f_2}{f_1} - 1}, \quad (3.3)$$

де f_1, f_2 – граничні фокусні відстані оптичної системи;

y_1, y_2 – розміри зображень об'єкта при f_1, f_2 .

Існують алгоритми, які дозволяють визначати відстань до об'єкта за допомогою стереопари. Стереопара – це пара плоских зображень одного і того ж сюжету, що має відмінності між зображеннями, створювані для отримання ефекту обсягу [13]. Даний ефект виникає внаслідок паралакса. Так один з алгоритмів використовує дві камери для побудови z-зображення або зображення глибини – це одноканальне зображення кожен піксель, якого відповідає дистанції до об'єкта. Даний підхід ґрунтується на тому, що для кожної точки на зображенні відбувається пошук парної їй точки на іншому зображенні.

По парі точок виконується триангуляція і визначається координата їх прообразу в тривимірному просторі [15]. Далі обчислюється глибина, як відстань до площини камери. Для кожного пікселя лівого зображення з координатами (x_0, y_0) проводиться пошук пікселя на правому зображенні.

Піксель на правому зображенні має координати $(x_0 - d, y_0)$, де d – величина зміщення камер, відносно один одного [15].

Інший алгоритм вимірювання відстані до об'єкта, ґрунтується на принципах фотограмметрії та кореляційній обробці цифрових зображень. Вимірювання дальності до об'єкта полягає в отриманні зображень об'єктів з різних ракурсів.

Коли відстань між камерою та сценою, яка спостерігається, істотно довшою за фокусну відстань оптичної системи, вважається, що зображення будується в її фокальній площині. У проєктованій моделі камери зображення тривимірного об'єкта виходить його проєктування в фокальній площині [15].

Дальність до об'єкта обчислюють за формулою:

$$Z = \frac{fL}{x_1 - x_2}, \quad (3.4)$$

де f – фокусна відстань;

L – величина бази (відстань між оптичними осями камер);

x_1 та x_2 – координата об'єкта в площині зображення на правому і лівому кадрах стереопари, а їх різниця – зміщення об'єкта.

Даний метод забезпечує високу точність вимірювань, так як в ньому відсутня механічна складова, що дозволяє досить швидко розраховувати відстань до об'єкта і інші параметри сцени

4. КЕРІВНИЦТВО ПРОГРАМІСТА

4.1 Розробка програмної реалізації системи обробки даних в Labview

Загальна схема даної програми приведена на рис.4.1.

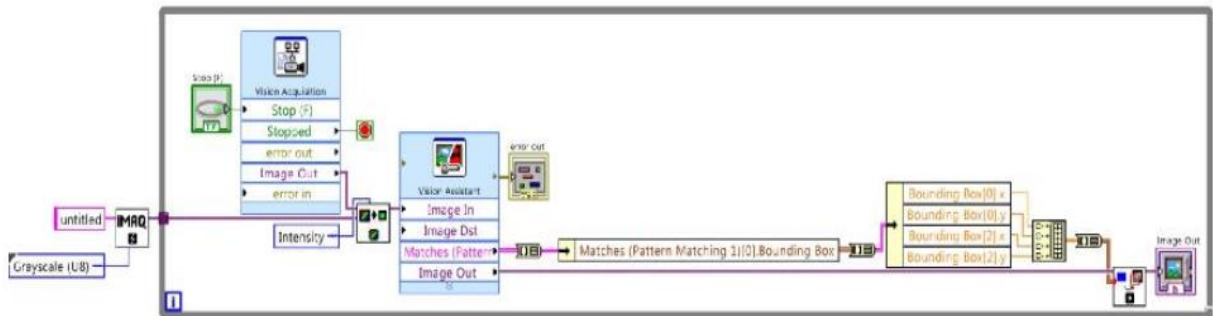


Рисунок 4.1 – Загальна схема програми

Для побудови програми в Labview використовуються вбудовані функції. Вони можуть бути знайдені шляхом натискання правої кнопки миші на панелі блок-діаграми (рис. 4.2).

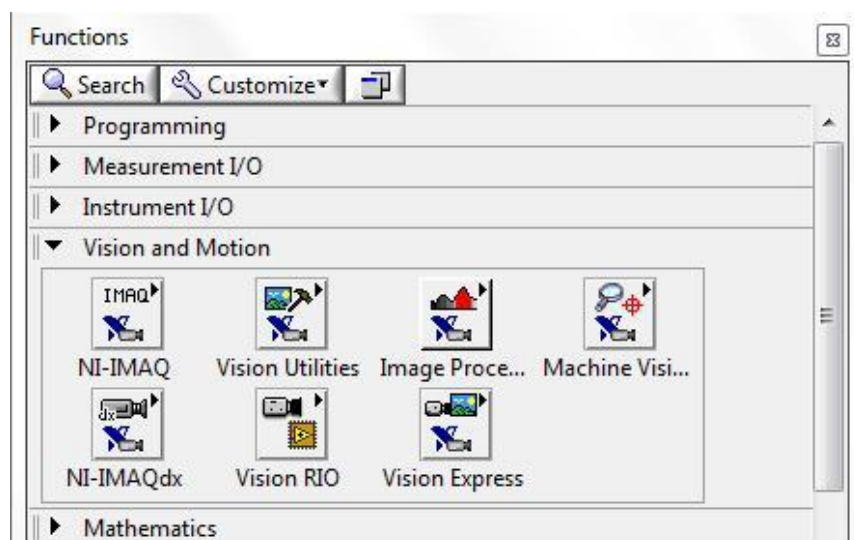


Рисунок 4.2 – Отримання зображення

На рис. 4.3 представлений блок Vision Acquisition, за допомогою якого відбувається отримання зображення.

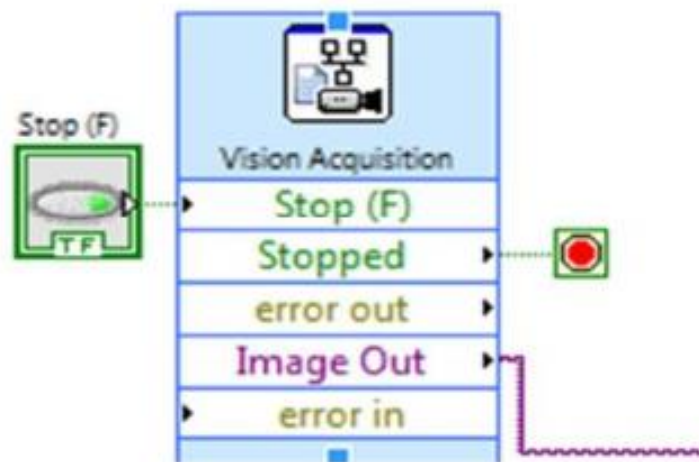


Рисунок 4.3 – Блок Vision Acquisition

Блок Vision Acquisition розташований Vision and motion - Vision Express - Vision Acquisition. За допомогою даного блоку можна управляти USB - пристроями (рис. 4.4) і налаштуваннями USB-камери (рис. 4.5). Для управління налаштуваннями необхідно двічі клацнути по іконці блоку Vision Acquisition.

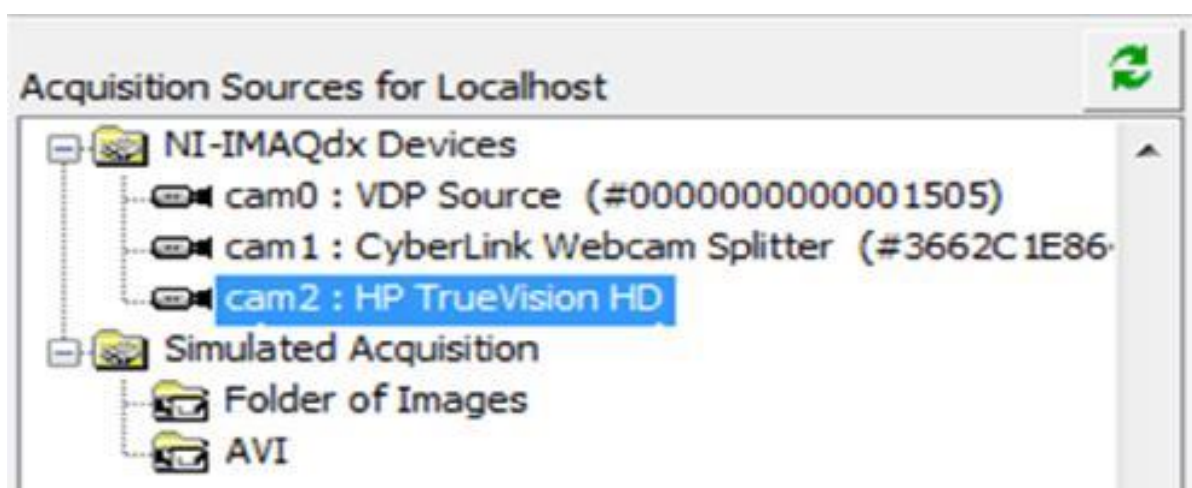


Рисунок 4.4 – Вибір пристрою отримання зображення

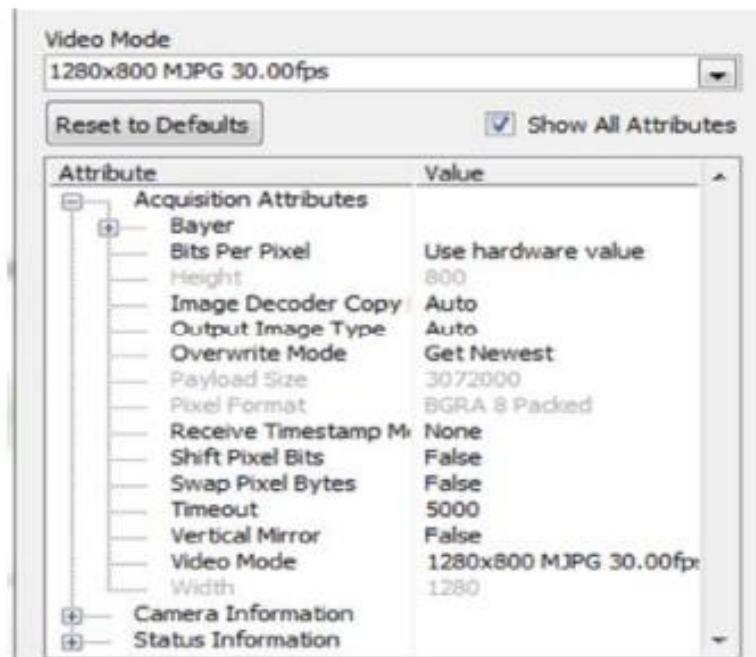


Рисунок 4.5 – Перетворення зображення в чорно-біле

Перетворення зображення в чорно-біле проводиться шляхом, вилучення компоненти інтенсивності з вхідного зображення. Фрагмент програми, що відповідає за перетворення, показаний на рис. 4.6

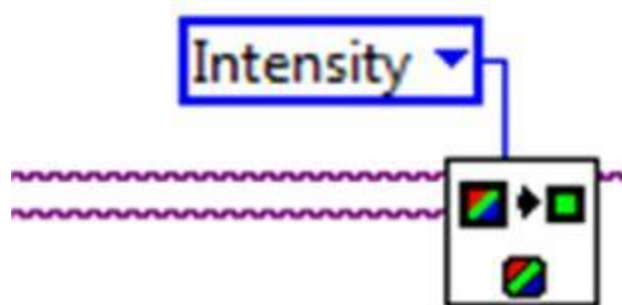


Рисунок 4.6 – Витяг компоненти інтенсивності

У середовищі Labview для вилучення будь-якої компоненти з зображення використовується блок IMAQ ExtractSingleColorPlane VI (рис. 4.7). До входу «Color plane», підключається константа, що відповідає за видобуту колірну компоненту. Даний блок розташований Vision and motion - Vision utilities - Color utilities - ExtractSingleColorPlane VI.

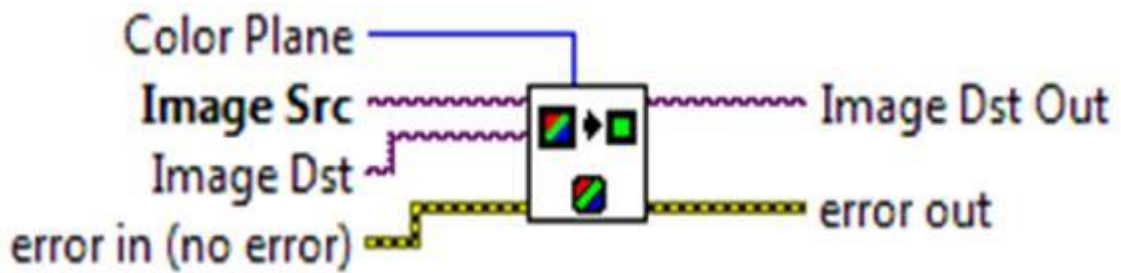


Рисунок 4.7 – Блок IMAQ ExtractSingleColorPlane VI

4.2 Створення шаблону

Створення шаблону проводиться за допомогою блоку Vision Assistant (рис. 4.8), який представляє собою додаткову програму, яка дозволяє істотно спростити процес обробки, перетворення і роботи з зображеннями. Блок Vision Assistant розташований Vision and motion - Vision Express - Vision Assistant.



Рисунок 4.8 – Блок Vision Assistant

Для створення шаблону необхідно двічі клацнути по даному блоку, у вікні, перейти у вкладку Processing Functions: Machine Vision, де вибрати пункт Pattern Matching (рис. 4.9).

У меню, надається можливість створити новий шаблон або використовувати заздалегідь заготовлений (рис. 4.9).



Pattern Matching: Checks the presence of a template in the entire image or in a region of interest based on its intensity.

Рисунок 4.9 – Вкладка Pattern Matching

У меню, надається можливість створити новий шаблон або використовувати заздалегідь заготовлений (рис. 4.10).

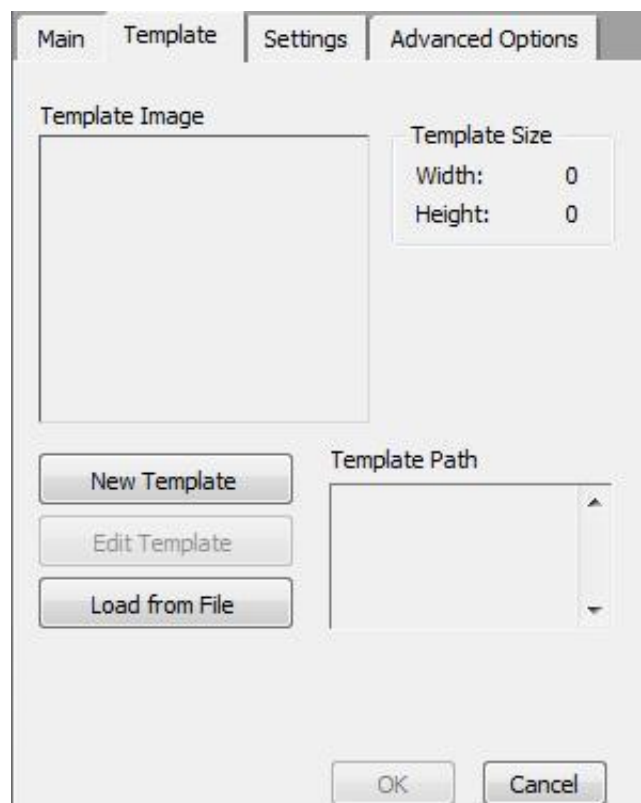


Рисунок 4.10 – Меню вибору шаблону

4.3 Вибір алгоритму пошуку

Вибір алгоритму виробляється в блоці Vision Assistant. Для вибору необхідно відкрити даний блок і вибрати в правому меню Pattern matching, потім перейти у вкладку settings і в підміню Algorithm вибрати потрібний алгоритм. А так само різні настройки пошуку такі як: число збігів, мінімальний рівень збігів, максимальний рівень «піраміди» (рис. 4.11).

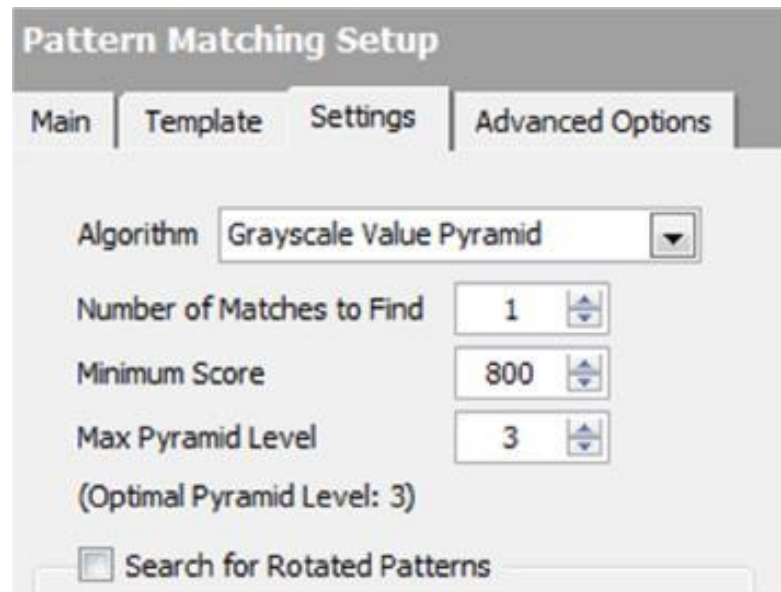


Рисунок 4.11 – Налаштування пошуку збігів

4.4 Створення масиву для пошуку за шаблоном

Процес створення масиву полягає в тому, що проводиться порівняння з шаблоном, збіги фіксуються і записуються в кластер, для подальшого використання. Щоб виділити потрібну область, необхідна частина кластера перетворюється в масив. Фрагмент програми, що реалізовує створення масиву для пошуку за шаблоном, зображений на рис. 4.12

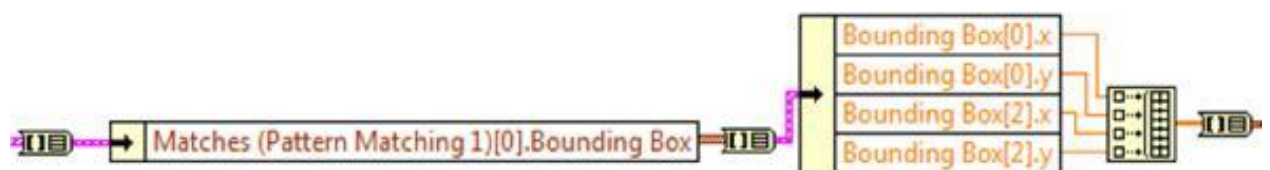


Рисунок 4.12 – Створення масиву для пошуку за шаблоном

Функції для роботи з масивами розташовані у вкладці «Cluster, class & Variant» і «Array»:

- Cluster, class & variant - array to cluster;

- Cluster, class & variant - unbundle by name;
- Array - build array.

4.5. Вилучення зображення з маркером стеження

Вилучення зображення з маркером стеження реалізується за допомогою вбудованої функції IMAQ Overlay Rectangle VI, зображеної на рис. 4.13.

Даний блок розташований Vision and motion - Vision Utilites - Overlay – IMAQ Overlay Rectangle VI.

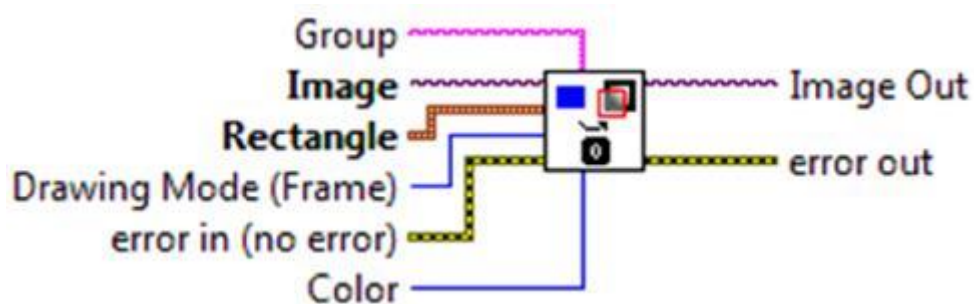


Рисунок 4.13 – Блок IMAQ Overlay Rectangle VI

Ця функція використовує створений раніше масив, для накладення квадрата на об'єкт спостереження. З масиву беруться координати на вхідному зображенні, які відповідають об'єкту стеження. Також використовується функція Image Out для створення форми на лицьовій панелі для виведення кінцевого зображення. Даний блок створюється шляхом натискання правою кнопкою на вихід Image out блоку Vision Acquisition і вибору у вкладці create пункту indicator.

Фрагмент програми, що відповідає за виведення зображення, показаний на рис.4.14

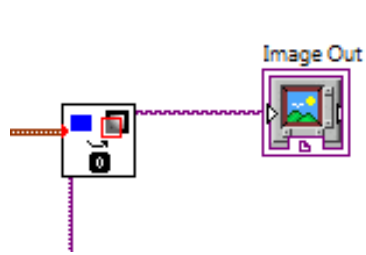


Рисунок 4.14 – Висновок зображення з маркером стеження

5 КЕРІВНИЦТВО КОРИСТУВАЧА

5.1 Запуск програми Labview

Запуск програми обробки даних проводиться кнопкою «Run», показаної на рис. 5.1. Далі з'являється інтерфейс програми, наведений на рис.5.2.



Рисунок 5.1 – Кнопка «Run»

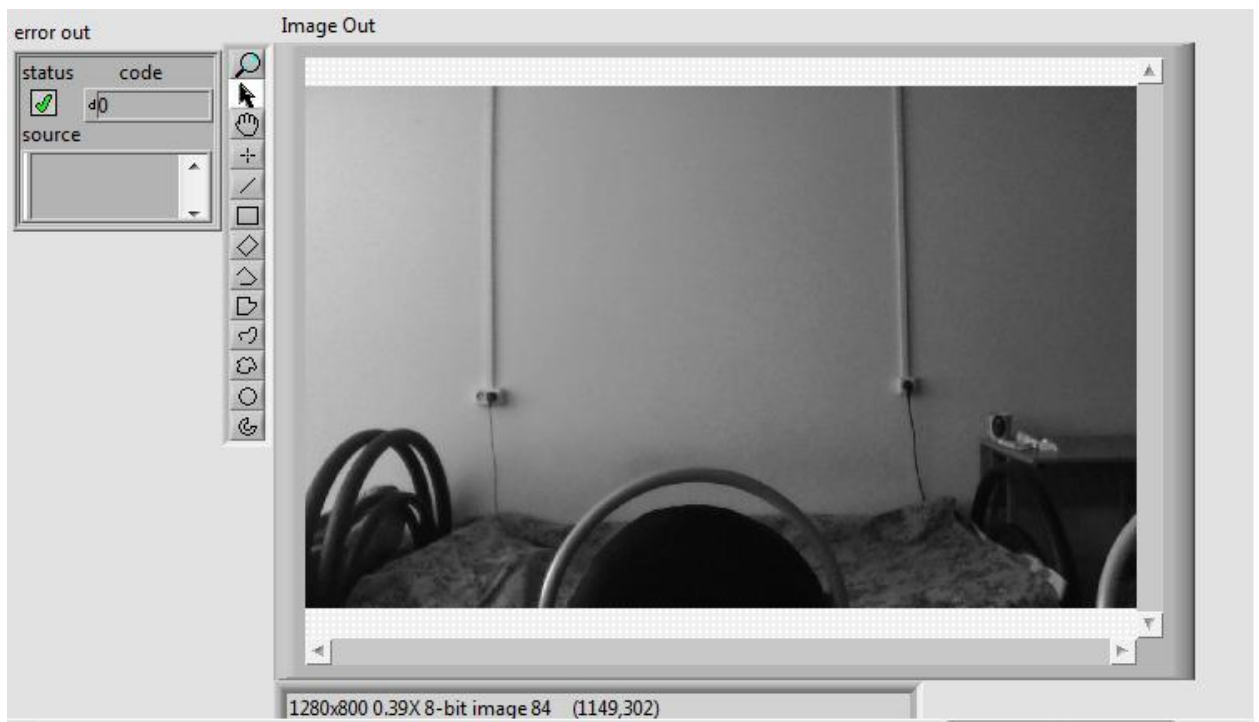


Рисунок 5.2 – Інтерфейс програми

Під час першого запуску програми необхідно, щоб в полі зору камери знаходився об'єкт, який в подальшому буде використовуватися як еталонний. Для роботи програми необхідно створити або завантажити заздалегідь заготовлений шаблон.

5.2 Створення шаблону

Створення шаблону відбувається наступним чином:

- подвійним клацанням необхідно відкрити блок vision assistant;
- вибрати в полі Script - pattern matching (рис.5.3);
- в розпочатому зліва меню «Pattern matching settings» натиснути кнопку «New» (рис.5.4);
- у вікні виділити область, що містить об'єкт трекінгу, і натиснути кнопку «Next» (рис.5.5);
- наступне вікно пропонує виділити область, що не є об'єктом стеження (рис. 5.6);
- потім необхідно вибрати центр об'єкта клацанням лівої кнопки миші;
- натиснути клавішу «next», після відкриється діалогове вікно, що пропонує зберегти файл шаблону, для подальшого використання;
- для закінчення створення шаблону необхідно натиснути кнопку «ok» в підміню «Pattern matching setting».



Рисунок 5.3 – Script - pattern matching

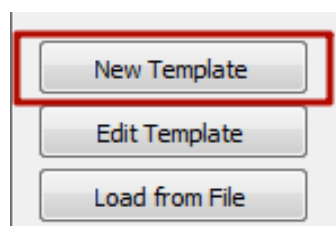


Рисунок 5.4 – Pattern matching settings

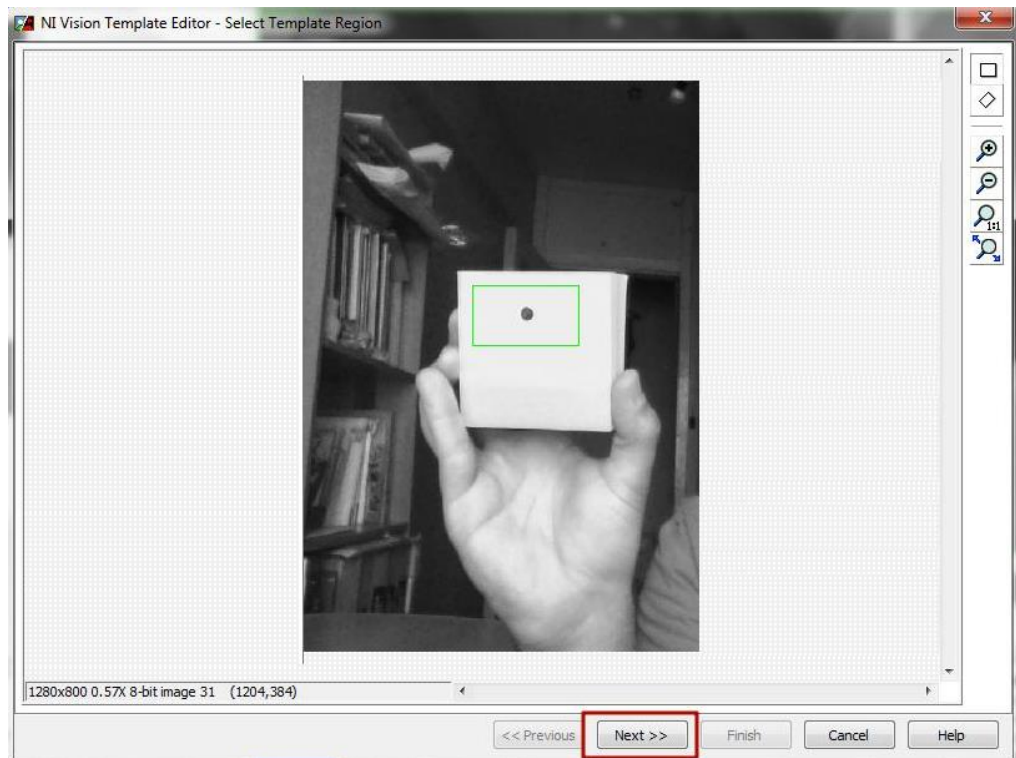


Рисунок 5.5 – Вибір області містить об'єкт

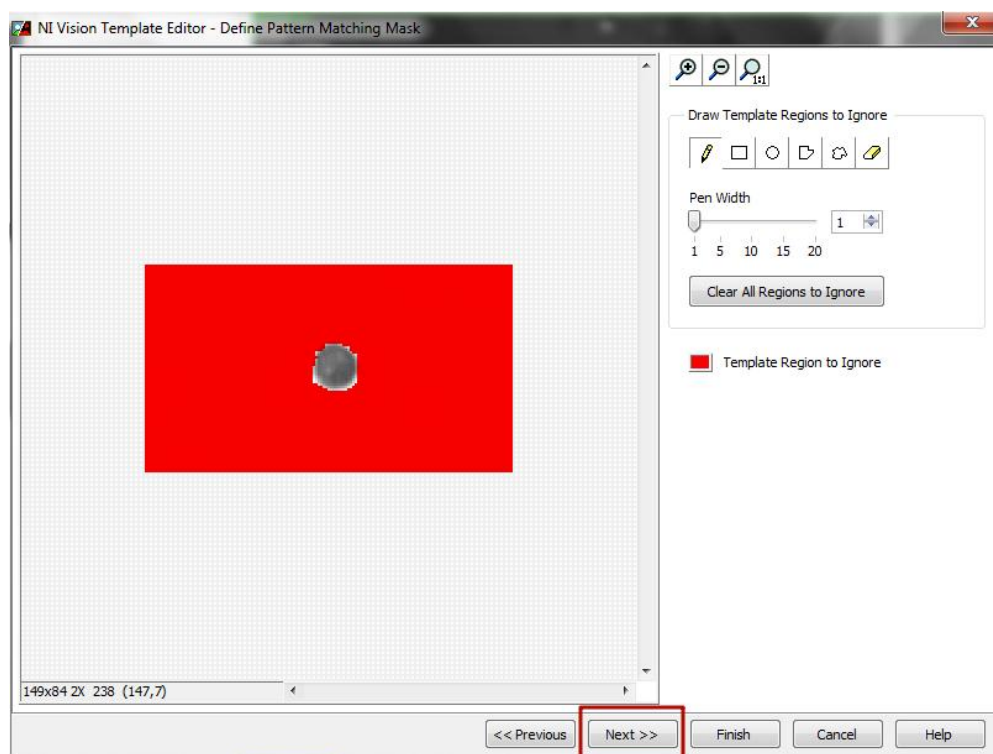


Рисунок 5.6 – Виділення об'єкта

В результаті роботи програми відбувається пошук та виділення необхідного об'єкта, як показано на рис. 5.7.

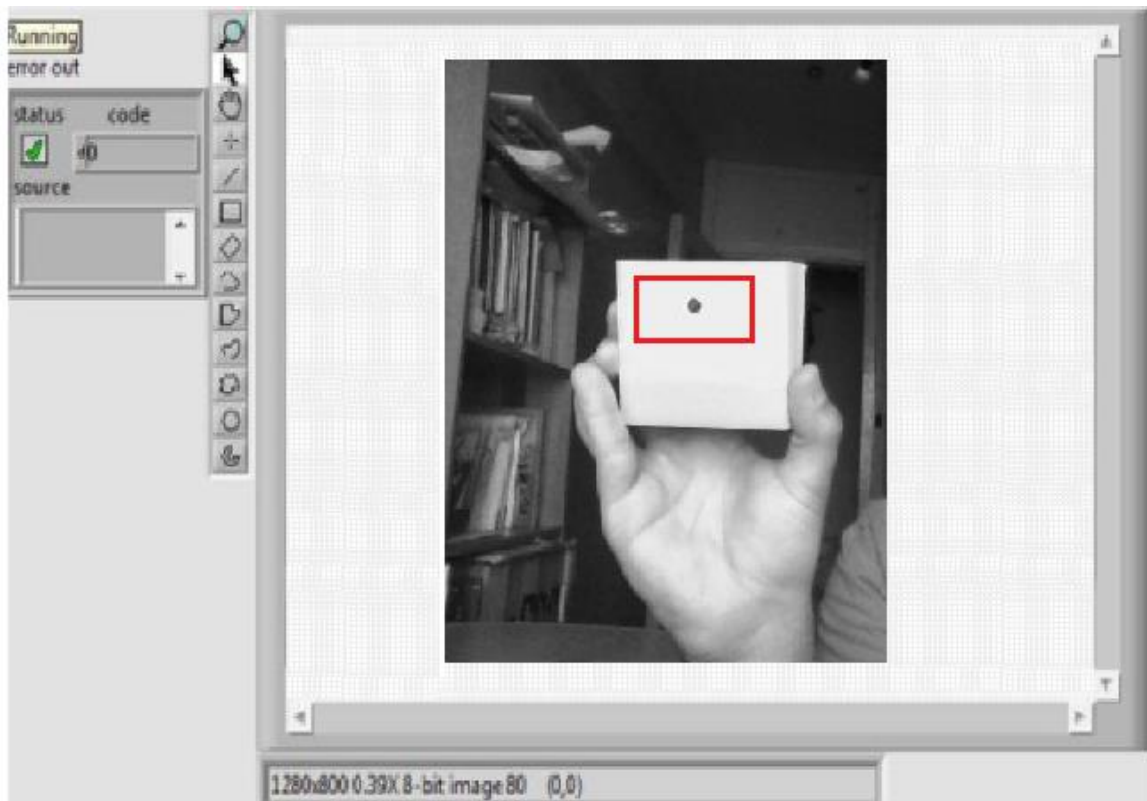


Рисунок 5.7 – Результат роботи програми

Для зупинки роботи програми необхідно натиснути кнопку «Stop» (рис. 5.8), розташовану на лицьовій панелі програми. Дана кнопка створена за допомогою інструментів «True constant» і «False constant», розташованих у вкладці Boolean.

Так само в інтерфейсі програми присутній статус наявності помилок (рис. 5.9), який виводить код помилки, що не дозволяє коректно працювати програмі. Для створення даного індикатора необхідно натиснути правою кнопкою миші на вихід «error out» блоку Vision Assistant, вибрати вкладку «create» і натиснути на панель «indicator».



Рисунок 5.8 – Кнопка «Стоп»

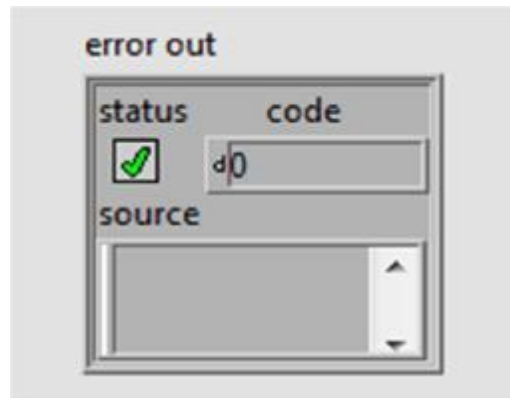


Рисунок 5.9 – Статус наявності помилок

В результаті виконання даної програми користувач може ознайомитися з алгоритмами обробки зображень та пошуку об'єкта на відеозображенні. Вбудовані алгоритми дозволяють відстежувати переміщення обраного об'єкта в реальному часі. Дані алгоритми працюють з підключеними до комп'ютера USB-камерами, або з вбудованою в ноутбук.

ВИСНОВКИ

Останнім часом рішення більшості завдань з автоматизації обробки даних різних процесів зводиться до використання підключених або вбудованих в персональний комп'ютер цифрових, плат вводу / виводу аналогових/цифрових даних, осцилографів, генераторів сигналів і інших пристроїв. У даному випадку проблема створення програмного забезпечення з необхідною функціональністю, яке взаємодіє з апаратними засобами та забезпечує обробку даних, виходить на перший план.

Створена програмна реалізація дає змогу проводити обробку графічних даних отриманих з відео та визначати задані елементи. Для досягнення мети були використані наступні інструменти: функція пошуку збігів. IMAQ Match Pattern 4, функція, що знаходить збіги з колірною шаблоном на вхідному зображенні IMAQ Match Color Pattern, інструменти Edge detector, Find Straight Edge, Adv.Straight Edge, Find Circular Edge.

Обробка проходить в наступній послідовності: спочатку необхідно отримати зображення, наступним кроком виникла необхідність перетворення зображення в чорно-біле, далі створення шаблону та вибір алгоритму пошуку, далі створення масиву для пошуку за шаблоном та виведення зображення з маркером стеження.

Для досягнення даного результату були виконані наступні завдання:

- було проведено аналіз та класифікацію технічних засобів обробки даних;
- проаналізовано методів для обробки зображень;
- проведено аналіз алгоритмів пошуку;
- розроблено програмну реалізацію системи обробки даних з використанням середовища графічного програмування Labview.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Рейнхард К. Компьютерное зрение. Теория и алгоритмы [Текст] / К. Рейнхард. – М.: ДМК Пресс, 2019. – 408 с.
2. Основные понятия режимов компьютерной обработки данных [Электрон. ресурс]. – Режим доступа: <http://bookash.pro/ru/t>.
3. Осипенко, Н. Б. Пакеты анализа и обработки данных: тексты лекций [Текст] / Н. Б. Осипенко. – Гомель: ГГУ им. Ф. Скорины, 2018. – 99 с.
4. Осипенко, Н. Б. Программные средства первичной статистической обработки экспериментальных данных: практическое руководство [Текст] / Н. Б. Осипенко, А. Н. Осипенко; М-во образования РБ, Гом. гос. ун-т им. Ф. Скорины. – Гомель: ГГУ им. Ф. Скорины, 2015. – 43 с. ISBN 978-985-439-991-1
5. Шепелев, К.В. Анализ режимов автоматизированной обработки данных [Текст] / К.В. Шепелев, Суркова Н.Е. Шувалова И.С. // Промышленные АСУ и контроллеры. – 2019. – № 12. – С. 48–53.
6. Колетвинов, Д. С. Обзор научных методов сбора и обработки информации [Текст] / Д. С. Колетвинов, А. А. Мельникова, К. В. Борзилов. — 2020. — № 3 (293). – С. 157–159.
7. Цветков, А. А Алгоритмы распознавания объектов [Текст] / А. А. Цветков, Д. К. Шорох, М. Г. Зубарева. // Технические науки: проблемы и перспективы: материалы IV Междунар. науч. конф. – 2016. – С. 20–28.
8. Исаев, А. Л. Способы классификации движущихся объектов на видео [Текст] / А. Л. Исаев, Д. А. Газаров, С. Д. Евсеев. –2016. – № 18 (122). – С. 44–47.
9. Черногорова, Ю. В. Методы распознавания образов [Текст] / Ю. В. Черногорова // Молодой ученый. – 2016. – № 28 (132). – С. 40–43.
10. Живрин, Я. Э. Методы определения объектов на изображении [Текст] / Я. Э. Живрин, Нафе Башар Алкзир // Молодой ученый. – 2018. – № 7 (193). – С. 8–19.

11. Боровик, И. Г. Обнаружение и ведение движущихся объектов на примере спортивного мероприятия [Текст] / И. Г. Боровик, Д. К. Шорох // Молодой ученый. – 2016. – № 6 (110). – С. 75–83.
12. Романов, С. А. Анализ методов сегментации изображений / С. А. Романов, О. М. Лепешкин, Ю. П. Стоянов. [Текст] // Молодой ученый. – 2010. – № 6 (17). – С. 26–28.
13. Кузин, Д. А. Создание модульной системы умного дома на базе аппаратного комплекса National Instruments [Текст] / Д. А. Кузин // Молодой ученый. – 2016. – № 19 (123). – С. 87–90.
14. Кожанов, Р. В. Модернизированная виртуальная лаборатория «Комбинационные цифровые устройства. Версия 2» [Текст] / Р. В. Кожанов, К. Ю. Котова, Е. Р. Кожанова // Исследования молодых ученых: материалы IV Междунар. науч. конф. – 2019. – С. 5–8.
15. Магеррамов, Р. В. Использование контрольно-диагностических стендов для тестирования микросхем [Текст] / Р. В. Магеррамов. // Молодой ученый. – 2016. – № 17 (121). – С. 53–57.
16. Визильтер, Ю.В. Обработка и анализ цифровых изображений с примерами на LabVIEW IMAQ Vision [Текст] / Ю.В. Визильтер, С.Ю. Желтов, В.А. Князь, А.Н. Ходарев, А.В. Моржин. – М.: ДМК-Пресс, 2009. – 465 с.

ДОДАТОК А
Текст програми

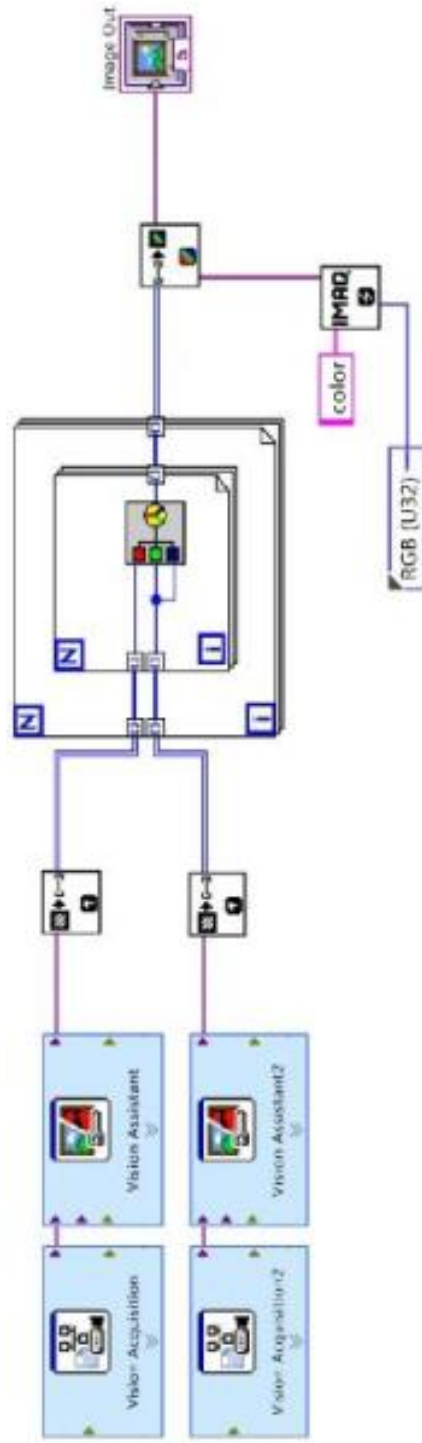


Рисунок А.1 - Отримання зображення

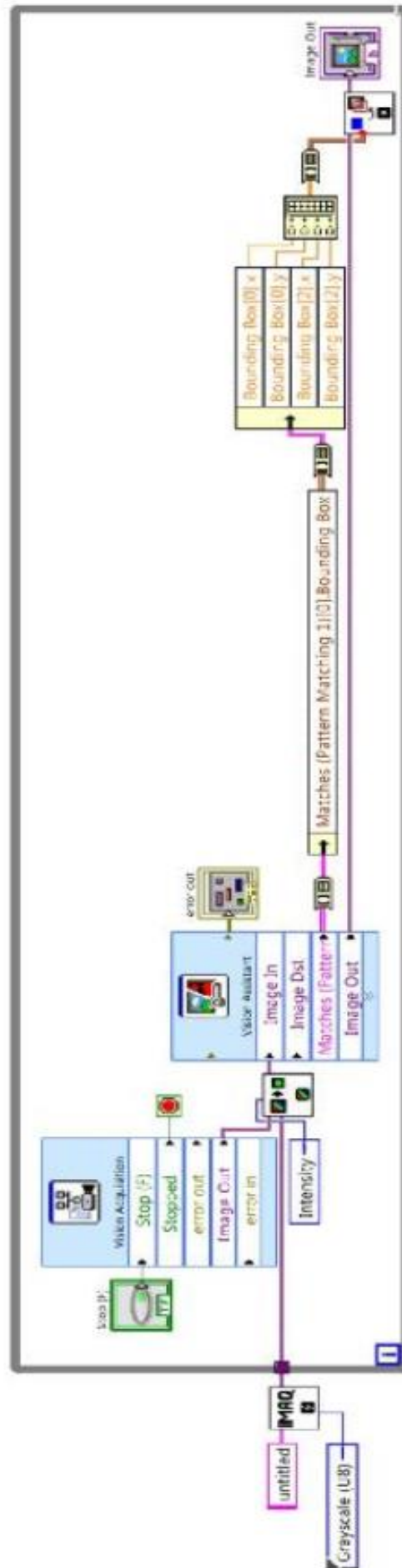


Рисунок А.2 – Обработка изображения

ДОДАТОК Б
Слайди презентації

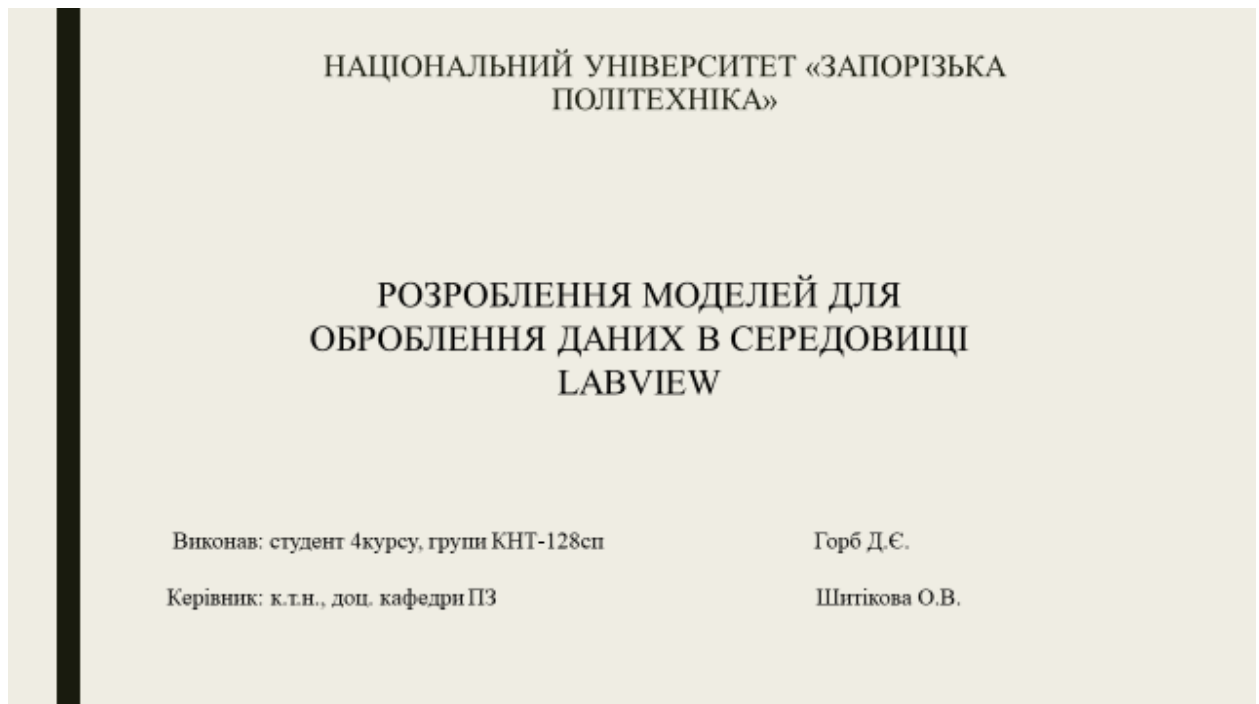


Рисунок Б.1 – Слайд №1

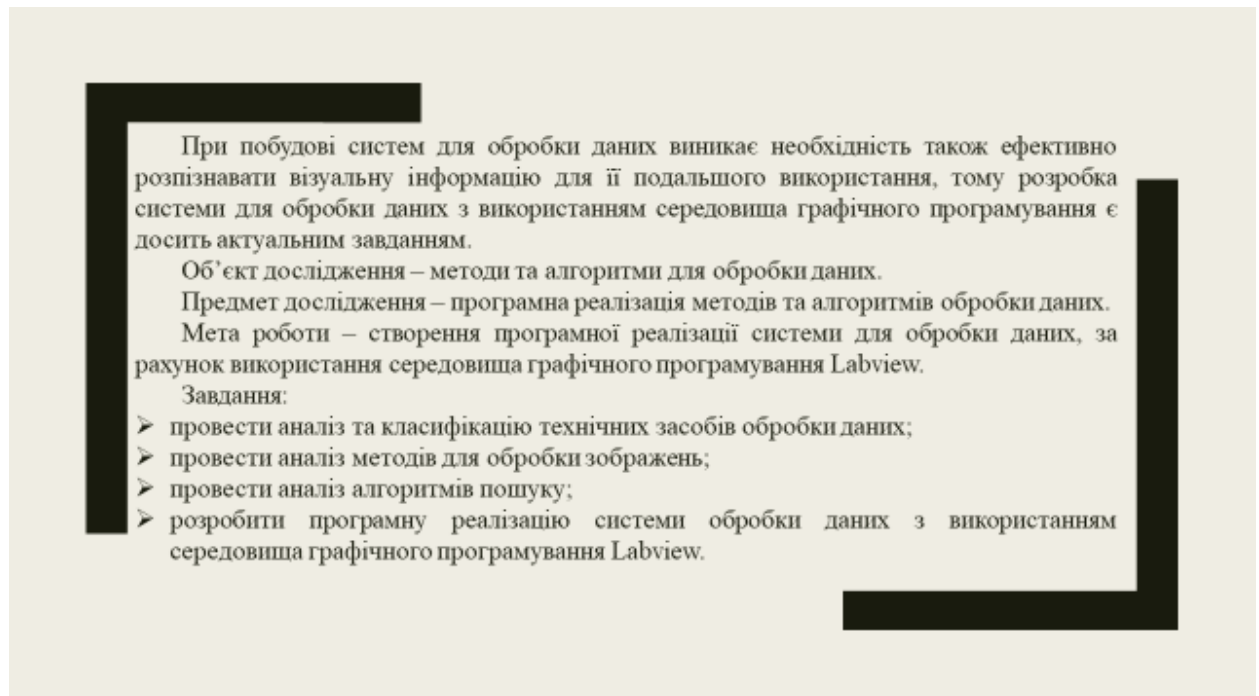


Рисунок Б.2 – Слайд №2

ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

На сьогоднішній день у зв'язку з високим рівнем автоматизації різних сфер людської діяльності виникає потреба не тільки швидко та точно виконувати обчислення, але також ефективно розпізнавати візуальну інформацію. Та на основі проведеного аналізу вирішувати складні завдання управління та контролю. Дані системи прийнято називати системами машинного зору.



Рисунок Б.3 – Слайд №3

РЕЖИМИ ОБРОБКИ ДАНИХ



Рисунок Б.4 – Слайд №4

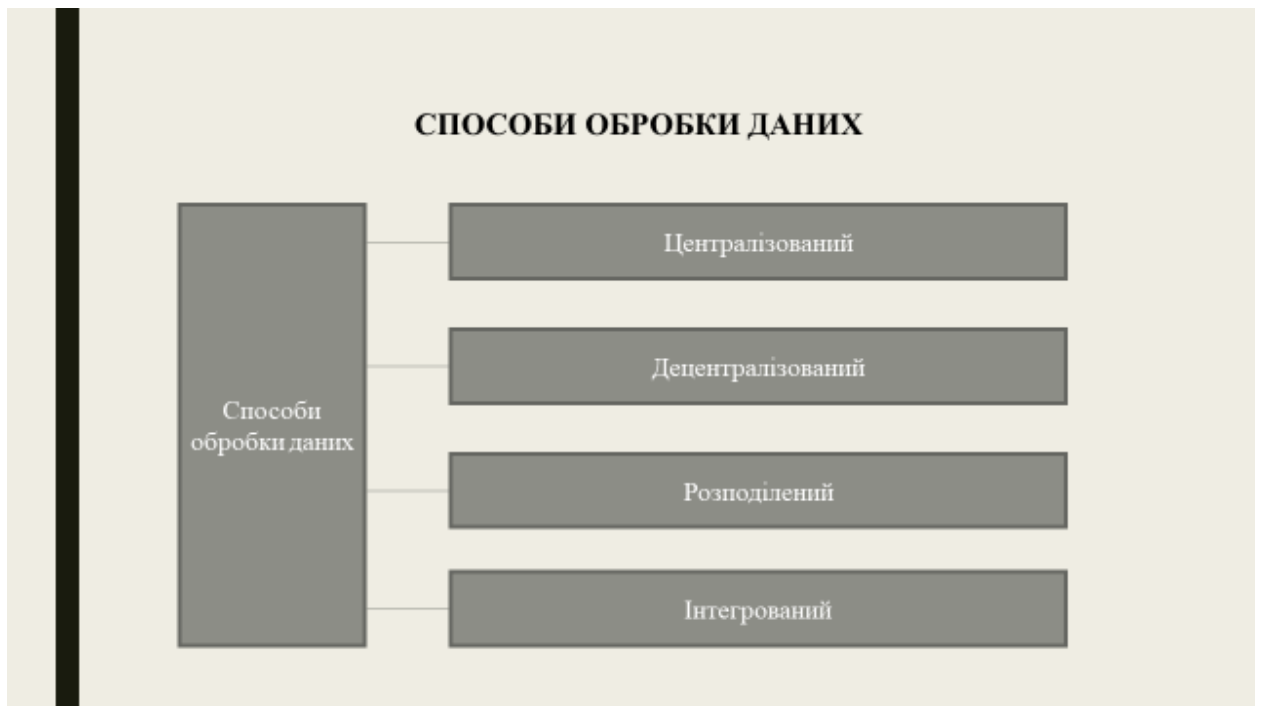


Рисунок Б.5 – Слайд №5

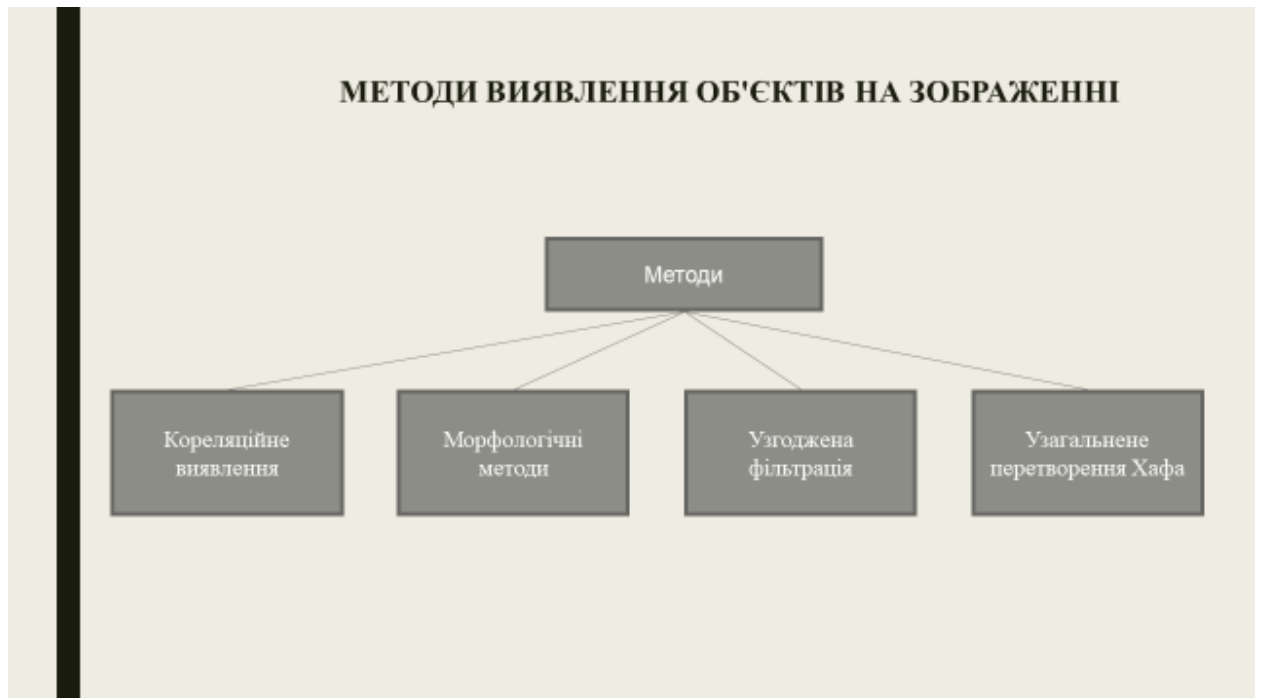


Рисунок Б.6 – Слайд №6

КОЛЬОРОВІ МОДЕЛІ, ВИКОРИСТОВУВАНІ В МАШИННОМУ ЗОРІ

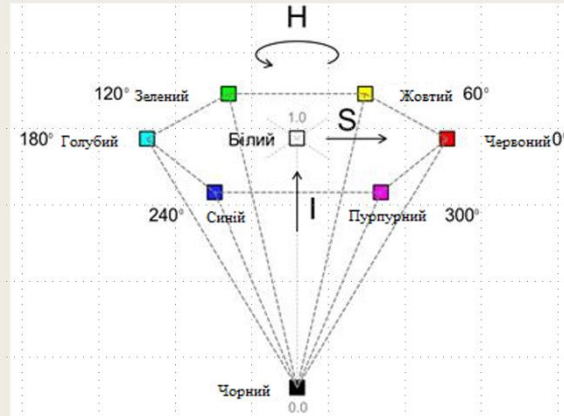


Рисунок 1 - Модель HSI

Рисунок Б.7 – Слайд №7

ІНСТРУМЕНТИ ПОШУКУ ЗБІГІВ З ЕТАЛОНОМ В LABVIEW

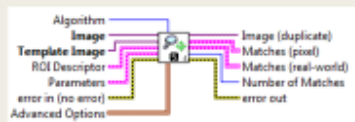


Рисунок 2 - IMAQ Match Pattern 4

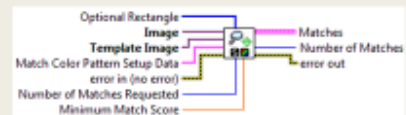


Рисунок 3 - IMAQ Match Color Pattern

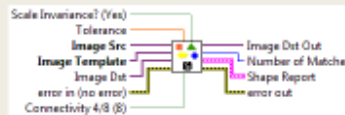


Рисунок 4 – Функція IMAQ Shape Match Tool

Рисунок Б.8 – Слайд №8

ІНСТРУМЕНТИ РЕАЛІЗАЦІЇ СТЕЖЕННЯ ЗА ОБ'ЄКТОМ В СЕРЕДОВИЩІ LABVIEW

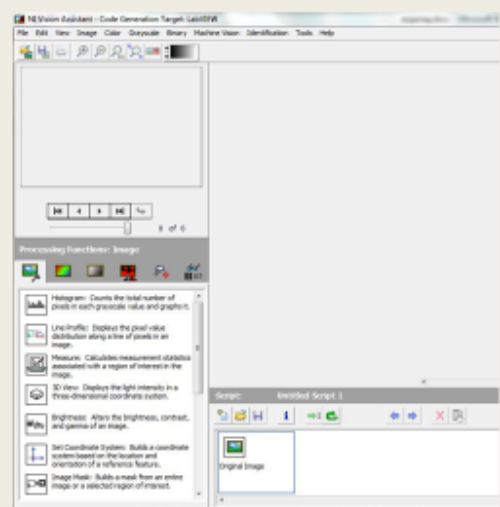


Рисунок 5 - Інтерфейс Vision Assistant

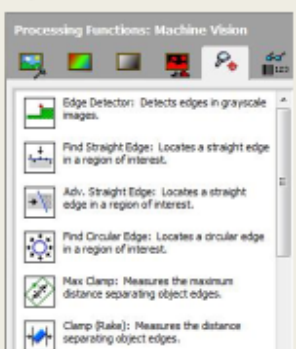


Рисунок 6 - Функції машинного зору

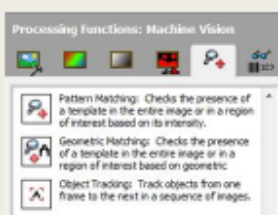


Рисунок 7 - Функції пошуку збігів



Рисунок 8 - Функції зчитування символічної інформації

Рисунок Б.9 – Слайд №9

ФУНКЦІОНАЛЬНА СХЕМА ДОСЛІДЖУВАНОГО АЛГОРИТМУ



Рисунок 9 - Функціональна схема алгоритму пошуку збігів

Рисунок Б.10 – Слайд №10

СИСТЕМА ОБРОБКИ ДАНИХ

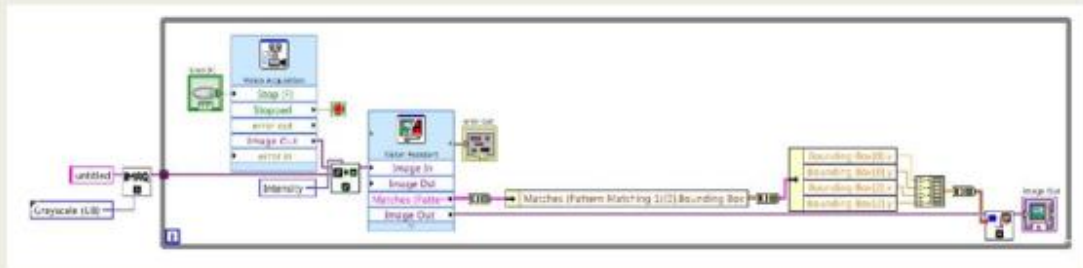


Рисунок 10 – Система обробки даних

Рисунок Б.11 – Слайд №11

ВИБІР ШАБЛОНУ ТА АЛГОРИТМУ ПОШУКУ



Рисунок 11 – Вибір шаблону

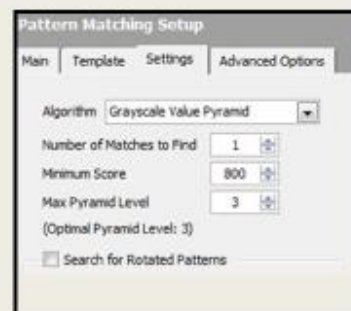


Рисунок 12 – Вибір алгоритму пошуку

Рисунок Б.12 – Слайд №12

СТВОРЕННЯ МАСИВУ ТА ВИЛУЧЕННЯ ЗОБРАЖЕННЯ



Рисунок 13 – Створення масиву

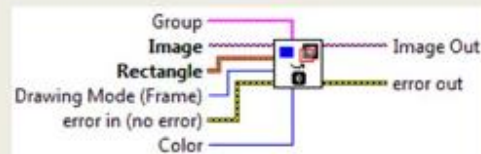


Рисунок 14 – Вилучення зображення

Рисунок Б.13 – Слайд №13

ЗАПУСК ПРОГРАМИ

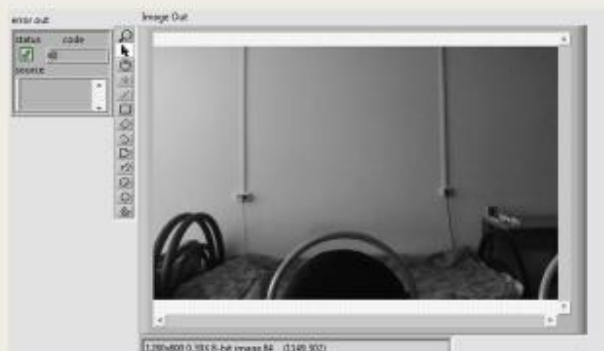


Рисунок 15 – Інтерфейс програми



Рисунок 16 – Script - pattern matching



Рисунок 17 – Pattern matching settings

Рисунок Б.14 – Слайд №14

ВИДІЛЕННЯ ОБ'ЄКТА



Рисунок 18 – Вибір області містить об'єкт

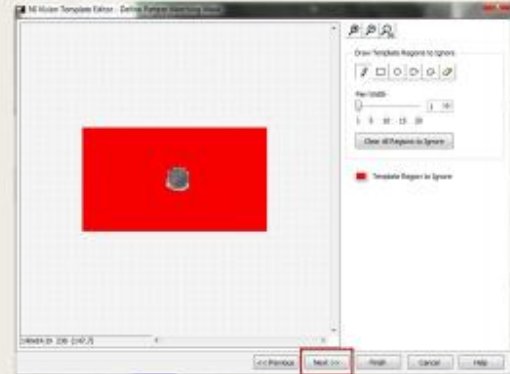


Рисунок 19 – Виділення об'єкта

Рисунок Б.15 – Слайд №15

ІНТЕРФЕЙС ПРОГРАМИ



Рисунок 20 – Інтерфейс програми

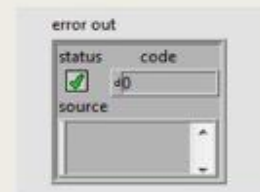


Рисунок 21 – Статус наявності помилок

Рисунок Б.16 – Слайд №16

ВИСНОВОК

Створена програмна реалізація дає змогу проводити обробку графічних даних отриманих з відео та визначати задані елементи. Для досягнення мети були використані наступні інструменти: функція пошуку збігів: IMAQ Match Pattern 4, функція, що знаходить збіги з кольірною шаблоном на вхідному зображенні IMAQ Match Color Pattern, інструменти Edge detector, Find Straight Edge, Adv.Straight Edge, Find Circular Edge.

Обробка проходить в наступній послідовності: спочатку необхідно отримати зображення, наступним кроком виникла необхідність перетворення зображення в чорно-біле, далі створення шаблону та вибір алгоритму пошуку, далі створення масиву для пошуку за шаблоном та виведення зображення з маркером стеження.

Для досягнення даного результату були виконані наступні завдання:

- було проведено аналіз та класифікацію технічних засобів обробки даних;
- проаналізовано методів для обробки зображень;
- проведено аналіз алгоритмів пошуку;
- розроблено програмну реалізацію системи обробки даних з використанням середовища графічного програмування Labview.

Рисунок Б.17 – Слайд №17