

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та
робототехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

Автоматизація процесу розпізнавання деталей на поточному виробництві
(тема)

Виконав:
студент 2 курсу, групи КІТПВм-22-2
Пашенко О. С.
(прізвище, ініціали)

Спеціальності 151 Автоматизація та
комп'ютерно-інтегровані технології
(код і повна назва спеціальності)

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

Освітня програма Комп'ютерно-
інтегровані технологічні процеси і
виробництва
(повна назва освітньої програми)

Керівник доц. Косенко Н. В.
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри КІТАР

(підпис)

Невлюдов І. Ш.
(прізвище, ініціали)

2024р.

Харківський національний університет радіоелектроніки

Факультет	Автоматики і комп'ютеризованих технологій
Кафедра	Комп'ютерно-інтегрованих технологій, автоматизації та роботехніки
Рівень вищої освіти	другий (магістерський)
Спеціальність	151 Автоматизація та комп'ютерно-інтегровані технології
Тип програми	освітньо-професійна
Освітня програма	Комп'ютерно-інтегровані технологічні процеси і виробництва

(код і повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри КІТАР _____

(підпис)

« ____ » _____ 20 ____ р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Студентові _____ Пащенко Олександр Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Автоматизація процесу розпізнавання деталей на поточному виробництві
затверджена наказом по університету від “ 03” листопада 2023р. №1287 Ст.
2. Термін подання студентом роботи “ 24 ” січня 2024р.
3. Вихідні дані до роботи 3.1 Промисловий робот з системою технічного зору;
3.2 Бібліотека Integrated Performance Primitives (IPP);
3.3 Бібліотека AviCap;
3.4 Бібліотека OpenCV;
3.4 Оформлення текстової документації згідно ДСТУ 3008-2015.
4. Перелік питань, що потрібно опрацювати в роботі 4.1 Вступ;
4.2 Методи обробки інформації у системах технічного зору роботів;
4.3 Методи ідентифікації деталей у роботизованих системах;
4.4 Практична реалізація методів обробки інформації у системах технічного зору;
4.5 Розробка програмного забезпечення для визначення параметрів деталей поточного виробництва за допомогою системи технічного зору;
4.6 Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій Демонстраційний матеріал представлений у форматі презентації PowerPoint (*.ppt) – 16 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	Термін виконання етапів роботи	Примітка
1	Методи обробки інформації у системах технічного зору роботів	06.11 – 09.11.23	виконано
2	Методи ідентифікації деталей у роботизованих система	10.11 – 26.11.23	виконано
3	Практична реалізація методів обробки інформації у системах технічного зору	27.11 – 18.12.23	виконано
4	Розробка програмного забезпечення для визначення параметрів деталей поточного виробництва за допомогою системи технічного зору	19.12 – 10.01.24	виконано
5	Промислова безпека та аналіз умов праці на робочому місці	11.01 – 12.01.24	виконано
6	Оформлення пояснювальної записки	13.01 – 15.01.24	виконано
7	Подання роботи на перевірку Інтернет-сервісом Unichesk	16.01 – 18.01.24	виконано
8	Подання роботи на рецензію	19.01 – 21.01.24	виконано
9	Подання роботи на підпис зав. кафедри	22.01 – 23.01.24	виконано
10	Подання кваліфікаційної роботи в ЕК	24.01.24	виконано

Дата видачі завдання 06 листопада 2023 р.

Студент _____ Пащенко О. С.
(підпис)

Керівник роботи _____ доц. Косенко Н. В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 104 с., 8 табл., 10 рис., 3 дод., 14 джерел.

ПРОМИСЛОВИЙ РОБОТ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, СИСТЕМА ТЕХНІЧНОГО ЗОРУ, РОЗПІЗНАВАННЯ, ІДЕНТИФІКАЦІЯ, МЕТОДИ ОБРОБКИ ІНФОРМАЦІЇ, ПОТОЧНЕ ВИРОБНИЦТВО

Об'єкт дослідження – розпізнавання та ідентифікації деталей поточного виробництва, за допомогою системи технічного зору промислового робота.

Предмет дослідження – програмне забезпечення для системи технічного зору промислового робота.

Мета роботи – розробка програмного забезпечення для розпізнавання та ідентифікації деталей у робочій зоні промислового робота на поточному виробництві.

Розглянуто методи розпізнавання та ідентифікації деталей поточного виробництва, проаналізовано методи обробки інформації у системах технічного зору роботів, побудована теоретико-множинна модель розпізнавання та ідентифікації, розглянута практична реалізація методів обробки інформації в робототехнічних системах, розроблено програмне забезпечення, що реалізує методи розпізнавання та ідентифікації деталей поточного виробництва.

До числа перспективних напрямків досліджень слід віднести подальший розвиток методів розпізнавання та ідентифікації, їх програмну реалізацію для промислових операційних систем реального часу. У комбінації з системою прийняття рішень такі розробки істотно прискорять реалізацію систем управління інтелектуальними роботами.

ABSTRACT

Explanatory note: 104 pp., 8 tab., 10 figs., 3 appendices, 14 sources.

INDUSTRIAL ROBOT, SOFTWARE, TECHNICAL VISION SYSTEM, RECOGNITION, IDENTIFICATION, INFORMATION PROCESSING METHODS, CURRENT PRODUCTION

The object of research is the recognition and identification of details of current production, using the technical vision system of an industrial robot.

The subject of research is software for the technical vision system of an industrial robot.

The purpose of the work is to develop software for recognition and identification of parts in the working area of an industrial robot in current production.

The methods of recognition and identification of details of current production were considered, the methods of information processing in systems of technical vision of robots were analyzed, a theoretical-multiple recognition and identification model was built, the practical implementation of information processing methods in robotic systems was considered, software was developed that implements methods of recognition and identification of details of current production

Further development of recognition and identification methods, their software implementation for real-time industrial operating systems should be included among the promising areas of research. In combination with a decision-making system, such developments will significantly accelerate the implementation of intelligent robot management systems.

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

«20» січня 2024 р.



A handwritten signature in black ink, appearing to read 'O. S. Pashchenko', is written over a horizontal line.

Пащенко О. С.

ЗМІСТ

Перелік скорочень	9
Вступ	10
1 Аналіз методів обробки інформації в системах технічного зору роботів	13
1.1 Класифікація методів обробки інформації у системах технічного зору	13
1.2 Методи попередньої обробки зображень	15
1.3 Сегментація деталей	15
1.4 Визначення порогового рівня зображень	20
1.5 Обласно-орієнтована сегментація	20
1.6 Проблема опису деталей	24
1.7 Дескриптори кордону	24
1.8 Дескриптори областей зображень	26
1.9 Сегментація і опис тривимірних структур	28
1.10 Опис тривимірної сцени плоскими ділянками	29
1.11 Техніка обробки візуальної інформації	30
2 Методи ідентифікації деталей в роботизованих системах	51
2.1 Метод порівняння з еталоном	51
2.2 Методи теорії графів і розпізнавання	54
2.3 Кореляційний метод	56
2.4 Розпізнавання через зв'язок шаблонів	58
2.5 Штучні нейронні мережі та їх використання при ідентифікації зображень	66
2.6 Розпізнавання деталей і їх інтерпретація	69
2.7 Теоретико-множинна модель розпізнавання та ідентифікації	70

3	Практична реалізація методів обробки інформації у системах технічного зору	74
3.1	Основні підходи до практичної реалізації методів обробки інформації	74
3.2	Бібліотека Integrated Performance Primitives (IPP)	76
3.3	Бібліотека AviCap	77
3.4	Бібліотека OpenCV	78
4	Розробка програмного забезпечення для визначення параметрів деталей	93
4.1	Основні особливості розробленого програмного забезпечення	93
4.2	Реалізація функції обробки зображень	95
4.3	Реалізація функцій розпізнавання та ідентифікації	96
4.4	Промислова безпека та аналіз умов праці на робочому місці	99
	Висновки	102
	Перелік джерел посилання	104
	Додаток А Апробація результатів досліджень.....	107
	Додаток Б Лістинг програми 1	108
	Додаток В Лістинг програми 2	113
	Додаток Г Демонстраційний матеріал	116

ПЕРЕЛІК СКОРОЧЕНЬ

- ЕОМ – електронно-обчислювальна машина;
- КПО – коефіцієнт природної освітленості;
- ПСО – перетворення середніх осей;
- РС – роботизована система;
- СТЗ – система технічного зору;
- ШНМ – штучні нейронні мережі;
- Avi Cap – avi capture;
- IPP – integrated performance primitives (бібліотека комп'ютерного зору);
- MLL – machine learning library;
- Open CV – open source computer vision library (бібліотека комп'ютерного зору з відкритим вихідним кодом).

ВСТУП

Зорові можливості робота, за аналогією до людських, забезпечуються складним чутливим механізмом, за допомогою якого відбувається гнучке реагування на зміни зовнішнього середовища. Застосування системи технічного зору (СТЗ) та інших методів очутливлення продиктовано постійною необхідністю підвищувати гнучкість та розширювати галузі використання робототехнічних систем. Хоча датчики відстані разом з тактильними, а також датчиками сили відіграють велику роль в удосконаленні функціонування робота, технічний зір вважається найбільш потужним джерелом інформації для робота. Таким чином, дослідження в галузі систем технічного зору роботів продовжують залишатися одними з найперспективніших напрямків сучасної робототехніки. Це підтверджує й актуальність даної роботи.

Зір робота можна схарактеризувати як процес виділення, ідентифікації та перетворення інформації, що надходить від тривимірних зображень. Даний процес також називають технічним або машинним зором. Розподіляється він на шість основних етапів:

- зчитування інформації;
- попередня обробка інформації;
- сегментація;
- опис;
- розпізнавання;
- інтерпретація.

Зчитування інформації полягає у процесі отримання візуального зображення. Попередня обробка інформації здійснюється шляхом застосування таких методів, як зниження шуму чи поліпшення зображення окремих деталей. Сегментацією називається процес виділення на зображенні деталей, які викликають зацікавлення. Під час опису окреслюються

характерні параметри (на кшталт, розмірів або форми), необхідні для виділення питомого об'єкта на тлі інших. Розпізнавання – це процес ідентифікації деталей (зокрема, гайкового ключа, болта, блоку двигуна). Інтерпретація, в свою чергу, визначає приналежність до групи розпізнаваних деталей.

Перелічені етапи зручно згрупувати відповідно до складності їхньої реалізації. Визначимо три рівні технічного зору: низький, середній та високий. Хоча чітких кордонів між цими рівнями не існує, проте, їхнє виділення є доцільним для класифікації різних процесів, які характерні системам технічного зору.

До низького рівня технічного зору належать такі процеси, що з точки зору здійснення автоматичних дій є простими, тобто не потребують наявності штучного інтелекту. До того ж, можемо віднести до зазначеного рівня і зчитування, а та попередню обробку інформації.

До середнього рівня технічного зору належать процеси виділення, ідентифікації та розмітки елементів зображення, що було отримано з нижнього рівня. З урахуванням визначеного розподілу технічного зору до середнього рівня належать сегментація, опис та розпізнавання окремих деталей.

До високого рівня технічного зору належать процеси, що пов'язані зі штучним інтелектом. На відміну від алгоритмів, які застосовують на нижньому та середньому рівнях технічного зору, і є добре розробленими, знань та розумінь процесів високого рівня поки що недостатньо.

Мета даної кваліфікаційної роботи магістра полягає у розробці програмного забезпечення для розпізнавання та ідентифікації деталей у робочій зоні промислового робота.

Для розв'язання поставленої мети треба виконати такі завдання:

- провести аналіз основних методів розпізнавання та ідентифікації;
- побудувати теоретико-множинну модель процесу розпізнавання та ідентифікації;

– розробити програмне забезпечення, що базується на розпізнаванні й ідентифікації простих деталей.

Робота виконується згідно з [1-4], як складова наукових досліджень, які здійснюються на кафедрі КІТАР Харківського національного університету радіоелектроніки, результати дослідження опубліковані у [5].

1 АНАЛІЗ МЕТОДІВ ОБРОБКИ ІНФОРМАЦІЇ В СИСТЕМАХ ТЕХНІЧНОГО ЗОРУ РОБОТІВ

1.1 Класифікація методів обробки інформації у системах технічного зору

Одним з найважливіших напрямків розвитку робототехнічних систем завжди була організація взаємодії маніпулятора і сенсорних аналізаторів. Використання сенсорів надає системі гнучкість, розширює її функціональні можливості і збільшує коло вирішуваних нею завдань. Більшість нині існуючих систем, що використовують взаємодію маніпулятора і сенсорів, побудовані шляхом зв'язування автономних сенсорної і робототехнічної підсистем. Ці системи продемонстрували спроможність концепції взаємодії системи управління роботом з сенсорною підсистемою, на основі якої розроблено концепцію візуального управління роботами [5-6]. Схема обробки інформації в СТЗ робота показана на рисунку 1.1.

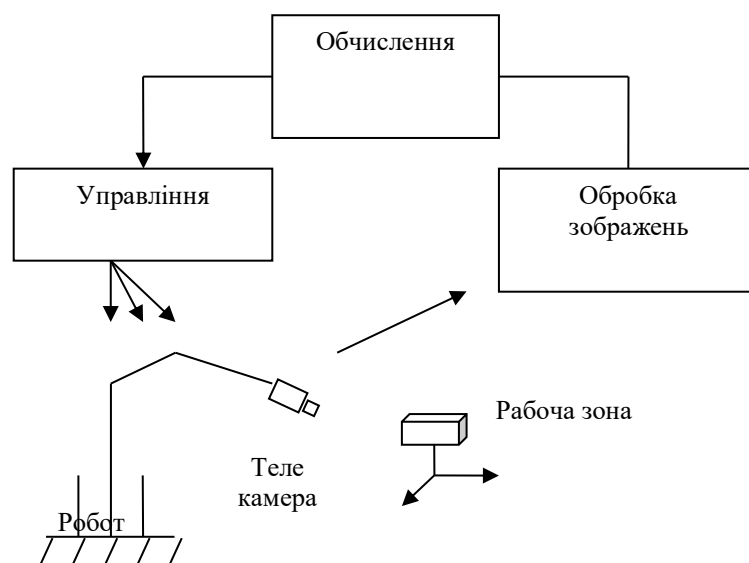


Рисунок 1.1 – Обробка інформації в СТЗ робота

Для класифікації методів і підходів, які застосовуються в системах технічного зору, зір доцільно розподілити на три основних підкласи: зір низького, середнього та високого рівнів. Системи технічного зору низького рівня виконують обробку інформації з датчиків зчутливлення [6].

Зазначені системи можна віднести до класу «інтелектуальних» машин, якщо вони наділені такими характеристиками (ознаками інтелектуальної поведінки):

- можливістю виділяти суттєву інформацію з безлічі незалежних ознак;
- здатністю до навчання на прикладах, а також узагальнення цих знань для їхнього подальшого застосування в нових ситуаціях;
- можливістю відновлювати події за наявності неповної інформації;
- здатністю визначати цілі та формулювати плани для досягнення поставлених цілей.

Створення систем технічного зору з властивостями для обмежених видів робочого простору в принципі є можливим, проте, характеристики таких систем далекі від можливостей людського зору. За основу технічного зору взято аналітичну формалізацію, що спрямовується на вирішення конкретних завдань. Машини з сенсорними характеристиками, що наближені до можливостей людини, ймовірно, ще не скоро з'являться. Проте зауважимо, що копіювання природи – це не єдине рішення для даної проблеми [6]. Сучасне розв'язання задачі щодо польоту у просторі докорінно відрізняється від рішень, які підказані природою. За швидкістю та і досяжністю висоти літаки набагато перевершують можливості птахів.

Системи технічного зору середнього рівня пов'язуються з завданнями сегментації, опису та розпізнавання окремих об'єктів. Вони охоплюють безліч підходів, які засновані на аналітичних поданнях. Системи технічного зору високого рівня вирішують проблеми, про які йшла мова вище. Для більш зрозумілого усвідомлення проблем технічного зору високого рівня, а

також його зв'язку з технічним зором низького та середнього рівнів зазвичай вводять низку обмежень, тим самим спрощують розв'язувану задачу.

1.2 Методи попередньої обробки зображень

Процес ідентифікації об'єктів, які розташовані в робочій зоні робота, здебільшого містить два етапи: визначення характерних ознак об'єктів; власне розпізнавання об'єктів за знайденою сукупністю характерних ознак. Відповідно до такої структури процесу ідентифікації алгоритми обробки інформації в СТЗ прийнято розподіляти на алгоритми попередньої обробки й алгоритми розпізнавання, що певною мірою мають умовний характер, оскільки в деяких практичних застосуваннях одні і ті ж за математичним призначенням алгоритми можуть застосовуватись на обох етапах розглянутого процесу.

Відповідно до спектрального діапазону перетворювачів світло-сигнал, а також принципом дії СТЗ датчик формує зображення робочої зони робота. Під зображенням, як правило, постає двовимірна картина поля інтенсивності випромінювань робочої зони. Формально зображення отримується шляхом визначення функціональної залежності інтенсивності випромінювань робочої зони від координат точок зображення x і y :

$$J = f(x, y). \quad (1.1)$$

Надалі під «зображенням» будемо розуміти саме функціональну залежність $f(x, y)$. З урахуванням прийнятої термінології завданням попередньої обробки є пошук будь-яких особливостей функції $f(x, y)$, які могли б вказати на тип об'єкту, що знаходиться в робочій зоні [6].

1.3 Сегментація деталей

1.3.1 Проведення контурів і визначення меж

Сегментація – це процес поділу сцени на складові частини або об'єкти. Сегментація вважається одним із основних елементів роботи автоматизованої системи технічного зору, оскільки саме на цьому етапі обробки об'єкти виокремлюються зі сцени для подальшого розпізнавання й аналізу. Алгоритми сегментації, зазвичай, засновані на двох фундаментальних принципах: розривності та подібності. У першому випадку основний підхід базується на визначенні контурів, а в другому – на визначенні порогового рівня та розширенні області. Такі поняття актуальні як до статичних, так і динамічних (залежних від часу) сцен. В останньому випадку рух може бути потужним засобом щодо поліпшення роботи алгоритмів сегментації.

Методами називають обчислення градієнта, граничний поділ, який визначає розриви в інтенсивності подання образу об'єкта. В ідеальному випадку такі методи визначають пікселі, що знаходяться на межі між об'єктом і фоном. На практиці такий ряд пікселів зрідка повністю характеризує межі через шум, розривів на межі внаслідок нерівномірної освітленості та інших ефектів, які спричиняють до розмиття зображення. Отже, алгоритми з визначення контурів супроводжуються процедурами побудови меж об'єктів із відповідних послідовностей пікселів [7]. Нижче розглянемо кілька методик, придатних для заданої мети.

1.3.2 Локальний аналіз

Одним із найпростіших підходів з'єднання точок контуру вважається аналіз характеристик пікселів у невеликому околі (наприклад, в околі розміром 3×3 або 5×5) кожної точки (x, y) образу, над яким вже виконали процедуру щодо виявлення контуру. Всі подібні точки (визначення критерію

подібності розглянемо нижче) з'єднуються, утворюючи межу з пікселів, які наділені деякими загальними властивостями.

Для такого аналізу зі встановлення подібності пікселів контуру необхідно з'ясувати:

- величину градієнта, що необхідна для побудови контурного пікселя;
- напрямок градієнта.

Перша характеристика позначається величиною $G\{f(x, y)\}$.

$$G[f(x, y)] = [G_x^2 + G_y^2]^{1/2} = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2} \quad (1.2)$$

$$G[f(x, y)] \cong |G_x| + |G_y|$$

Отже, піксель контуру з координатами (x', y') подібний за величиною в певному раніше околі (x, y) пікселя з координатами (x, y) , якщо нерівність справедлива

$$|G[f(x, y)] - G[f(x', y')]| \leq T \quad (1.3)$$

де T – порогове значення.

Напрямок градієнта визначається за кутом вектора градієнта, визначеного в рівнянні

$$G[f(x, y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} . \quad (1.4)$$

$$\theta = \operatorname{arctg} \begin{bmatrix} G_x \\ G_y \end{bmatrix} \quad (1.5)$$

де θ – кут (щодо осі x), уздовж якого швидкість зміни набуває найбільшого значення.

Виходить, що кут пікселя контуру з координатами $\{x', y'\}$ в деякому околі (x, y) подібний куту пікселя з координатами $\{x, y\}$ за умови виконання такої нерівності:

$$|\theta - \theta'| < A, \quad (1.6)$$

де A – порогове значення кута.

Зазначимо, що напрямок контуру в точці $\{x, y\}$ в дійсності є перпендикулярним напрямку вектора градієнта в цій точці. Проте для порівняння напрямків нерівність дає еквівалентні результати.

Беручи за основу дані припущення, з'єднуємо точку в деякому околі (x, y) з пікселем, який має координати $\{x, y\}$, за умови, що за величиною і напрямком задовольняються критерії. Рухаючись від пікселя до пікселя, при цьому враховуючи кожен приєднувану точку як центр околу, процес повторюється для кожної точки образу. З метою встановлення відповідності між рівнями інтенсивності освітлення та послідовностями пікселів контуру використовується стандартна бібліотечна процедура.

Необхідно визначити розміри прямокутників, за допомогою яких можна створити якісне зображення. Побудова таких прямокутників реалізується шляхом визначення строго горизонтальних і вертикальних контурів. Подальший процес полягає у поєднанні сегментів контуру, що розділені невеликими проміжками, а також в об'єднанні окремих коротких сегментів.

1.3.3 Глобальний аналіз за допомогою перетворення Хоуга

Опрацюємо метод з'єднання граничних точок завдяки визначенню їхнього розміщення на кривій спеціального виду. Спершу припустимо, що на площині xu образу дано n точок, потрібно визначити підпоследності

точок, які лежать на прямих лініях. Одним із можливих рішень може бути процес побудови всіх ліній, які проходять через кожен пару точок, а потім – знаходження всіх підпоследовностей точок, близьких до певних ліній. Завданням, що пов'язане з цією процедурою, є знаходження $n(n-1)/2 \sim n^2$ ліній, після чого – проведення $n[n(n-1)]/2 \sim n^3$ порівнянь кожної точки з усіма лініями. Цей процес (з обчислювальної точки зору) є трудомістким за винятком найпростіших додатків [7].

Існуючу задачу можна розв'язати по-іншому, використовуючи підхід, запропонований Хоугом, який ще називають перетворенням Хоуга. Обчислювальна перевага перетворення Хоуга полягає в поділі простору параметрів на так звані збираючі елементи, де (a_{\max}, a_{\min}) та (b_{\max}, b_{\min}) – допустимі величини параметрів ліній. Збираючий елемент $A(i, j)$ відповідає площі, що пов'язується з координатами простору параметрів (a_i, b_j) . Спочатку такі елементи вважаються рівними нулю. Тоді для кожної точки (x_k, y_k) в у площині образу ми припускаємо параметр a дорівнює кожному з допустимих значень на осі a і обчислюємо відповідне b , використовуючи рівняння $b = -x_k + y_k$. Після чого отримане значення b округляється до найближчого допустимого значення на осі b . Якщо вибір a_p призводить до обчислення b_q , ми розраховуємо $A(p, q) = A(p, q) + 1$. Після проведення цієї процедури значення M в елементі $A(i, j)$ відповідає M точкам у площині xu , що розташовані на лінії $y = ax + b$. Власне точність розташування даних точок на одній прямій залежить від числа розбиття площини ab . Якщо ми розіб'ємо вісь a на K частин, відповідно, для кожної точки (x_k, y_k) ми отримаємо K значень b , відповідних K можливих значенням a . Оскільки є n точок образу, процес складається з nK обчислювальних операцій. Отже, процедура, наведена вище, лінійна щодо n і має меншу кількість обчислювальних операцій за вже розглянуту процедуру, якщо $K \leq n$.

1.4 Визначення порогового рівня зображень

Поняття порогового рівня (порога) тест виду

$$T = T[x, y, p(x, y), f(x, y)], \quad (1.7)$$

де $f(x, y)$ – інтенсивність в точці (x, y) ;

$p(x, y)$ – деяка локальна властивість, яка визначається в околі цієї точки.

Граничне зображення дається наступним виразом:

$$g(x, y) = \begin{cases} 1, & \text{если } f(x, y) > T \\ 0, & \text{если } f(x, y) \leq T \end{cases}, \quad (1.8)$$

так що пікселі в $g(x, y)$, зі значенням 1, відповідають об'єктам, а пікселі зі значенням 0, відповідають фону. У рівнянні передбачається, що інтенсивність об'єктів є більшою за інтенсивність фону. Протилежна умова реалізується шляхом зміни знаків у нерівностях.

1.5 Обласна-орієнтована сегментація

1.5.1 Основні визначення

Метою сегментації полягає у розподілі образу на області. Розглянемо методи сегментації, що ґрунтуються на прямому знаходженні областей [7].

Нехай R – область образу. Розглянемо сегментацію як процес розбиття R на n подобластей R_1, R_2, \dots, R_n , так що:

$$\bigcup_{i=1}^n R_i = R,$$

- P_i – зв'язна область, $i = 1, 2, \dots, n$;
- $R_i = \emptyset$ для всіх i та j , $i \neq j$;
- $P(R_i) \in \text{ІСТИНА}$ для $i = 1, 2, \dots, n$;

– $P(R_iUR_j)$ є ХИБНІСТЬ для $i \neq j$, де $P(R_i)$ – логічний предикат, визначений на точках з множини R_i , $i \in \emptyset$ – порожня множина.

Умова 1 зобов'язує, що сегментація має бути повною, іншими словами, кожен піксель має перебувати в образі. Умова 2 зобов'язує, щоб точки в області були зв'язковими. Умова 3 вимагає, щоб області не перетиналися. Умова 4 формулює властивості, що мають задовольняти пікселі в сегментованій області. Елементарний приклад: $P(R_i) = \text{ІСТИНА}$, якщо всі пікселі в R_i мають однакову інтенсивність. Умова 5 окреслює, що області R_i та R_j розрізняються за предикатом P .

1.5.2 Розширення області за рахунок об'єднання пікселів

Розширення області реалізується процедурою групування пікселів або підобластей у великі об'єднання. Найпростішою є агрегування пікселів. Починається процес із вибору багатьох вузлових точок, за допомогою яких відбувається розширення області шляхом приєднання до вузлових точок сусідніх пікселів зі схожими характеристиками (інтенсивність, текстура чи колір). Проте існують дві проблемами: вибір початкових вузлів для правильного уявлення областей, що викликають інтерес, а також визначення відповідних властивостей для включення точок до різних областей у процесі розширення. Вибір множини, що містить одну чи кілька початкових точок, починається з постановки задачі. Наприклад, у військових програмах об'єкти, що викликають інтерес, мають більш високу температуру, ніж фон, у зв'язку з цим окреслюються більш яскраво. Вибір найяскравіших пікселів є природним початковим кроком в алгоритмі щодо розширення області. За умови відсутності апріорної інформації можна починати з обчислення для кожного пікселя набору властивостей, який, імовірно, буде використаний для встановлення відповідності пікселя тій чи іншій області в процесі розширення. Якщо результат обчислень – групи точок (кластери), тоді за вузлові беруть ті пікселі, властивості яких є близькими до властивостей центроїдів цих груп. Так, наприклад, гістограма інтенсивностей показала б,

що точки з інтенсивністю від одного до семи - домінуючі. Вибір критерію подібності залежить як від завдання, так і виду наявних даних про образ. Зокрема, аналіз інформації, що надходить зі супутників, істотно залежить від кольору. Завдання аналізу буде значно ускладненим, якщо використовувати тільки монохроматичні образи. На жаль, для промислового технічного зору можливість отримання мультиспектральних та інших додаткових даних про спосіб – це скоріше виняток, ніж правило [7]. Як правило, аналіз області повинен виконуватися за допомогою набору дескрипторів, які містять інтенсивність і просторові характеристики (моменти, текстуру) одного джерела зображення. Зауважимо, що використання тільки одних дескрипторів може призвести до неправильних результатів, якщо не застосовувати інформацію про умови зв'язку в процесі розширення області. Це легко продемонструвати під час розгляду випадкового розташування пікселів із трьома різними значеннями інтенсивності. Об'єднання в «область» пікселів за ознаками однакової інтенсивності без урахування умов зв'язку призведе до безглузлого результату під час сегментації.

Інша важлива проблема, що виникає під час розширення області, - формулювання умови закінчення процесу. Як правило, процес розширення області закінчується, якщо більше немає пікселів, які відповідають критерію приналежності тій чи іншій області. Вже були опрацьовані такі критерії, як інтенсивність, текстура і колір, які за своєю природою є локальними, до того ж, не враховують «історію» процесу розширення області. Додатковий критерій, який підвищує потужність алгоритму розширення області, ґрунтується на поняттях розміру, схожості між пікселем-кандидатом та щойно створеними пікселями (порівняння інтенсивності кандидата та середньої інтенсивності області), а також форми області, що підлягає розширенню. Застосування цих типів дескрипторів базується на припущенні, що інформація про очікувані результати є неповною.

1.5.3 Розбиття та об'єднання області

Розглянута процедура розширення області розпочинає роботу з заданої множини вузлових точок. Проте спочатку можна розбити образ на низку довільних непересічних областей, після чого об'єднувати та/або розбивати ці області з метою задоволення умов. Ітеративні алгоритми розбиття та об'єднання, реалізація яких спрямована на виконання цих обмежень, можуть бути викладені таким чином [6]. На кожному кроці виконуються наступні операції:

- розбиття області R_i , для якої $P(R_i) = \text{ХИБНІСТЬ}$, на чотири непересічних квадранта;
- об'єднання сусідніх областей R_i і R_k , для яких $P(R_i \cup R_k) = \text{ІСТИНА}$;
- вихід на останок, якщо подальше об'єднання або розбиття неможливе.

1.5.4 Аналіз руху об'єктів

Рух є потужним засобом, який використовується людиною і тваринами для виокремлення їхніх об'єктів із фону. У системах технічного зору роботів рух активізується під час виконання різних операцій на конвеєрі, під час переміщення руки, обладнаної датчиком, рідше – у процесі переміщення всієї робототехнічної системи.

Одним із найпростіших підходів для визначення змін між двома кадрами зображення (образами) $f(x, y, t_i)$ і $f(x, y, t_j)$, взятими, відповідно, в моменти часу t_i і t_j , вважається порівняння відповідних пікселів цих двох образів. Для цього використовується процедура, що ґрунтується на формуванні так званої різниці образів.

Припустимо, що існує еталонний образ, який містить тільки стаціонарні компоненти. Якщо порівняємо цей образ із таким, який складається з рухомих об'єктів, то різниця двох образів виявляється в

результаті викреслювання стаціонарних компонент (тобто залишаються тільки ненульові записи, що відповідають нестаціонарним компонентам зображення) [7].

1.6 Проблема опису об'єктів

У СТЗ проблема опису полягає у виділенні властивостей (деталей) об'єкта з метою розпізнавання. В ідеальному випадку дескриптори не мають залежати від розмірів, розташування, також орієнтації об'єкта, проте, повинні містити необхідну кількість інформації для надійної ідентифікації об'єктів. Опис є головним результатом під час конструювання систем технічного зору в тому значенні, що дескриптори повинні впливати не тільки на складність алгоритмів розпізнавання, але й на їхню роботу. Звернемо увагу на три основні категорії дескрипторів: дескриптори межі, дескриптори області та дескриптори для опису тривимірних структур.

1.7 Дескриптори межі

Ланцюгові коди використовують для подання межі як послідовність відрізків прямих ліній певної довжини та напрямку. Як правило, за основу цього уявлення обирають 4 або 8-зв'язкову прямокутну решітку. Довжина кожного відрізка окреслюється роздільною здатністю решітки, а напрямки – обраним кодом. Зауважимо, що для представлення всіх напрямків у 4-направленому ланцюговому коді достатньо лише 2 біт, а для 8-спрямованого ланцюгового коду – 3 біта. Для створення ланцюгового коду заданої межі спочатку необхідно обрати решітку. Звернемо увагу, якщо площа комірки, розташованої усередині межі, є більшою за визначене число (як правило, 50 %), їй присвоюється значення 1, в іншому випадку – значення 0. Остаточню ми кодуємо межу між двома областями за допомогою напрямків. Результат кодування знаходиться в напрямку за годинниковою стрілкою з

початком у місці, що позначене точкою. Альтернативна процедура ґрунтується на розбитті межі на ділянки рівної довжини (кожна ділянка має одне і те ж число пікселів), а також з'єднанні граничних точок кожної ділянки прямою лінією. Після цього відбувається присвоєвання кожній лінії напрямку. Він є найближчим до одного з допустимих напрямів ланцюгового коду. Зауважимо, що ланцюговий код межі залежить від початкової точки. Проте його можна нормувати за допомогою простої процедури. Для створення ланцюгового коду початкова точка на решітці обирається довільно. Вивчаючи ланцюговий код як замкнуту послідовність індексів напрямків, необхідно обрати початкову точку таким чином, щоб вислідна послідовність індексів була цілим числом з мінімальною величиною. Крім того, можна нормувати повороти, якщо замість ланцюгового коду застосовувати його першу різницю. Вона обчислюється в результаті відліку (у напрямку проти годинникової стрілки) кількості напрямків, які розділяють два сусідніх елементи коду. Нормування можна виконати шляхом розбиття всіх кордонів об'єкта на однакову кількість рівних відрізків із подальшою підгонкою довжин сегментів коду з метою їхньої відповідності для цього розбивання.

Сигнатурою називається одномірне функціональне уявлення межі [8]. Існує кілька способів створення сигнатур. Одним із найпростіших є побудова відрізка від центру до межі як функції кута. Як видно, такі сигнатури залежать від периметра області та початкової точки. Нормування периметра можна виконати, пронормувавши криву $r(\theta)$ максимальним значенням. Проблема щодо вибору початкової точки можна розв'язати, спочатку визначивши ланцюговий код межі, після чого використавши метод із попереднього розділу. Певна річ, відстань, яка залежить від кута, це не єдиний спосіб визначення сигнатури. Як варіант, через межу можна провести пряму лінію і визначити кут між дотичною до межі та даною лінією як функцію положення вздовж меж. Отримана сигнатура відрізнятиметься від кривої $r(\theta)$, проте, міститиме інформацію про основні характеристики форми межі. Наприклад, горизонтальні ділянки кривої відповідали б прямим лініям

вздовж меж, тому що кут дотичної тут постійний. Один із варіантів цього методу за сигнатуру застосовує так звану функцію щільності нахилу. Дана функція – гістограма значень кута дотичної. У зв'язку з тим, що гістограма є мірою концентрації величин, то функція щільності нахилу строго відповідає ділянкам межі з постійними кутами дотичній (прямі або майже прямі ділянки, до того ж, має глибокі провали для ділянок, які відповідають за швидкі зміни кутів (виступи або інші види вигинів)).

У задачі апроксимації багатокутниками використовують методи об'єднання, що ґрунтуються на помилці чи інших умовах. Один із підходів заснований на з'єднанні точок межі лінією за методом найменших квадратів. Лінія проводиться доти, поки помилка апроксимації не перевищить раніше заданий поріг. Коли поріг перевищується, параметри лінії фіксуються у пам'яті, помилка покладається рівною нулю, а процедура повторюється; нові точки межі з'єднуються доти, поки помилка знову не перевищить поріг. У кінці процедури утворюються вершини багатокутника шляхом перетину сусідніх ліній. Одне з основних ускладнень, пов'язується з цим підходом, через те, що вершини, як правило, не відповідають вигинам межі (таким, як кути), оскільки нова лінія починається лише тоді, коли помилка перевищить поріг. Якщо, наприклад, довга пряма лінія перетинає кут, то кількістю (що залежать від порога) точок, побудованих після перетину, можна знехтувати раніше, ніж відбудеться перевищення значення порогового рівня. Проте для усунення цієї проблеми поряд із методами об'єднання можна застосувати методи розбиття.

Один із методів розбиття сегментів межі полягає в послідовному розподілі сегмента на дві частини до тих пір, поки задовольнятиметься заданий критерій.

1.8 Дескриптори областей зображень

1.8.1 Деякі прості дескриптори

Наявні системи технічного зору ґрунтуються на значно простих дескрипторах області, що наділяє їх більшою привабливістю з обчислювальної точки зору. Як можна з'ясувати, використання цих дескрипторів обмежено ситуаціями, в яких об'єкти, викликаючи інтерес, розрізняються настільки, що для їхньої ідентифікації достатньо кілька основних дескрипторів.

Площа області визначається як число пікселів, розташованих у межах її кордонів. Такий дескриптор є корисним для збору інформації про взаємне розташування та форми об'єктів, від яких камера знаходиться приблизно на однаковій відстані. Показовим прикладом може бути розпізнавання системою технічного зору об'єктів, які рухаються конвеєром.

Велика і мала осі області є корисними для визначення орієнтації об'єкта. Відношення довжин цих осей, іншими словами, ексцентриситет області, теж є важливим дескриптором для опису форми області.

Периметром області називають довжину її межі. Хоча іноді периметр використовують як дескриптор, частіше його застосовують для визначення міри компактності області, що дорівнює квадрату периметра, поділеному на площу. Зауважимо, що компактність – це безрозмірна величина (отже, інваріантна до зміни масштабу) і мінімальна для поверхні, що має форму диска.

Зв'язковою називають область, в якій будь-яка пара точок може з'єднуватись кривою, повністю лежить в цій області. Для множини зв'язкових областей (деякі з них мають отвори) за дескриптор корисно брати число Ейлера, що визначається як різниця між числом зв'язкових областей і числом отворів. Зокрема, числа Ейлера для букв *A* і *B* відповідно рівні 0-1. Інші дескриптори області розглянемо нижче.

1.8.2 Текстура

У багатьох випадках ідентифікацію об'єктів або областей образу можна виконати за допомогою дескрипторів текстури. Хоча й не існує формального визначення текстури, проте, інтуїтивно цей дескриптор можна сформулювати як опис властивостей поверхні (однорідність, шорсткість, регулярність). Двома основними підходами для опису текстури називають статистичний і структурний. Статистичні методи відповідають за такі характеристики текстури, як однорідність, шорсткість, зернистість тощо. Структурні методи встановлюють взаємне розташування елементарних частин образу, на кшталт, опису текстури, що ґрунтується на регулярному розташуванні паралельних ліній [10].

1.8.3 Схема області

Важливим підходом для опису виду структури плоскої області є її представлення у вигляді графа. У багатьох випадках необхідно визначити схему (скелет) області за допомогою проріджуючих (чи скорочуючих) алгоритмів. Проріджуючі процедури відіграють ключову роль у широкому діапазоні завдань комп'ютерного зору: від автоматичної перевірки друкованих плат до підрахунку азбестових волокон у повітряних фільтрах. Скелет області можна визначити за допомогою перетворення середніх осей (ПСО), запропонованого в роботі. Хоча ПСО окреслює досить задовільний скелет області, його пряме використання є складним із обчислювальної точки зору, оскільки необхідно визначити відстані між кожною точкою області і межею. Було запропоновано низку алгоритмів побудови середніх осей, які наділені більшою обчислювальною ефективністю. Як правило, це алгоритми проріджування, що ітеративно усувають з опрацювання точки контуру області так, щоб виконувалися такі обмеження:

- не усувати крайніх точок;
- не призводити до порушення зв'язності;
- не викликати надмірного розмивання області.

1.9 Сегментація й опис тривимірних структур

Фактично зір є тривимірною проблемою, тому за основу розробки багатофункціональних систем технічного зору, що придатні для роботи в різних середовищах, взято процес обробки інформації про тривимірні сцени. Хоча дослідження цієї галузі проводяться вже більше 10 років, такі чинники, як вартість, швидкість, а також складність, гальмують впровадження обробки тривимірної зорової інформації до промислових додатків.

Існують три основні форми щодо подання інформації про тривимірну сцену. Якщо застосовувати датчики для вимірювання відстані, то отримаємо координати (x, y, z) точок поверхонь об'єктів [10]. Використання пристроїв, які створюють стереозображення, надає тривимірні координати, а також інформацію про освітленість у кожній точці. У даному випадку кожна точка подається як функція $f(x, y, z)$, де значення останньої в точці з координатами (x, y, z) набувають значення інтенсивності в цій точці (для позначення точки в тривимірному просторі та її інтенсивності, зазвичай, використовують термін воксель). Отже, можна встановити тривимірні зв'язки на базі одного двовимірного образу сцени, тобто можна виводити такі зв'язки між об'єктами, як «над», «за», «перед». Оскільки точне тривимірне розташування точок сцени здебільшого не може бути обчислено на основі одного зображення, зв'язки, що отримані шляхом цього виду аналізу, іноді належать до так званої 2,5-мірної інформації.

1.10 Опис тривимірної сцени плоскими ділянками

Один із найпростіших підходів для сегментації та опису тривимірних структур за допомогою координат точок (x, y, z) поєднується у разі необхідності розділення сцени на невеликі плоскі «ділянки» з подальшим їхнім об'єднанням у більші елементи поверхні відповідно до деякого

критерію. Такий метод особливо придатний для ідентифікації багатогранних об'єктів, поверхні яких досить гладкі щодо роздільної здатності.

Коли сцена задана вокселем, то її можна описати плоскими ділянками за рахунок тривимірного градієнта. У зазначеному випадку дескриптори поверхні також з'являються в результаті об'єднання цих плоских ділянок. Вектор градієнта вказує напрямок максимальної швидкості зміни функції, а його величина відповідає величині цієї зміни. Такі поняття застосовані для тривимірного випадку, крім того, можуть використовуватись для розбиття на сегменти тривимірних структур у такий же способом, як для двовимірних даних.

1.11 Техніка обробки візуальної інформації

Форма і розмір. Будемо вважати всі об'єкти, що задіяні в системі автоматичного складання, точними проекціями відповідних реальних об'єктів. Об'єкти можуть бути будь-якої форми, а їхня кількість – необмеженою. Будь-яке виявлене аналізується системою відхилення від форми або розміру, що може сприяти появі помилки збірки, служить підставою для відбраковування деталі (це не означає, що буде виконано перевірку 100 % деталей). У зв'язку з цим отриманий у результаті аналізу опис об'єкта повинен містити всю існуючу в образі інформацію про розміри і форми цього об'єкта. Навпаки, система розпізнавання символів у загальному випадку повинна допускати значні відхилення (різне накреслення) в розпізнаваних символах [10]. У найскладнішому випадку розпізнаються рукописні символи. Проте число типів розпізнаваних символів завжди є обмеженим. Найбільша вірогідність розпізнавання досягається під час застосування шаблону для аналізу друкованих шрифтів, спеціально призначених для оптичного зчитування, зокрема, ОКР-А й ОКР-Б (OCR-A и OCR-B).

Параметри стану. Деталі, з якими працює робот-складальник, можуть знаходитися в будь-якій точці в межах поля зору, а також мати довільну орієнтацію. З огляду на те, що для розпізнавання деталі потрібен її опис, інваріантне до параметрів положення, то визначення значень вказаних параметрів також є важливою функцією аналізатора візуальної інформації. Значення параметрів положення застосовуються під час виконання таких операцій. Задача розпізнавання символів спрощується за рахунок того, що, хоча рядки символів можуть бути похилими, окремі символи, як правило, потрапляють до поля зору, що розпізнає системи в доволі жорстко фіксованому положенні, вимірювати параметри якого немає необхідності.

Кількість об'єктів. Природно вимагати від зорового аналізатора робота-складальника вміння працювати одночасно з декількома деталями, що знаходяться в полі зору та розташовані випадковим чином. Розв'язання завдання щодо розподілу об'єктів під час розпізнавання символів полегшується тим, що попередньо відомі розміри та курсив символу (в разі розпізнавання друкованих символів). Дана інформація дозволяє створити прийоми поділу дотичних символів [10]. У нашій роботі не ставиться завдання розрізнити дотичні об'єкти. Їхній сукупний силует сприйматиметься як єдиний, імовірно, невідомий об'єкт.

Відсутність обмеження на кількість типорозмірів об'єктів, із якими доводиться працювати розпізнавальній системі, призводить до необхідності створювати повний опис форми для кожного образу. Проте існує кілька основних параметрів, які визначаються однозначно для контуру будь-якої конфігурації. Використовуючи їх, можна отримати необхідну інформацію про тип і стан об'єкта.

До числа цих параметрів належать: площа, периметр, мінімальний охоплюючий прямокутник (розміри), центр площі (центр ваги), радіус-вектор мінімальної довжини (довжина та напрям), радіус-вектор максимальної довжини (довжина та напрям), характеристики отворів (кількість, розмір, розташування).

Значення площі та периметра є найпростішими інваріантними до орієнтації та розташування характеристики. Під час розв'язання задачі розпізнавання об'єкта було застосовано безрозмірний "коефіцієнт форми" – площа/периметр. Координати вершин, охоплюючого мінімального прямокутника, подають деяку інформацію щодо форми та розміру об'єкта, проте їхнє значення залежить від його орієнтації. Центр площі - це точка, що легко визначається для будь-якого об'єкта незалежно від його орієнтації. У зв'язку з цим центр площі відіграє важливу роль під час виявлення та розпізнавання об'єкта. Він же задає початок радіус-вектора, що з'єднує центр площі з точкою, що лежить на межі об'єкта. В основному радіус-вектори максимальної та мінімальної довжини можуть бути корисними для розпізнавання та визначення параметрів положення об'єкта.

Як правило, в промислових деталях є отвори. Їхня кількість також є корисною характеристикою. Отвори можуть розпізнаватися як об'єкти певної форми та розміру, що певним чином орієнтовані на основний об'єкт, в якому вони виявлені.

Найбільш зручним способом отримання докладного опису довільного геометричного об'єкта та встановлення зв'язності образу можна назвати простежування межі. Відстеження починається з вибору довільної точки, що лежить на чорно-білій межі проекції об'єкта. Потім виконується алгоритм, який послідовно визначає суміжні точки, що розташовані на межі, доти, поки не буде виконано повний обхід контуру. Записуючи послідовно напрямки руху вздовж межі від кожної поточної точки до наступної, отримуємо одновимірний опис проекції об'єкта, що містить інформацію про його форму. Такі ланцюжки напрямків всебічно досліджувалися Фріменом [10]. Коли вже простежено межу, легко визначити площу, периметр, центр площі та охоплюючий прямокутник. Опис межі також можна застосовувати і для визначення радіус-векторів.

Відстеження межі встановлює зв'язність аналізованого об'єкта. Продовжуючи перегляди у пошуках наступних об'єктів, можна повторно виявити вже відстежену межу.

Щоб уникнути цю дію, Розенфельд пропонує у всіх точках, які вже переглянуті програмою відстеження межі, замінювати одиниці на двійки для першого оброблюваного об'єкта, на трійки - для другого і далі таким чином (одиниця відповідає об'єкту або "чорному" елементу растра в вихідному образі, а нуль – фону або "білому" елементу). Однак такий підхід потребує більше одного біта для представлення кожної точки вихідного образу, тому не придатний для систем із малим розміром пам'яті.

Для вирішення цього завдання автори пропонують залучати оператор поточечного диференціювання або виділення межі, відразу ж застосовуючи його після зчитування образу телекамерою. Оператор замінює всі одиниці в образі на нулі, за винятком точок, які лежать на межі чорного та білого полів. Отримані за допомогою ЕОМ контури квадрата і кола, що розташовані в полі зору камери, наведено на рисунку 1.2. Тут показано результат застосування до цього зображення оператора виділення межі. Процедура простеження межі EDGETRACE може працювати з проміжним зображенням за аналогією до вихідних об'єктів. Вона побудована тааким чином, що елементи растра переглянуті нею "обнуляються", іншими словами, простежування межі об'єкта видаляє його із зображення, при цьому робить неможливим повторне виявлення.

Згладжування. Метод обробки зображень об'єктів, який ґрунтується на простежуванні меж, може виявитися непрацездатним, у разі, якщо на межі об'єкта буде виявлено різкі скачки контрастності. У системах розпізнавання символів простежування меж здебільшого випереджується операцією згладжування (локального усереднення), що намагається знищити просвіти (білі крапки, що мають бути чорними), а також помилкові чорні точки, крім того, з'єднати невеликі розриви. В операції згладжування за Дайніном на кожен елемент растра по черзі накладається вікно розміром $n \times n$ елементів і

підраховується кількість чорних елементів, які потрапили до вікна. Створюється нове растрове уявлення, де кожен елемент відповідає окремій позиції вікна. У такому растровому поданні елемент буде чорним тільки в тоді, коли кількість чорних елементів, які потрапили до відповідного йому вікна, перевищує деяке заздалегідь задане число. Найменший розмір вікна, що застосовується на практиці, 3×3 елемента. Унгер запропонував процедуру згладжування, в якій замість усереднення вмісту вікна розміром 3×3 елемента до нього застосовують деякі логічні операції для визначення значення елемента нового образу. Недолік обох методів полягає у досить великому обсязі операцій із пам'яттю: в обох випадках усі визначення значення елемента нового способу доводиться виконати за допомогою приблизно 100 машинних операцій.

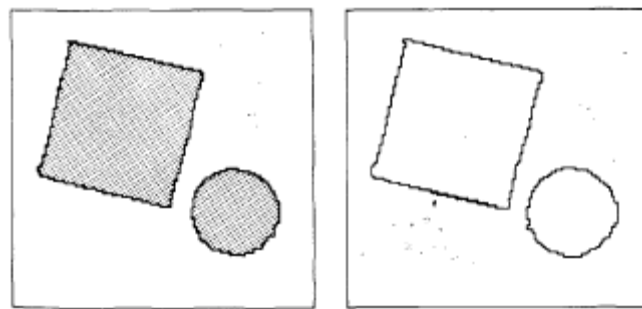


Рисунок 1.2 – Вміст растру після процедури INPUT-FRAME
і вміст растру після виділення меж

Об'єкти, з якими має справу робот-складальник, відрізняються від друкованих символів тим, що будь-які просвіти в відповідних їм образах означають наявність дефектів і служать достатньою підставою для відбраковування такого об'єкта [10]. Зрідка виникають ізольовані шумові точки в загальному випадку не впливають на роботу алгоритму простеження межі. Тому спеціальні згладжувальні операції не використовуються, хоча в процедурі простеження вживаються заходи для зменшення впливу шумових точок.

Виділення межі – процедура `OUTLINEFRAME`. Оператор виділення межі повинен створювати безперервну послідовність граничних точок, забезпечуючи можливість простеження межі. Для мінімізації часу обчислення при створенні кожного елемента нового способу слід використовувати доступ до мінімально можливої кількості елементів вихідного образу.

Варіанти визначення контуру об'єкта: Розенфельд розрізняє поняття "край" і "межа". Край – це сукупність зовнішніх елементів растрового представлення об'єкта, межа же проходить посередині між горизонтально або вертикально суміжними точками, одна з яких належить об'єкту, а інша – фону. Для наших цілей будемо використовувати концепцію межі, так як вона більш точно представляє лінію, реально розділяючу чорні і білі області растра.

Ця концепція забезпечує однозначність рішення в разі лінії товщиною в одну точку. Використання концепції краю призведе до отримання результату, непридатного для подальшої роботи алгоритму простеження, так як два края лінії будуть збігатися.

Недолік визначення межі по Розенфельду полягає в тому, що підсумковий масив має вдвічі більшу точкову щільність, ніж вихідний, при збереженні взаємно однозначної відповідності двох масивів. Новий масив граничних точок створюється з такою ж відстанню між точками, як вихідний, але при цьому кожна точка зсувається за координатами x і y аполовину цієї відстані. Кожна точка нового масиву лежить в центрі квадрата, в вершинах якого знаходяться точки вихідного масиву, а її стан (чорне або біле, межа або не межа) визначається станом цих чотирьох точок.

Шість основних (з числа шістнадцяти можливих) поєднань станів чотирьох суміжних елементів вихідного образу. Там же наведені інтуїтивні певні значення відповідних елементів масиву, що містить опис межі. Значення для конфігурації d вибрано нульовим для забезпечення однозначності в разі лінії товщиною в одну точку [10].

Відстеження межі – процедура EDGETRACE.

Параметри визначувані процедурою EDGETRACE наведено у таблиці 1.1.

Таблиця 1.1 – Параметри, визначувані процедурою EDGETRACE

Змінна	Значення
PERIMETER	Довжина простеженого контуру, позитивне ціле число
AREA	Площа, обмежена контуром, позитивне ціле число
XMAX XMIN YMAX YMIN	Максимальне і мінімальне значення координат точок, що належать контуру, негативні цілі числа
XCENTROID YCENTROID	Координати центру площі, обмеженою контуром, негативні цілі числа
CHAINCOUNT	Кількість елементарних векторів, що утворюють контур, що збігається з кількістю переглянутих граничних точок, від'ємне ціле число
CHAIN	Впорядкований масив значень, задають напрямки елементарних векторів, що утворюють контур. Масив складається з елементів CHAINCOUNT

EDGETRACE – процедура, яка визначає, чи належить межі об'єкта точка, виявлена в диференційованому способі. Координати початкової точки вводяться в глобальні змінні цілого типу STARTX, STARTY. Процедура EDGETRACE викликається як логічна процедура – функція, наприклад, в операторі

```
IF EDGETRACE THEN ... ELSE
```

Значення "істина" виробляється процедурою тоді, коли знайдена замкнута межа, що включає в себе більше заздалегідь певного числа точок.

Процедура EDGETRACE визначає значення сукупності параметрів, що відносяться до контуру, а також створює опис самого контуру у формі списку векторів, представлених глобальними змінними.

Визначення значень XM AX, XMIN, YMAX, YMIN зроблено необов'язковим. Якщо воно бажано, слід встановити ознака слова EDGEF-LAG

(тобто зробити значення відповідного слова ненульовим). Точно так же за допомогою ознаки CHAINFLAG можна включати і вимикати запис значень в лічильник CHAINCOUNT і масив CHAIN.

Ланцюговий запис і основний алгоритм. У даній роботі точка матриці, відповідної образу, вважається пов'язаною з іншою точкою, якщо остання знаходиться в одній з восьми позицій, безпосередньо суміжних з першою точкою.

З'єднання будь-якої пари пов'язаних точок може бути представлено одним з восьми елементарних векторів, напрямки яких позначені значеннями від -8 до -1 (негативні значення взяті для поєднання зі схемою індексації EOM PL-516).

Алгоритм простежування межі "обнуляє" значення знайденої точки, що належить контуру, до переходу до наступної точки. Отже, з матриці FRAME видаляються простежені нариси. Базова точка пошуку переміщається по контуру проти годинникової стрілки, що при пошуку контурів за допомогою розгортки забезпечує перше входження в контур ззовні. Увійти в контур зсередини можна тільки в тому випадку, якщо він перетинає межу матриці. При цьому процедура EDGETRACE видасть повідомлення про помилку.

Немає необхідності перевіряти в кожній точці контуру об'єкта всі вісім сусідніх точок – деякі з них уже були перевірені на попередньому кроці, коли базовою точкою пошуку була попередня. До початку локального пошуку в залежності від напрямку попереднього вектора встановлюються змінні SEARCHDIRN (перше перевіряється напрямком) і ENDDIRN (останнє перевіряється напрямком).

На кожному кроці пошуку проглядається до шести точок, але зазвичай їх число менше. На прямолінійних ділянках контура до знаходження наступної його точки найчастіше досить переглянути одну-дві точки.

Повний опис нарисового контуру формується у вигляді списку напрямків векторів, що з'єднують кожен точку контуру з наступною в тому порядку, в якому вони були знайдені (масив CHAIN). Результат застосування процедури простежування межі до простого контуру і отриманий "ланцюговий" опис показані на рисунку 1.3.

x (-7, -8, -1) – негативні (Y має негативне значення). При таких умовах площа, обмежена контуром, прослідковується проти годинникової стрілки, матиме позитивне значення.

Номери та значення ланцюжка наведено у таблиці 1.2.

Таблиця 1.2 – Номери та значення ланцюжка

Номер елемента ланцюжка	Значення елемента ланцюжка
-1	-2
-2	-2
-3	-2
-4	-8
-5	-1
-6	-2
-7	-8
-8	-8
-9	-8
-10	-8
-11	-6
-12	-6
-13	-6
-14	-6
-15	-5
-16	-5
-17	-4
-18	-4
-19	-3
-20	-3

Щоб уникнути зайвих маніпуляцій з коефіцієнтом $1/2$ процедура EDGETRACE накопичує подвоєне значення площі (AREA2) і в кінці обчислень ділить його навпіл.

Визначення перших моментів площі і центрів. Аналогічний метод використовується для обчислення моментів обмеженої площі першого порядку щодо осей x і y . Вісім елементарних векторів і перші моменти для площ, укладених між векторами і віссю x (AW_x). Сума всіх компонентів $ДЛ/x$ для контуру дасть негативний ціле число, яке після поділу на значення обмеженої площі дасть координату її центра по осі y .

Значення площ і прирощень наведено у таблиці 1.3.

Таблиця 1.3 – Значення площ і прирощень

Напрямок вектора	Прирощення площі	Подвоєне прирощення
-8	Y	$2Y$
-7	$Y-1/2$	$2Y-1$
-6	0	0
-5	$(Y-1/2)(-1)$	$-2Y+1$
-4	$Y(-1)$	$-2Y$
-3	$(Y+1/2)(-1)$	$-2Y-1$
-2	0	0
-1	$Y+1/2$	$2Y+1$

Вираз для AM_x у разі векторів з напрямками, що мають непарні номери, містить постійний член $1/6$ або $-1/6$. У замкнутому контурі, що складається з великого числа елементарних векторів, можна вважати кількості векторів всіх напрямків приблизно однаковим. Це призводить до взаємного знищення постійних членів. Крім того, постійний член досить малий у порівнянні з Y^2 і з площею, особливо у тому випадку, коли об'єкт розглядається поблизу і його образ займає більшу частину поля зору [10]. Тому можна спростити обчислення, нехтуючи постійним членом (рисунок 1.4).

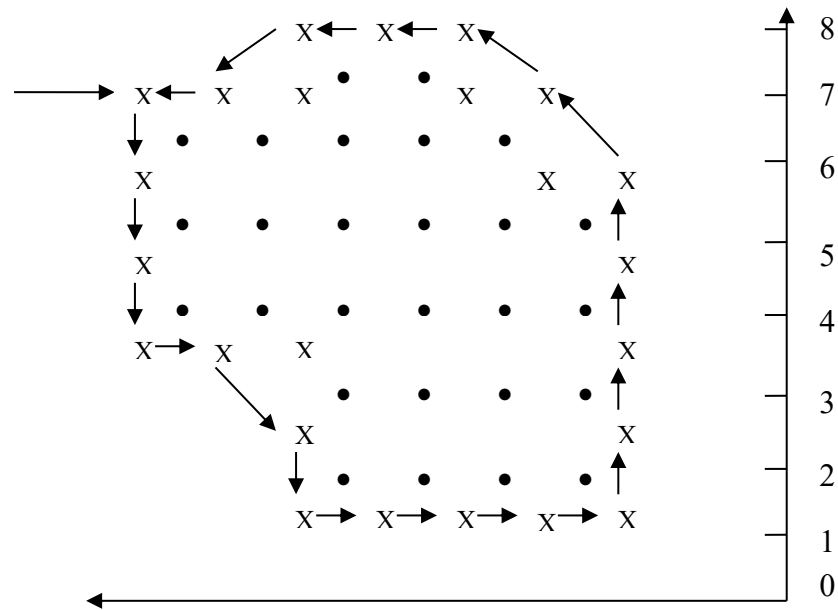


Рисунок 1.4 – Обчислення площі, обмеженою контуром

Операцій з коефіцієнтом $1/2$ уникають так само, як при обчисленні площі, – обчислюючи значення 2 ДМ_x . Це значення обчислюється подвоєною довжиною XMOMENT процедурою SUMXMOMENT.

При завершенні процедури EDGETRACE XMOMENT ділиться на значення AREA2. Округляючи результат до найближчого цілого, отримуємо координату Y центру площі – YCENTROID.

Точно так же, підсумовуючи моменти площі щодо осі у отримуємо значення XCENTROID.

Одне або невелике число суміжних "шумових" точок, відповідних, наприклад, що випадково потрапило у поле зору уламку стружки, утворюють контури, що складаються з малого числа елементарних векторів. Відповідним значенням для змінної MINCHAIN виявилось -10, тобто, будь-який замкнутий контур, довжина якого не перевищує 10 векторів, просто не розглядається.

Знаходження сумарної площі наведені у таблиці 1.4.

Таблиця 1.4 – Знаходження сумарної площі

Номер елемента ланцюжка	Значення елемента ланцюжка	Подвоєне прирощення	Подвоєна площа
-1	-2	0	0
-2	-2	0	0
-3	-2	0	0
-4	-8	-8	-8
-5	-1	-7	-15
-6	-2	0	-15
-7	-8	-4	-19
-8	-8	-4	-23
-9	-8	-4	-31
-10	-8	0	-31
-11	-6	0	-31
-12	-6	0	-31
-13	-6	0	-31
-14	-6	+13	-31
-15	-5	+15	-18
-16	-5	+15	-3
-17	-4	+16	+13
-18	-4	+16	+29
-19	-3	+15	+44
-20	-4	+14	+58
Сумарна площа = 29			

Значення моментів відносно осі X і Y наведено у таблицях 1.5-1.6.

Таблиця 1.5 – Значення моментів відносно осі X

Напрямок вектора	Прирошення площі	Подвоєне прирошення
-8	$Y^2/2$	Y^2
-7	$Y^2/2 - Y/2 + 1/6$	$Y(Y-1)$
-6	0	0
-5	$-Y^2/2 + Y/2 - 1/6$	$Y(-Y+1)$
-4	$-Y^2/2$	Y^2
-3	$-Y^2/2 + Y/2 - 1/6$	$Y(-Y+1)$
-2	0	0
-1	$Y^2/2 - Y/2 + 1/6$	$Y(Y-1)$

Таблиця 1.6 – Значення моментів відносно осі Y

Напрямок вектора	Прирошення площі	Подвоєне прирошення
1	2	3
-8	0	0
-7	$X^2/2 + X/2 + 1/6$	$X(X+1)$
-6	$X^2/2$	X^2
-5	$X^2/2 - X/2 + 1/6$	$X(X-1)$
-4	0	0
-3	$-X^2/2 + X/2 - 1/6$	$X(-X+1)$
-2	$-X^2/2$	$-X^2$
-1	$-X^2/2 + X/2 - 1/6$	$X(-X-1)$

Матриця образу після процедури EDGETRACE. Описаний алгоритм простеження меж, обробляючи диференційовану матрицю, обнуляє всі переглянуті точки, але це не означає, що будуть переглянуті і обнулені всі точки контуру. Що залишилися в матриці одиничні точки зазвичай не становлять повного контуру і при подальших операціях виступають як "шумові". Для мінімізації їх впливу повинні вживатися спеціальні заходи. На рисунку 1.5 показаний вид матриці образу FRAME після того, як

процедура EDGETRACE опрацювала обидва контури. Видно, що залишилися шумові точки.

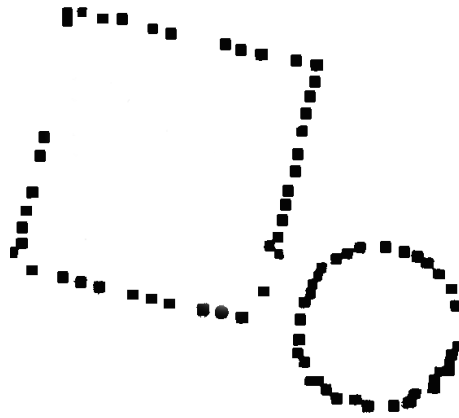


Рисунок 1.5 – Матриця зображення після простежування обох меж

Ідентифікація отворів – процедура INSIDE. Процедура INSIDE визначає, всередині або зовні певного контуру лежить деяка точка, що належить матриці образу FRAME [10].

Передбачається, що контур, виділений процедурою EDGETRACE, зберігається у вигляді масиву CHAIN і змінна CHAINCOUNT визначає його довжину. Початкова точка ланцюжка задається цілими змінними PARTXSTART и PARTYSTART.

Координати перевіряється точки вводяться в змінні STARTX і STARTY і викликається логічна процедура-функція INSIDE. Вона виробляє значення "істина", якщо перевіряється точка лежить всередині контуру. При цьому ніякі значення в пам'яті не змінюються.

Використовуваний алгоритм заснований на наступній справедливою для площині теоремі: якщо точка P , що лежить поза замкнутої кривої S , з'єднана відрізком прямої з точкою Q , то точка Q теж лежить поза кривої S тоді і тільки тоді, коли відрізок PQ перетинає криву S парне число раз. Якщо PQ перетинає криву S непарне число раз, точка Q лежить всередині замкнутої кривої S (рисунок 1.6). Доказ приводиться Курантом і Роббінсом.

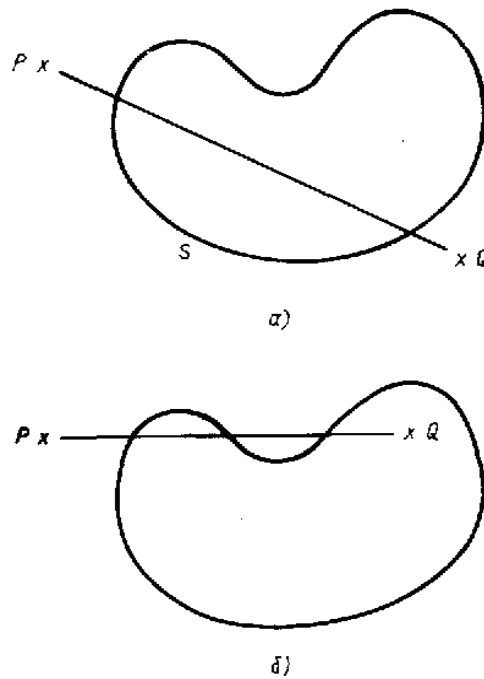


Рисунок 1.6 – До теореми про перетин прямої і замкнутої лінії

Пара локальних змінних (XCOORD і YCOORD) спочатку встановлюються рівними координатами початку зберігання контуру. З масиву CHAIN послідовно зчитуються елементарні вектори. Залежно від їхнього значення модифікуються XCOORD і YCOORD (процедура UNCHAIN). Отже, XCOORD та YCOORD послідовно приймають значення координат усіх точок, які прослідковані у вихідному контурі [10].

На кожному кроці перевіряється, чи не лежить точка контуру на промені $Y = \text{STARTY}$, $x > \text{STARTX}$. Кількість перетинів контуру з цим променем підраховується в цілій змінній CROSSCOUNT. Множинні перетину, за умови, що лінія контуру збігається з тестовим променем в кількох суміжних точках, не враховуються. В момент входження лінії контуру тестовий промінь у змінній STRIKEDIRN записується індекс напрямку вектора, що входить в промінь. Для наступних точок, які лежать на промені, ніяких дій не виконується. Коли черговий вектор контуру виходить із тестового променя, поточний напрямок порівнюється зі значенням, яке записано у STRIKEDIRN для перевірки: чи був пересічений контуром

тестовий промінь (CROSSCOUNT збільшується на одиницю) або контур пройшов відносно до променя (CROSSCOUNT не змінюється).

Коли всі точки контуру зберігаються, вони будуть відновлені та перевірені. Перевіряється значення CROSSCOUNT і процедура завершується з відповідним результатом. Неоднозначні граничні випадки, коли перевірені точки лежать на самому контурі, усуваються, тому що в процесі простежування межі все-таки точки обнулені.

Після того, як виявлені контури об'єкта, необхідно визначити його орієнтацію та положення. Ця інформація є необхідною для правильної роботи з деталлю в процесі складання.

Завдання визначення орієнтації. Основні процедури обробки візуального образу містять в ньому контури (об'єкта й отворів у ньому) в параметрах площі, периметра, положення центру площі, що охоплюють прямокутник, опису контуру ланцюжком векторів. Такі параметри придатні для виконання початкової обробки, проте для розпізнавання образу та визначення його орієнтації необхідно більше опрацьовувати інформації про отвори та контури.

Модель об'єкта з отворами. Рисунок, який утворюється отворами, наявними в об'єкті, можна аналізувати із використанням відстані центрів площ отворів від центру площі об'єкта та їхніх відносних кутових координат. До того ж, спиратися на вже виміряні площі та периметри отворів.

Ця модель інваріантна до орієнтації основного об'єкта, тому може безпосередньо порівнюватися з моделлю-зразком. Коли буде встановлено відповідність між окремими отворами розглянутого об'єкта та моделлю, значення її абсолютних кутових координат можна застосовувати для визначення орієнтації об'єкта.

Виділення особливостей контуру. Модель контуру подається як список – ланцюжок векторів. Однак порівнювати вхідну інформацію, представлену такою моделлю, зі зразком, як це робить, зокрема, Фрімен, довго і складно. Головна складність полягає в тому, що початкова точка ланцюжка, що

виражає контур об'єкта, розташована на ньому випадковим чином. Її конкретне положення залежить від орієнтації об'єкта та напрямків сканування під час перегляду. Порівнювані ланцюжки доводиться циклічно зміщувати відносно один одного до збігу. На жаль, опис контуру ланцюжком векторів є не дуже зручним для проведення операції повороту на кут, не кратний 45° . Встановлення відповідності ще більш ускладнюється, оскільки еталон та вхідний ланцюжок у загальному випадку складаються з різного числа елементарних векторів.

Розроблено низку методів згладжування ланцюгових списків і нормалізації їхніх довжин, які забезпечують можливість порівняння. Зах і Монтанари описали методи отримання аппроксимацій шляхом використання прямих мінімальної довжини, що знижують обсяг даних, які підлягають обробці. Але такі методи занадто складні для застосування в системі автоматичного складання. У такій системі основною вимогою є не забезпечення 100 % збігу з еталонами, за умови порівняння моделі контурів, виділених із вхідного зображення, а підтвердження результатів пробних перевірок і визначення орієнтації об'єкта.

Необхідно обрати модель, яка дозволить найбільш швидко та легко визначати орієнтацію об'єкта. Найбільш підходящим для такої моделі може бути підхід, який ґрунтується на виділенні особливостей нарисового контуру. Значна кількість методів розпізнавання символів, наприклад, за Хоскінгом, використовують повторювані особливості структури штрихів, які створюють символ. Такими особливостями є кінці штрихів, точки їхнього з'єднання або зламу, петлі. Аналогічні особливості, характерні для матеріальних об'єктів, можуть проявлятися через різкі зміни напрямку контуру (кути) і точки, максимально і мінімально віддалені від центру площі фігури. На жаль, відсутність обмежень щодо форми деталей, з якими доводиться мати справу роботу-складальнику, обмежує цінність методів, які застосовують раніше складений набір стандартних особливостей.

Модель із використанням кіл. Кутові координати радіус-векторів максимальної та мінімальної довжини можуть бути зручним критерієм для розпізнавання об'єкта, а також засобом визначення його орієнтації. Їхнє значення визначається безпосередньо з інформації, що генерується процедурою EDGETRACE. Але якщо в околиці точки, що екстремально віддалена від центру площі, довжина радіус-вектора мало змінюється (рис. 18), координати R_{MAX} та R_{MIN} будуть ненадійними. З огляду на це, більш доцільним під час налаштування системи є варіант задавати такі значення довжини радіус-вектора R , для яких відповідна позиція кінцевої точки може бути чітко визначеною. Фактично, це зводиться до накладання на образ кола заданого радіуса, центр якого збігається з центром площі об'єкта. Точки перетину цієї окружності з контуром об'єкта окреслюються як його особливості. На рисунку 19 особливі точки визначаються параметрами R , θ_2 , θ_3 , θ_4 . Різниці кутів становлять інваріантний до орієнтації набір параметрів для розпізнавання, а їхні абсолютні значення можуть застосовуватися для визначення орієнтації об'єкта.

Точки перетину визначаються за інформацією, згенерованою процедурою EDGETRACE: для кожної точки контуру, представленого ланцюговим списком векторів, обчислюється її відстань від центра площі. Окреслюються точки, для яких ця відстань дорівнює заданому радіусу, та обчислюються їхні кутові координати відносно центру площі. Отже, точки перетину подаються у вигляді впорядкованого списку. Яка з цих точок знаходиться на початку зазначеного списку - залежить від того, з якого місця контуру починалася процедура EDGETRACE.

Критеріями з вибору значення радіуса кола можуть бути такі положення:

- точки перетину повинні визначати орієнтацію об'єкта однозначно;
- кількість точок перетину не повинно бути занадто великою (наприклад, не більше восьми);

- для підвищення точності вимірювань радіус повинен бути великим за можливістю;
- незначні зміни довжини радіуса не мають суттєво впливати на кутові координати точок перетину;
- незначні зміни довжини радіуса не повинні призводити до зникнення або появи нових точок перетину.

Останні два пункти необхідні для зниження впливу незначних спотворень у вхідному поданні образу.

Щоб однозначно визначити орієнтацію деталі або відрізнити її від такої ж іншої, може виявитися за необхідне поставити більше однієї окружності. У такому разі точки перетину, знайдені шляхом обходу контуру, будуть представлені у вигляді впорядкованого списку пар значень радіус-кут.

Порівняння значень вимірних кутів необхідно виконувати тільки тоді, коли наявний повний збіг списків позначень радіусів. Це спрощує процедуру порівняння. Коли встановлено відповідність обох списків, можна обчислювати різниці абсолютних кутових координат для кожної пари суміжних точок перетину. Шість значень призначені для визначення орієнтації одного об'єкта щодо іншого. Ще однією властивістю моделі з використанням кіл – можливість фіксувати випадок, коли деталь лежить зворотною стороною вгору. Отже, така модель являє собою потужний засіб визначення особливих точок на рисовому контурі об'єкта довільної форми. Такі точки можуть застосовуватись для розпізнавання та визначення орієнтації. Обсяг пам'яті, необхідний для зберігання моделі-еталона, становить всього два слова на кожну точку перетину, а необхідні дані про скановані деталі можуть надходити з інформації, що генерується процедурою EDGETRACE. У поєднанні з моделлю, що використовує отвори, вона реалізується на нижчому рівні розпізнавання та визначення орієнтації деталей у системі керування робота-складальника.

Отже, в першій частині магістерської кваліфікаційної роботи проаналізовано методи обробки інформації в системах технічного зору

роботів. Процес ідентифікації об'єктів, які розташовані в робочій зоні робота, містить два етапи: виділення характерних ознак об'єктів і розпізнавання об'єктів шляхом знайденої сукупності характерних ознак. Основними методами обробки інформації є сегментація (процес розподілу сцени на складові частини чи об'єкти), визначення порогового рівня, обласно-орієнтована сегментація, дескриптори меж і областей зображень, опис тривимірних сцен і структур, обробка візуальної інформації.

2 МЕТОДИ ІДЕНТИФІКАЦІЇ ДЕТАЛЕЙ У РОБОТИЗОВАНИХ СИСТЕМАХ

2.1 Метод порівняння з еталоном

Ідентифікація об'єктів у СТЗ найчастіше відбувається із залученням методів порівняння з еталоном. При цьому, зазвичай, СТЗ вирішують одне з двох завдань. Перше завдання – отримання зображення одного об'єкта та порівняння з усіма зразками заданого класу. За збігом обирається найкращий еталон та виконується ідентифікація об'єкта, а потім за необхідності знаходяться параметри його положення й орієнтації [11].

Друге завдання – отримання зображення декількох об'єктів та почергове їхнє порівняння з еталоном об'єкта, що необхідно виділити. Далі обчислюються його положення й орієнтація в робочій зоні робота.

Метод ґрунтується на встановленні збігу двох точкових зображень. Якщо збіг присутній, то вважається, що предмет розпізнано. Перевага методу накладення еталона полягає в незалежності від конкретного виду та складності об'єкта. У випадку двовимірних систем розпізнати ключ не складніше за коло чи будь-яку іншу фігуру. Проте даний метод розпізнавання може забрати багато часу під час використання обчислювальної машини. Можливі труднощі продемонструємо на прикладі. На малюнку зображено контури, відповідні кубу і піраміді. Припустимо, ставиться завдання визначити методом збігу зображень, чи містить ця фігура квадрат де-небудь. Виготовимо еталон з ряду чорних клітин, які утворюють квадрат, оточений зовні та всередині білими клітинами. Накладемо його на фігуру та переміщатимемо до встановлення збігу. Труднощі можуть виникати з багатьох причин:

– еталон повинен бути правильного розміру, в іншому випадку необхідно виготовити безліч еталонів, які відрізнятимуться один від одного лише масштабним збільшенням;

– невідомо, в якій частині фігури знаходиться передбачуваний квадрат. Отже, пошук необхідно проводити двояко: за зрушенням та обертанням;

– позначається просторове квантування. Навіть за умови правильного накладення еталона збіг не ідеальний, а найкращий з можливих. У зв'язку з цим метод порівняння з еталоном для роботів є актуальним лише за наявності інформації, що дозволяє заздалегідь обчислити розмір еталона, а також привести число порівнянь до реально здійсненого.

Метод порівняння з еталоном є цікавим в окремих випадках. Оскільки способи виготовлення оптичних кореляторів (і навіть електронних кореляторів, на кшталт дисектора зображень) відомі, весь час цей метод застосовувався в оптичних пристроях, які не встановлювалися на роботів. Проте відомі розміри використовуваних еталонів завжди заздалегідь, а предмети розташовані й орієнтовані належним чином. Йдеться лише про те, щоб розпізнати об'єкт із заданої множини об'єктів. Робот повинен вирішувати більш складні завдання.

Повного збігу об'єкта з еталоном у просторі вибраних ознак, зазвичай, не досягається. Отже, задається допустиме розходження між еталоном і зображенням, у межах цієї відмінності і перевіряється їхній збіг.

Брагін і Войлов [11] використовують таке трактування даного методу. Якщо позначити початкове зображення об'єкта $F(i, j)$, еталон $E(j, i)$, а обчислену різницю T , то процедуру порівняння можна представити як

$$T = \sum_i \sum_j |F(i, j) - E(i, j)|^2 ; \quad (2.1)$$

$$T = \sum_i \sum_j |F(i, j) - E(i, j)| . \quad (2.2)$$

Тут індекси i та j характеризують положення елементів в дискретному цифровому зображенні.

Збіг еталона з об'єктом перевіряється за правилом

$$D \geq T, \quad (2.3)$$

де D – задане граничне розходження.

Якщо умова не виконується, то необхідно замінити еталон або перейти до іншого зображення.

Покажемо особливості методу порівняння з еталонами при використанні деяких систем ознак. Якщо в якості ознак обрані площа і периметр зображення, розміри вписаних і описаних фігур, момент інерції і подібні геометричні властивості, то слід врахувати масштабування і на етапі попередньої обробки нормувати зображення по якомусь параметру. Наприклад, площа описаного прямокутника або кола нормується квадратом периметра зображення, а периметр – значенням кореня квадратного з площі зображення [11].

Зображення акумулятивної різниці формується в результаті порівняння еталонного образу з кожним чином в даній послідовності. У процедурі побудови зображення акумулятивної різниці є лічильник, призначений для обліку розташування пікселів. Його значення збільшується кожен раз, коли виникає відмінність у розташуванні відповідних пікселів еталонного образу і образу з розглянутої послідовності. Таким чином, коли k -й кадр порівнюється з еталонним, запис у цьому пікселі акумулятивної різниці означає, у скільки разів інтенсивність пікселя k -го кадру відрізняється від інтенсивності пікселя еталонного образу. Відмінності встановлюються, наприклад, з допомогою рівняння.

Нерідко корисно розглядати три типи зображень акумулятивний різниці: абсолютне, позитивне і негативне.

Успіх застосування методів залежить від еталонного образу, щодо якого проводяться подальші порівняння. Як вже говорилося вище, відмінність між двома образами в задачі розпізнавання рухомих деталей визначається шляхом виключення стаціонарних компонент при збереженні елементів, відповідних шуму і рухомих деталей. Проблема виділення образу з шуму вирішується методом фільтрації або за допомогою формування зображення акумулятивний різниці.

На практиці не завжди можна отримати еталонний образ, який має тільки стаціонарні елементи, і це призводить до необхідності побудови еталону з набору образів, що містять один або більше рухомих деталей. Це особливо характерно для ситуацій, що описують сцени з багатьма швидкоплинними об'єктами або у випадках, коли виникають часті зміни сцен. Розглянемо наступну процедуру генерації еталонного образу. Припустимо, що ми розглядаємо перший образ послідовності в якості еталонного. Коли нестаціонарна компонента повністю вийшла зі свого становища в еталонному кадрі, відповідний фон в даному кадрі може бути перенесений в положення, спочатку займане об'єктом в еталонному кадрі. Коли всі рухомі об'єкти повністю покинули свої початкові положення, в результаті цієї операції відтворюється еталонний образ, який містить лише стаціонарні компоненти. Переміщення об'єкта можна визначити за допомогою операції розширення позитивного зображення акумулятивний різниці.

2.2 Методи теорії графів і розпізнавання

При визначенні країв і контурів зображень застосовують методи графів. Розглянемо глобальний підхід, заснований на поданні сегментів контуру у вигляді графа і пошуку на графі шляху найменшої вартості, який

відповідає значущим контурам, описаний в книзі Брагіна і Войлова [11]. Цей підхід представляє наближений метод, ефективний при наявності шуму. Як і слід було очікувати, ця процедура значно складніше і вимагає більше часу обробки, ніж методи, викладені вище.

Спочатку дамо кілька простих визначень. Граф $G = (N, A)$ являє собою кінцеве, непорожня множина вершин N разом з множиною A неупорядкованих пар різних елементів з N . Кожна пара з A називається дугою.

Граф, в якому дуги є спрямованими, називається спрямованим графом. Якщо дуга виходить з вершини n_i , до вершини n_j , тоді n_j називається наступником вершини n_i . В цьому випадку вершина n_i називається попередником вершини n_j . Процес ідентифікації наступників кожної вершини називається розширенням цієї вершини. У кожному графі визначаються рівні таким чином, щоб нульовий рівень складався з єдиною вершини, званої початковою, а останній рівень – з вершин, званих цільовими. Кожній дузі $(n_i n_j)$ приписується вартість $c(n_i n_j)$. Послідовність вершин n_1, n_2, \dots, n_k , де кожна вершина n_i є наступником вершини n_{i-1} , називається шляхом від n_1 до n_k , а вартість шляху визначається формулою

$$c = \sum_{i=2}^k c(n_{i-1}, n_i). \quad (2.4)$$

Елемент контуру ми визначимо як межу між двома пікселями p та q . У даному контексті під контуром розуміється послідовність елементів контуру.

Важливим методом ідентифікації зображень з геометричними або іншими ознаками служить метод побудови графів рішень. Його успішно застосовують у тих випадках, коли в заданому класі зображень є об'єкти, які неможливо розрізнити за однією ознакою зображення, і для правильного розпізнавання необхідно використовувати кілька ознак. Від методу порівняння зображення й еталона по векторах ознак метод графів відрізняється тим, що в ньому на кожному етапі порівняння відбувається

відбір можливих рішень. Таким чином, число можливих рішень задачі розпізнавання зменшується на кожному етапі порівняння.

Граф (або дерево) розпізнавання по геометричними ознаками представлений на рисунку 2.1.

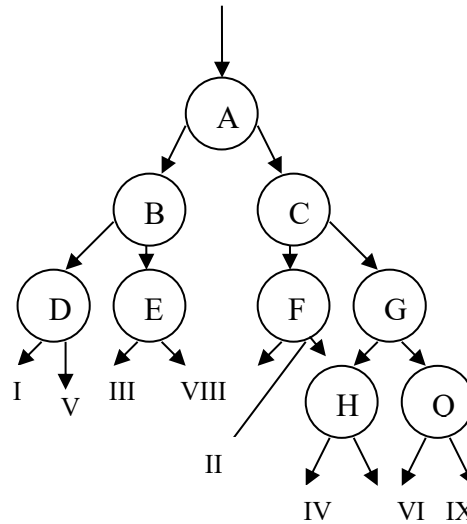


Рисунок 2.1 – Дерево розпізнавання

Цифрами I, II, ..., X позначені можливі рішення – номери розпізнаваних деталей. Букви A, B, ..., Q вершинах графа позначають оператори, які виділяють певні ознаки зображення. Наприклад, оператор A проводить класифікацію зображення по довжині і висоті описаного прямокутника, оператори B і C – за площею, DEFG можуть бути операторами, які проводять класифікацію за кількістю кутів, H і Q – по устоянню кутів один від одного. Граф може мати більше або менше рівнів, і зміст операторів може бути різним.

2.3 Кореляційний метод

Модифікацією методу порівняння з еталоном є кореляційний метод, заснований на обчисленні взаємкорреляційною функцією між еталоном і зображенням.

Кореляція – статистичний взаємозв'язок двох або декількох випадкових величин (або величин, які можна з деякою допустимою ступенем точності

вважати такими). При цьому, зміни однієї або декількох з цих величин призводять до систематичної зміни іншої або інших величин. Математичною мірою кореляції двох випадкових величин служить коефіцієнт кореляції.

Кореляційний аналіз – метод обробки статистичних даних, що полягає у вивченні коефіцієнтів кореляції між змінними. При цьому порівнюються коефіцієнти кореляції між однією парою або множиною пар ознак для встановлення між ними статистичних взаємозв'язків.

Мета кореляційного аналізу – забезпечити отримання деякої інформації про однієї змінної з допомогою іншої змінної. У випадках, коли можливе досягнення мети, кажуть, що змінні корелюють. В самому загальному вигляді прийняття гіпотези про наявність кореляції означає, що зміна значення змінної A , відбудеться одночасно з пропорційним зміною значення B .

Класифікація зображень проводиться по результату: чим більше значення функції взаємної кореляції, тим з більшою ймовірністю еталон збігається із зображенням. Використовуючи позначення, прийняті в вираженні, формулу для обчислення взаємкорреляційної функції K можна представити у вигляді

$$K = \sum_i \sum_j F(i, j) E(i, j)^2. \quad (2.5)$$

Максимальне значення взаємкорреляційної функції дорівнює,

$$\sum_i \sum_j |E(i, j)|^2, \quad (2.6)$$

і досягається при повному збігу зображення з еталоном. Нормована взаємкорреляційна функція

$$R = \frac{\sum_i \sum_j F(i, j)E(i, j)}{\sum_i \sum_j |F(i, j)|^2}, \quad (2.7)$$

при збігу еталона із зображенням досягає максимального значення, рівного одиниці.

Використання кореляційного методу і методу прямого порівняння з еталоном пред'являє до процесу попередньої обробки зображень загальні вимоги. Вони полягають в тому, що зображення і живий взірець повинні бути однаково орієнтовані, мати рівний масштаб і не бути зсунутими один щодо одного в полі зображення. Іншим властивістю цих методів, яке слід враховувати, є необхідність використання великої кількості еталонів. Це особливо важливо в тих випадках, коли вирішуються завдання розпізнавання деталей зміною їх проекції.

2.4 Розпізнавання через зв'язок шаблонів

2.4.1 Пошук деталей зазначенням зв'язків між шаблонами

Часто спостережуваний об'єкт володіє внутрішніми ступенями волі, а це означає, що його зовнішній вигляд може сильно варіюватися (наприклад, люди можуть рухати руками і ногами, риби деформуються при плаванні, звиваються змії тощо). Дане явище може надзвичайно ускладнити порівняння з шаблоном, оскільки потрібен або класифікатор з гнучкими межами (і множину зразків), або багато різних шаблонів.

Багато деталей названого типу містять невелику кількість компонентів, досить строго впорядкованих. Можна спробувати узгодити дані компоненти як шаблони, а потім визначити, які об'єкти присутні, вивчивши запропоновані зв'язку між знайденими шаблонами. Наприклад, замість пошуку особи по одному повного шаблоном особи, можна шукати очі, ніс і рот з прийнятним взаємним розташуванням.

Даний підхід має кілька потенційних переваг. По-перше, дізнатися шаблон очі може бути легше, ніж дізнатися шаблон особи, оскільки перша структура очевидно простіше. По-друге, можна отримати і використовувати відносно прості імовірнісні моделі, оскільки можуть існувати деякі властивості незалежності, які можна буде використовувати. По-трет'є, можливо, вдасться узгодити велике число деталей з відносно невеликим числом шаблонів. Хороший приклад цього явища – обличчя тварин; майже всі тварини з характерними обличчями мають очі, ніс і рот, відрізняється лише просторове розташування цих елементів. Нарешті, зі сказаного випливає, що для побудови складних деталей можна використовувати прості окремі шаблони. Наприклад, люди можуть рухати руками і ногами, і схоже, що навчити цілісний явний шаблон виявляти людей цілком значно складніше, ніж отримати окремі шаблони для частин тіла і вірогідну модель, що описує їх ступеня свободи.

Вже згадана тема не настільки добре вивчена, щоб до неї виробився якийсь стандартний підхід. У той же час основне питання досить очевидний – як закодувати набір зв'язків між шаблонами в форму, з якої легко працювати. В цьому розділі вивчається ряд різних підходів до даної задачі. По-перше, кожен шаблон може вказувати на об'єкти, які він може представляти, а потім якимось чином вважається число покажчиків. Якщо побудувати деяку явну вірогідну модель, для опису деталей просторових відносин можна використовувати більше вагових коефіцієнтів. Дану модель можна отримати з функцій правдоподібності; по суті, потрібна функція розподілу ймовірностей, що дає велике значення, коли конфігурація компонентів подібна об'єкту, і мале – в іншому випадку. Тоді пошук деталей перетворюється в пошук шаблонів, які при підстановці в вірогідну модель дають великі значення. Потрібно відзначити, що слід уважно ставитися до скорочення пошуку. Складність цього підходу полягає в тому, що навіть при скороченні пошук може бути дорогим. Як стверджують Форсайт і Понс, в

той же час при певному класі імовірнісних моделей можна провести ефективний пошук [11].

Прості моделі деталей можуть забезпечувати досить ефективне розпізнавання. Найпростіша модель – це розглядати об'єкт як набір фрагментів зображення (невеликих околиць елементів характерного вигляду) кількох різних типів, які формують образ (pattern). Щоб визначити, який образ спостерігається, знаходяться всі фрагменти, кожен з яких вказує на все образи, в які він входить. Те зображення, на яке було зазначено найбільше число, і вважається присутнім. Хоча дана стратегія проста, вона досить ефективна. Нижче описуються методи пошуку фрагментів, а потім представляється ряд послідовно ускладнюючихся реалізацій даної стратегії.

2.4.2 Опис фрагментів зображення

Невеликі фрагменти зображення можуть мати досить характерний вигляд, якщо мають багато ненульових похідних (наприклад, в кутах). Автори [11] шукали кути на зображенні, що їх називають точками інтересу. Далі оцінювався набір похідних функції яскравості в цих кутах і набір похідних, інваріантних щодо обертання, трансляції, визначеного зміни масштабу і зміни освітлення. Дані ознаки називалися інваріантними локальними особливостями (invariant local jets).

Далі будемо припускати, що фрагменти зображення можна розбити на кілька класів. Представники кожного класу можуть бути отримані використанням декількох зображень кожного об'єкта – як правило, відповідні фрагменти будуть ставитися до одного класу, але, можливо, внаслідок шуму мати дещо відмінні інваріантні локальні потоки. Відповідний набір класів можна визначити або за допомогою ручної класифікації фрагментів, або за допомогою кластеризації фрагментів-зразків (кілька кращий метод). Тепер потрібно визначити, коли два набори інваріантних локальних потоків представляють один клас фрагментів зображення. Шмід (Schmid) і Мор (Mohr) для перевірки використовували відстань Махаланобіса (Mahalanobis

distance) між векторами ознак тестованого фрагмента і фрагмента-зразка; якщо завмер був менше деякого порога, тестований фрагмент вважався ідентичним зразку.

Звернемо увагу на те, що по суті описаний класифікатор (він розподіляє фрагменти по класах, представленим зразками, або відмовляється від класифікації), а фрагменти представляють шаблони. Програму узгодження шаблонів можна побудувати за допомогою будь-якої техніки, не використовуючи ознаки, інваріантні щодо обертання, як говорили Форсайт і Понс [11]. Для цього можна використовувати набір налаштувань, що містить версії кожного фрагмента-зразка, які відрізняються обертанням і зміною масштабу, при варіюються умовах освітлення, так щоб класифікатор міг визначити, що обертання, зміна масштабу і освітлення не впливає на ідентифікацію фрагмента. Перевага використання інваріантних ознак полягає в тому, що класифікатором не потрібно дізнаватися про інваріантності з набору налаштувань.

2.4.3 Вказівка ознак і проста породжувальна модель

Нехай дано зображення, в якому знаходяться цікаві для нас точки і в кожній точці зображення класифікується фрагмент зображення. Виникає наступне питання: які образи знаходяться в зображенні? Для відповіді на це питання можна побудувати відповідність між фрагментами зображення і образами. Припустимо, що за все на зображенні є N_i фрагментів. Більш того, припустимо, що на зображенні або присутній один образ з даної колекції, або жодного. Окремий фрагмент може породжуватися або єдиним присутнім чином, або шумом. Втім, образи, як правило, містять фрагменти не всіх класів. Це означає, що твердження про присутність певного способу рівнозначно твердженню про те, що деякі фрагменти зображення породжені шумом (оскільки може бути присутнім тільки один образ, а є фрагменти зображення, які належать класам, які не містяться в поточному образі).

Остаточно приходимо до простої породжуючої моделі зображення. Коли образ присутній, можуть виникнути тільки фрагменти деяких певних класів. Розробляючи дану модель, отримуємо ряд алгоритмів зіставлення образів з спостережуваними картинами.

Найпростіша версія даної моделі виходить в припущенні, що образ породжує всі фрагменти класів, які він може породити, а значить потрібно, щоб шумом породжувалася мінімальне число фрагментів. Дане припущення зводиться до простої схеми вказівки. Береться кожен фрагмент зображення і вказується кожен образ, який може породити фрагменти цього класу. Вибирається спосіб, який отримав найбільшу кількість вказівок, він же вважається присутнім на зображенні. Дана стратегія може бути ефективною, хоча її використання приводить до деяких проблем.

2.4.4 Імовірнісні моделі для вказівки

Отриманий простий процес голосування можна інтерпретувати через імовірнісну модель, оскільки при цьому виявляються сильні і слабкі сторони описаного підходу. Фосайт і Понс [11] пишуть, що використовується породжує модель можна перетворити в імовірнісну, припустивши, що фрагменти генеруються незалежно і випадковим чином, причому також передбачається, що об'єкт присутній. Запишемо

$$P\{\text{фрагмент } i\text{-го типу є на зображенні} \mid \text{присутній } j\text{-й образ}\} = p_{ij}$$

$$P\{\text{фрагмент } i\text{-го типу} \mid \text{немає образу}\} = p_{ix}$$

У простій моделі передбачається, що для кожного образу j , $p_{ij} = \mu$, якщо образ може породити даний фрагмент, і 0 – в іншому випадку. Більш того, передбачається, що $p_{ix} = \lambda < \mu$ для всіх i . Нарешті, передбачається, що кожен спостережуваний фрагмент на зображенні може породжуватися або окремим чином, або шумом. Всього на зображенні n_i фрагментів. При таких припущеннях для обчислення функції правдоподібності потрібно знати тільки, які фрагменти породжені чином, а які – шумом. Зокрема, якщо взяти

функцію правдоподібності зображення при даному способі і припустити, що n_p фрагментів породжені даними чином, а $n_p - n_i$ фрагментів породжені шумом, можна записати:

$$P(\text{інтерпретація / образ}) = \lambda^{n_p} \mu^{(n_i - n_p)}, \quad (2.8)$$

причому дана величина більше для великих значень n_p . У той же час, оскільки не кожен образ може породити будь-який фрагмент, максимальний доступний вибір n_p залежить від обраного способу. Прийнятий метод вказівки рівнозначний вибору способу з максимальним можливим правдоподібністю при даній (простий) породжувальній моделі.

Звідси бачиться джерело певних складнощів: якщо образ мало правдоподібний, слід врахувати додаткову (апріорну) інформацію. Більш того, внаслідок шуму деякі фрагменти можуть породжуватися легше, ніж інші – якщо не врахувати цей факт, при підрахунку голосів деякі образи будуть в більш вигідному становищі. Нарешті, при даному об'єкті деякі фрагменти можуть бути більш ймовірними, ніж інші. Наприклад, кути значно частіше зустрічатимуться на образі шахової дошки, ніж на зображенні смуг зебри.

2.4.5 Доопрацювання породжувальної моделі

Щодо просто врахувати всі вищесказане і відповідним чином удосконалити найпростішу модель (фрагменти з'являються незалежно за умови наявності образу). Припустимо, що є N різних типів фрагментів. Форсайт і Понс [11] пишуть, що фрагменти кожного типу генеруються зі своєю ймовірністю усіма зображеннями і шумом. Тепер припустимо, що на зображенні присутні n_{il} примірників фрагментів l -го типу. Більш того, n_k з них генерується чином, а решта – шумом. Тепер оскільки фрагменти

виникають незалежно при даному способі і шум не залежить від способу, правдоподібність дорівнює

$$\frac{P(\text{фрагменти породжені образом} \mid j\text{-й образ})}{P(\text{фрагментів породжено шумом})}. \quad (2.9)$$

Перший член дорівнює

$$P(\text{тип 1} \mid j\text{-й образ})^{n_1} P(\text{тип 2} \mid j\text{-й образ})^{n_2} \dots \quad (2.10)$$

$$P(\text{тип } N \mid j\text{-й образ})^{n_N} P(\text{шум}), \quad (2.11)$$

що можна оцінити як

$$P_{1j}^{n_1} P_{2j}^{n_2} \dots P_{Nj}^{n_N}. \quad (2.12)$$

Тепер припустимо, що фрагменти породжуються шумом незалежно один від одного. Це означає, що шумову складову можна записати в такий спосіб:

$$P(\text{тип 1} \mid \text{шум})^{(n_1 - n_1)} \dots P(\text{тип } N \mid \text{шум})^{(n_N - n_N)}, \quad (2.13)$$

що можна оцінити як

$$P_{1x}^{(n_1 - n_1)} \dots P_{Nx}^{(n_N - n_N)}. \quad (2.14)$$

Це означає, що правдоподібність можна записати як

$$P_{1j}^{n_1} P_{2j}^{n_2} \dots P_{Nj}^{n_N} P_{1x}^{(n_1 - n_1)} \dots P_{Nx}^{(n_N - n_N)}. \quad (2.15)$$

Для кожного типу фрагментів k можливі два випадки: якщо $p_{kj} > p_{kx}$, то максимум досягається при $n_k = n_{ik}$; в іншому випадку максимум досягається при $n_k = 0$. Позначимо через π_i апіорну ймовірність того, що містить j -й образ, а через π_0 — апіорну ймовірність того, що воно не містить жодного об'єкта. Це означає, що для кожного типу образів j максимальне значення апостеріорної ймовірності запишеться в такий спосіб:

$$\left(\prod_m p_{mj}^{n_{im}} \prod_l p_{lx}^{n_{il}} \right) \pi_j, \quad (2.16)$$

де m — індекс, який служить для підрахунку значень ознак, для яких $p_{mj} > p_{mx}$; l — для підрахунку значень ознак, для яких $p_{lj} < p_{lx}$. Дана величина формується для кожного об'єкта, а потім вибирається та з них, яка має найбільшу апостеріорне значення. Треба звернути увагу, що дана модель є реляційної, хоча в дійсності ми не обчислюємо геометричні ознаки, що зв'язують образи. Це пояснюється тим, що фрагменти зв'язуються умовними ймовірностями їх появи при цьому образі, які відрізняються для різних образів.

2.4.6 Зазначення зв'язків

Для покращання простої стратегії зазначення можна використовувати геометричні зв'язки. Фрагмент співставляється з об'єктом, тільки коли з об'єктом співставлені сусідні фрагменти і всі ці фрагменти утворюють відповідну конфігурацію. Термін "відповідна конфігурація" може бути джерелом складнощів, але припустимо поки, що об'єкти співставлені з точністю до плоского обертання, трансляції і масштабу (це припущення розумно, наприклад, для осіб анфас). Припустимо тепер, що даний фрагмент, відповідний деякому об'єкту. Далі беремо p найближчих фрагментів і перевіряємо, чи відповідає більш 50 % їх з тим же об'єктом і чи співпадають

кути між трійками узгоджених фрагментів з відповідними кутами об'єкту — все це називається полулокальними умовами. Якщо ці дві перевірки пройдені, покажчик на об'єкт, поданий цим фрагментом, реєструється. Відзначимо, що ймовірнісну інтерпретацію даного підходу побудувати доволі важко — ймовірності появи фрагментів тепер залежать від того, де вони розташовані в образі, а також від того, який спосіб їх генерує.

2.4.7 Зазначення на тривимірні об'єкти

Хоча підхід Шміда і Мора був описаний з позиції двовимірних образів, його відносно легко розширити на розпізнавання тривимірних деталей. Для цього кожен набір проекцій об'єкту Форсайт і Понс [11] розглядають як інший двовимірний образ. При наявності достатнього числа проекцій цей підхід працює, оскільки невеликі зміни в двовимірному зображенні, викликані малими змінами кута спостереження, будуть компенсуватися дозволеним рівнем помилок процесу узгодження інваріантних локальних потоків і кутів.

Дана стратегія відомостей тривимірного розпізнавання до двовимірного застосовується доволі широко, але має і свої складності. Основна проблема — це велике число моделей, що може привести до ускладнення процедури зазначення. Невідомо також, яке мінімальне число проекцій, необхідних для узгодження в подібній схемі.

2.5 Штучні нейронні мережі та їх використання при ідентифікації зображень

Штучні нейронні мережі (ШНМ) — математичні моделі, а також їх програмні або апаратні реалізації, побудовані за принципом організації і функціонування біологічних нейронних мереж — мереж нервових клітин живого організму. Це поняття виникло при вивченні процесів, що протікають в мозку, і під час спроби змодельювати ці процеси.

ШНМ представляють собою систему з'єднаних і взаємодіючих між собою простих процесорів (штучних нейронів). Такі процесори зазвичай доволі прості, особливо в порівнянні з процесорами, використовуваними в персональних комп'ютерах. Кожен процесор подібної мережі має справу тільки з сигналами, які він періодично отримує, і сигналами, які він періодично посилає іншим процесорам. І тим не менше, будучи з'єднаними в достатньо велику мережу з керованим взаємодією, такі локально прості процесори разом здатні виконувати доволі складні завдання. Проста штучна нейронна мережа представлена на рисунку 2.2.

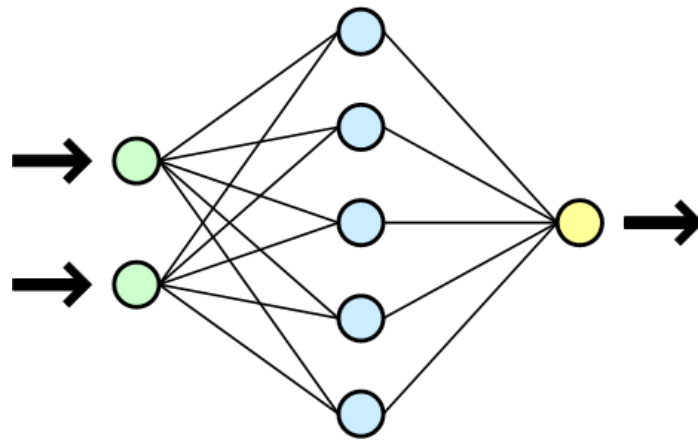


Рисунок 2.2 – Проста нейронна мережа

З точки зору машинного навчання, нейронна мережа – це окремий випадок методів розпізнавання образів, дискримінантного аналізу, методів кластеризації тощо. З математичної точки зору, навчання нейронних мереж вважається багатопараметричною задачею нелінійної оптимізації. З точки зору кібернетики, нейронна мережа застосовується в задачах адаптивного керування, а також як алгоритми для робототехніки. З точки зору розвитку обчислювальної техніки та програмування, нейронна мережа – це метод розв'язання проблеми ефективного паралелізму. А з точки зору штучного інтелекту, ШНМ є ґрунтом для філософської течії конективізму й основним напрямком у структурному підході щодо вивчення можливості побудови

(моделювання) природного інтелекту за допомогою комп'ютерних алгоритмів.

Нейронні мережі не програмуються в прямому сенсі цього слова, вони навчаються. Можливість навчання є однією з головних переваг нейронних мереж перед традиційними алгоритмами. Технічно навчання реалізується через знаходження коефіцієнтів зв'язків між нейронами. Під час навчання нейронна мережа здатна виявляти складні залежності між вхідними та вихідними даними, а також виконувати узагальнення. Це означає, що за умови успішного навчання мережа зможе повернути правильний результат на підставі даних, які були відсутні в навчальній вибірці, а також даних, які вважаються неповними і/або «зашумленими», частково викривленими.

За приклад образів можуть обиратися різні за своєю природою об'єкти: символи тексту, зображення, зразки звуків тощо. Під час навчання мережі пропонуються різні зразки образів із зазначенням того, до якого класу вони належать. Зразок, зазвичай, подається як вектор значень ознак. Зауважимо, що сукупність усіх ознак повинна однозначно визначати клас, до якого зразок належить. Якщо ж ознак недостатньо, мережа може співвідносити один і той же зразок із декількома класами, це є хибним варіантом. Після закінчення навчання мережі можна пред'явити раніше невідомі образи й отримувати відповідь про належність до певного класу.

Топологія такої мережі характеризується тим, що кількість нейронів у вихідному шарі, зазвичай, дорівнює кількості визначених класів. Крім того, встановлюється відповідність між виходом нейронної мережі та класом, який він представляє. Коли мережі подається якийсь образ, на одному з її виходів повинна з'явитися ознака того, що образ належить цьому класу. До того ж, на інших виходах повинна бути ознака того, що образ не належить даному класу. Якщо на двох або більше виходах є ознака приналежності до класу, то це сигналізує про те, що мережа «не впевнена» у своїй відповіді [22].

2.6 Розпізнавання об'єктів і їхня інтерпретація

Розпізнавання – це процес розмітки, в якому алгоритми розпізнавання ідентифікують кожен об'єкт сцени і привласнюють йому мітки (гайковий ключ, перемичка). Як правило, в більшості промислових систем технічного зору передбачається, що об'єкти сцени сегментовані у вигляді окремих елементів. Інше загальне обмеження стосується розташування пристроїв збору інформації щодо досліджуваної сцени (зазвичай вони розміщуються перпендикулярно робочій поверхні). Це призводить до зменшення відхилень у характеристиках форми. Крім того, спрощує процес сегментації й опису в результаті зменшення ймовірності загороджування одних об'єктів іншими. Керування відхиленнями в орієнтації об'єкта реалізується шляхом вибору дескрипторів, інваріантних до обертання, або шляхом використання головних осей об'єкта для орієнтування його у напрямку, що вже було визначено.

Сучасні методи розпізнавання можна розподілити на дві основні категорії: теоретичні та структурні методи. Теоретичні методи засновані на кількісному описі (статична структура), а за основу структурних методів взято символічні описи та їхні зв'язки (послідовності напрямків у межі, закодованої за допомогою ланцюгового коду).

Інтерпретація – це процес, який дозволяє системі технічного зору набути більш глибокі знання про навколишнє середовище в порівнянні зі знаннями, що були отримані за допомогою методів, які ми розглядали вище. Вже згадана з цієї точки зору інтерпретація охоплює дані методи як невід'ємну частину процесу розуміння зорової сцени. Хоча в області технічного зору вона і є об'єктом активних досліджень, значних досягнень поки не визначено. Далі ми коротко розглянемо проблеми, що визначають сучасні дослідження в цій області технічного зору.

Потужність СТЗ характеризується її здатністю виділяти зі сцени значиму інформацію за різних умов спостереження, а також використанні мінімальних знань про об'єкти сцени. Через низку причин (нерівномірне

висвітлення, наявність тіл, які загороджують об'єкти, геометрії спостереження) такий тип обробки є важкою задачею. Багато уваги приділяється методам зменшення розкиду в інтенсивності. Способи зворотного та структурованого освітлення дозволяють усунути труднощі, що пов'язані з довільним освітленням робочого простору, зокрема, мова йде про тіньові афекти, що ускладнюють процес визначення контурів, а також неоднорідності на гладких поверхнях. Як правило, це призводить до того, що вони розпізнаються як окремі об'єкти. Очевидно, що більшість цих проблем обумовлена тим, що відносно мало інформації про моделювання властивостей освітлення і відображення тривимірних сцен. Методи розмітки ліній і з'єднань є незначними спробами в цьому напрямку, проте вони не в змозі кількісно пояснити ефекти взаємодії освітлення і відображення. Більш перспективний підхід засновано на математичних моделях, які описують пріоритетні зв'язки між освітленням, відображенням і характеристиками поверхні, наприклад, як орієнтація.

2.7 Теоретико-множинна модель розпізнавання та ідентифікації

Розглянемо теоретико-множинну модель процесу розпізнавання та ідентифікації деталей поточного виробництва.

Нехай задано простір S в якому існує множина деталей $X = \{x_1 \dots x_n\}$, також в ньому знаходиться робот R , оснащений системою технічного зору S_R . Нехай кожному об'єкту x_i відповідає набір ознак $\{f_{li} \dots f_{ki}\}$.

Тоді множину деталей $X = \{x_1, x_2 \dots x_n\}$ можна охарактеризувати матрицею ознак:

$$F = \begin{bmatrix} f_{11} & f_{12} & \dots & f_{1n} \\ f_{21} & f_{22} & \dots & f_{2n} \\ \dots & \dots & \dots & \dots \\ f_{n1} & f_{n2} & \dots & f_{nn} \end{bmatrix}.$$

Процес ідентифікації буде полягати в тому, що об'єкт x_i може бути розпізнаний по підмножині ознак f'_i з множини ознак F . Для всієї множини деталей X набір підмножин f'_i формує множину F' :

$$F' = \begin{bmatrix} f'_{11} & f'_{12} & \dots & f'_{1k} \\ f'_{21} & f'_{22} & \dots & f'_{2k} \\ \dots & \dots & \dots & \dots \\ f'_{n1} & f'_{n2} & \dots & f'_{nk} \end{bmatrix}.$$

Загальний вигляд процедури розпізнавання деталей формально представляється у вигляді:

$$R(S) \times X \Rightarrow X',$$

де $R(S)$ – робот, оснащений системою датчиків S ;

X – множина деталей;

X' – множину розпізнаних деталей.

Процедура розпізнавання та ідентифікації деталей представлена у вигляді

$$R(S) \times X \equiv \begin{bmatrix} S_1 \\ S_2 \\ \dots \\ S_n \end{bmatrix} \times \begin{bmatrix} f_{11} & f_{12} & \dots & f_{1m} \\ f_{21} & f_{22} & \dots & f_{2m} \\ \dots & \dots & \dots & \dots \\ f_{m1} & f_{m2} & \dots & f_{mm} \end{bmatrix} \Rightarrow \begin{bmatrix} f'_{11} & f'_{12} & \dots & f'_{1k} \\ f'_{21} & f'_{22} & \dots & f'_{2k} \\ \dots & \dots & \dots & \dots \\ f'_{m1} & f'_{m2} & \dots & f'_{mk} \end{bmatrix} \Rightarrow X',$$

де $\begin{bmatrix} S_1 \\ S_2 \\ \dots \\ S_n \end{bmatrix}$ – множина сенсорів;

$\begin{bmatrix} f_{11} & f_{12} & \dots & f_{1m} \\ f_{21} & f_{22} & \dots & f_{2m} \\ \dots & \dots & \dots & \dots \\ f_{m1} & f_{m2} & \dots & f_{mm} \end{bmatrix}$ – множина ознак;

$$\begin{bmatrix} f_{11} & f_{12} & \dots & f_{1m} \\ f_{21} & f_{22} & \dots & f_{2m} \\ \dots & \dots & \dots & \dots \\ f_{m1} & f_{m2} & \dots & f_{mm} \end{bmatrix} - \text{множина розпізнаних ознак};$$

X' – множина розпізнаних деталей.

Таким чином, процедура ідентифікації буде полягати у визначенні в матриці розпізнаних ознак тих з них, які є достатніми для ідентифікації деталей:

$$I \times \begin{bmatrix} f'_{11} & f'_{12} & \dots & f'_{1k} \\ f'_{21} & f'_{22} & \dots & f'_{2k} \\ \dots & \dots & \dots & \dots \\ f'_{m1} & f'_{m2} & \dots & f'_{mk} \end{bmatrix} \Rightarrow \begin{bmatrix} f''_{11} & f''_{12} & \dots & f''_{1l} \\ f''_{21} & f''_{22} & \dots & f''_{2l} \\ \dots & \dots & \dots & \dots \\ f''_{m1} & f''_{m2} & \dots & f''_{ml} \end{bmatrix} = X'',$$

де $\begin{bmatrix} f''_{11} & f''_{12} & \dots & f''_{1l} \\ f''_{21} & f''_{22} & \dots & f''_{2l} \\ \dots & \dots & \dots & \dots \\ f''_{m1} & f''_{m2} & \dots & f''_{ml} \end{bmatrix}$ – матриця ідентифікованих ознак, що характеризує

множину ідентифікованих деталей X'' .

Дана модель може застосовуватись під час розробки підсистем розпізнавання, а також ідентифікації в складі систем технічного зору. Її використання дозволить забезпечити розв'язання завдань розпізнавання й ідентифікації об'єктів у робочій зоні маніпуляційних і мобільних роботів.

У другій частині роботи проаналізовано методи ідентифікації об'єктів у робототехнічних системах і побудовано теоретико-множинну модель розпізнавання й ідентифікації.

Основними методами є:

– метод порівняння з еталоном (встановлення збігу двох точкових зображень);

– методи теорії графів і розпізнавання (подання сегментів контуру у вигляді графа і пошуку на графі шляху найменшого значення, що відповідає значущим контурам);

- кореляційний метод (обчислення взаємкореляційної функції між еталоном і зображенням);
- розпізнавання шляхом зв'язку шаблонів (узгодження компонентів зображення як шаблон і визначення, які об'єкти присутні, проаналізувавши запропоновані зв'язки між знайденими шаблонами);
- штучні нейронні мережі (навчання мережі за різними зразками образів із зазначенням того, до якого класу вони належать).

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ МЕТОДІВ ОБРОБКИ ІНФОРМАЦІЇ В СИСТЕМАХ ТЕХНІЧНОГО ЗОРУ

3.1 Основні підходи до практичної реалізації методів обробки інформації

3.1.1 Методи просторової області

Розглянемо два основні підходи до попередньої обробки інформації. Перший підхід заснований на методах просторової області, а другий – на методах частотної області з використанням перетворення Фур'є. Разом ці підходи охоплюють більшість з існуючих алгоритмів попередньої обробки інформації, що застосовуються в системах технічного зору роботів [12].

До просторової області відноситься сукупність пікселів, що становлять зображення. Методи просторової області є процедурами, які оперують безпосередньо з цими пікселями. Функції попередньої обробки в просторовій області записуються у вигляді

$$g(x, y) = h[f(x, y)], \quad (3.1)$$

де $f(x, y)$ – вхідне зображення;

$g(x, y)$ – вихідна (оброблене) зображення;

h – оператор функції, певний в деякій області (x, y) . Оператор h можна застосовувати також до ряду вхідних зображень для формування, наприклад, суми пікселів K зображень при зменшенні шуму.

Основним підходом при визначенні околиці точки (x, y) є використання квадратної або прямокутної області частини зображення з центром в точці (x, y) . Центр цієї частини зображення переміщається від пікселя до пікселя, починаючи, наприклад, від лівого верхнього кута, при цьому для отримання $g(x, y)$ оператор застосовується для кожного положення (x, y) . Хоча іноді

застосовуються й інші форми околиці (наприклад, коло), квадратні форми є більш кращими завдяки простоті їхньої реалізації.

Один із найчастіше використовуваних методів просторової області ґрунтується на використанні так званих масок згортки (чи шаблонів, вікон або фільтрів). Як правило, під маскою розуміють невелику (на кшталт, розмірність 3×3) двовимірну систему, коефіцієнти якої вибираються таким чином, щоб виявити задану властивість зображення.

Припустимо для початку, що дано зображення з постійною інтенсивністю, що складається з окремо віддалених один від одного пікселів із відмінною від фону інтенсивністю. У ході такого процесу центр маски переміщується певним чином по зображенню. За умови збігу центру маски з положенням кожного пікселя проводиться множення значень усіх пікселів, які знаходяться під маскою, на відповідний коефіцієнт на масці. Потім підсумовуються результати цих дев'яти множень. Якщо всі пікселі під маскою мають однакові значення (постійний фон), то сума дорівнюватиме нулю. Якщо ж центр маски розміститься над точкою з іншою інтенсивністю, сума буде відмінною від нуля. У разі розміщення зазначеної точки поза центром сума також буде відмінною від нуля, проте на меншу величину. Цю різницю можна усунути шляхом порівняння значення суми з граничним значенням.

Зауважимо, що використання околиці не обмежується областями розмірністю 3×3 і випадками, що розглянемо в подальшому, зокрема, зниження шуму, отримання змінних порогів зображення, підрахунок вимірювань параметрів зображення та формування структури об'єкта.

3.1.2 Методи частотної області

До частотної області належить сукупність комплексних пікселів у вигляді перетворення Фур'є від зображення. Поняття «частота» застосовується під час інтерпретації перетворення Фур'є і виходить з того факту, що результат даного перетворення є сумою синусоїд. У зв'язку з

підвищенням вимог щодо обробки результатів методи частотної області не так активно використовуються в технічному зорі роботів, як методи просторової області [12].

Проте перетворення Фур'є грає важливу роль під час аналізу руху об'єкта, а також описі об'єкта. Крім того, значна кількість просторових методів для поліпшення якості та відновлення зображення базуються на концепціях перетворення Фур'є.

Дозволяється проводити пряму процедуру обчислення двовимірного перетворення Фур'є, застосовуючи тільки однорозмірні БПФ-алгоритми. Перетворення Фур'є може бути актуальним для вирішення багатьох завдань систем технічного зору. Наприклад, за допомогою уявлення межі об'єкта у вигляді одновимірного масиву точок і обчислення їх перетворення Фур'є отримані значення можна використати як опис форми кордону. Одномірне перетворення Фур'є також є ефективним для виявлення руху об'єкта.

Застосування дискретного двовимірного перетворення Фур'є можливо за умов зміни, збільшення та відновлення зображень, проте, як вже було зауважено, застосування вказаного методу в промислових системах технічного зору досі обмежено через складність необхідних обчислень. Наостанок відзначимо, що двовимірне аналогове перетворення Фур'є може бути реалізовано (зі швидкістю світла) оптичними засобами. Такий підхід, який вимагає застосування прецизійного оптичного устаткування, використовується в промислових умовах для вирішення певних завдань, зокрема, перевірки якості поверхні металу після фінішної обробки.

3.2 Бібліотека Integrated Performance Primitives (IPP)

Можливості Intel IPP полягають у таких функціях:

- кодуванні та декодуванні відео;
- кодуванні та декодуванні аудіо;
- JPEG/JPEG2000;

- машиному зорі;
- криптографії;
- стисненні даних;
- перетворенні кольору;
- обробці зображення;
- трасуванні променя / візуалізації;
- обробці сигналів;
- кодуванні мови;
- розпізнаванні мови;
- обробці рядків;
- векторної / матричної математики.

Бібліотека застосовує розширені набори інструкцій процесора MMX, SSE, SSE2, SSE3, SSSE3, SSE4, а також багатоядерні процесори.

Intel IPP розподілено на три основні групи: сигнали (лінійний масив даних або вектор), зображення (двомірний масив для типових колірних просторів), а також матриці (nхm масиви для матричних операцій).

Половина функцій - для матричних операцій, третина - для обробки сигналів і залишок - для зображень. Функції Intel IPP розподілено на 4 типи даних: 8u (8 - бітові беззнакові), 8s (8- бітові зі знаком), 16s, 32f (32 - бітові з плаваючою точкою), 64f тощо. Зазвичай, розробники застосунків працюють тільки з одним домінуючим типом даних для більшості функцій обробки. Тільки в кінці обробки виробляють перетворення у вихідний формат [26].

3.3 Бібліотека AviCap

Застосунки, призначені для запису звукових даних і відеоданих, можуть використовувати зручні високорівневі інтерфейси, які надаються бібліотекою avicap.dll. Створюючи застосунок для запису відео, не доведеться піклуватися про внутрішню структуру avi-файлів, компресії (стисненню) даних при записі, інтерфейс із драйверами пристрою (або

пристроїв) записів. За потреби, можна скористатися інтерфейсом нижчого рівня, що забезпечується бібліотекою `avifile.dll`.

Для створення застосунків для запису відео, найкраще вибрати клас вікна `AVI Cap`, визначений у бібліотеці `avicap.dll`. Створивши вікно, використавши клас `AVI Cap`, застосунок отримає в своє розпорядження простий інтерфейс для запису відео та звукових даних до `avi`-файлу, для попереднього перегляду відео та виконання інших операцій.

У класі `AVI Cap` передбачені засоби динамічного перемикання пристроїв запису відео та звуку, що є зручним у тих випадках, коли можливо по чергово використовувати декілька таких пристроїв, установлених у комп'ютері. Застосунок може створити `avi`-файл, призначений для запису, скопіювати вміст одного `avi`-файлу до іншого, встановити частоту кадрів, вивести на екран діалогову панель, завдяки якій користувач зможе задати формат запису. Крім того, є засоби для роботи з палітрами й універсальним буфером обміну `Clipboard`.

Для запису звуку клас вікна `AVI Cap` використовує засоби бібліотеки `mmsystem.dll`.

3.4 Бібліотека OpenCV

3.4.1 Оператори для визначення зображень за допомогою бібліотеки OpenCV

`Open CV` (англ. `Open Source Computer Vision Library`, бібліотека комп'ютерного зору з відкритим вихідним кодом) являє собою бібліотеку алгоритмів комп'ютерного зору, обробки зображень та чисельних алгоритмів загального призначення з відкритим кодом. Створена на базі `C / C ++`, за аналогією розробляється для `Python`, `Ruby`, `Matlab` та інших мов.

`Open CV` призначена для підвищення обчислювальної ефективності процедур обробки відеозображення з особливим акцентуванням на застосуванні в задачах реального часу.

Open CV написана на C, є добре оптимізованою та може застосовувати переваги багатоядерних процесорів. Для використання розширених можливостей бібліотеки рекомендується встановити Intel Performance Primitives (IPP), що дозволить підвищити продуктивність процедур бібліотеки.

Вона дозволяє досить швидко й ефективно реалізовувати складні алгоритми машинного зору. Бібліотека містить більше 500 функцій для реалізації програм у багатьох областях, зокрема:

- контролю якості продукції, що випускається;
- обробці зображень у медицині;
- забезпеченні безпеки;
- інтерфейсі користувача;
- робототехніці.

OpenCV складається з бібліотеки загальних функцій штучного інтелекту «Machine Learning Library» (MLL). Вона використовується переважно для розпізнавання фрагментів зображення та кластеризації.

Дану бібліотеку використовують:

- для затвердження загального стандартного інтерфейсу комп'ютерного зору для застосунків у цій області;
- для сприяння зростанню числа таких застосунків і створення нових моделей використання РС;
- зробити платформи Intel привабливими для розробників таких застосунків шляхом додаткового прискорення Open CV за допомогою Intel® Performance Libraries (Зараз містять IPP (засоби низького рівня бібліотеки для обробки сигналів, зображень, а також медіа-кодеки) та MKL (спеціальна версія LAPACK и FFTPack));
- Open CV автоматично визначає присутність IPP і MKL і застосовує їх для прискорення обробки.

Ключові модулі.

Ядро sxscore виробляє:

- базові операції над багатовимірними числовими масивами;
- матрична алгебра, математичні функції, генератори випадкових чисел DFT, DCT;
- запис / відновлення структур даних в / з XML/YAML;
- базові функції 2D-графіки;
- підтримка більш складних структур даних: розріджені масиви, динамічно зростаючі послідовності, графи;

CV вважається модулем обробки зображень і комп'ютерного зору, функції якого полягають у:

- базових операціях над зображеннями (фільтрація, геометричні перетворення, перетворення колірних просторів тощо);
- аналізі зображень (вибір відмінних ознак, морфологія, пошук контурів, гістограми);
- структурному аналізі (опис форм, плоскі розбиття);
- аналізі руху, спостереження за об'єктами;
- виявленні об'єктів, зокрема осіб;
- калібруванні камер, елементів відновлення просторової структури;

Highgui – це модуль для введення / виведення зображень та відео, створення призначеного для користувача інтерфейсу.

Його застосовують для виконання наступних операцій:

- отримання відео з камер та відеофайлів, зчитування / запис статичних зображень;
- функції для організації простого UI (наразі всі демо-застосунки використовують HighGUI).

Сваux містить у собі експериментальні та застарілі функції.

Даний модуль виконує:

- просторовий зір: стереокалібрацію, власне калібрацію;
- пошук стереовідповідності, кліки в графах;
- пошук та опис рис обличчя;
- порівняння форм, побудова скелетонів;

- приховані Марківські ланцюги;
- опис текстур.

Оператор Собеля обчислює перші, другі, треті чи змішані похідні зображення за допомогою розширеного оператора.

Функція обчислює похідну зображення, скручуючи зображення з відповідним ядром.

$$dst(x, y) = d^{xorder+yorder} src / dx^{xorder} dy^{yorder} |_{(x,y)} \quad (3.2)$$

Оператори Собеля комбінують Гаусове згладжування і диференціювання, таким чином результат більш-менш стійкий до спотворення. Найчастіше, функція викликана, щоб обчислити спочатку x-або y- похідну зображення. Перший випадок (X-оператор) відповідає

$$\begin{vmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{vmatrix},$$

другий випадок (Y -оператор)

$$\begin{vmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{vmatrix},$$

або

$$\begin{vmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{vmatrix}.$$

в залежності від початку координат зображення. Так як масштабування не зроблено, зображення адресата зазвичай має більший розмір абсолютної величини, ніж вихідне зображення.

Ядра застосовуються до кожного пікселя зображення: він поміщається в центр ядра, і значення інтенсивності в сусідніх точках множаться на відповідні коефіцієнти ядра, після чого отримані значення підсумовуються. X -оператор Собеля, застосований до 3×3 матриці вихідного зображення, дає величину горизонтальної складової градієнта інтенсивності в центральній точці цієї матриці, а Y -оператор Собеля дає величину вертикальної складової градієнта [12]. Коефіцієнти ядра обрані так, щоб при його застосуванні одночасно виконувалися згладжування в одному напрямку і обчислення просторової похідної – в іншому. Величина градієнта визначається як квадратний корінь з суми квадратів значень горизонтальної та вертикальної складових градієнта.

В результаті утворюється масив чисел, що характеризують зміни яскравості в різних точках зображення. Потім виконується операція порівняння з порогом і визначається положення елементів зображення з найбільш сильними перепадами яскравості. Вибір порога є одним з ключових питань виділення перепадів [12].

В результаті обробки виходить бінарна матриця, де одиницям відповідають точки зі значним перепадом яскравості, нулях – всі інші. В якості додаткової міри в боротьбі з шумом і ліквідації можливих розривів в контурах застосовуються морфологічні операції.

У бібліотеці комп'ютерного зору OpenCV цей оператор представлений функцією

```
void cvSobel( const CvArr* src, CvArr* dst, int xorder, int yorder, int aperture_size=3 ),
```

де src – вихідне зображення;

dst – одержуване зображення;

$xorder$ – похідна по координаті x ;

yorder – похідна по координаті y ;

aperture_size – розмір апертури зображення (апаратура в оптиці – чинне отвір оптичного приладу, що визначається розмірами лінз або діафрагмами).

Оператор Лапласа. Знайдемо ймовірність попадання випадкової величини, розподіленої за нормальним законом, в заданий інтервал.

$$P(a < X < b) = \int_a^b f(x) dx = \frac{1}{\sigma\sqrt{2\pi}} \int_a^b e^{-\frac{(x-m)^2}{2\sigma^2}}, \quad (3.3)$$

позначимо

$$\frac{x-m}{\sigma\sqrt{2}} = t; \quad \frac{a-m}{\sigma\sqrt{2}} = \alpha; \quad \frac{b-m}{\sigma\sqrt{2}} = \beta; \quad (3.4)$$

тоді

$$P(a < X < b) = \frac{1}{\sigma\sqrt{2\pi}} \int_a^b e^{-t^2} \sigma\sqrt{2} dt = \frac{1}{\sqrt{\pi}} \int_a^b e^{-t^2} dt = \frac{1}{2} [\Phi(\beta) - \Phi(\alpha)]. \quad (3.5)$$

Так як інтеграл $\int_a^b e^{-t^2} dt$ не виражає через елементарні функції, то вводиться в розгляд функція

$$\Phi(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt, \quad (3.6)$$

яка називається функцією Лапласа або інтегралом ймовірностей.

Значення цієї функції при різних значеннях x пораховані і наводяться в спеціальних таблицях.

На рисунку 3.1 показаний графік функції Лапласа.

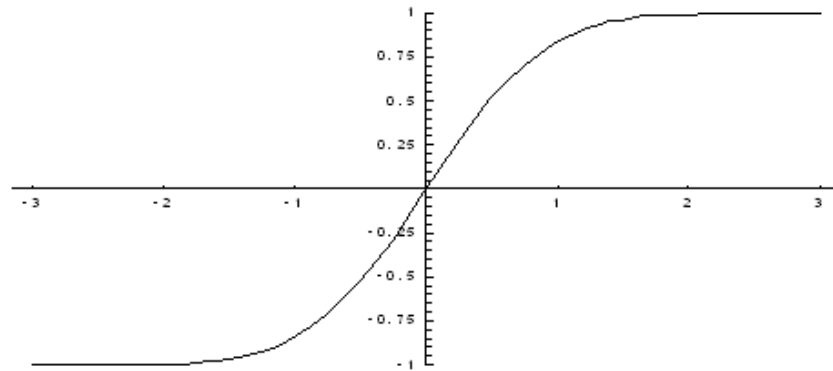


Рисунок 3.1 – Графік функції Лапласа

Функція Лапласа має такі властивості:

$$\Phi(0) = 0;$$

$$\Phi(-x) = -\Phi(x);$$

$$\Phi(\infty) = 1.$$

Функцію Лапласа також називають функцією помилок і позначають $\text{erf } x$.

Ще використовується нормована функція Лапласа, яка пов'язана з функцією Лапласа, нижче показаний графік нормованої функції Лапласа (рисунок 3.2).

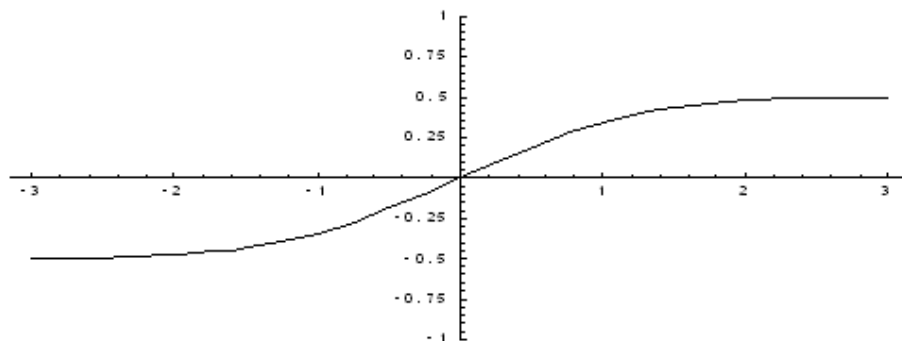


Рисунок 3.2 – графік нормованої функції Лапласа

У бібліотеці комп'ютерного зору OpenCV даний оператор представлений функцією

```
void cvLaplace( const CvArr* src, CvArr* dst, int aperture_size=3 ),
```

де `src` – вихідне зображення;

`dst` – отримане зображення;

`aperture_size` – розмір апертури зображення (за аналогією до функції `cvSobel`). Відмінність функції Лапласа від функції Собеля визначається тим, що в функції Лапласа знаходиться друга похідна, а в функції Собеля – перша.

Алгоритм Кенні. Кути характеризують контури об'єктів на зображенні, тому є дуже важливою проблемою під час обробки зображень.

Кути на зображенні – це області з різкими стрибками інтенсивності кольору за умови переходу від одного пікселя до іншого. Виділення кутів на зображенні виконується для значного скорочення обсягу даних під час обробки, при цьому зберігаючи ключові структурні властивості вихідного зображення.

Алгоритм Кенні виділення кутів на зображенні відомий багатьом як оптимальний детектор кутових точок. Мета Кенні полягала в поліпшенні детекторів, які було створено ще до нього на момент розробки. Науковець успішно досягнув своєї мети, виклавши свої ідеї та методи.

Цей алгоритм є актуальним і на сьогодні, отже, лежить в основі багатьох сучасних алгоритмів виділення кутів.

У своїй роботі Кенні дотримувався деякого списку критеріїв для поліпшення сучасних методів виділення кутів.

Перший і найбільш очевидний критерій можна визначити як низький рівень похибки. Дуже важливо, щоб точність кутових точок зображення не було втрачено і, відповідно, щоб не виникало посилань на НЕ кутові точки. Другий критерій - хороший рівень локалізації кутових точок: відстань між знайденими кутовими точками та дійсними кутами повинна зводитись до

мінімуму. Третій критерій полягає в тому, щоб кожному куту відповідало не більше однієї точки. Такий критерій виділили, оскільки перших двох було недостатньо для повного виключення можливості множинного співвідношення різних точок одному і тому ж куті.

Базуючись на цих критеріях, детектор кутів Кенні на початку згладжує зображення для зменшення шуму. Далі знаходить градієнт зображення в кожній точці для виділення областей з найбільшою величиною просторової похідної. Потім алгоритм проходить цими областями та глушить пікселі з не максимальним значенням градієнта (процедура *nonmaximum suppression*). Масив градієнтів скорочується процедурою гістерезису (лат. *hysteresis*), що застосовується для обробки залишку пікселів, які не були приглушені. Процедура гістерезису використовує два порога. Якщо величина градієнта менша за перший поріг, то піксель встановлюється в нуль (не кут). Якщо величина градієнта більша за другий поріг, то піксель встановлюється в максимальне значення (робиться кутом). Якщо величина знаходиться між двома порогамі, то піксель не встановлюється в нуль, поки не знайдений шлях від даного пікселя до пікселя з величиною градієнта більший за другий поріг.

У бібліотеці комп'ютерного зору OpenCV даний оператор представлений функцією

```
void cvCanny( const CvArr* image, CvArr* edges, double threshold1,
             double threshold2, int aperture_size=3 ),
```

де *image* – вхідне зображення;

edges – зображення для виділення країв за допомогою функції;

threshold1 – перший поріг зображення;

threshold2 – другий поріг зображення; *aperture_size* – розмір апертури зображення.

Детектор Харріса. За останні двадцять років було створено безліч різних детекторів точкових особливостей зображень. Найпопулярнішими з

них є детектор Харріса та детектор за мінімальним власним значенням. У даній роботі реалізовано обидва.

Розглянемо детектор за мінімальним власним значенням. Візьмемо перше зображення послідовності. Для кожного пікселя (x_0, y_0) розраховуємо наступну матрицю:

$$M = \sum_{x,y \in S} w(x,y) \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \left(\frac{\partial I}{\partial x}\right)\left(\frac{\partial I}{\partial y}\right) \\ \left(\frac{\partial I}{\partial x}\right)\left(\frac{\partial I}{\partial y}\right) & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix} = \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix} \quad (3.7)$$

де $I(x, y)$ – яскравість в точці (x, y) ;

$w(x, y)$ – вагова функція по околі $S(x_0, y_0)$, в якості якої зазвичай беруть розподіл Гаусса.

У точки, околиця якої схожа на кут, матриця M буде мати два великих позитивних власних значення. Таким чином, умова схожості точки на кут запишеться в такий спосіб:

$$F = \min(\lambda_1, \lambda_2) > 1 \quad (3.8)$$

Тепер залишається знайти локальні максимуми функції відгуку кута $F(x, y)$, які і будуть відповідати особливим точкам.

Щоб спростити обчислення, Харріс запропонував відмовитися від підрахунку власних значень матриці, і замість них ввів таку функцію відгуку кута:

$$R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 = \det(M) - k(\text{Trace}(M))^2 \quad (3.9)$$

У бібліотеці комп'ютерного зору OpenCV цей оператор представлений функцією

```
void cvCornerHarris( const CvArr* image, CvArr* harris_responce,
                    int block_size, int aperture_size=3, double k=0.04 ),
```

де *image* – вхідне зображення;

harris_responce – зображення для виведення функції, яке повинно мати такий же розмір як і вхідне зображення;

block_size – розмір блоків;

aperture_size – розмір апертури зображення;

k – вільний параметр детектора Харріса, його величина зазвичай лежить в межах $(0,04 \div 0,15)$ і визначається емпірично.

Далі над нею виконують аналогічні дії, пошук локальних максимумів, і введення порогового значення.

На другому етапі (його називають етапом спостереження (Tracking)) визначаються зміщення для кожної особливої точки між сусідніми кадрами, а також розраховується середній за всіма точками вектор зсуву для кожної пари кадрів.

Основною ідеєю завдання стеження є пошук околиці в поточному кадрі, максимально схожою на околицю точки в попередньому кадрі. Наступний основний момент полягає в тому, що даний пошук околиці виконується не по всьому зображенню, а в деякому радіусі від початкового положення особливої точки. Тут доцільно ввести припущення, що за час між двома кадрами в сцені великих змін не відбувається, тому немає сенсу шукати точку далеко від поточного її положення.

Отже, якщо в попередньому кадрі координати особливої точки позначити (x_0, y_0) , то в області пошуку R шукається такий вектор $(\Delta x, \Delta y)$, що сума по околиці $S(x_0, y_0)$ буде мінімальна:

$$\min_{\Delta x, \Delta y < R} \left[\sum_{x, y \in S(x_0, y_0)} (I_n(x, y) - I_{n-1}(x + \Delta x, y + \Delta y))^2 \right] \quad (3.10)$$

де $S(x_0, y_0)$ – околиця порівняння;

R – область пошуку особливості,

I_n і I_{n-1} – яскравість точки в поточному і попередньому кадрі.

Функцією пошуку контурів є функція `cvFindContours` (таблиця 3.1):

```
int cvFindContours( CvArr* image, CvMemStorage* storage, CvSeq**
first_contour, int header_size=sizeof(CvContour), int mode=CV_RETR_LIST, int
method=CV_CHAIN_APPROX_SIMPLE, CvPoint offset=cvPoint(0,0) ),
```

де `image` – вихідне 8-бітове одноканальне зображення. Ненульові пікселі вважаються 1, нульові залишаються 0 – розглядається як бінарний файл. Для отримання такого бінарного зображення з відтінками сірого, можна використовувати функції `cvThreshold`, `cvAdaptiveThreshold` або `cvCanny`. Функція змінює вміст вихідного зображення;

`Storage` – комірка пам'яті, яка містить отримані контура;

`first_contour` – вихідний параметр, що містить вказівник на перший знайдений контур;

`header_size` – розмір послідовності заголовка $\geq \text{sizeof}(\text{CvChain})$, якщо метод `CV_CHAIN_CODE`, і $\geq \text{sizeof}(\text{CvContour})$ в іншому випадку;

`Mode` – мода.

Таблиця 3.1 – Параметри функції виділення контурів

<code>CV_RETR_EXTERNAL</code>	якщо необхідні тільки крайні зовнішні контури
<code>CV_RETR_LIST</code>	виділення всіх контурів і приміщення їх в список
<code>CV_RETR_CCOMP</code>	виділення всіх контурів і приміщення їх в подвійну ієрархію
<code>CV_RETR_TREE</code>	отримати все контура і реконструювати повну ієрархію вкладених контурів

Method – метод апроксимації (для всіх модів, окрім CV_RETR_RUNS, який застосовує вбудовану апроксимацію);

I offset – кожен контур точки зрушився. Залучається, якщо контури витягуються із зображень, після чого їх необхідно проаналізувати в контексті цілого зображення.

Методи апроксимації для функції визначення контурів наведені у таблиці 3.2.

Таблиця 3.2 – Методи апроксимації для функції визначення контурів

CV_CHAIN_APPROX_NONE	перетворює всі крапки з ланцюжка в точки
CV_CHAIN_APPROX_SIMPLE	тискає горизонтальних, вертикальних і діагональних сегментах, тобто функція залишається тільки припинення їх центри
CV_CHAIN_CODE	вихідні контуру в ланцюжковому коді Фрімена
CV_CHAIN_APPROX_TC89_L1, CV_CHAIN_APPROX_TC89_KCOS	застосування одного з апроксимаційних ланцюжкових кодів Тех-Чину
CV_LINK_RUNS	використовувати інший алгоритм пошуку контуру за допомогою горизонтальних зв'язків сегментів в один. Тільки CV_RETR_LIST може бути використаний з цим методом

Функція Наближення полігональних кривої (кривих) із заданою точністю.

```
CvSeq* cvApproxPoly( const void* src_seq, int header_size,
CvMemStorage* storage, int method, double parameter, int parameter2=0 );
```

src_seq – послідовність ряду точок; header_size – розмір заголовка кривої; storage- Контейнер з апроксимується контурами. Якщо NULL, то використовується осередок з вхідними послідовностями; method – метод апроксимації; тільки CV_POLY_APPROX_DP, який описує метод Дугласа-

Пеукера; `parameter` – параметр спеціального методу; `CV_POLY_APPROX_DP` для точної апроксимації; `parameter2` – якщо `src_seq` є одиничною послідовністю, то він повинен бути наближений для всіх послідовностей на тому ж рівні або нижче `src_seq`. Якщо `src_seq` є масив (`CvMat *`) точок, параметр визначає, чи буде крива замкнута.

Функції зменшення і збільшення розміру зображень:

```
Void cvPyrUp(const CvArr* src, CvArr* dst, int
filter=CV_GAUSSIAN_5x5);
```

де `src` – вихідне зображення;

`dst` – входить зображення (має містити довжину і ширину);

`filter` – тип фільтра; тільки `CV_GAUSSIAN_5x5` підтримується.

Порогова функція в бібліотеці комп'ютерного зору Open CV визначається як.

```
void cvThreshold( const CvArr* src, CvArr* dst, double threshold,
double max_value, int threshold_type );
```

де `src` – вхідний масив (8-бітний, 32-бітний);

`dst` – вихідний масив (такого ж типу як і вхідний);

`threshold` – порогове значення;

`max_value` – максимальне значення, яке використовується з `CV_THRESH_BINARY` і `CV_THRESH_BINARY_INV`;

`threshold_type` – тип порогової функції.

Знаходження кіл в відтинках сірого зображення, використовуючи перетворення Хафа;

```
CvSeq* cvHoughCircles( CvArr* image, void* circle_storage,
int method, double dp, double min_dist, double param1=100, double
param2=100, int min_radius=0, int max_radius=0 );
```

`Image` – вхідне 8-бітне зображення; `circle_storage` – комірка, що зберігає виявлені кола. Це може бути сховищем (у даному випадку послідовність кіл створюється за умови зберігання та повертається

функцією), або один рядок / один стовпець матриці (CvMat *) типу CV_32FC3, до якого пишуться параметри кіл. Якщо circle_storage є матрицею, а фактичне число ліній перевищує її розмір, максимально можлива кількість кіл повертається. Кожне коло кодується 3 числами з плаваючою точкою: центр координатами (x, y) і радіусу; Method – реалізований тільки метод CV_HOUGH_GRADIENT; Dp – як встановити акумулятор, який застосовується для виявлення центрів кіл. Як приклад наведемо, якщо він дорівнює 1, акумулятор матиме той же дозвіл, як у вихідного зображення, якщо це 2 – акумулятор буде в два рази меншим шириною та і висотою тощо; min_dist – мінімальна відстань між центрами виявлених кіл. Якщо параметр буде занадто малим, то кілька сусідніх кіл може бути помилково виявлено на додаток до дійсних. Якщо він буде занадто великим, то деякі кола можуть бути втрачені; param1 – перший метод конкретних параметрів. У разі CV_HOUGH_GRADIENT це вищий поріг, край детектора (нижче 1 буде в два рази менше); param2 – другий метод конкретних параметрів. У разі CV_HOUGH_GRADIENT – акумулятор. Чим меншим є параметр, тим більше помилкових кіл може бути виявлено; min_radius – мінімальний радіус шуканих кіл; max_radius - максимальний радіус.

У третій частині роботи розглянуто практичну реалізацію методів обробки інформації в робототехнічних системах. Є два основні підходи до попередньої обробки інформації. Перший ґрунтується на методах просторової області, а другий – на методах частотної області з використанням перетворення Фур'є. У комплексі ці підходи охоплюють більшість з наявних алгоритмів попередньої обробки інформації, що застосовуються в системах технічного зору роботів. Провідними бібліотеками, що працюють із методами розпізнавання та ідентифікації, є: бібліотека Integrated Performance Primitives (IPP), бібліотека AviCap, бібліотека комп'ютерного зору з відкритим кодом OpenCV. Розглянуто основні функції та їхня реалізація в бібліотеці OpenCV.

4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИЗНАЧЕННЯ ПАРАМЕТРІВ ДЕТАЛЕЙ

4.1 Основні особливості розробленого програмного забезпечення

Метою розробки даного програмного забезпечення є дослідження програмних методів розпізнавання та ідентифікації простих деталей за допомогою бібліотеки комп'ютерного зору OpenCV.

З точки зору MFC проект буде складатися з двох класів: CApp (клас додатка) і CMainWin (головне вікно) з відповідною першому функцією InitInstance(); конструктором і картою повідомлень для класу вікна.

У програмі вказується глобальне оголошення функції обробки кадру - void mycallback (IplImage *img).

Вказівниками на зображення будуть: IplImage *image1,*src2,*dst,*dst2,*dst3,*dst4,*gray,*dst5,*dst6,*dst7,*dst8.

Далі виділяються змінні для збереження пам'яті. В даному випадку це CvMemStorage *storage,*storage2;

Конструктор головного вікна програми необхідно описати в такий спосіб:

```
CMainWin::CMainWin()
{ Create (NULL, "OpenCV");
```

Далі в конструкторі вказується дескриптор головного вікна програми:

```
HWND w= this->GetSafeHwnd();
```

Також в конструкторі визначається кількість камер за допомогою команди int ncams=cvcamGetCamerasCount();

Далі йде перевірка наявності камер, якщо їх кількість нульове, виконується ряд функцій, пов'язаних з ініціалізацією камери, основних вікон програми і властивостей зображень:

```

if (ncams) {      bCreate=true;
VidFormat vidFmt={800,600,20.0};
cvcamSetProperty(0,CVCAM_PROP_ENABLE,CVCAMTRUE);
cvcamSetProperty(0,CVCAM_PROP_CALLBACK,mycallback);
cvcamSetProperty(0,CVCAM_PROP_WINDOW,&w);
cvcamSetProperty(0,CVCAM_PROP_SETFORMAT,&vidFmt);
cvNamedWindow(cvGetWindowName(w),CV_WINDOW_AUTOSIZE);
}

```

Для організації виведення результатів обробки візуальної інформації використовується функція `cvNamedWindow("Canny", 1)`, де 1 – ідентифікатор вікна. Результат кожного перетворення вихідного зображення виводиться за допомогою команди `cvShowImage()`.

За допомогою функції `cvCreateTrackbar` створюється смуга прокрутки `cvCreateTrackbar("CannyTrack","Canny",&cannyt,200,NULL)` і задається розмір вікна для функції `cvResizeWindow("Canny",320,200)`.

Аналогічним чином створюється вікно для виведення функції визначення контурів.

Потім проводиться ініціалізація камери, або виводиться повідомлення про помилку:

```
if(!cvcamInit())  MessageBox("Error");
```

в іншому випадку проводиться запуск камери:

```
else  cvcamStart();}
```

або видається повідомлення про те, що вона не знайдена.

Закриття головного вікна програми має забезпечувати закриття всіх програмних вікон (в іншому випадку вони можуть залишитися в пам'яті). Для цього необхідно використовувати функцію `void CMainWin::OnClose()`, в якій проводиться зупинка камери (`cvcamStop()`), закриття всіх вікон (`cvDestroyAllWindows()`), вихід з режиму камери (`cvcamExit()`) і закриття головного вікна:

```
if(!bCreate) cvReleaseImage(&dst);
    DestroyWindow(); }.
```

4.2 Реалізація функції обробки зображень

Обробка зображень в OpenCV проводиться кадр за кадром, а функції зворотного виклику (mycallback), яка в розробленому програмному забезпеченні виглядає так: `void mycallback(IplImage *src)`

Зображення, отримане з камери в тексті програми позначено змінною `src` і виводиться в основне вікно програми.

У додатку використовуються різні типи моделей пам'яті, які відповідають різним типам зображень, наприклад:

```
src2=cvCreateImage(cvSize(src->width,src->height),IPL_DEPTH_8U,3);
dst2=cvCreateImage(cvSize(src->width,src->height),IPL_DEPTH_32F,3);
gray=cvCreateImage(cvSize(src->width,src->height),IPL_DEPTH_8U,1);
```

Для повороту зображення використовувалася функція `cvFlip(src, src2)`.

Для нормальної роботи функції Кенні, необхідно перетворити вихідне зображення в чорно-біле (використовується функція `cvCvtColor(src2,gray,CV_RGB2GRAY)`), після чого виконати саме перетворення функцією `cvCanny(gray,dst3,25,100+cannyt,3)`.

Далі виконується висновок цього зображення у вікно `dst3` за допомогою функції `cvShowImage("Canny",dst3)`.

Однією з найпростіших функцій обробки зображень є порогова функція. У бібліотеці OpenCV вона представлена у вигляді:

```
cvThreshold( tgray, gray, 100, 255, CV_THRESH_BINARY ).
```

Тут `tgray` – вхідний масив (8- або 32- бітний); `gray` – вихідний масив такого ж типу як і `tgray`; 100 та 255 – мінімальне і максимальне значення, яке використовується пороговою функцією; `CV_THRESH_BINARY` – показує тип порогової функції.

Більш покращеною різновидністю порогової функції є функція `cvAdaptiveThreshold` (`dst3`, `dst4`, `5+thresh2`, `CV_ADAPTIVE_THRESH_MEAN_C`, `CV_THRESH_BINARY`,3,5).

4.3 Реалізація функцій розпізнавання та ідентифікації

Результатом виконання функції Кенні є зображення, що містить края деталей у вихідному кадрі, і є основою для подальших перетворень, зокрема для виділення контурів.

Розглянемо функцію `cvFindContours`, яка знаходить контури деталей, що знаходяться у полі зору камери:

```
cvFindContours(gray, storage, &contours, sizeof(CvContour),
CV_RETR_LIST, CV_CHAIN_APPROX_SIMPLE, cvPoint(0,0) );
```

Тут `gray` є вихідним зображенням у 8-бітному форматі, у області пам'яті `storage` зберігаються отримані контури; у масиві `contours` міститься інформація про першому знайденому контурі; `CvContour` показує розмір структури даних контуру; параметр `CV_RETR_LIST` означає, що всі виділені контури поміщаються у список; параметр `CV_CHAIN_APPROX_SIMPLE` означає, що функція виділяє центри контурів; `cvPoint(0,0)` показує, що контури витягуються з зображень і потім повинні бути проаналізовані у контексті цілого зображення.

Для знаходження декількох контурів використовується цикл, представлений нижче:

```
while(contours)
{
    result = cvApproxPoly( contours, sizeof( CvContour), storage,
CV_POLY_APPROX_DP, cvContourPerimeter( contours)*0.02, 0 );
    if( result->total == 4 && fabs( cvContourArea( result,
CV_WHOLE_SEQ)) > 1000 && fabs( cvContourArea( result,
```

```

CV_WHOLE_SEQ)) <( img->height * img->width/2 ) &&
cvCheckContourConvexity( result) )
{ s = 0; for( int i = 0; i < 5; i++ )
    {if( i >= 2) { t = fabs( angle( ( CvPoint*)cvGetSeqElem( result, i), (
    CvPoint*)cvGetSeqElem( result, i-2 ),
    ( CvPoint*)cvGetSeqElem( result, i-1 ))) ); s = s > t ? s : t; }
    }
if( s < 0.5 )for( int i = 0; i < 4; i++ )cvSeqPush( squares, (
    CvPoint*)cvGetSeqElem( result, i ));
contours = contours->h_next;
}

```

Функція `cvHoughCircles` знаходить окружності на сірому зображенні, використовуючи перетворення Х'ю:

```

cvHoughCircles( gray, cstorage, CV_HOUGH_GRADIENT, 1, gray->height/16,
8, 10, 4, 50 ).

```

даній функції, використаної у програмному забезпеченні параметр `gray` є 8-бітим зображенням; `cstorage` – область пам'яті, у якій зберігаються окружності, виявлені функцією; `CV_HOUGH_GRADIENT` – метод реалізації функції; `1` – стек, який використовується для виявлення центрів кіл з тим же розділенням, що і вихідне зображення; `gray->height/16` – обчислюється мінімальна відстань між центрами виявлених окружностей; наступні 2 параметра відповідають за накопичення порогів у виявлених окружностях; останні 2 параметра відповідають за мінімальний і максимальний радіус знаходять окружностей.

Для відображення кіл використовується цикл, який виглядає так:

```

for( int i = 0; i < circles-> total; i++ )
{
    float* p = ( float*)cvGetSeqElem( circles, i );
    cvCircle( out, cvPoint( cvRound( p[0]), cvRound( p[1])),
2, CV_RGB( 200, 0, 0), -1, 8, 0 );
    cvCircle( out, cvPoint( cvRound( p[0]), cvRound( p[1])), cvRound(
p[2]), CV_RGB( 200, 0, 0), 1, 8, 0 );
}

```

Відображення прямокутних зображень здійснюється командою `void drawSquares(IplImage *img, CvSeq* squares);`

Прямокутні об'єкти визначаються за допомогою функції поліліній `cvPolyLine(img, &rect, &count, 1, 1, CV_RGB(200,0,0), 1, CV_AA, 0);`

Далі читається послідовність 4 ліній

```
CV_READ_SEQ_ELEM( pt[0], reader );
```

```
CV_READ_SEQ_ELEM( pt[1], reader );
```

```
CV_READ_SEQ_ELEM( pt[2], reader );
```

```
CV_READ_SEQ_ELEM( pt[3], reader ).
```

Кут повороту ліній полілінії заданий за допомогою функції:

```
double angle = abs(pt[1].y-pt[2].y)/sqrt((pt[1].x-pt[2].x)*
```

```
(pt[1].x-pt[2].x)+(pt[1].y-pt[2].y)*(pt[1].y-pt[2].y)+0.00001);
```

Самі лінії полілінії промальовуються за допомогою функцій:

```
cvLine( img, cvPoint(0,img->height/2), cvPoint(img->width,img->height/2),
```

```
CV_RGB(200,200,200),1, 8, 0 );
```

```
cvLine( img, cvPoint(img->width/3,0), cvPoint(img->width/3,img->height),
```

```
CV_RGB(200,200,200),1, 8, 0 );
```

```
cvLine( img, cvPoint(img->width/3*2,0), cvPoint(img->width/3*2,img->height),
```

```
CV_RGB(200,200,200),1, 8, 0 );    cvPutText( img, st, pt[1], &font,
```

```
CV_RGB(200,0,0).
```

Виведення прямокутних деталей здійснюється функцією `drawSquares(out,findSquares4(gray, mainStorage))`, де `out` – зображення, що містить прямокутні фігури.

Отже розроблене програмне забезпечення виконує знімання й обробку інформації, що надходить з WEB-камери (СТЗ робота), забезпечує виконання меж, контурів, ідентифікацію простих деталей.

Лістинг програм наведено у Додатках А та Б.

У четвертій частині досліджено та програмно реалізовано методи розпізнавання та ідентифікації простих об'єктів. Пріоритетною бібліотекою розпізнавання обрано бібліотеку комп'ютерного зору OpenCV. Отримані

результати реалізують виділення меж та контурів об'єктів, визначення центрів мас. Для простих об'єктів залучені функції розпізнавання та ідентифікації.

4.4 Промислова безпека та аналіз умов праці на робочому місці

Розміри лабораторії, в якій виконувалась робота, складають 5×6 м. Робоче місце складається з стола, стільця і персонального комп'ютера. У приміщенні працює 4 людини. Площа приміщення 30 м², об'єм – 90 м³. Згідно ДСанПин 3.3.2.007-98 площа на одне робоче місце має становити не менше 6 м², а об'єм – 20 м³. Для даного приміщення робоча площа і об'єм на одну людину відповідає нормам, так як в нашому випадку площа на одне робоче місце становить 10 м², а об'єм – 30 м³.

Живлення комп'ютерів здійснюється від трифазної чотирьох провідної електричної мережі змінного струму з глухо-заземленою нейтраллю і напругою 220 В, частотою 50 Гц.

Згідно НПАОП 40.1-1.21-98 лабораторію можна віднести до категорії без підвищеної небезпеки, так як в приміщенні відсутні чинники, які викликають підвищену або особливу небезпеку.

Для створення безпечних умов праці необхідно провести ряд організаційних і технічних заходів. Згідно НПАОП 40.1-1.32-01 для запобігання ураження людини електричним струмом в приміщенні застосовується система занулення.

Згідно з вимогами НПАОП 0.00-4.12-05 необхідно провести вступний, первинний на робочому місці, повторний, цільовий та позаплановий інструктажі. Зміст інструктажу відповідає вимогам НПАОП 0.00-4.12-05. Інструктаж відзначається в відповідних журналах з підписами інструктованих і інструктора.

Робота в лабораторії проводиться сидячи і не вимагає фізичної напруги. Тому вона відноситься до категорії Ia (легкі фізичні роботи,

енерговитрати до 120 ккал / год). З метою забезпечити комфортні умови для працівників та відповідно до ДСН 3.3.6.042-99 у відвідуванні встановлені наступні метеорологічні параметри:

а) для холодного періоду:

- 1) температура повітря від 22 до 24 °С;
- 2) вологість повітря від 40 до 60 %;
- 3) швидкість руху повітря оптимальна до 0,1 м/с;

б) для теплого періоду року:

- 1) температура повітря від 23 до 25 °С;
- 2) вологість повітря від 40 до 60 %;
- 3) швидкість руху повітря оптимальна до 0,1 м/с.

Для освітлення робочих місць і приміщення в цілому застосовується як природне бічне освітлення, так і штучне освітлення.

Приміщення з ЕОМ повинні мати природне і штучне освітлення відповідно до ДБН В.25-28-2006 «Природне і штучне освітлення». Природне світло повинно проникати через бічні світлові прорізи, зорієнтовані, як правило, на північ або північний схід, і забезпечувати коефіцієнт природної освітленості (КПО) не нижче 1,5%:

$e^{IV} = 1.35$, де $e^{IV}_{\text{норм}}$ – нормоване значення КПО для 4-го поясу світлового клімату СНД.

Згідно ДСН 3.3.6.037-99 рівень шуму в лабораторії не перевищує 50 дБ.

Загальний рівень штучного освітлення приміщення можна перевірити за допомогою методу питомої потужності.

Розрахункова формула методу [14]:

$$W = \frac{W_{\Sigma}}{S}, \quad (4.1)$$

де W – питома потужність, Вт/м²;

S – площа приміщення, м²;

W_{Σ} – загальна потужність освітлювальної установки, Вт, яка розраховується

за формулою

$$W_{\Sigma} = W_{cv} \cdot n_{cv}, \quad (4.2)$$

де W_{cv} – потужність одного світильника, $W_{cv} = 80$ Вт;

n_{cv} – кількість світильників у приміщенні, $n_{cv} = 4$ шт.

Дане приміщення має площу 30 м², в якому розташовано шість світильників потужністю 80 Вт.

$$W_{\Sigma} = 4 \cdot 80 = 320 \text{ Вт},$$

$$W = \frac{320}{30} = 11 \text{ Вт/м}^2.$$

Табличне значення для отриманого результат освітленість складе 200 лк, коли відповідно до стандарту ДБН В.2.5-28-2006. в лабораторії освітленість повинна бути 300-500лк. Для отримання освітленості в 400 лк необхідна питома потужність 21 Вт/м².

Для поліпшення умов роботи в лабораторії необхідно в денний час застосовувати додаткове освітлення.

ВИСНОВКИ

У даній кваліфікаційній роботі основна увага приділена вивченню методів розпізнавання та ідентифікації деталей у СТЗ роботів.

У першій частині роботи ми проаналізували методи обробки інформації у системах технічного зору роботів. Процес ідентифікації деталей, що лежить у робочій зоні робота, включає два етапи: виділення характерних ознак деталей і розпізнавання деталей за визначеною сукупністю характерних ознак. Ключовими методами обробки інформації є сегментація (процес розподілу сцени на складові частини або об'єкти), визначення порогового рівня, обласна-орієнтована сегментація, дескриптори меж та областей зображень, опис тривимірних сцен і структур, обробка візуальної інформації.

У другій частині проаналізували методи ідентифікації деталей у робототехнічних системах, а також побудували теоретико-множинну модель розпізнавання та ідентифікації.

Головними методами є:

- метод порівняння з еталоном (встановлення збігу двох точкових зображень);
- методи теорії графів та розпізнавання (представлення сегментів контуру у виді графа та пошуку на графі шляху найменшого значення, що відповідає значимим контурам);
- кореляційний метод (обчислення взаємкореляційної функції між еталоном і зображенням);
- розпізнавання через зв'язки шаблонів (узгодження компонентів зображення як шаблон та визначення, які об'єкти є присутніми, опрацювавши запропоновані зв'язки між знайденими шаблонами);
- штучні нейронні мережі (навчання мережі з різних зразків образів із зазначенням того, до якого класу вони належать).

У третій частині роботи розглянули практичну реалізацію методів обробки інформації у робототехнічних системах. Існують два основні підходи до попередньої обробки інформації. Перший підхід ґрунтується на методах просторової області, а другий – на методах частотної області з використанням перетворення Фур'є. Комплексно ці підходи охоплюють більшість з чинних алгоритмів попередньої обробки інформації, що застосовуються у СТЗ роботів. Ключовими бібліотеками, що працюють з методами розпізнавання та ідентифікації, є бібліотека Integrated Performance Primitives (IPP), бібліотека AviCap, бібліотека комп'ютерного зору з відкритим кодом OpenCV. Розглянули основні функції та їхню реалізацію у бібліотеці OpenCV.

У четвертій частині дослідили програмно реалізовані методи розпізнавання та ідентифікації простих деталей. Основною бібліотекою розпізнавання визначили бібліотеку комп'ютерного зору OpenCV. За результатами забезпечуються виділення меж та контурів деталей, визначення центрів мас. Для простих деталей реалізовано функції розпізнавання та ідентифікації.

До перспективних напрямків досліджень слід віднести подальший розвиток методів розпізнавання та ідентифікації, їхню програмну реалізацію для промислових операційних систем реального часу. Разом із системою прийняття рішень такі розробки істотно прискорять реалізацію систем керування інтелектуальними роботами.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки з підготовки й оформлення кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології, освітньо-професійних програм: «Автоматизоване управління технологічними процесами», «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи» / Упоряд. І.Ш. Невлюдов, Р.В. Артюх, Н.П. Демська, В.В. Євсєєв, О.І. Филипенко, О.М. Цимбал. – Харків: ХНУРЕ, 2021. – 50 с.

2. Дипломне проектування для студентів усіх форм навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» : навч. посіб. / за ред. І.Ш. Невлюдов, А.О. Андрусевич, О.В. Токарева, Г.В. Пономарьова. Київ, 2016. 320 с.

3. ДСТУ 3008:2015 Інформація та документація «Звіти у сфері науки і техніки». Структура та правила оформлювання. / В. Земцева; Ю. Поліщук, канд. фіз.-мат. наук; Р. Санченко, канд. техн. наук; Л. Шрамко; А. Ямчук (науковий керівник) ДП «УкрНДНЦ» від 22 червня 2015р. № 61 з 2017-07-01.

4. Положення про кваліфікаційну роботу здобувача вищої освіти на другому (магістерському) рівні [Електронний ресурс] : Наказ ХНУРЕ від 06 травня 2021 р. No 143. – Режим доступу: https://nure.ua/wpcontent/uploads/Main_Docs_NURE/143-vid-06.05.2021-pro-vvedennja-v-dijurishennja-vchenoi-radi-universitetu.pdf.

5. Пащенко О.С., Зозуля К.О. Сучасне виробництво з використанням комп'ютерного управління та інформаційних технологій / «AUTOMATION AND DEVELOPMENT OF ELECTRONIC DEVICES» ADED-2023 Part 2. Харків, ХНУРЕ.: 2023. С. 394-400.

6. Невлюдов І.Ш. Основи виробництва електронних апаратів / І.Ш. Невлюдов. – Харків: Компанія СМІТ, 2005. – 592 с.
7. Кравченко Л.С. Розробка технологічного процесу складання: Навч. посібник для студентів машинобудівних спеціальностей. – Харків: НТУ«ХПІ», 2004 р. – 146 с.
8. Невлюдов І. Ш. Виробничі процеси та обладнання об'єктів автоматизації: Підручник для студентів вищих навчальних закладів / І. Ш. Невлюдов. – Кривий Ріг: Криворізький коледж НАУ, 2017р. – 444 с.
9. Невлюдов І.Ш. Технічні засоби автоматизації: Підручник / І.Ш. Невлюдов, А.О. Андрусевич, О.І. Филипенко, Н.П. Демська, С.П. Новоселов. – Кривий Ріг : Криворізький коледж НАУ, 2019. – 366 с.
10. Невлюдов І.Ш. Людино-машинний інтерфейс в технічних засобах автоматизації: Навчальний посібник / І.Ш. Невлюдов, О.І. Филипенко, Б.О. Шостак. – Харків : «ХТМТ», 2019. – 244 с.
11. Невлюдов І.Ш. Основи наукових досліджень / І.Ш. Невлюдов, Ю.М. Олександров, А.О. Андрусевич, О.О. Чала. – Кривий Ріг : КК НАУ, 2017. – 344 с.
12. Лубко Д.В., Шаров С.В. Методи та системи штучного інтелекту: навч. посібник – Мелітополь: ФОП Однорог Т.В., 2019. – 264с
13. Діагностика та контроль робочих процесів: навч. посіб. для студентів спеціальності «Прикладна механіка» денної та дистанційної форм навчання / В. М. Доля – Харків: НТУ «ХПІ», 2019. – 129 с.
14. Sung K. K. Learning Human Face Detection in Cluttered Scene / K. K. Sung, T. Poggio // Lecture Notes in Computer Science – Computer Analysis of Images and Patterns, 2019. P. 432-439.
15. Rosenblum M. Human Emotion Recognition from Motion Using a Radial Basis Function Network Architecture / M. Rosenblum, Y.Yacoob, L.Davis // IEEE Workshop on Motion of Non-Rigid and Articulated Objects, 2018. – 257 с.

16. Комплекс навчально-методичного забезпечення навчальної дисципліни "Організація керування умовами праці" підготовки освітнього рівня бакалавр усіх спеціальностей та усіх напрямів університету [Електронний ресурс] / ХНУРЕ; розроб.: Т. Є. Стищенко, Г. В. Пронюк, Н. М. Сердюк. – Харків, 2017. – 108 с.