МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Магістерської підготовки

Кафедра Автоматизованих систем моніторингу навколишнього середовища

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: Методи ідентифікації об’єктів у робочій зоні сканера-маніпулятора

Виконала студентка 2 курсу групи МІС-18
спеціальності 122 Комп’ютерні науки

Вісханова Дженєт Гусейнівна

Керівник професор каф. АСМНС, к. т. н., доцент Великодний Станіслав Сергійович

Консультант

Рецензент к. т. н., доцент
Гнатовська Ганна Арнольдівна

Одеса 2019

АНОТАЦІЯ

Методи ідентифікації об’єктів у робочій зоні сканера-маніпулятора.

Вісханова Дженєт Гусейнівна.

Мета роботи – дослідження методів ідентифікації об'єктів у робочій зоні сканера-маніпулятора. Об'єктом роботи – дослідження є система технічного зору робота, що встановлений у митній зоні аеропорту, для сканування валіз пасажирів на транспортерній стрічці.

У роботі проаналізовано методи обробки інформації в системах технічного зору роботів. Основними методами обробки інформації є сегментація, визначення порогового рівня, обласним-орієнтована сегментація, дескриптори кордонів і областей зображень, опис тривимірних сцен і структур, обробка візуальної інформації. Проаналізовано методи ідентифікації об'єктів в робототехнічних системах і побудована теоретико-множинна модель розпізнавання та ідентифікації.

Розглянута практична реалізація методів обробки інформації в робототехнічних системах. Розглянуто основні функції та їх реалізація в бібліотеці OpenCV. Програмно реалізовані методи розпізнавання та ідентифікації простих об'єктів. Базовою бібліотекою розпізнавання обрана бібліотека комп'ютерного зору OpenCV. Для простих об'єктів реалізовані функції розпізнавання та ідентифікації.

До числа перспективних напрямів 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 сканерами.

Магістерська кваліфікаційна робота містить: 73 сторінки, 13 рисунків, 2 таблиці, перелік посилань з 18 найменувань.

КЛЮЧОВІ СЛОВА: СИСТЕМА ТЕХНІЧНОГО ЗОРУ, ОБ’ЄКТ, КОНТУР, РОЗПІЗНАВАННЯ, ПІКСЕЛЬ, КОМП'ЮТЕРНА БІБЛІОТЕКА

SUMMARY

Methods for identification of objects in the work area of a scanner manipulator. Viskhanova Dzheniet.

The purpose of the work is to investigate methods of object identification in the work area of the scanner manipulator. The object of work – the study is a system of technical vision of the work, installed in the customs zone of the airport, for scanning the suitcases of passengers on the conveyor belt.

The methods of information processing in systems of technical vision of robots are analyzed. The main methods of information processing are segmentation, definition of threshold level, regional-oriented segmentation, descriptors of borders and areas of images, description of three-dimensional scenes and structures, processing of visual information. Methods of object identification in robotic systems are analyzed and a theoretical and multiple model of recognition and identification is constructed.

Practical implementation of information processing methods in robotic systems is considered. The basic functions and their implementation in the OpenCV library are considered. Methods for recognizing and identifying simple objects are programmatically implemented. The OpenCV computer vision library was selected as the base recognition library. For simple objects recognition and identification functions are implemented.

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

The master's qualification work contains: 73 pages, 13 drawings, 2 tables, a list of references of 18 titles.

KEYWORDS: TECHNICAL VISION SYSTEM, OBJECT, BOOK, RECOGNITION, PIXEL, COMPUTER LIBRARY

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАК

МКР – магістерська кваліфікаційна робота.

ПСО – перетворення середніх осей.

СТЗ – система технічного зору.

ШНМ – штучні нейронні мережі.

IPP – Integrated Performance Primitives.

MLL – Machine Learning Library.

OpenCV – Open Source Computer Vision Library.

ВСТУП

Аеропорти відіграють найважливішу роль у житті людини, що переміщується на далекі (понад 1 тис. км) відстані. Як правило, з аеропорту починається держава та починаються проблеми в того, хто не поважає законів конкретної держави.

На вході у правове поле держави при прибутті іноземних пасажирів стає митниця із своїми методами виявлення забороненого вантажу. У найбільших аеропортах Світу пасажиропотік становить до 100 тис. пасажирів за добу. Звісно при таких обсягах необхідно застосовувати найшвидші методи виявлення заборонених об’єктів.

Найшвидшим методом контролю вмісту великих валіз є сканування валіз на транспортерній стрічці, проте оператор контролю є живою людиною із притаманними людині властивості втоми та перенапруження. Саме тут людині може прийти на допомогу робот-маніпулятор, який виконує функцію сканера із розпізнаванням об’єктів транспортування, що робить тему магістерської роботи актуальною.

Зорові можливості робота, як і людей, забезпечуються складним чутливим механізмом, який дозволяє гнучко реагувати на зміни зовнішнього середовища. Використання технічного зору та інших методів розпізнавання диктується постійною необхідністю підвищувати гнучкість та розширювати сфери застосування робототехнічних систем. Хоча датчики відстані, тактильні датчики і датчики сили відіграють велику роль в удосконаленні функціонування робота, технічний зір є найбільш потужним джерелом інформації для робота. В силу сказаного, дослідження в області систем технічного зору роботів продовжують залишатися одним з найбільш перспективних напрямків сучасної робототехніки, що і підтверджує актуальність даної роботи.

Зір робота можна визначити як процес виділення, ідентифікації і перетворення інформації, отриманої з тривимірних зображень. Цей процес, що

зветься також технічним або машинним зором, розділяється на шість основних етапів:

а) зняття інформації;

б) попередня обробка інформації;

в) сегментація;

г) опис;

д) розпізнавання;

е) інтерпретація.

Зняття інформації являє собою процес отримання візуального зображення. Попередня обробка інформації полягає у використанні таких методів як зниження шуму або поліпшення зображення окремих деталей. Сегментація – процес виділення на зображенні об'єктів, що цікавлять. При описі визначаються характерні параметри (наприклад, розміри або форма), необхідні для виділення необхідного об'єкта на тлі інших. Розпізнавання являє собою процес ідентифікації об'єктів (наприклад, гайкового ключа, болта, блоку двигуна). Нарешті, інтерпретація виявляє приналежність до групи розпізнаваних об'єктів.

Названі етапи зручно згрупувати відповідно до складності їх реалізації. Розділяють три рівні технічного зору:

– низький;

– середній;

– високий.

Хоча чітких кордонів між цими рівнями не існує, їх виділення доцільно для класифікації різних процесів, характерних для систем технічного зору.

Під низьким рівнем технічного зору розуміються такі процеси, які є простими з точки зору здійснення автоматичних дій, які не потребують наявності штучного інтелекту. До низького рівня технічного зору ми будемо також відносити зняття і попередню обробку інформації.

Під середнім рівнем технічного зору розуміються процеси виділення, ідентифікації та розмітки елементів зображення, отриманого на нижньому рі-

вні. З урахуванням прийнятого підрозділу технічного зору до середнього рівня відносяться сегментація, опис та розпізнавання окремих об'єктів.

Під високим рівнем технічного зору розуміються процеси, пов'язані з штучного інтелекту. У той час як алгоритми, використовувані на нижньому і середньому рівнях технічного зору, розроблені досить добре, знання і розуміння процесів високого рівня поки недостатні, що робить їх особливо цікавими для дослідження.

Метою даної магістерської кваліфікаційної роботи (МКР) є дослідження методів ідентифікації об'єктів у робочій зоні сканера-маніпулятора.

Об'єктом дослідження є система технічного зору робота, що встановлений у митній зоні аеропорту, для сканування валіз пасажирів на транспортерній стрічці.

1 МЕТОДИ ОБРОБКИ ІНФОРМАЦІЇ В СИСТЕМАХ ТЕХНІЧНОГО ЗОРУ РОБОТІВ

1.1 Класифікація методів обробки інформації в системах технічного зору

Одним з найважливіших напрямків розвитку робототехнічних систем завжди була організація взаємодії маніпулятора і сенсорних аналізаторів. Використання сенсорів надає системі гнучкості, 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 сенсорної підсистемою, на основі якої розроблено концепцію візуального управління роботами [1] [1)]. Схема обробки інформації в системі технічного зору (СТЗ) робота показана на рис. 1.

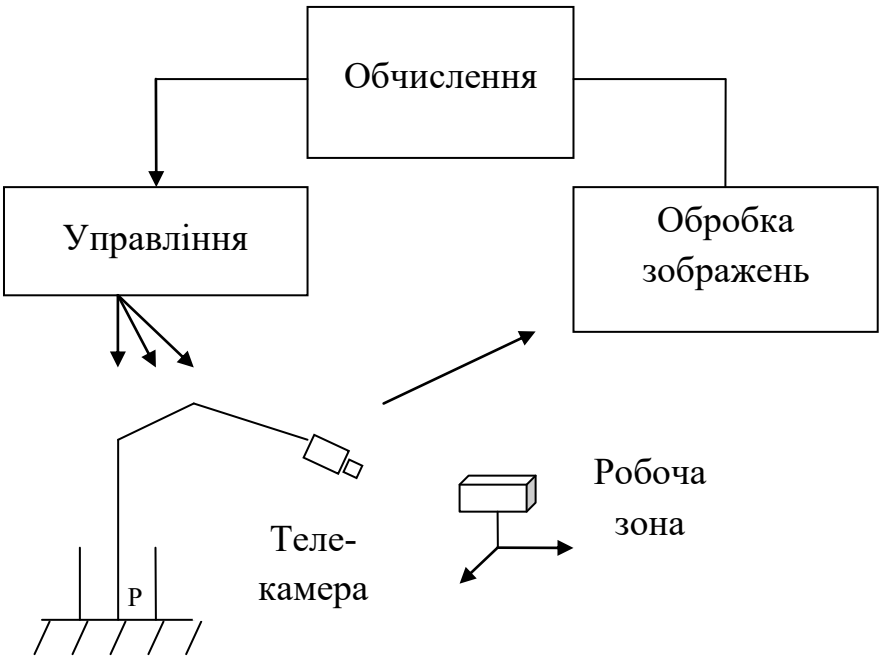

Рисунок 1 – Обробка інформації у СТЗ робота

[1)] [1] Анисимов Б. В., Курганов В. Д. Распознавание и цифровая обработка изображений. Москва: Мир, 1991. 295с.

З метою класифікації методів і підходів, що використовуються в СТЗ, зір розбитий на три основних підкласи: зір низького, середнього і високого рівнів. СТЗ низького рівня призначені для обробки інформації з датчиків.

1.2 Методи попередньої обробки зображень

Процес ідентифікації об'єктів, що знаходяться в робочій зоні робота, зазвичай включає два етапи: виділення характерних ознак об'єктів; власне розпізнавання об'єктів по знайденої сукупності характерних ознак. Відповідно з такою структурою процесу ідентифікації алгоритми обробки інформації у СТЗ прийнято ділити на алгоритми попередньої обробки і алгоритми розпізнавання, що носить, певною мірою, умовний характер, так як в деяких практичних застосуваннях одні і ті ж за математичною сутю алгоритми можуть бути використані на обох етапах розглянутого процесу.

Відповідно до спектральних діапазонів перетворювачів світло-сигнал і принципом дії СТЗ датчик формує зображення робочої зони робота. Під зображенням, зазвичай, розуміється двовимірна картина поля інтенсивності випромінювань робочої зони. Формально отримання зображення полягає у визначенні функціональної залежності інтенсивності випромінювань робочої зони (J) від координат точок зображення x і y:

$$J = f(x, y).$$

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 [2] [1)].

[1)] [2] Фу К., Гонсалес К. Робототехника. Москва: Мир, 2001. 624 с.

1.3 Сегментація об'єктів

1.3.1 Проведення контурів і визначення кордонів

Сегментацією називається процес підрозділу сцени на складові частини або об'єкти. Сегментація є одним з основних елементів роботи автоматизованої системи технічного зору, так як саме на цій стадії обробки об'єкти виділяються зі сцени для подальшого розпізнавання і аналізу. Алгоритми сегментації, як правило, ґрунтуються на двох фундаментальних принципах: розривності і подобі. У першому випадку основний підхід ґрунтується на визначенні контурів, а в другому – на визначенні порогового рівня і розширенні області. Ці поняття застосовні як до статичних, так і до динамічним (залежних від часу) сценам. В останньому випадку рух може служити потужним засобом для поліпшення роботи алгоритмів сегментації.

Методи обчислення градієнта та граничний поділ – визначають розриви в інтенсивності представлення образу об'єкта. В ідеальному випадку ці методи визначають пікселі, що лежать на кордоні між об'єктом і фоном. На практиці даний ряд пікселів рідко повністю характеризує кордон через шум, розривів на кордоні внаслідок нерівномірного освітленості та інших ефектів, що призводять до розмиття зображення. Таким чином, алгоритми виявлення контурів супроводжуються процедурами побудови меж об'єктів з відповідних послідовностей пікселів [3] [1)]. Нижче розглянуто кілька методик, придатних для цієї мети.

1.3.2 Локальний аналіз

Одним з найбільш простих підходів з'єднання точок контуру є аналіз характеристик пікселів у невеликій околиці (наприклад, в околиці розміром 3x3 або 5x5) кожної точки *(x, y)* образу, який вже піддався процедурі вияв-

[1)] [3] Катыс Г. П. Техническое зрение роботов. Москва: Машиностроение, 1989. 176 с.

лення контуру. Всі точки, які є подібними (визначення критерію подібності дано нижче), з'єднуються, утворюючи кордон з пікселів, що володіють деякими загальними властивостями.

При такому аналізі для встановлення подібності пікселів контуру необхідно визначити:

а) величину градієнта, необхідного для побудови контурного пікселя;

б) напрямок градієнта.

Перша характеристика позначається величиною градієнту: *G {f (x, y)}*.

$$G[f(x,y)] = [G^2{}_x + G_y^2]^{\frac{1}{2}} = [(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial x})^2]^{\frac{1}{2}};$$

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

Таким чином, піксель контуру з координатами *(x', y')* подібний за величиною в певній раніше околиці *(x, y)* пікселя з координатами *(x, y)*, якщо справедлива нерівність:

$$|G[f(x,y)] - G[f(x',y')]| \le T,$$

де T – порогове значення.

Напрямок градієнта встановлюється за кутом вектора градієнта, визначеного в рівняннях:

$$G[f(x,y)] = \begin{bmatrix} G \\ G_y \end{bmatrix};\ \theta = arctg \begin{bmatrix} G_x \\ G_y \end{bmatrix},$$

де θ – кут (щодо осі *x*), уздовж якого швидкість зміни має найбільше значення. Тоді можна сказати, що кут пікселя контуру з координатами *(x ', y')*

в деякому околі *(x, y)* подібний до кута пікселя з координатами *(x, y)* при виконанні наступної нерівності:

$$|\theta - \theta'| < A,$$

де *A* – порогове значення кута.

Необхідно відзначити, що напрямок контуру в точці *(x, y)* в дійсності перпендикулярний до напрямку вектора градієнта в цій точці. Однак для порівняння напрямків нерівність дає еквівалентні результати.

Ґрунтуючись на цих припущеннях, ми з'єднуємо точку в деякій околиці *(x, y)* з пікселем, які мають координати *(x, y)*, якщо задовольняються критерії за величиною і напрямком. Рухаючись від пікселя до пікселя і представляючи кожну приєднувану точку як центр міста, процес повторюється для кожної точки образу. Для встановлення відповідності між рівнями інтенсивності освітлення і послідовностями пікселів контуру застосовується стандартна бібліотечна процедура.

Мета полягає у визначенні розмірів прямокутників, за допомогою яких можна побудувати якісне зображення. Побудова таких прямокутників здійснюється в результаті визначення строго горизонтальних і вертикальних контурів. Подальший процес полягав у поєднанні сегментів контуру, розділених невеликими проміжками, і в об'єднанні окремих коротких сегментів.

1.3.3 Глобальний аналіз за допомогою перетворення Хоуга

Розглянемо метод з'єднання граничних точок шляхом визначення їх розташування на кривій спеціального виду. Спочатку припускаючи, що на площині *XY* образу, дано *n* точок, потрібно знайти підпослідовності точок, що лежать на прямих лініях. Одне з можливих рішень полягає у побудові всіх ліній, що проходять через кожну пару точок, а потім у знаходженні всіх підпослідовностей точок, близьких до певних ліній. Завдання, пов'язане з цією

процедурою, полягає у знаходженні: $[n(n–1)]/2 \sim n^2$ ліній і потім в здійсненні $\{n[n(n–1)]\}/2 \sim n^3$ порівнянь кожної точки з усіма лініями. Цей процес трудомісткий, з обчислювальної точки зору, за винятком найпростіших додатків [4] [1)].

Дану задачу можна вирішити по-іншому, застосовуючи підхід, запропонований Хоугом і званий перетворенням Хоуга. Обчислювальна привабливість перетворення Хоуга полягає в поділі простору параметрів на збиральні елементи, де (a_{max}, a_{min}) і (b_{max}, b_{min}) – припустимі величини параметрів ліній.

Елемент $A\ (i, j)$ відповідає площі, пов'язаної з координатами простору параметрів (a_i, b_j). Спочатку ці елементи вважаються рівними нулю. Тоді для кожної точки (x_k, y_k) у площині способу ми вважаємо параметр a рівним кожному з допустимих значень на осі a і обчислюємо відповідне b, використовуючи рівняння:

$$b = –xk + yk.$$

Отримане значення b потім округляється до найближчого допустимого значення на осі b. Якщо вибір a_p призводить до обчислення b_q, ми вважаємо:

$$A\ (p, q) = A\ (p, q) + 1.$$

Після завершення цієї процедури значення M в елементі $A\ (i, j)$ відповідає M точкам в площині XY, що лежать на лінії:

$$y = a_i x + b.$$

Точність розташування цих точок на одній прямій залежить від числа розбиття площині AB. Якщо ми розбиваємо вісь a на k частин, тоді для кожної точки (x_k, y_k) ми отримуємо k значень b, відповідних k можливим значен-

[1)] [4] Хорн Б. К. Зрение роботов. Москва: Мир, 1989. 488 с.

ням *a*. Оскільки є *n* точок образу, процес складається з *nk* обчислювальних операцій. Тому наведена вище процедура лінійна щодо *n* і має менше число обчислювальних операцій, ніж процедура, описана вище, якщо *k* <= n.

1.3.4 Визначення порогового рівня зображень

Поняття порогового рівня (порогу) тест виду:

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

де *f (x, y)* інтенсивність в точці *(x, y)*;

p (x, y) – деякі локальні властивості, які визначаються в околиці цієї точки.

Граничне зображення дається наступним виразом:

$$g(x,y) = \begin{cases} 1, \textit{якщо } f(x,y) > T \\ 0, \textit{якщо } f(x,y) \le T \end{cases},$$

де пікселі у *g (x, y)*, що мають значення 1, відповідають об'єктам, а пікселі, що мають значення 0, відповідають фону.

У рівнянні передбачається, що інтенсивність об'єктів більше інтенсивності фону. Протилежне умова виходить шляхом зміни знаків в нерівностях.

1.4 Обласно-орієнтована сегментація

1.4.1 Основні визначення

Метою сегментації є поділ образу на області. Розглянемо методи сегментації, засновані на прямому знаходженні областей [5] [1)].

[1)] [5] Sung K. K., Poggio T. Learning Human Face Detection in Cluttered Scene. Lecture Notes in Computer Science. *Computer Analysis of Images and Patterns*. 1995. P. 432–439.

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

а) $\bigcup_{i=1}^{n} Ri = R$;

б) P_i – область зв'язку , i = 1, 2, ..., n;

в) $R_i = \varnothing$ для усіх i та j, $i \neq j$;

г) $P(R_i)$ є істиною для i= 1, 2, ..., n;

д) $P(R_i \ U \ R_i)$ є брехнею для $i \neq j$, де $P(R_i)$ – логічний предикат, визначений на точках з множини R_i, та $\varnothing$ – порожня множина.

Умова 1 означає, що сегментація повинна бути повною, тобто кожен піксель повинен перебувати в образі. Друга умова вимагає, щоб точки в області були зв'язковими. Умова 3 вказує на те, що області не повинні перетинатися. Умова 4 визначає властивості, яким повинні задовольняти пікселі у сегментованій області. Простий приклад: P (R_i) = ІСТИНА, якщо усі пікселі у R_i мають однакову інтенсивність. Умова 5 означає, що області R_i і R_i розрізняються по предикату P.

1.4.2 Розбиття і об'єднання області

Викладена вище процедура розширення області починає роботу з заданої множини вузлових точок. Однак можна спочатку розбити образ на ряд довільних непересічних областей і потім об'єднувати та / або розбивати ці області з метою задоволення умов. Ітеративні алгоритми розбиття та об'єднання, робота яких спрямована на виконання цих обмежень, можуть бути викладені таким чином [6] [1)].

На кожному кроці виконуються наступні операції:

[1)] [6] Rosenblum M., Yacoob Y., Davis L. Human Emotion Recognition from Motion Using a Radial Basis Function Network Architecture. *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, 1994. 257 p.

а) розбиття області R_i, для якої $P\{R_i\}$ = КРИВДА, на чотири непересічних квадранта;

б) об'єднання сусідніх областей R_i і R_k, для яких $P(R_i\ U\ R_k)$ = ІСТИНА;

в) вихід на зупинку, коли подальше об'єднання або розбиття неможливе.

1.4.3 Аналіз руху об'єктів

Рух є потужним засобом, який використовується людиною і тваринами для виділення їхніх об'єктів з фонових. У системах технічного зору роботів рух використовується при виконані різних операцій на конвеєрі, при переміщенні руки, оснащеної датчиком, більш рідко при переміщенні всієї робототехнічної системи.

Один з найбільш простих підходів для визначення змін між двома кадрами зображення (образами) $f(x, y, t_i)$ і $f(x, y, t_j)$, що узяті, відповідно, у моменти часу t_i і t_j, ґрунтується на порівнянні відповідних пікселів цих двох образів. Для цього застосовується процедура, яка полягає у формуванні так званої різниці образів.

Припустимо, що ми маємо еталонний образ, який має тільки стаціонарні компоненти. Якщо порівняємо цей образ з таким же чином, що мають рухомі об'єкти, то різниця двох образів виходить в результаті викреслювання стаціонарних компонент (тобто залишаються тільки ненульові записи, які відповідають нестаціонарним компонентам зображення) [7] [1)].

1.4.4 Проблема опису об'єктів

У системах технічного зору проблемою опису називається виділення властивостей (деталей) об'єкта з метою розпізнавання. В ідеальному випадку

[1)] [7] Пентланд А. С., Чаудхари Т. Распознавание лиц для интеллектуальных сред. Открытые Системы. № 3. 2009. С. 86–95.

дескриптори не повинні залежати від розмірів, розташування і орієнтації об'єкта, але повинні містити достатню кількість інформації для надійної ідентифікації об'єктів. Опис є основним результатом при конструюванні систем технічного зору в тому сенсі, що дескриптори повинні впливати не тільки на складність алгоритмів розпізнавання, але також і на їх роботу [8] [1)]. Існує три основні категорії дескрипторів: дескриптори кордону, дескриптори області та дескриптори для опису тривимірних структур [9] [2)].

1.5 Дескриптори областей зображень

1.5.1 Деякі прості дескриптори

Існуючі СТЗ ґрунтуються на досить простих дескрипторах області, що робить їх більш привабливими з обчислювальної точки зору. Як слід очікувати, застосування цих дескрипторів обмежена ситуаціями, у яких об'єкти, що становлять зацікавлення розрізняються настільки, що для їх ідентифікації досить кілька основних дескрипторів.

Площа області визначається як число пікселів, що містяться в межах її кордону. Цей дескриптор корисний при зборі інформації про взаємне розташування і формі об'єктів, від яких камера розташовується приблизно на одному і тому ж відстані. Типовим прикладом може служити розпізнавання СТЗ об'єктів, що рухаються по конвеєру.

Велика й мала осі області корисні для визначення орієнтації об'єкта. Відношення довжин цих осей, зване ексцентриситетом області, також є важливим дескриптором для опису форми області.

Периметром області називається довжина її межі. Хоча іноді периметр застосовується як дескриптор, частіше він використовується для визначення міри компактності області, яка дорівнює квадрату периметра, поділеній на

[1)] [8] Техническое зрение роботов. Пер с англ. / под. ред. А. Пью. Москва: Машиностроение, 1990. 320 с.

[2)] [9] Глазунов А. Компьютерное распознавание человеческих лиц. Открытые Системы. № 3. 2014. С. 107–114.

площу. Відзначимо, що компактність є безрозмірною величиною (тому інваріантна до зміни масштабу) і мінімальної для поверхні, що має форму диска.

Зв'язковою називається область, у якій будь-яка пара точок може бути з'єднана кривої, повністю лежить в цій області. Для безлічі зв'язкових областей (деякі з них мають отвори) в якості дескриптора корисно використовувати число Ейлера, яке визначається як різниця між числом зв'язкових областей і числом отворів. Наприклад, числа Ейлера для літер «А» і «В» відповідно рівні «0» і «1». Інші дескриптори області розглядаються нижче.

1.5.2 Текстура

У багатьох випадках ідентифікацію об'єктів або областей образу можна здійснити, використовуючи дескриптори текстури. Хоча не існує формального визначення текстури, інтуїтивно цей дескриптор можна розглядати як опис властивостей поверхні (однорідність, шорсткість, регулярність). Двома основними підходами для опису текстури є статистичний і структурний. Статистичні методи дають такі характеристики текстури, як однорідність, шорсткість, зернистість тощо. Структурні методи встановлюють взаємне розташування елементарних частин образу, як, наприклад, опис текстури, заснованої на регулярному розташуванні паралельних ліній [10] [1)].

1.5.3 Схема області

Важливим підходом для опису виду структури плоскої області є її представлення у вигляді графа. У багатьох випадках для цього визначається схема (скелет) області за допомогою так званих проріджують (або ж скорочують) алгоритмів. Проріджувати процедури грають основну роль в широкому діапазоні завдань комп'ютерного зору – від автоматичної перевірки дру-

[1)] [10] Абламейко С. В., Лагуновский Д. М.. Обработка изображений: Технология, методы, применение. Минск: Техника, 2009. 302 с.

кованих плат до підрахунку азбестових волокон в повітряних фільтрах. Скелет області можна визначити через перетворення середніх осей (ПСО), запропоноване в роботі. Хоча ПСО дає досить задовільний скелет області, його пряме застосування важко з обчислювальної точки зору, оскільки потрібно визначення відстані між кожною точкою області і кордону. Було запропоновано ряд алгоритмів побудови середніх осей, що володіють більшою обчислювальною ефективністю. Зазвичай це алгоритми проріджування, які інтерактивно усувають з розгляду точки контуру області так, щоб виконувалися наступні обмеження:

а) не усувати крайні точки;

б) не приводити до порушення зв'язності;

в) не викликати надмірного розмивання області.

1.6 Сегментація і опис тривимірних структур

По суті зір є тривимірною проблемою, тому в основі розробки багатофункціональних систем технічного зору, придатних для роботи в різних середовищах, лежить процес обробки інформації про тривимірних сценах. Хоча дослідження в цій області мають більш ніж 10-річну історію, такі фактори, як вартість, швидкість і складність, гальмують впровадження обробки тривимірної зорової інформації в промислових додатках.

Можливі три основні форми подання інформації про тривимірній сцені. Якщо застосовуються датчики, які вимірюють відстань, то ми отримуємо координати (*x, y, z*) точок поверхонь об'єктів [11] [1)]. Застосування пристроїв, що створюють стереозображення, дає тривимірні координати, а також інформацію про освітленість в кожній точці. У цьому випадку кожна точка представляється функцією *f* (*x, y, z*), де значення останньої в точці з координатами (*x, y, z*) дають значення інтенсивності в цій точці (для позначення точки в три-

[1)] [11] Соломатин Н. М. Логические элементы ЭВМ. Москва: Высшая школа. 1990. 160 с.

вимірному просторі і її інтенсивності часто застосовується термін воксель). Нарешті, можна встановити тривимірні зв'язку на основі одного двовимірного образу сцени, тобто можна виводити зв'язки між об'єктами, такі, як «над», «за», «перед». Оскільки точне тривимірне розташування точок сцени зазвичай не може бути обчислено на основі одного зображення, зв'язку, наповнені за допомогою цього виду аналізу, іноді відносяться до так званої 2,5-мірної інформації.

Один з найбільш простих підходів для сегментації і опису тривимірних структур за допомогою координат точок (*x, y, z*) складається у разі необхідності розділення сцени на невеликі плоскі «ділянки» з подальшим їх об'єднанням в більші елементи поверхні відповідно до деякого критерію. Цей метод особливо зручний для ідентифікації багатогранних об'єктів, поверхні яких досить гладкі щодо роздільної здатності.

Коли сцена задана вокселями, її можна описати плоскими ділянками за допомогою тривимірного градієнта. В цьому випадку дескриптори поверхні також виходять в результаті об'єднання цих плоских ділянок. Вектор градієнта вказує напрямок максимальної швидкості зміни функції, а його величина відповідає величині цієї зміни. Ці поняття застосовні для тривимірного випадку і також можуть бути використані для розбиття на сегменти тривимірних структур тим же способом, який застосовувався для двовимірних даних.

2 МЕТОДИ ІДЕНТИФІКАЦІЇ ОБ'ЄКТІВ У РОБОТИЗОВАНОМУ КОМПЛЕКСІ

2.1 Метод порівняння з еталоном

Ідентифікація об'єктів у СТЗ найчастіше проводиться методами порівняння з еталоном. При цьому, як правило, СТЗ вирішують одну з двох завдань. Перше завдання полягає в отриманні зображення одного об'єкта і порівняно з усіма зразками заданого класу. За збігом вибирається найкращий еталон і здійснюється ідентифікація об'єкта, а потім при необхідності знаходяться параметри його положення і орієнтації [12] [1)].

Друге завдання полягає в отриманні зображення декількох об'єктів і почерговому їх порівнянні з еталоном того об'єкта, який необхідно виділити. Після цього обчислюються його положення і орієнтація в робочій зоні робота.

Метод полягає у встановленні збігу двох точкових зображень. Якщо збіг має місце, вважається, що предмет пізнаний. Перевага методу накладення еталона полягає в незалежності від конкретного виду і складності об'єкта. У разі двовимірних систем розпізнати ключ не складніше, ніж коло або будь-яку іншу фігуру. З іншого боку, цей метод розпізнавання може забрати багато часу при використанні обчислювальної машини. Покажемо виникають труднощі на прикладі. На малюнку зображені контури, відповідні кубу і піраміді. Нехай ставиться завдання методом збігу зображень визначити, чи містить ця фігура де-небудь квадрат. Виготовимо еталон з ряду чорних клітин, що утворюють квадрат, оточений зовні і всередині білими клітинами. Накладемо його на фігуру і будемо переміщувати до встановлення збігу. Можуть виникнути труднощі з багатьох причин:

[1)] [12] Форсайт П. Компьютерное зрение. Современный подход. Москва: Издательский дом «Вильямс». 2004. 928 с.

- еталон повинен мати правильний розмір, інакше необхідно виготовити безліч еталонів, що відрізняються один від одного лише масштабним збільшенням;
- невідомо, в якому місці фігури знаходиться передбачуваний квадрат, тому пошук необхідно проводити подвійно – за зрушенням та обертанням;
- позначається просторове квантування: навіть при правильному накладення еталона, збіг не ідеальний, а найкращий з можливих.

З цих причин метод порівняння з еталоном для роботів становить інтерес лише при наявності інформації, що дозволяє заздалегідь обчислити розмір еталона і привести число порівнянь до реально здійсненним.

Метод порівняння з еталоном виявляється цікавим в окремих випадках. Завдяки тому, що способи виготовлення оптичних кореляторов (та навіть електронних кореляторов) відомі, до сих пір цей метод застосовувався в оптичних пристроях, які на СТЗ не встановлювалися. Однак при цьому завжди заздалегідь відомі розміри використовуваних еталонів, а предмети розташовані і орієнтовані належним чином. Йдеться лише про те, щоб пізнати об'єкт із заданої множини об'єктів. Робот повинен вирішувати більш складні завдання.

Повний збіг об'єкта з еталоном в просторі вибраних ознак, як правило, не досягається, тому задається допустиме розходження між еталоном і зображенням, в межах цієї відмінності і перевіряється їх збіг.

Брагін у [13] [1)] дає таке трактування цього методу. Якщо позначити вихідне зображення об'єкта $F(i, j)$, еталон $E(j, i)$, а обчислену відмінність T, то процедуру порівняння можна формально записати як:

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

[1)] [13] Брагин В., Войлов Ю. Системы очувствления и адаптивные промышленные роботы. Москва: Машиностроение, 1985. 256 с.

де індекси *i* і *j* характеризують стан елементів у діскретному цифровому зображенні.

Збіг еталона з об'єктом перевіряється за правилом:

$$D \geq T,$$

де D – задане граничне розходження.

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

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 [13] [1)]. 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 кінцеву, непорожню множину вершин N, разом із безліччю A невпорядкованих 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 вер-

[1)] [13] Брагин В., Войлов Ю. Системы очувствления и адаптивные промышленные роботы. Москва: Машиностроение, 1985. 256 с.

шина 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, ..., n_k, де кожна вершина n_i є наступником вершини r_{i-1}, називається шляхом від n_i до n_k, а вартість шляху визначається формулою:

$$c = \sum_{i=2}^{k} c(ni-1, nj)$$

Елемент контуру ми визначимо як кордон між двома пікселями p і q. В даному контексті під контуром розуміється послідовність елементів контуру.

Важливим методом ідентифікації зображень по геометричним або іншими ознаками служить метод побудови графів рішень. Його успішно застосовують в 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 і 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 рис. 2. Цифрами I, II, ..., X позначені можливі рішення – номери розпізнаних об'єктів. Літери *A, B, ..., Q* у 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 відстоянню кутів один від одного.

Граф може мати більше або менше рівнів та зміст операторів може бути різним.

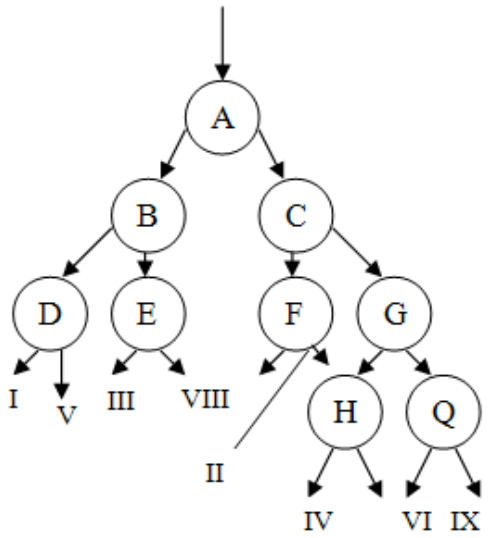

Рисунок 2 – Дерево розпізнавання

2.3 Кореляційний метод

Модифікацією методу порівняння з еталоном є кореляційний метод, заснований на обчисленні взаємокореляційної функції між еталоном та зображенням.

Кореляція – статистичний взаємозв'язок двох або кількох випадкових величин (або величин, які можна з деякою допустимої ступенем точності вважати такими). При цьому, зміни однієї або декількох з цих величин призводять до систематичного зміни іншої або інших величин. Математичної 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 великою кількістю пар ознак для встановлення між ними статистичних взаємозв'язків.

Мета кореляційного аналізу – забезпечити отримання деякої інформації про однієї змінної за допомогою іншої змінної. У випадках, коли можливе досягнення мети, кажуть, що змінні корелюють. У найзагальнішому вигляді

прийняття гіпотези про наявність кореляції означає, що зміна значення змінної *А*, відбудеться одночасно з пропорційною зміною значення *Б*.

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 взаємокореляційної функції *К* можна представити у вигляді:

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

Максимальне значення взаємокорелюючої функції дорівнює:

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

та досягається при повному збігу зображення з еталоном.

Нормована функція:

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

при збігу еталону 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. Іншою властивістю цих методів, яке необхідно

враховувати, є необхідність використання великої кількості еталонів. Це особливо важливо у тих випадках, коли вирішуються завдання розпізнавання об'єктів зміною їх проекції.

2.4 Розпізнавання через зв'язок шаблонів

2.4.1 Опис фрагментів зображення

Невеликі фрагменти зображення можуть мати досить характерний вигляд, якщо мають багато ненульових похідних (наприклад, в кутах). Автори [14] [1)] 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 (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 фрагменту і фрагмента-зразка; якщо замір був менше деякого порогу, тестований фрагмент вважався ідентичним зразку.

Звернемо увагу на те, що по суті описаний класифікатор (він розподіляє фрагменти по класах, представленим зразками, або відмовляється від

[1)] [14] Smith S. M., Brady J. M. SUSAN – a new approach to Low Level Image Processing. Technical Report. 1995. 57 p.

класифікації), а фрагменти представляють шаблони. Програму узгодження шаблонів можна побудувати за допомогою будь-якої техніки, не використовуючи ознаки, інваріантні щодо обертання, як говорили Форсайт та Понс [12][1)]. 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.

2.4.2 Зазначення ознак та породжуюча 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 у 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 породжує моделі зображення. Коли образ присутній, виникнути можуть тільки фрагменти деяких певних класів.

[1)] [12] Форсайт П. Компьютерное зрение. Современный подход. Москва: Издательский дом «Вильямс». 2004. 928 с.

Розробляючи дану модель, отримуємо ряд алгоритмів зіставлення образів з спостерігаються картинами.

Найпростіша версія даної моделі виходить в припущенні, що образ породжує все фрагменти класів, які він може породити, а значить потрібно, щоб шумом породжувалося 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 образ, який отримав найбільшу кількість вказівок, він же вважається присутнім на зображенні. Дана стратегія може бути ефективною, хоча її використання призводить до деяких проблем.

2.4.3 Імовірнісні моделі для вказівки

Отриманий простий процес можна інтерпретувати через вірогідну модель, оскільки при цьому виявляються сильні і слабкі сторони описаного підходу. Фосайт [12] [1)] пише, що використовувану породжувальну модель можна перетворити в вірогідну, 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 припу-

[1)] [12] Форсайт П. Компьютерное зрение. Современный подход. Москва: Издательский дом «Вильямс». 2004. 928 с.

щеннях для обчислення функції правдоподібності потрібно знати тільки, які фрагменти породжені образом, 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:

$$P\,(\textit{інтерпретація} \mid \textit{образ}) = \lambda^{n_p}\,\mu^{(n_i - n_p)},$$

де 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) породжує 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.

2.4.4 Доопрацювання породжує моделі

Відносно просто враховувати всі вищесказане і, відповідним чином, удосконалити найпростішу модель (фрагменти з'являються незалежно за умови наявності образу). Припустимо, що є N різних типів фрагментів. Фрагменти кожного типу генеруються зі своєю ймовірністю усіма зображеннями і шумом. Тепер припустимо, що на зображенні присутній n_{il} примірників фрагментів l-го типу. Більш того, n_k з них генерується чином, а решта – шу-

мом. Тепер оскільки фрагменти виникають незалежно при даному способі і шум не залежить від способу, правдоподібність дорівнює:

P (фрагменти породжені образом | j-й образ)
P (фрагментів породжено шумом).

Перший член дорівнює:

P (тип 1 | j-й образ); P (тип 2 | j-й образ); P (тип N | j-й образ);ˈP (шум),

що можна оцінити як:

$$P_{1j}^{n_1} P_{2j}^{n_2} \ldots P_{Nj}^{n_N} .$$

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

$$P\,(тип1|\ шиу)^{(n_{i1}-n_1)} \ldots P\,(тип1|шум)^{(n_{iN}-n_N)},$$

що можна оцінити як:

$$P_{1x}^{(n_{i1}-n_1)} \ldots P_{Nx}^{(n_{iN}-n_N)}.$$

Це можна записати у такий спосіб:

$$P_{1j}^{n_1} P_{2j}^{n_2} \ldots P_{Nj}^{n_N}\ P_{1x}^{(n_{i1}-n_1)} \ldots P_{Nx}^{(n_{iN}-n_N)}.$$

2.4.5 Зазначення на тривимірні об'єкти

Хоч підхід Шміда і Мора був описаний з позиції двовимірних образів, його відносно просто розширити на розпізнавання тривимірних об'єктів. Для цього кожен набір проекцій об'єкта розглядаються 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 відомості тривимірного розпізнавання до двовимірного застосовується досить широко, але має і свої складності. Основна проблема - це велике число моделей, що може привести до утруднення процедури вказівки. Невідомо також, яке мінімальне число проекцій, необхідних для погодження 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 та функціонування біологічних нейронних мереж – мереж нервових клітин живого організму. Це поняття виникло при вивченні процесів, що протікають в мозку, і при спробі змоделювати ці процеси.

ШНМ є систему з'єднаних і взаємодіючих між собою простих процесорів (штучних нейронів). Такі процесори зазвичай досить 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 досить велику мережу з керованим вза-

ємодією, такі локально прості процесори разом здатні виконувати досить складні завдання. Проста штучна нейронна мережа представлена на рис. 3.

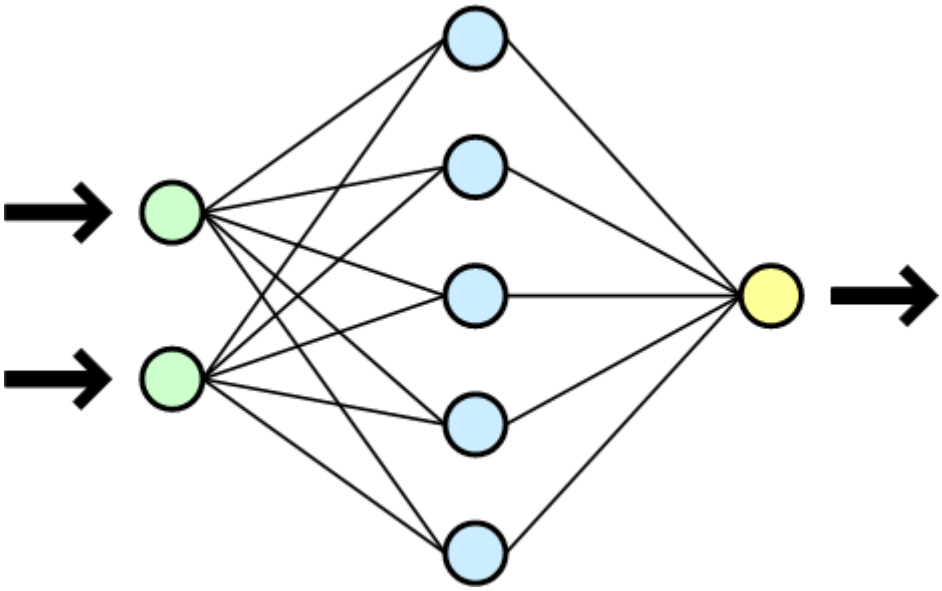

Рисунок 3 – Проста нейронна мережа

З точки зору машинного навчання, нейронна мережа являє собою окремий випадок методів розпізнавання образів, дискримінантного аналізу, методів кластеризації тощо. З математичної точки зору, навчання нейронних мереж – це багатопараметрична завдання нелінійної оптимізації. З точки зору кібернетики, нейронна мережа використовується в задачах адаптивного управління і як алгоритми для робототехніки.

З точки зору розвитку обчислювальної техніки та програмування, нейронна мережа – спосіб вирішення проблеми ефективного паралелізму. А з точки зору штучного інтелекту, ШНМ є основою філософської течії коннективізма і основним напрямком в структурному підході з вивчення можливості побудови (моделювання) природного інтелекту за допомогою комп'ютерних алгоритмів.

Нейронні мережі не програмуються в звичному сенсі цього слова, вони навчаються. Можливість навчання – одне з головних переваг нейронних мереж перед традиційними алгоритмами. Технічно навчання полягає в знахо-

дженні коефіцієнтів зв'язків між нейронами. В процесі навчання нейронна мережа здатна виявляти складні залежності між вхідними даними і вихідними, а також виконувати узагальнення. Це означає, що в разі успішного навчання мережа зможе повернути вірний результат на підставі даних, які були відсутні в навчальній вибірці, а також неповних і / або «зашумлених», частково перекручених даних.

Як образів можуть виступати різні за своєю природою об'єкти: символи тексту, зображення, зразки звуків тощо. При навчанні мережі пропонуються різні зразки образів із зазначенням того, до якого класу вони відносяться. Зразок, як правило, представляється як вектор значень ознак. При цьому сукупність усіх ознак повинна однозначно визначати клас, до якого належить зразок. У разі якщо ознак недостатньо, мережа може співвіднести один і той же зразок з декількома класами, що невірно. Після закінчення навчання мережі їй можна пред'являти невідомі раніше образи і отримувати відповідь про належність до певного класу.

Топологія такої мережі характеризується тим, що кількість нейронів у вихідному шарі, як правило, дорівнює кількості визначених класів. При цьому встановлюється відповідність між виходом нейронної мережі і класом, який він представляє. Коли мережі пред'являється якийсь образ, на одному з її виходів повинен з'явитися ознака того, що образ належить цьому класу. У той же час на інших виходах повинен бути ознака того, що образ даного класу не належить. Якщо на двох або більше виходах є ознака приналежності до класу, вважається що мережа «не впевнена» в своїй відповіді [15] [1)].

2.6 Розпізнавання об'єктів і їх інтерпретація

Розпізнаванням називається процес розмітки, тобто алгоритми розпізнавання ідентифікують кожен об'єкт сцени і привласнюють йому мітки

[1)] [15] Введение в компьютерное зрение. URL: http://graphics.cs.msu.ru/ru/library/cv/cv_intro.html (дата звернення: 28.10.2019).

(гайковий ключ, перемичка). Зазвичай в більшості промислових систем технічного зору передбачається, що об'єкти сцени сегментовані як окремі елементи. Інша загальне обмеження відноситься до розташування пристроїв збору інформації щодо досліджуваної сцени (зазвичай вони розташовуються перпендикулярно робочої поверхні). Це призводить до зменшення відхилень в характеристиках форми, а також спрощує процес сегментації та опису в результаті зменшення ймовірності загороджування одних об'єктів іншими. Управління відхиленнями в орієнтації об'єкта проводиться шляхом вибору дескрипторів, інваріантних до обертання, або шляхом використання головних осей об'єкта для орієнтування його в попередньо визначеному напрямку.

Сучасні методи розпізнавання діляться на дві основні категорії: теоретичні та структурні методи. Теоретичні методи ґрунтуються на кількісному описі (статична структура), а в основі структурних методів лежать символічні описи і їх зв'язку (послідовності напрямків в кордоні, закодованої за допомогою ланцюгового коду).

Інтерпретацію – процес, який дозволяє системі технічного зору придбати більш глибокі знання про навколишнє середовище в порівнянні зі знаннями, отриманими за допомогою методів, викладених 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, досягнення поки досить незначні.

2.7 Теоретико-множинна модель розпізнавання та ідентифікації

Розглянемо теоретико-множинну модель процесу розпізнавання та ідентифікації об'єктів роботизованого виробництва.

Нехай задано простір S в якому існує безліч об'єктів $X = \{x_1 \dots x_n\}$, також в ньому знаходиться робот R, оснащений системою технічного зору S_R. Нехай кожному об'єкту x_i відповідає набір ознак $\{f_{1i} \dots f_{ki}\}$.

Тоді безліч об'єктів $X = \{x_1, x_2 \ldots x_n\}$ можна охарактеризувати матрицею ознак:

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

Процес ідентифікації буде полягати в тому, що об'єкт x_i може бути розпізнаний по підмножині ознак f_i' з безлічі ознак F. Для всієї множини об'єктів X набір підмножин f_i' формує безліч F':

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

Загальний вигляд процедури розпізнавання об'єктів формально представляється у вигляді:

$$R(S) \times \mathrm{X} \Rightarrow \mathrm{X}',$$

де $R(S)$ – робот, оснащений системою датчиків S; X – безліч об'єктів; X' – безліч розпізнаних об'єктів.

Процедура розпізнавання та ідентифікації об'єктів представлена у вигляді:

$$\mathrm{R(S)} \times \mathrm{X} \equiv \begin{bmatrix} \mathrm{S}_1 \\ \mathrm{S}_2 \\ \cdots \\ \mathrm{S}_n \end{bmatrix} \times \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1m} \\ f_{21} & f_{22} & \cdots & f_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ f_{m1} & f_{m2} & \cdots & f_{mm} \end{bmatrix} \Rightarrow \begin{bmatrix} f'_{11} & f'_{12} & \cdots & f'_{1k} \\ f'_{21} & f'_{22} & \cdots & f'_{2k} \\ \cdots & \cdots & \cdots & \cdots \\ f'_{m1} & f'_{m2} & \cdots & f'_{mk} \end{bmatrix} \Rightarrow \mathrm{X}'$$

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 властивостей;

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

X' – множина розпізнаних об’єктів.

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

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

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 властивостей, що характеризує множину ідентифікованих об’єктів X''.

Дана модель може бути використана при розробці підсистем розпізнавання та ідентифікації в складі систем технічного зору. Її застосування дозволить забезпечити вирішення завдань розпізнавання та ідентифікації об'єктів в робочій зоні сканера-маніпулятора.

У другій частині роботи проаналізовані методи ідентифікації об'єктів в робототехнічних системах і побудована теоретико-множинна модель розпізнавання та 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 найменшої вартості, який відповідає значущим контурам);
- кореляційний метод (обчислення взаємокореляційної функції між еталоном і зображенням);
- розпізнавання через зв'язку шаблонів (узгодження компонентів зображення як шаблон і визначення, які об'єкти присутні, вивчивши запропоновані зв'язку між знайденими шаблонами);
- штучні нейронні мережі (навчання мережі по різних зразків образів із зазначенням того, до якого класу вони відносяться).

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 СТЗ роботів [16] [1)].

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:

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

де $f(x, y)$ – вхідне зображення;

$g(x_y, y)$ – вихідне (оброблене) зображення;

h – оператор функції, визначений у 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 $g(x_y, y)$ оператор застосовується для кожного положення (x, y). Хоча іноді

[1)] [16] Цифровое компьютерное зрение. URL: http://graphics.cs.msu.ru/courses/cg02b/library/ (дата звернення: 28.10.2019).

використовуються і інші форми околиці (наприклад, коло), квадратні форми більш кращі з-за простоти їх реалізації.

Один з найбільш часто зустрічаються методів просторової області заснований на використанні так званих масок згортки (або шаблонів, вікон або фільтрів). Зазвичай маска являє собою невелику (наприклад, розмірність 3x3) двовимірну систему, коефіцієнти якої вибираються таким чином, щоб виявити заданий властивість зображення.

3.1.2 Методи частотної області

До частотної області відноситься сукупність комплексних пікселів у вигляді перетворення Фур'є від зображення. Поняття «частота» використовується при інтерпретації перетворення Фур'є і випливає з того факту, що результат цього перетворення є сумою синусоїд. Через підвищення вимог до обробки результатів методи частотної області не так широко використовуються в технічному зорі роботів, як методи просторової області [17] [1)].

Однак перетворення Фур'є грає важливу роль при аналізі руху об'єкта і при описі об'єкта. Крім того, багато просторові методи для поліпшення якості і відновлення зображення базуються на концепціях перетворення Фур'є.

Дозволяється проводити пряму процедуру обчислення двовимірного перетворення Фур'є, використовуючи тільки однорозмірних БПФ-алгоритм. Перетворення Фур'є можна використовувати при вирішенні багатьох завдань систем технічного зору. Наприклад, за допомогою уявлення кордону об'єкта у вигляді одновимірного масиву точок і обчислення їх перетворення Фур'є отримані значення можуть бути використані в якості опису форми кордону. Одномірне перетворення Фур'є є також ефективним при виявленні руху об'єкта.

[1)] [17] Open Source Computer Vision Library. URL: http://www.intel.com/technology/computing/opencv/index.htm (дата звернення: 28.10.2019).

Використання дискретного двовимірного перетворення Фур'є можливо при зміні, збільшенні і відновленні зображень, хоча, як вже зазначалося, застосування цього методу в промислових системах технічного зору досі обмежено складністю необхідних обчислень. На закінчення відзначимо, що двовимірне аналогове перетворення Фур'є може бути здійснено (зі швидкістю світла) оптичними засобами. Цей підхід, що вимагає використання прецизійного оптичного устаткування, застосовується в промислових умовах для вирішення таких завдань, як перевірка якості поверхні металу після фінішної обробки.

3.2 Бібліотеки для обробки зображень

3.2.1 Бібліотека Integrated Performance Primitives

Бібліотека Integrated Performance Primitives (IPP) використовує розширені набори інструкцій процесора: MMX, SSE, SSE2, SSE3, SSSE3, SSE4 і багатоядерні процесори.

Intel IPP розділений на три основні групи: сигнали (лінійний масив даних або вектор), зображення (двомірний масив для типових колірних просторів) і матриці (*n* x *m* масиви для матричних операцій).

Половина функцій для матричних операцій, третина для обробки сигналів і залишилися для зображень. Функції Intel IPP розділені на 4 типи даних: 8u (8-бітові беззнакові), 8s (8-бітові зі знаком), 16s, 32f (32-биті з плаваючою точкою), 64f тощо. Як правило, розробники додатків працює тільки з одним домінуючим типом даних для більшості функцій обробки, і тільки в кінці обробки виробляють перетворення у вихідний формат [18] [1)].

Можливостями Intel IPP є наступні функції:

– кодування і декодування відео;

– кодування і декодування аудіо;

[1)] [18] Компьютерное зрение. URL: http://www.roboforum.ru/viewforum.php?f=51&sid=29ea9ba7faf446dda7cbdbcbd5a45d42 (дата звернення: 30.10.2019).

– JPEG / JPG;
– машинний зір;
– криптографія;
– стиснення даних;
– перетворення кольору;
– обробка зображення;
– трасування променів / візуалізація;
– обробка сигналів;
– кодування мови;
– розпізнавання мови;
– обробка рядків;
– векторна / матрична математика.

3.2.2 Бібліотека AviCap

Додатки, призначені для запису звукових даних і відеоданих, можуть скористатися зручним високорівневим інтерфейсом, що надаються бібліотекою «avicap.dll». Створюючи додатки для запису відео, не доведеться піклуватися про внутрішню структуру «avi-файлів», про компресії (стиску) даних при записі, про інтерфейс з драйверами пристрою (або пристроїв) записи. При необхідності, тим не менш, можна скористатися інтерфейсом нижчого рівня, який забезпечується бібліотекою «avifile.dll».

Для створення додатків, записуючих відео, найкраще скористатися класом вікна AVICap, визначеному в бібліотеці «avicap.dll». Створивши вікно на базі класу AVICap, додаток отримає в своє розпорядження простий інтерфейс для запису відео і звукових даних в «avi-файл», для попереднього перегляду відео і виконання інших операцій.

У класі AVICap передбачені кошти динамічного перемикання пристроїв запису відео та звуку, що зручно в тих випадках, коли можливо почергове використання декількох таких пристроїв, встановлених в комп'ютері. Дода-

ток може створити «avi-файл», призначений для запису, скопіювати вміст одного «avi-файлу» в інший, встановити частоту кадрів, вивести на екран діалогову панель, за допомогою якої користувач зможе задати формат запису. Присутні засоби для роботи з палітрами і універсальним буфером обміну Clipboard.

Для запису звуку клас вікна AVICap користає засоби бібліотеки «mmsystem.dll».

3.2.3 Бібліотека OpenCV

OpenCV (англ.: Open Source Computer Vision Library), бібліотека комп'ютерного зору з відкритим вихідним кодом) – бібліотека алгоритмів комп'ютерного зору, обробки зображень та чисельних алгоритмів загального призначення з відкритим кодом. Реалізована на C / C++, так само розробляється для Python, Ruby, Matlab і інших мов.

OpenCV призначена для підвищення обчислювальної ефективності процедур обробки відеозображення з особливим наголосом на застосування в задачах реального часу.

OpenCV написана на C добре оптимізована і може використовувати переваги багатоядерних процесорів. Для більш повного використання можливостей бібліотеки рекомендується встановити IPP – це дозволить підвищити продуктивність процедур бібліотеки.

Дозволяє досить швидко і ефективно реалізовувати складні алгоритми машинного зору. Бібліотека містить більше 500 функцій, які дозволяють реалізовувати програми працюють у багатьох областях, в тому числі:

- контроль якості продукції, що випускається;
- обробці зображень в медицині;
- забезпечення безпеки;
- інтерфейсі користувача;
- робототехніці.

OpenCV містить бібліотеку загальних функцій штучного інтелекту Machine Learning Library (MLL). Вона служить, в основному, для розпізнавання фрагментів зображення і кластеризації.

Дану бібліотеку застосовують:

– для затвердження загального стандартного інтерфейсу комп'ютерного зору для додатків в цій області;
– для сприяння зростанню числа таких додатків і створення нових моделей використання PC;
– для виконання платформи Intel привабливими для розробників таких додатків за рахунок додаткового прискорення OpenCV за допомогою Intel® Performance Libraries – зараз включають IPP (засоби низького рівня бібліотеки для обробки сигналів, зображень, а також медіакодеки) і MKL (спеціальна версія LAPACK і FFTPack);
– OpenCV автоматично визначає присутність IPP і MKL і використовувати їх для прискорення обробки.

Основними модулями є наступні модулі.

Ядро CXcore виробляє:

– базові операції над багатовимірними числовими масивами;
– матричну 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 – є модулем обробки зображень і комп'ютерного зору, його функціями є:

– базові операції над зображеннями (фільтрація, геометричні перетворення, перетворення колірних просторів тощо)
– аналіз зображень (вибір відмінних ознак, морфологія, пошук контурів, гістограми);

– структурний аналіз (опис форм, пласкі розбиття);

– аналіз руху, спостереження за об'єктами;

– виявлення об'єктів, зокрема осіб;

– калібрування камер, елементи відновлення просторової структури.

Highgui – модуль для введення / виведення зображень і відео, створення призначеного для користувача інтерфейсу, необхідний для виконання наступних операцій:

– захоплення відео з камер і з відео файлів, читання / запис статичних зображень;

– функції для організації простого UI (зараз все демо додатки використовують HighGUI).

Cvaux – експериментальні і застарілі функції

Даний модуль виконує:

– просторовий зір: стерео калібрації, автокалібрації;

– пошук стерео-відповідності, кліки в графах;

– знаходження і опис рис обличчя;

– порівняння форм, побудова скелетонів;

– приховані Марковські ланцюги;

– опис текстур.

Детектор Харріса. За останні двадцять років було створено багато різних детекторів точкових особливостей зображень. Найчастіше використовуються детектор Харріса і детектор за мінімальними власним значеннями. У даній роботі реалізовані обидва.

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

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

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

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

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

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

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:

$$R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 = \det(M) - k(Trace(M))^2$$

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

Далі з нею проводяться аналогічні дії, пошук локальних максимумів, і введення порогового значення.

На другому етапі – етапі спостереження (Tracking) – визначаються зміщення для кожної особливої точки між сусідніми кадрами і розраховується середній по всіх точках вектор зсуву для кожної пари кадрів.

Основна ідея завдання стеження полягає в пошуку околиці в поточному кадрі, максимально схожою на околицю точки в попередньому кадрі. Наступний ключовий момент полягає в тому, що цей пошук околиці проводиться не по всьому зображенню, а в деякому радіусі від початкового положення особливої точки. Тут ми вводимо припущення, що за час між двома кадрами в сцені не відбувається великих змін, тому немає сенсу шукати точку далеко від поточного її положення.

Таким чином, 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:

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

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

R – область пошуку особливості;

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

Функцією пошуку контурів є функція cvFindContours:

```
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 одноканальное 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 – осередок пам'яті, який містить отримані контуру;
– first_contour – вих.ідний параметр, що містить покажчик на перший знайдений контур;
– header_size – розмір послідовності заголовка > = sizeof (CvChain), якщо метод CV_CHAIN_CODE, та > = sizeof (CvContour) в іншому випадку;
– mode – режим;
– method – метод апроксимації (для всіх режимів, крім CV_RETR_RUNS, який використовує вбудовану апроксимацію);
– offset – зрушення для кожної точки контуру, використовується якщо контури витягуються із зображень, потім вони повинні бути проаналізовані в контексті цілого зображення.

Параметри функції виділення контурів, зведено до табл. 1, а методи для її апроксимації – до табл. 2.

Таблиця 1 – Параметри функції виділення контурів

Функція	Параметр
CV_RETR_EXTERNAL	якщо необхідні тільки крайні зовнішні контури
CV_RETR_LIST	виділення всіх контурів і приміщення їх в список
CV_RETR_CCOMP	виділення всіх контурів і приміщення їх в подвійну ієрархію
CV_RETR_TREE	отримати всі контуру і реконструювати повну ієрархію вкладених контурів

Таблиця 2 – Методи апроксимації для функції визначення контурів

Функція	Метод
CV_CHAIN_APPROX_NONE	перетворює усі ланцюжки у точки
CV_CHAIN_APPROX_SIMPLE	стискає горизонтальних, вертикальних і діагональних сегментах, тобто функція залишається тільки у центрі
CV_CHAIN_CODE	вихідні контури у ланцюжковому коді Фрімена
CV_CHAIN_APPROX_TC89_L1, CV_CHAIN_APPROX_TC89_KCOS	застосування одного з апроксимаційних ланцюжкових кодів методом Тех-Чину
CV_LINK_RUNS	використовувати інший алгоритм пошуку контуру за допомогою горизонтальних зв'язків сегментів в один

Функція наближення полігональних кривих із заданою точністю:

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

де src_seq – послідовність ряду точок;

header_size – розмір заголовка кривої;

storage – контейнер з апроксимуючими контурами (якщо NULL, то використовується комірка 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 та збільшення розміру зображень:

```
void cvPyrDown (const CvArr * src, CvArr * dst, int filter =
                        CV_GAUSSIAN_5x5);
```

де src – вихідне зображення;

dst – вхідне зображення (має містити довжину і ширину);

filter – тип фільтра (підтримується тільки CV_GAUSSIAN_5x5).

Порогова функція в бібліотеці комп'ютерного зору OpenCV 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 у відтінках 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);
```

де Image – вхідне 8-бітове зображення;

circle_storage – осередок, що зберігає виявлені кола, може бути сховищем (в даному випадку послідовність кіл створюється при зберіганні та повертається функцією), або один рядок / один стовбчик матриці (CvMat *) типу CV_32FC3, до якого пишуться параметри кіл;

circle_storage – матриця та фактичне число ліній, що перевищує розмір матриці, з максимально можливою кількістю кіл, кожне коло кодується у 3 числа з плаваючою точкою: центр з координатами (*x, y*) та радіусом;

mthod – реалізована тільки методом CV_HOUGH_GRADIENT;

Dp – установка акумулятора, який використовується для виявлення центрів кіл (наприклад, якщо він дорівнює 1, акумулятор буде мати той же дозвіл, як у вихідного зображення, якщо це 2 – акумулятор буде в два рази менше ширини і висоти тощо;

min_dist – мінімальна відстань між центрами виявлених кіл, якщо параметр занадто малий, то кілька сусідніх кіл може бути помилково виявлено на додаток до дійсних, якщо він занадто великий, 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 менше параметр, тим більше помилкових кіл можуть бути виявлені;

min_radius – мінімальний радіус шуканих кіл;

max_radius – максимальний радіус.

У третьому розділі роботи розглянуто практичну реалізацію методів обробки інформації у робототехнічних системах. Існують два основні підходи до попередньої обробки інформації. Перший підхід заснований на методах просторової області, а другий – на методах частотної області із використанням перетворення Фур'є.

Разом ці підходи охоплюють більшість з існуючих алгоритмів попередньої обробки інформації, що застосовуються у СТЗ роботів. Основними бібліотеками, що працюють з методами розпізнавання та ідентифікації є бібліотека Integrated Performance Primitives (IPP), бібліотека AviCap, бібліотека комп'ютерного зору із відкритим кодом OpenCV. Також розглянуто основні функції та їх реалізація у бібліотеці OpenCV.

4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИЗНАЧЕННЯ ПАРАМЕТРІВ ОБ'ЄКТІВ ЗА ДОПОМОГОЮ СИСТЕМИ ТЕХНІЧНОГО ЗОРУ

4.1 Основні особливості розробленого програмного забезпечення

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

З точки зору MFC проект буде складатися з двох класів: CApp (клас додатки) 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).

Покажчиками 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 наступним чином:

```
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);
}
```

На рис. 4 зазначено головне вікно програми, яке містить приклад вихідного зображення, отриманого камерою СТЗ робота-маніпулятора.

Рисунок 4 – Приклад зображення в головному вікні програми

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 ().

На рис. 5 наведено приклад застосування функції Canny, що забезпечує виділення кордонів зображень.

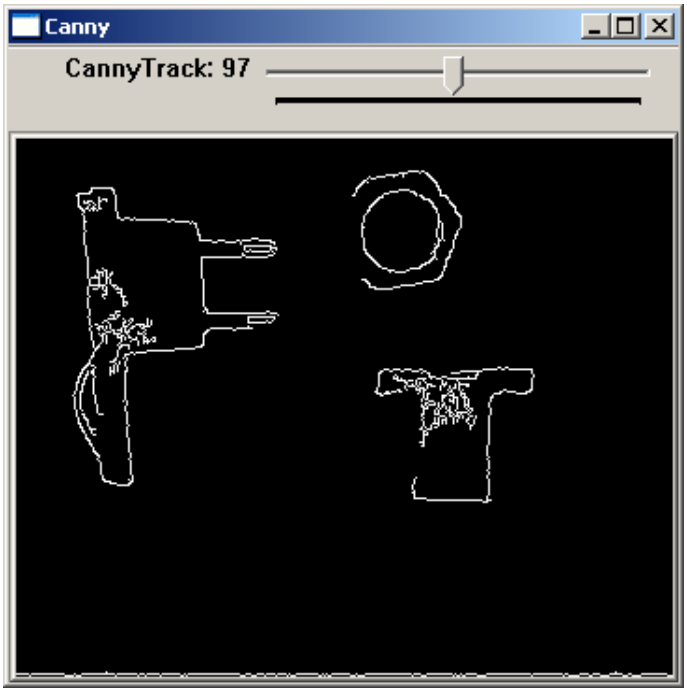

Рисунок 5 – Вікно програми з функцією Canny

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. На рис. 6 приведено виконання функції.

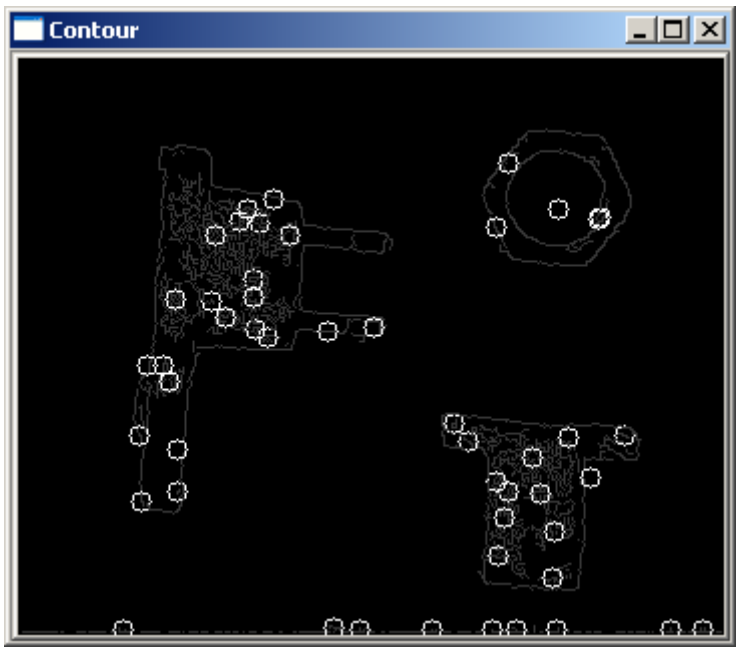

Рисунок 6 – Визначення контурів зображень

Згодом 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 у 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 (рис. 7).

Рисунок 7 – Основне вікно програми (зменшений вигляд)

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 у чорно-біле (використовується функція cvCvtColor (src2, gray, CV_RGB2GRAY)), після чого виконати саме перетворення функцією:

```
cvCanny (gray, dst3,25,100 + cannyt, 3).
```

Далі проводиться висновок цього зображення у вікно dst3 за допомогою функції:

```
cvShowImage ( "Canny", dst3).
```

На рис. 8 представлений приклад виконання функції Кенні до вихідного зображення.

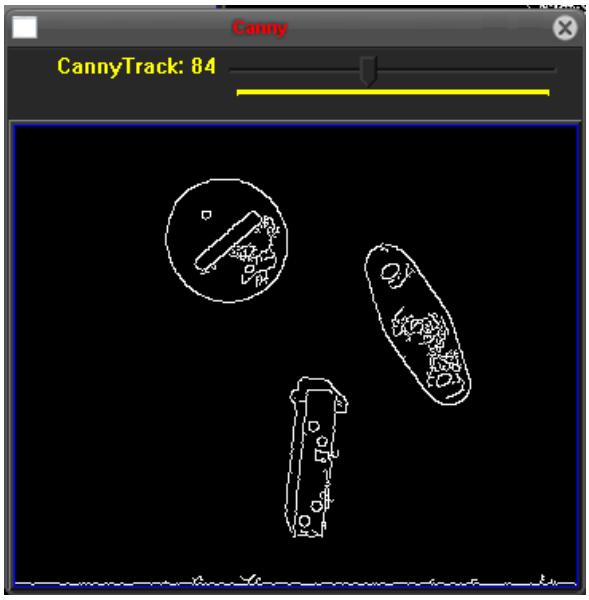

Рисунок 8 – Приклад перетворення Кенні

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

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

де tgray – вхідний масив (8- або 32-бітний);

gray – вихідний масив такого ж типу як і tgray;

100 та 255 – мінімальне і максимально значення, яке використовується пороговою функцією;

CV_THRESH_BINARY – показує тип порогової функції.

Приклад використання порогової функції представлений на рис. 9.

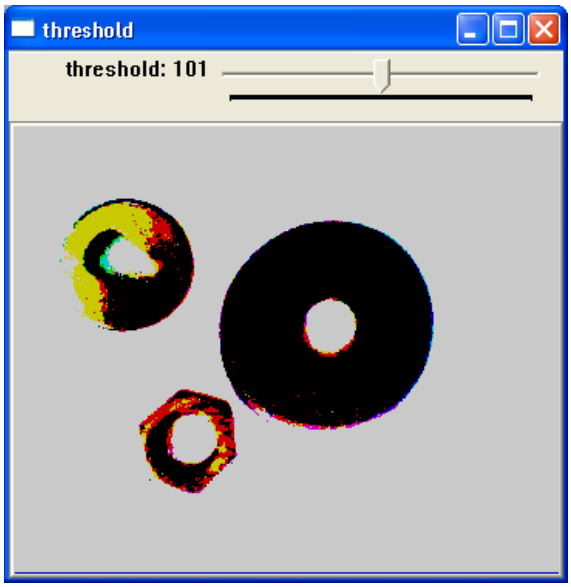

Рисунок 9 – Застосування порогової функції

Більш поліпшеною різновидом порогової функції є функція:

```
cvAdaptiveThreshold (dst3, dst4, 5 + thresh2,
CV_ADAPTIVE_THRESH_MEAN_C, CV_THRESH_BINARY, 3,5).
```

Приклад використання функції наведено на рис. 10.

Рисунок 10 – Застосування адаптивної порогової функції

4.3 Реалізація функцій розпізнавання та ідентифікації

Результатом виконання функції Кенні є зображення, що містить краю об'єктів у вихідному кадрі, і є основою для подальших перетворень, зокрема для виділення контурів.

Розглянемо функцію cvFindContours, яка знаходить контуру об'єктів, що знаходяться у полі зору камери:

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

де gray є вихідним зображенням у 8-бітному форматі у області пам'яті;

storage – зберігає отримані контуру;

у масиві contours міститься інформація про перший знайдений контур;

CvContour показує розмір структури даних контуру;

параметр CV_RETR_LIST означає, що усі 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, а потім повинні бути проаналізовані у контексті цілого зображення.

Для знаходження декількох контурів 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;     }.
```

На рис. 11 представлений результат застосування функції виділення контурів.

Функція cvHoughCircles знаходить окружності на сірому зображенні, використовуючи перетворення Хью:

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

де параметр gray є 8-битим зображенням;

cstorage – область пам'яті, в якій зберігаються окружності, виявлені функцією;

CV_HOUGH_GRADIENT – метод реалізації функції;

1 – стек, який використовується для виявлення центрів кіл з тим же дозволом, що і вихідне зображення;

gray-> height / 16 – обчислюється мінімальна відстань між центрами виявлених кіл;

наступні 2 параметра відповідають за накопичення порогів у виявлених колах;

останні 2 параметра відповідають за мінімальний і максимальний радіус знайдених кіл.

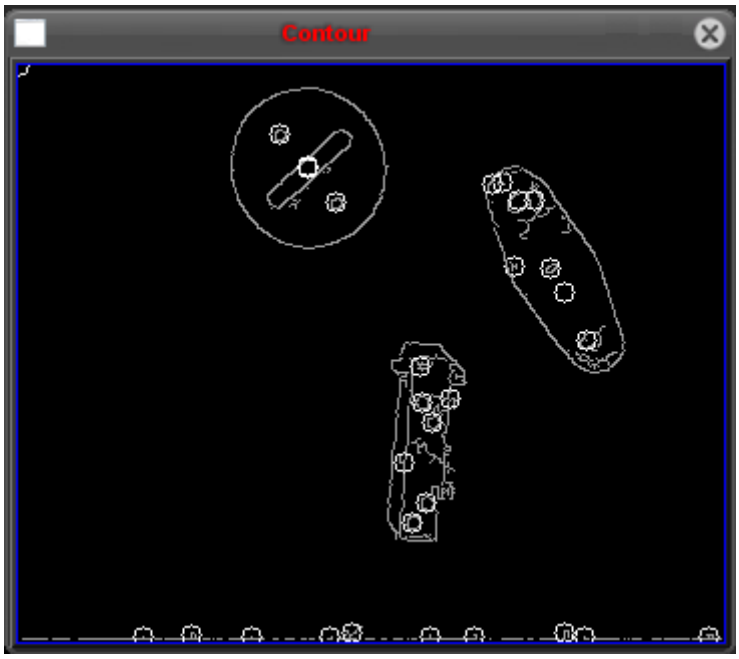

Рисунок 11 – Приклад виділення контурів зображень

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); }
```

Приклад виконання функції представлений на рис. 12.

Рисунок 12 – Застосування функції виділення кіл

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 поліліній задано 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 промальовувалися 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).
```

Вивід прямокутних об'єктів проводиться функцією:

```
drawSquares (out, findSquares4 (gray, mainStorage)),
```

де out – зображення, що містить прямокутні фігури.

Приклад виконання виділення прямокутних об'єктів показаний на рис. 13.

Таким чином розроблене програмне забезпечення виконує схоплення та обробку інформації, що надходить із камери маніпулятора (СТЗ робота), забезпечує виконання підрисовування кордонів, контурів та ідентифікацію простих об'єктів.

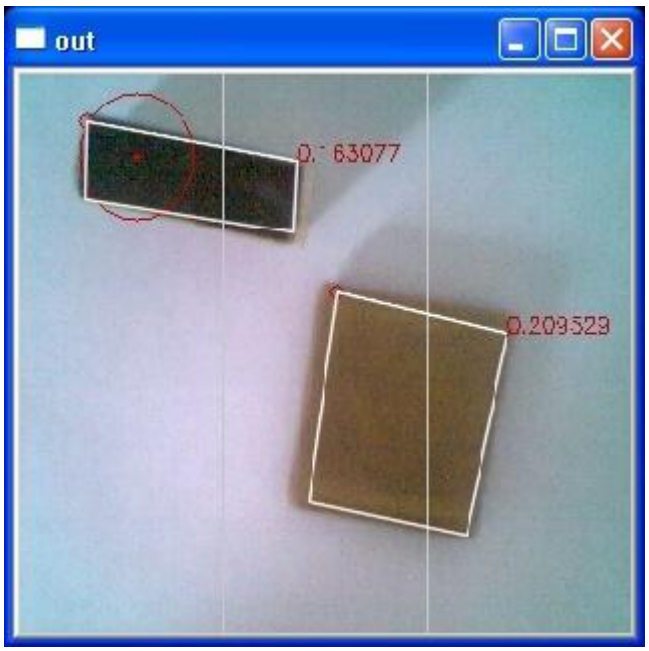

Рисунок 13 – Виконання виділення прямокутних об'єктів

У четвертій частині роботи досліджено та програмно реалізовано методи розпізнавання та ідентифікації простих об'єктів. Базовою бібліотекою розпізнавання обрана бібліотека комп'ютерного зору OpenCV. Отримані результати забезпечують виділення кордонів і контурів об'єктів, визначення центрів мас. Для простих об'єктів реалізовані функції потокового розпізнавання та ідентифікації.

ВИСНОВКИ

У даній магістерській кваліфікаційній роботі – основну увагу приділено розгляду методів розпізнавання та ідентифікації об'єктів в СТЗ роботів для забезпечення вимог безпеки при здійсненні повітряних перелетів та проходження митного контролю у аеропортах.

У першій частині МКР було проаналізовані методи обробки інформації в СТЗ роботів. Процес ідентифікації об'єктів, що знаходяться в робочій зоні робота, включає два етапи, такі як виділення характерних ознак об'єктів і розпізнавання об'єктів по знайденої сукупності характерних ознак.

Основними методами обробки інформації є сегментація (процес підрозділу сцени на складові частини або об'єкти), визначення порогового рівня, обласно-орієнтованої сегментації, дескрипторів кордонів та областей зображень, опису тривимірних сцен та структур, обробка візуальної інформації.

У другій частині проаналізовані методи ідентифікації об'єктів в робототехнічних системах і побудована теоретико-множинна модель розпізнавання та 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 найменшої вартості, який відповідає значущим контурам);
- кореляційний метод (обчислення взаємокореляційної функції між еталоном та зображенням);
- розпізнавання через зв'язку шаблонів (узгодження компонентів зображення як шаблон і визначення, які об'єкти присутні, вивчивши запропоновані зв'язку між знайденими шаблонами);
- штучні нейронні мережі (навчання мережі по різних зразків образів із зазначенням того, до якого класу вони відносяться).

У третій частині роботи була розглянута практична реалізація методів обробки інформації в робототехнічних системах. Існують 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. Основними бібліотеками, що працюють з методами розпізнавання та ідентифікації є бібліотека Integrated Performance Primitives (IPP), бібліотека AviCap, бібліотека комп'ютерного зору з відкритим кодом OpenCV. Розглянуто основні функції та їх реалізація в бібліотеці OpenCV.

У четвертій частині досліджено та програмно реалізовано методи розпізнавання та ідентифікації простих об'єктів. Базовою бібліотекою розпізнавання обрана бібліотека комп'ютерного зору OpenCV. Отримані результати забезпечують виділення кордонів та контурів об'єктів, визначення центрів мас. Для простих об'єктів реалізовані функції розпізнавання та ідентифікації.

До числа перспективних напрямків досліджень слід віднести подальший розвиток методів розпізнавання та ідентифікації, їх програмну реалізацію для митних та транспортних операційних систем реального часу. У комбінації з системою прийняття рішень такі розробки істотно прискорять реалізацію систем інтелектуального контролю на митному та транспортному кордонах держави.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Анисимов Б. В., Курганов В. Д. Распознавание и цифровая обработка изображений. Москва: Мир, 1991. 295с.
2. Фу К., Гонсалес К. Робототехника. Москва: Мир, 2001. 624 с.
3. Катыс Г. П. Техническое зрение роботов. Москва: Машиностроение, 1989. 176 с.
4. Хорн Б. К. Зрение роботов. Москва: Мир, 1989. 488 с.
5. Sung K. K., Poggio T. Learning Human Face Detection in Cluttered Scene. Lecture Notes in Computer Science. Computer Analysis of Images and Patterns. 1995. P. 432–439.
6. Rosenblum M., Yacoob Y., Davis L. Human Emotion Recognition from Motion Using a Radial Basis Function Network Architecture. IEEE Workshop on Motion of Non-Rigid and Articulated Objects, 1994. 257 p.
7. Пентланд А. С., Чаудхари Т. Распознавание лиц для интеллектуальных сред. Открытые Системы. № 3. 2009. С. 86–95.
8. Техническое зрение роботов. Пер с англ. / под. ред. А. Пью. Москва: Машиностроение, 1990. 320 с.
9. Глазунов А. Компьютерное распознавание человеческих лиц. Открытые Системы. № 3. 2014. С. 107–114.
10. Абламейко С. В., Лагуновский Д. М. Обработка изображений: Технология, методы, применение. Минск: Техника, 2009. 302 с.
11. Соломатин Н. М. Логические элементы ЭВМ. Москва: Высшая школа. 1990. 160 с.
12. Форсайт П. Компьютерное зрение. Современный подход. Москва: Издательский дом «Вильямс». 2004. 928 с.
13. Брагин В., Войлов Ю. Системы очувствления и адаптивные промышленные роботы. Москва: Машиностроение, 1985. 256 с.
14. Smith S. M., Brady J. M. SUSAN – a new approach to Low Level Image Processing. Technical Report. 1995. 57 p.

15. Введение в компьютерное зрение. URL: http://graphics.cs.msu.ru/ru/library/cv/cv_intro.html (дата звернення: 28.10.2019).
16. Цифровое компьютерное зрение. URL: http://graphics.cs.msu.ru/courses/cg02b/ library/ (дата звернення: 28.10.2019).
17. Open Source Computer Vision Library. URL: http://www.intel.com/technology/computing/opencv/index.htm (дата звернення: 28.10.2019).
18. Компьютерное зрение. URL: http://www.roboforum.ru/viewforum.php?f=51&sid=29ea9ba7faf446dda7cbdbcbd5a45d42 (дата звернення: 30.10.2019).