

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Кібербезпеки _____
(повна назва)

Кафедра _____ Інформаційно-мережної інженерії _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти _____ перший (бакалаврський) _____
Визначення центру ваги плоских фігур за їх зображеннями в системах
комп'ютерного зору _____
(тема)

Виконав:
здобувач _____ 4 _____ року навчання,
групи _____ ТРІМІ-22-2 _____
_____ Тетенів Данііл _____
(власне ім'я, прізвище)

Спеціальність _____ 172 Телекомунікації та _____
_____ радіотехніка _____
(код і повна назва спеціальності)
Тип програми _____ освітньо-професійна _____
Освітня програма _____ Інформаційно-мережна _____
_____ інженерія _____
(повна назва освітньої програми)

Керівник _____ ст. викл. Олексій Федоров _____
(посада, власне ім'я, прізвище)

Допускається до захисту

В. о. зав. кафедри _____ ІМІ _____

(підпис)

_____ Дарія Чеботарьова _____
(власне ім'я, прізвище)

2026 р.

Не містить відомостей, заборонених до відкритого публікування

Студент _____ / Тетенов Д.Д. /
(підпис) (прізвище та ініціали)

Керівник _____ / Федоров О.В /
(підпис) (прізвище та ініціали)

Харківський національний університет радіоелектроніки

Факультет _____ Кібербезпеки _____
Кафедра _____ Інформаційно мережної інженерії _____
Рівень вищої освіти _____ перший (бакалаврський) _____
Спеціальність _____ 172 Телекомунікації та Радіотехніка _____
(код і повна назва)
Тип програми _____ освітньо-професійна _____
Освітня програма _____ Інформаційно-мережна інженерія _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

« _____ » _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Тетенову Даніілу Дмитровичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Визначення центру ваги плоских фігур за їх зображеннями в системах комп'ютерного зору

затверджена наказом університету від 02 червня 2026 р. № 466Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 18 липня 2026 р.

3. Вихідні дані до роботи Дослідити методи визначення центру ваги плоских фігур за їх зображеннями в системах комп'ютерного зору. З'ясувати фактори, які впливають на точність визначення контурів на зображенні. Порівняти отримані результати з тими, що випливають з теоретичних міркувань. Під час проведення експериментальних досліджень використовувати функції бібліотеки OpenCV.

4. Перелік питань, що потрібно опрацювати в роботі _____
Вступ

1 Базові концепції комп'ютерного зору

2 Цифрові зображення. Способи та формати кодування цифрових зображень

3 Методи виділення границь і контурів на зображенні

4 Центр мас плоских фігур: теоретичні основи та практичні методи обчислення

Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) _____

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1.	Ознайомлення із завданням. Уточнення ТЗ.	03.06.2026	вик.
2.	Підбір літератури за темою роботи	03.06 – 03.07.2026	вик.
3.	Виконання розділу 1	10.06 – 25.06.2026	вик.
4.	Виконання розділу 2	20.06 – 03.07.2026	вик.
5.	Виконання розділу 3	16.06 – 05.07.2026	вик.
6.	Виконання розділу 4	01.07 – 10.07.2026	вик.
7.	Оформлення презентаційного матеріалу, підготовка до захисту в ЕК	05.07 – 17.07.2026	вик.

Дата видачі завдання 03 червня 2026р.

Здобувач _____
 (підпис)

Керівник роботи _____ ст. викл. Олексій Федоров
 (підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 60 с., 14 рис., 1 таб., 9 джерел, 1 додаток.

Об'єктом дослідження є методи визначення центру ваги плоских фігур за їх зображеннями.

Предметом дослідження є методи виділення контурів на зображенні, методи обчислення центру ваги.

Мета кваліфікаційної роботи – створення програмної моделі системи визначення центру ваги плоских фігур за їх зображеннями.

В кваліфікаційній роботі детально розглянуто питання визначення центра ваги плоских фігур за їх графічними зображеннями. На мові програмування Python було створено скрипт, який засобами бібліотеки OpenCV розв'язував задачу виділення контурів на зображенні із подальшим обчисленням центру ваги цієї фігури.

КОНТУР, ГРАНИЦЯ, ЦЕНТР ВАГ, ОБЧИСЛЕННЯ, СИСТЕМА, ЗІР, РОБОТИЗАЦІЯ, OPENCV

ABSTRACT

Explanatory note: 60 p., 14 fig., 1 tabs, 9 sources, 1 app.

Object of study is methods for determining the center of gravity of flat figures based on their images.

Subject of study is methods for detecting contours in an image and methods for calculating the center of gravity.

Purpose of the qualification work is to develop a software model for determining the center of gravity of flat shapes based on their images.

This thesis examines in detail the problem of determining the center of gravity of flat shapes based on their graphical representations. A script was written in the Python programming language that used the OpenCV library to solve the problem of detecting contours in an image and subsequently calculating the center of gravity of the shape.

CONTOUR, EDGE, CENTER OF GRAVITY, CALCULATION,
SYSTEM, VISION, ROBOTICS, OPENCV

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	8
ВСТУП.....	9
1 БАЗОВІ КОНЦЕПЦІЇ КОМП'ЮТЕРНОГО ЗОРУ	10
1.1 Комп'ютерний зір	10
1.2 Спектр завдань, доступних системам комп'ютерного зору	11
1.3 Взаємодія із суміжними науково-технічними дисциплінами.....	13
2 ЦИФРОВІ ЗОБРАЖЕННЯ. СПОСОБИ ТА ФОРМАТИ КОДИРУВАННЯ ЦИФРОВИХ ЗОБРАЖЕНЬ	16
2.1 Характеристики зорової системи людини	16
2.2 Цифрові графічні формати.....	19
3 МЕТОДИ ВИЗНАЧЕННЯ ГРАНИЦЬ І КОНТУРІВ НА ЗОБРАЖЕННІ.....	24
3.1 Детальне визначення контуру.....	29
3.2 Алгоритм градієнтного виділення.....	35
3.3 Градієнтний оператор Робертса.....	36
3.4 Градієнтний оператор Превітта	37
3.5 Градієнтний оператор Собеля.....	38
4 ЦЕНТР МАС ПЛОСКИХ ФІГУР: ТЕОРЕТИЧНІ ОСНОВИ ТА ПРАКТИЧНІ МЕТОДИ ОБЧИСЛЕННЯ	39
4.1 Загальна характеристика проблеми визначення цунтра мас	39
4.2 Центр мас дискретної системи точок	39
4.3 Центр мас суцільної плоскої фігури.....	40
4.3.1. Однорідні плоскі фігури.....	41
4.4 Статичні моменти плоских фігур	41
4.5 Практичні методи визначення центру мас	42
4.5.1 Метод симетрії.....	42

4.5.2	Метод розбиття на прості фігури.....	43
4.5.3	Метод від'ємних площ (метод доповнення).....	43
4.5.4	Теореми Паппа-Гюльдена	43
4.5.5	Координати центрів мас основних геометричних фігур	44
4.6	Інтегральний метод: обчислення для фігур, обмежених кривими	45
4.7	Практичне застосування	46
	ВИСНОВКИ	50
	ПЕРЕЛІК ПОСИЛАНЬ	51
	ДОДАТОК А СЛАЙДИ ПРЕЗЕНТАЦІЇ	52

ПЕРЕЛІК СКОРОЧЕНЬ

ЗСЛ — зорова система людини.

ДКП — дискретне косинусне перетворення.

ЦОС — цифрова обробка сигналів.

JPEG - Joint Photographer Expert Group.

PNG – Portable Network Graphics.

ВСТУП

Комп'ютерний зір — це відносно нова галузь наукових досліджень. За останні 25 років ця галузь зазнала значного розвитку. Раніше комп'ютерний зір використовувався лише для чітко обмежених і цілеспрямованих завдань. Однак зараз він має безліч сфер застосування одною з яких є системи зору роботів.

Роботизований зір - це застосування комп'ютерного зору в робототехніці. Такий зір дає машинам змогу сприймати навколишнє середовище, інтерпретувати його та взаємодіяти з визначеним середовищем. Завдяки використанню цифрових камер, датчиків та алгоритмів на основі штучного інтелекту роботизований зір допомагає роботам розпізнавати візуальні об'єкти, орієнтуватися в просторі та приймати обґрунтовані рішення.

Дана кваліфікаційна робота присвячена розробці однієї з таких систем роботизованого зору, а саме – системі визначення центру ваги плоских фігур за їх зображеннями. Такі системи дозволяють роботам взаємодіяти з навколишнім середовищем

1 БАЗОВІ КОНЦЕПЦІЇ КОМП'ЮТЕРНОГО ЗОРУ

1.1 Комп'ютерний зір

Зоровий апарат людини по праву вважається найбільш досконалим та інформативним із усіх доступних нам органів чуття. Він безперервно забезпечує нас колосальним потоком даних про навколишню реальність та надає здатність безперешкодно контактувати з фізичним світом, причому здійснювати це на відстані, без необхідності прямого тактильного контакту. Саме завдяки візуальному сприйняттю ми здатні оцінювати просторове становище фізичних тіл, здійснювати їхню ідентифікацію, встановлювати зв'язки між різними об'єктами. Втративши цю здатність, людина відчуває високий рівень дискомфорту і дезорієнтації. У світлі цього зовсім не дивно, що в міру того, як електронно-обчислювальні машини ставали все більш продуктивними та доступними, дослідники почали робити активні кроки щодо створення алгоритмів, здатних наділити комп'ютери повноцінним «зором» [1].

Одночасно з цим необхідно розуміти, що зір є найбільш комплексним та заплутаним із усіх сенсорних механізмів. Існуюча база знань щодо того, яким саме чином функціонує біологічний зоровий апарат, досі залишається дуже фрагментарною. Наші знання багато в чому обмежуються лише первинними фазами обробки даних, які відбуваються відразу після того, як світловий сигнал потрапляє на рецептори сітківки. Одне можна стверджувати, напевно: біологічна система зору неймовірно складна. Саме тому не варто дивуватися з того факту, що безліч ранніх спроб створити штучний «комп'ютерний зір» зазнавали краху. І все ж таки дослідникам вдалося досягти вражаючих результатів. Сьогодні ми повсюдно стикаємося зі складними зоровими алгоритмами, інтегрованими в комп'ютерні архітектури, які чудово справляються зі своїми завданнями

навіть у умовах середовища, що постійно змінюються. Такі комплекси комп'ютерного зору базуються як у класичних математичних моделях, і на використанні сучасних штучних нейронних мереж. Нейромережі, своєю чергою, дедалі активніше впроваджуються у область інтелектуального дата-майнінгу, окремим випадком якого за своєю суттю є комп'ютерний зір [1].

1.2 Спектр завдань, доступних системам комп'ютерного зору

Будь-яка система комп'ютерного зору функціонує за принципом аналізу графічних файлів, що надходять, з подальшим формуванням детального опису відображеної на них сцени. Даний сформований опис має включати специфічні властивості знайдених об'єктів, які потрібні для вирішення конкретної прикладної задачі. Таким чином, ми розглядаємо блок машинного зору як інтегрований компонент більш масштабного комплексу, що безперервно взаємодіє з навколишнім середовищем. Систему зору можна інтерпретувати як інформаційний сенсор у замкнутому контурі зворотного зв'язку, що відповідає безпосередньо за сприйняття, тоді як інші суміжні модулі комплексу несуть відповідальність за вироблення рішень та їхнє фізичне виконання. Функціонування системи комп'ютерного зору пояснено на рис. 1.1 [1].

Як початкові дані на вхід системи машинного зору подається одне або відразу кілька візуальних зображень. На виході ж формується інформаційний опис, який обов'язково має відповідати двом ключовим критеріям [1]:

- 1) він повинен мати пряму смислову та логічну прив'язку до об'єктів, присутнім на аналізованому зображенні;
- 2) він має включати в собі абсолютно всю інформацію, яка критично важлива для успішного виконання поставленої мети.

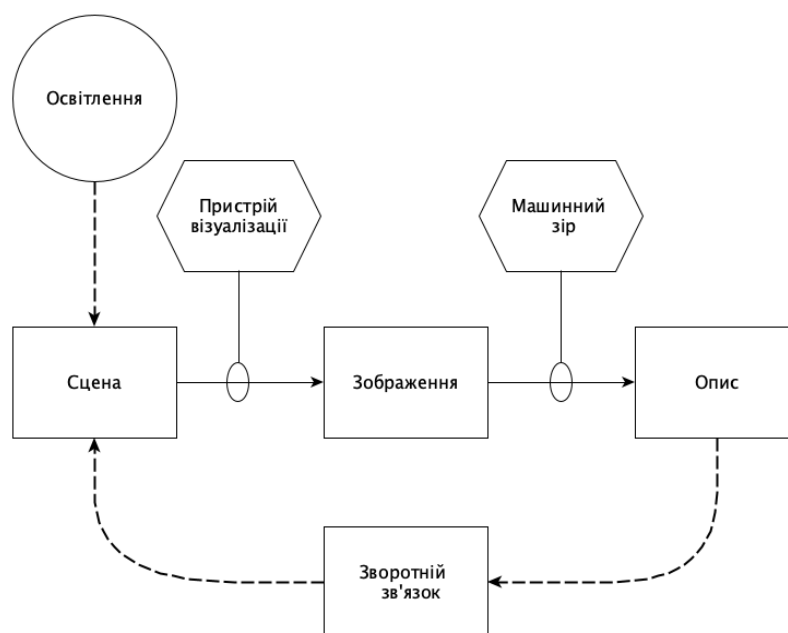


Рисунок 1.1 – Принцип функціонування системи комп'ютерного зору

Дотримання першого критерію гарантує, що підсумковий опис безпосередньо залежатиме від візуальних вхідних даних, а виконання другого критерію підтверджує практичну значущість отриманих відомостей [1].

Сама собою концепція «опису об'єкта» може трактуватися дуже неоднозначно, оскільки існує безліч можливих рівнів деталізації і ракурсів розгляду. Скласти абсолютно вичерпний та всеосяжний опис будь-якого об'єкта фізично неможливо. Однак, завдяки чіткій концентрації уваги виключно на тих аспектах, які потрібні для конкретного завдання, ми можемо обминути потенційні логічні пастки та протиріччя. Іншими словами, нам не потрібний абстрактний і нескінченно докладний опис, нам потрібен лише певний набір даних, який дозволить системі виконати коректну цільову дію.

1.3 Взаємодія із суміжними науково-технічними дисциплінами

Галузь комп'ютерного зору має надзвичайно тісні точки дотику з трьома суміжними галузями: безпосередньою цифровою обробкою зображень, методологіями класифікації образів та алгоритмами аналізу сцен (див. рис. 1.2) [1].

Дисципліна цифрової обробки зображень (Image Processing), як правило, фокусується на генерації нових, покращених картинок на основі вже існуючих оригіналів. Переважна більшість алгоритмів у цій сфері базується на фундаментальній теорії лінійних систем. В процесі генерації нового графічного файлу можна ефективно видалити цифровий шум, компенсувати ефект розмиття або акцентувати увагу на контурах об'єктів. Підсумковий результат зазвичай є модифікованим зображенням, оптимізованим для його подальшої комфортної інтерпретації або людиною, або обчислювальною машиною [1].

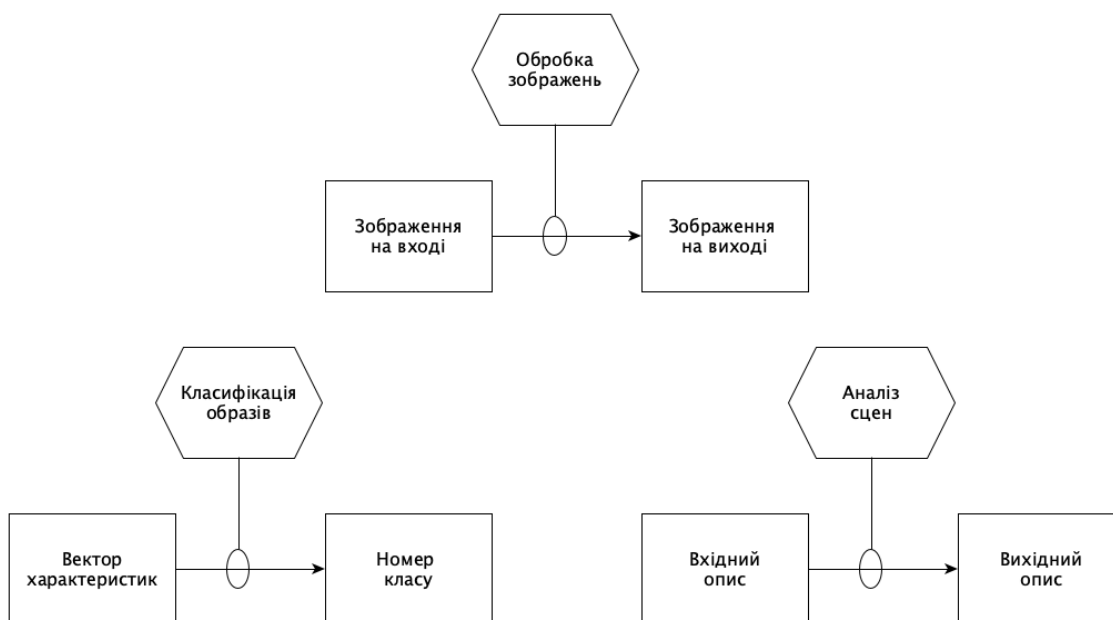


Рисунок 1.2 – Зв'язки машинного зору з іншими суміжними галузями інформатики

Головна мета розпізнавання образів (Pattern Classification) зводиться до співвіднесення аналізованого образу до певного заздалегідь заданого класу. При цьому сам образ найчастіше представляється у форматі математичного вектора або набору чисел, які описують виміряні фізичні характеристики об'єкта, що вивчається (наприклад, його ширину, висоту, площу). Незважаючи на те, що на вхід програмного класифікатора зазвичай подається не сама картинка як така, а набір цих характеристик, методики класифікації найчастіше виявляються вкрай затребуваними при розборі зображень, отриманих сенсорами зорової системи. Процедура розпізнавання об'єкта фактично означає його успішне зарахування до однієї з відомих системі категорій. При цьому варто особливо наголосити на тому факті, що саме розпізнавання — це лише мала частина глобального спектра завдань, покладених на систему зору. Дослідники, які присвятили себе проблемам розпізнавання, спроектували безліч різнопланових методів вилучення значних ознак (властивостей) безпосередньо з графічних файлів [1].

У рамках напряму аналізу сцен (Scene Analysis) детально вивчається проблема трансформації найпростіших низькорівневих описів, отриманих прямо із зображення, у більш глибокі та комплексні моделі. Ці ускладнені моделі видаються у форматі, максимально адаптованому для вирішення конкретного прикладного завдання. Класичним прикладом такого процесу може слугувати семантична інтерпретація векторних контурних малюнків. У цьому випадку початковий опис графічної сцени, що складається з групи багатогранників, виражається як простий набір прямих ліній (відрізків). Однак перед тим як приступити до використання цієї інформації, алгоритму необхідно обчислити, які ділянки, обмежені даними лініями, належать тому самому фізичному тілу. З іншого боку, вкрай бажано з'ясувати просторове розташування цих тіл щодо одне одного. Саме шляхом подібних маніпуляцій із примітивного набору ліній можна сконструювати комплексний символічний опис усієї сцени [1].

Переважна більшість методологій графічного аналізу орієнтована працювати не стільки з картинкою у її глобальному масштабі, скільки з конкретними ізольованими областями, що належать окремим об'єктам. У зв'язку з тим, що в об'єктив камери або сенсора зазвичай потрапляє велике скупчення різних поверхонь навколишнього світу, перед запуском аналітичних алгоритмів зображення обов'язково потрібно розділити на специфічні області, кожна з яких відповідатиме конкретному фізичному об'єкту. Цей процес називається сегментацією зображення. Процес сегментації є невід'ємною складовою роботизованого зору, наприклад маніпулятора типу рука який за результатами виділення контурів визначатиме центри ваги фігур з метою краще за них вхопитись. Саме таким задачам і буде присвячена дана кваліфікаційна робота [1].

2 ЦИФРОВІ ЗОБРАЖЕННЯ. СПОСОБИ ТА ФОРМАТИ КОДИРУВАННЯ ЦИФРОВИХ ЗОБРАЖЕНЬ

2.1 Характеристики зорової системи людини

Специфічні особливості функціонування зорової системи людини (ЗСЛ) прийнято класифікувати на дві великі категорії: низькорівневі (які можна назвати суто «фізіологічними») і високорівневі (відповідають за «психофізіологічний» аспект сприйняття). Аж до середини 1990-х років наукова спільнота акцентувала свою увагу переважно на низькорівневих механізмах роботи зору. Однак, останнім часом ситуація змінилась. Наприклад, намітився явний тренд на розробку стеганографічних алгоритмів, які повною мірою враховують складні високорівневі властивості ЗСЛ [2].

Фахівці виділяють три фундаментальні низькорівневі параметри, які безпосередньо впливають на те, наскільки людське око здатне помітити сторонній шум або спотворення на картинці [2]:

- 1) сприйнятливість зорового апарату до флуктуацій яскравості та загального розмаїття картинки;
- 2) частотна сприйнятливість (залежність сприйняття просторових частот);
- 3) феномен візуального маскування.

Частотна чутливість нашого зору виявляється у тому феномені, що людське око значно сприйнятливіше до низькочастотних (НЧ) візуальних шумів, ніж до високочастотних (ВЧ) завад. Цей факт безпосередньо зумовлений нелінійним характером амплітудно-частотної реакції нашої зорової системи.

Рецепторні елементи ЗСЛ розкладають вхідний світловий сигнал на незалежні частотні компоненти, і кожна з цих компонентів стимулює

нейронні шляхи ока за допомогою певних підканалів. Ці складові, що виділяються зоровим апаратом, володіють різними просторовими і частотними характеристиками, а також відрізняються своєю просторовою орієнтацією (вони можуть бути горизонтальними, вертикальними або розташовуватися по діагоналі).

Якщо на зоровий апарат одночасно впливають дві візуальні складові, що мають схожі параметри, відбувається активація одних і тих же нейронних підканалів. Саме цей біологічний механізм породжує ефект маскування. Суть маскування полягає в тому, що поріг виявлення одного візуального сигналу істотно підвищується (сигнал стає менш помітним), якщо він накладається поверх іншого, сильнішого сигналу з аналогічними параметрами. Як наслідок, будь-який впроваджений адитивний шум кидатиметься в очі на низькочастотних (монотонних, однорідних за кольором) областях картинки набагато сильніше, ніж на високочастотних (сильно деталізованих) ділянках. Тобто на високочастотних ділянках завада успішно «маскується» великою кількістю дрібних деталей. Цей ефект досягає своєї максимальної сили в тому випадку, коли обидва візуальні патерни мають ідентичну просторову орієнтацію та локалізацію.

Частотна чутливість ока перебуває у нерозривному зв'язку зі сприйняттям яскравості. У науковому середовищі виведено математичне співвідношення, що дозволяє розрахувати поріг візуального маскування, базуючись вже відомої чутливості до перепадів яскравості. Це, у свою чергу, дає можливість розробити таку метрику оцінки спотворень зображення, яка адекватно враховуватиме біологічні обмеження ЗСЛ. Подібні математичні формули та моделі чудово опрацьовані для алгоритмів квантування коефіцієнтів дискретного косинусного перетворення, - саме тієї математичної основи, на якій базується найпопулярніший стандарт стиснення JPEG [2].

Візуальне маскування в просторовій площині можна інтерпретувати, створюючи стохастичні (імовірнісні) моделі картинки. У рамках такого

підходу цифрове зображення описується як випадкове марковське поле, де розподіл ймовірностей піксельних значень підпорядковується, наприклад, узагальненому розподілу Гауса.

Що ж стосується високорівневих властивостей ЗСЛ, то їх принципова відмінність від низькорівневих криється у «вторинності» їх прояву: після того, як людський мозок проводить первинну обробку інформації, що надійшла від рецепторів, він генерує сигнали на адаптацію і своєрідне «підстроювання» всього зорового апарату під конкретний [2].

Серед ключових високорівневих характеристик зору слід виділити такі:

1) чутливість до контрасту – ділянки картинки з високим перепадом яскравості та вираженим контрастом рефлекторно привертають до себе максимальну увагу спостерігача;

2) чутливість до розміру – масивні графічні об'єкти виглядають набагато «помітнішими» і натомість дрібних деталей. У цьому спостерігається ефект візуального насичення: починаючи з певного габариту, подальше збільшення об'єкта не призводить до пропорційного зростання уваги щодо нього;

3) чутливість до форми – об'єкти з видовженою, тонкою геометрією притягують людський погляд сильніше, ніж об'єкти з плавними закругленнями чи однорідною текстурою;

4) чутливість до кольору – певні відтінки колірному діапазону (наприклад, насичений червоний) мають вищої психологічної «помітністю», ніж інші цвета. Даний візуальний ефект багаторазово посилюється в тих ситуаціях, коли фонове заповнення різко контрастує з кольором геометричних фігур, що розташовані на ньому;

5) чутливість до позиціонування – людська психологія влаштована в такий спосіб, що з першому погляді картинку ми рефлекторно фокусуємося її центральної частини. Крім того, об'єкти, що знаходяться на

яскраво вираженому передньому плані, вивчаються мозком набагато скрупульозніше, ніж елементи фонового задника;

б) чутливість до зовнішніх стимулів – траєкторія руху очей людини багато в чому детермінована довкіллям, а також тими інструкціями або фоновою інформацією, яку спостерігач отримав до або безпосередньо під час вивчення зображення.

2.2 Цифрові графічні формати

Розглянемо детальніше ключові особливості найпопулярніших форматів [3].

RAW -специфічний формат файлів, що включає «сирі», абсолютно не модифіковані дані, які зчитуються безпосередньо з матриці цифрового фотоапарата. На відміну від файлів JPEG, дані в контейнері RAW не піддаються внутрішньокамерній компресії та програмній обробці процесором пристрою, завдяки чому зберігають первозданну інформацію про умови зйомки. Файли формату RAW можуть бути піддані алгоритмам стиснення, які гарантують повну відсутність втрат початкової якості.

Переваги формату RAW незаперечні: на противагу стиснутому та відкоригованому автоматикою камери JPG, формат RAW надає фотографам та дизайнерам воістину безмежні перспективи для глибокої постобробки графічного матеріалу, гарантуючи при цьому збереження максимально можливої якості пікселів.

TIFF (Tagged Image File Format) – формат, створений для зберігання растрових графічних файлів. Формат TIFF завоював величезну популярність завдяки своїй здатності зберігати картинки із феноменально високою глибиною кольору. Його повсюдно застосовують у процесах сканування паперових носіїв, при передачі факсимільних повідомлень, алгоритмах оптичного розпізнавання символів (OCR), а також у

професійній поліграфії. Формат підтримується переважною більшістю графічних редакторів [3].

Формат TIFF дуже популярний для архівації цифрових знімків. Його структура дозволяє записувати фотоматеріали у величезному спектрі колірних просторів (таких як RGB, CMYK, YCbCr, CIE Lab та багатьох інших), підтримуючи при цьому колосальну глибину передачі кольору (від 8 до 64 біт на канал).

Кардинальною відмінністю TIFF від алгоритму JPG є те, що зображення в контейнері TIFF абсолютно не схильне до деградації якості при багаторазових циклах перезапису та збереження файлу. Однак розплатою за цю безпеку інформації є те, що підсумкові розміри файлів TIFF багаторазово перевищують вагу аналогічних зображень у форматі JPG.

BMP (Bitmap Picture) - історично один із найстаріших і базових растрових форматів. Файли з розширенням BMP без проблем відкриваються і розпізнаються буквально будь-якою програмою для перегляду графіки, а його підтримка на фундаментальному рівні вбудована в ядро операційних систем сімейства Windows.

GIF (Graphics Interchange Format) - неймовірно затребуваний формат обміну графікою. Його ключова особливість полягає у здатності зберігати попередньо стислі масиви графічних даних без спотворення початкової якості, проте палітра при цьому жорстко обмежена максимумом 256 відтінків. Цей кроссплатформенний формат, незалежний від апаратних архітектур. Безперечною перевагою GIF є підтримка динамічних анімованих зображень. Структурно така анімація komponується з низки статичних графічних кадрів (кадрів). У файл також зашивається інформація про часові інтервали (затримки), що визначають, як довго конкретний кадр буде демонструватися користувачеві [3].

PNG (Portable Network Graphics) — сучасний формат, архітектура якого створювалася з прицілом на повну заміну морально застарілого та

технічно обмеженого GIF, а також як легковажнішу альтернативу громіздкому формату TIFF. Насамперед формат PNG позиціонується як ідеальний вибір для інтеграції у веб-інфраструктуру та для проведення растрового редагування графіки. Специфікації PNG включають повноцінну підтримку трьох базових різновидів растрових сіток [3]:

- 1) монохромні напівтонові зображення (з підвищеною градацією яскравості до 16 біт);
- 2) кольорові зображення на базі індексованої палітри (з обмеженням палітри до 8 біт та підтримкою 24-бітного повноцвіту);
- 3) зображення з технологією TrueColor, або повнокольорові (підтримують глибину до 48 біт).

Будь-яка графічна інформація у файлах PNG зберігається у запакованому вигляді. Ключовою перевагою алгоритму є те, що ця компресія відбувається абсолютно без втрати графічної інформації, на відміну від популярного формату JPEG, який відкидає частину даних задля економії місця.

Якщо порівнювати PNG зі старим стандартом GIF, перший демонструє цілу низку незаперечних переваг:

- 1) підтримка фактично нескінченного числа кольірних варіацій на одному полотні (у той час як GIF технічно замкнений у жорстких рамках 8-бітового кольору);
- 2) наявність інтегрованого опціонального альфа-каналу, що забезпечує підтримку плавних переходів напівпрозорості;
- 3) апаратна підтримка алгоритмів гамма-корекції, що дозволяють досягти однакового відображення яскравості на різних моніторах;
- 4) просунута система двовимірної черезрядкової розгортки (interlacing), що дозволяє браузерам поступово покращувати якість картинки, що завантажується;
- 5) відкрита архітектура, що допускає впровадження у файл додаткових користувацьких метаданих.

Важливо відзначити, що PNG здатний забезпечити значно сильніший коефіцієнт компресії для тих зображень, які рясніють складними кольоровими градієнтами, порівняно з GIF. Однак цей вииграш у розмірі зазвичай коливається в районі 5-25%, що не дозволяє PNG повністю витіснити конкурента з ринку.

Формат PNG зарекомендував себе як чудовий інструмент потреб редагування графіки. Він ідеально підходить на роль формату для збереження проміжних робочих копій, оскільки алгоритми розпакування та подальшого перезапису файлів протікають без втрати якості пікселів. Крім того, в пік формату TIFF, стандарти PNG неймовірно суворі і не дають розробникам програмного забезпечення свободи вибору в тому, які саме функції формату вони хочуть впроваджувати у свої програми, а які ігнорувати. Завдяки цій жорсткій уніфікації абсолютно будь-який файл, збережений у форматі PNG, гарантовано і без помилок відкриється в будь-якому іншому програмному забезпеченні, що заявило про підтримку стандарту PNG.

Зведена інформація про перераховані вище формати даних міститься в таблиці 2.1.

Таблиця 2.1 - Формати растрової та векторної графіки

Растрові формати	Опис растрових форматів	Векторні формати	Опис векторних форматів
.bmp	Класичний стандарт Windows. Не має на увазі алгоритмів компресії.	.cdr	Специфічний формат CorelDRAW.

Продовження таблиці 2.1

Растрові формати	Опис растрових форматів	Векторні формати	Опис векторних форматів
.jpg, .jpeg	Оптимізовано для зберігання фотографій та багатобарвних зображень (зі стисненням).	.wmf, .emf	Векторні стандарти середовища Windows.
.gif	Підтримує базову анімацію та ефект прозорості підкладки.	.ai	Пропрієтарний формат Adobe Illustrator.

3 МЕТОДИ ВИЗНАЧЕННЯ ГРАНИЦЬ І КОНТУРІВ НА ЗОБРАЖЕННІ

Детектування та виділення границь об'єктів виконує критично важливу функцію у процесах всебічного аналізу візуальних даних та систем розпізнавання образів, що лежать у фундаменті глобальної проблематики комп'ютерного зору. З фізичної точки зору границі являють собою зони різких стрибків чи перепадів параметрів піксельної яскравості у межах напівтонового графічного поля, що робить їх одними з найважливіших інформаційних дескрипторів будь-якого зображення. Ці локації різких перепадів безпомилково маркують геометричні контури та форми предметів, присутніх у кадрі. Перехід від повнокольорового полотна до подання сцени виключно у вигляді контурів фізичних об'єктів дає можливість колосально знизити обсяг обчислювальних даних, принагідно зберігаючи левову частку значущої інформації про просторову структуру та форми досліджуваної сцени. Алгоритми пошуку границь по праву вважаються найбільш затребуваними та масовими технологічними рішеннями у сфері програмної обробки растрових файлів [5].

У ході роботи алгоритму локалізації границь цільового об'єкта здійснюється прорахунок двовимірного просторового градієнта яскравості по всьому полю зображення, після чого автоматика виявляє конкретні кластери пікселів, які максимально відповідають характеристикам контурів. У процесі цієї операції проводиться точкова оцінка модуля вектора градієнта для абсолютно кожного пікселя, що становить напівтонове зображення. У термінології теорії графічної обробки та комп'ютерного (машинного) зору термін «виділення границь» тісно стикається із завданнями локалізації та сегментації об'єктів. У його основі лежить пул алгоритмів, які сканують цифровий масив і маркують ті точки, у яких фіксується аномально різке коливання рівня яскравості чи інші форми візуальної неоднорідності сигналу. Фінальним результатом роботи алгоритмів контурного виділення стає генерація взаємопов'язаної мережі

векторних чи растрових кривих ліній. Ці лінії описують фізичні границі предметів, окреслюють грані поверхонь, різні текстурні відбитки, а також репрезентують зони зламу або зміни просторового положення поверхонь, що візуалізуються. Будь-який реальний об'єкт у кадрі скомпонований з величезної кількості сегментів, що мають різну колірну або яскраву заливку. У свою чергу, перепад (стрибок) яскравості є неперервним і зв'язним масивом піксельних точок, які розташовуються строго на стику між двома різними за інтенсивністю областями [5].

Поняття границі в цифровому масиві сприймається як математично значиме локальне відхилення лише у рівні інтенсивності піксельної заливки. Це відхилення зазвичай асоціюється із різкою просторовою неоднорідністю візуального сигналу. Як правило, математичний опис подібної зміни інтенсивності проводиться через обчислення першої похідної, яка характеризує безпосередньо скалярну величину локального градієнта [4, 5]. Якщо ми хочемо визначити передбачуваний напрямок, яким пролягає границя об'єкта на площині растру, ми повинні побудувати лінію, перпендикулярну до розрахованого вектора градієнта. Самі ж аномалії та стрибки піксельної інтенсивності можуть набувати абсолютно різноманітних фізичних форм. Наприклад, існують границі типу "математичний крок" (сходінка), в зоні яких значення інтенсивності здійснює миттєвий і різкий стрибок від однієї константи до іншої. Існують також границі типу "лінія", за яких інтенсивність різко змінює свій рівень, але практично миттєво повертається до початкового фонових значення. Важливо, що за умов аналізу реальних, знятих камерою фотографій, ідеальні границі типу кроку чи чіткої лінії зустрічаються дуже рідко. Причина цього полягає в тому, що в апаратному сигналі завжди домінують низькочастотні спектральні компоненти, що є наслідком неминучого оптичного та апаратного згладжування світла на матриці відеодатчика. Через цих оптичних спотворень границя у вигляді різкого кроку деградує більш плавну границю типу "скат", а вузька границя типу "лінія"

перетворюється на границю типу "дах". У таких умовах перепади світлової інтенсивності втрачають свою миттєвість та розтягуються на певний просторовий піксельний інтервал [4, 5].

Стандартна процедура контурного виділення традиційно розбивається на 3 послідовні обчислювальні етапи [4, 5]:

- 1) початкова фільтрація,
- 2) фаза покращення та
- 3) безпосереднє виділення контурів.

Дамо характеристику згаданим етапам.

Фаза фільтрації Найпершим і абсолютно необхідним кроком є придушення паразитного шуму, яким рясніє оригінальне зображення. Під терміном "шум" прийнято розуміти випадкові, хаотичні осциляції значень інтенсивності піксельної сітці. У цифровій фотографії найчастіше зустрічаються такі шумові патерни: імпульсний шум (звані перешкоди типу «сіль і перець»), і навіть класичний гаусовський (білий) шум. У більшості випадків програмні алгоритми шумозаглушення мають вузьку спеціалізацію та орієнтовані на нейтралізацію строго певної категорії перешкод. На даний час в арсеналі програмістів ще не з'явилося абсолютно універсальних математичних фільтрів, здатних зі 100% ефективністю знаходити і ліквідувати всі без винятку види шумів одночасно [4, 5].

Фаза покращення. Для того, щоб максимально спростити процес локалізації шуканого краю, системі вкрай важливо чітко зафіксувати всі коливання інтенсивності, що відбуваються в найближчих піксельних околицях. Процедура покращення (або підкреслення) спрямована на те, щоб штучно посилити контрастність тих пікселів, які демонструють значне відхилення значення своєї інтенсивності щодо сусідів. Як правило, саме в ході виконання цього етапу процесором реалізується ресурсомісткий прорахунок вектора градієнта для необхідних точок зображення.

Фаза виділення. На практиці безліч пікселів аналізованої картини можуть мати розраховані значення градієнта, які значно відрізняються від

нуля, проте далеко не всі ці пікселі в реальності належать контуру або границі предмета. У зв'язку з цим код інтегруються додаткові евристичні і математичні методи (наприклад, порогова обробка), метою яких є фінальна відбраковування і точне визначення того, які саме градієнтні аномалії формують справжні краї об'єктів [4, 5].

Завдання щодо знаходження та виділення границь візуальних предметів цілком допустимо розглядати як самодостатню та незалежну прикладну проблему, яка може і не мати прямої прив'язки до подальшої тотальної сегментації сцени. Виявлені контури є фундаментальною базою для конструювання різноманітних інформаційних ознак та математичних граматики, необхідних у процесах автоматизованого розпізнавання образів. На даний момент відомо безліч різноманітних методик з виявлення границь і аналітики контурів, але далеко не кожна з цих методик є доцільною для впровадження, скажімо, в мобільну робототехніку. Головним обмежуючим чинником тут є колосальний обсяг необхідних математичних обчислень, який виснажує процесорні потужності. Саме собою поняття "границі фізичного тіла", як і саме філософське визначення "тіла" чи "об'єкта", не піддається абсолютно точної і однозначної математичної формалізації через призму піксельного масиву $B(i, j)$. Керуючись виключно евристичною логікою, алгоритми намагаються намацати граничні пікселі, інтерпретуючи їх як локації, де функція яскравості відчуває найбільш різкий і очевидний стрибок.

Якщо ми математично опишемо цифровий растр D як певну функцію $B(x, y)$, задану в двовимірній системі координат (x, y) , і при цьому умовимося, що функція $B(x, y)$ є гладкою і диференційованою по обох своїх аргументах, то ми зможемо визначити стрибок яскравості в конкретній точці з координатами (x_0, y_0) . У цьому випадку перепад яскравості обчислюватиметься як норма вектора градієнта для функції

$B(x, y)$ строго в цій локації (x_0, y_0) , тобто це буде значення норми вектора, описаного формулою:

$$\left\| \left(\left. \frac{\partial B(x, y)}{\partial x} \right|_{x_0, y_0}, \left. \frac{\partial B(x, y)}{\partial y} \right|_{x_0, y_0} \right) \right\|. \quad (3.1)$$

Проте, враховуючи фундаментальну дискретність цифрової матриці (поля зору) D , зробити пряме, безперервне обчислення для $B(x_0, y_0)$ фізично неможливо. Для того щоб все-таки вдатися до математичного апарату диференціювання та розрахувати силу яскравого стрибка в заданій точці, програмістам доводиться вибирати одну з двох базових стратегій.

По-перше, використання складного алгоритму інтерполяції для трансформації дискретної функції $B(i, j)$ в безперервну функцію $B(x, y)$, яка буде математично задана в межах прямокутної зони координат (x, y) , що охоплює всю матрицю растру D .

По-друге, застосування специфічних прийомів чисельного диференціювання, що дозволяють знаходити похідні для функцій, які у вигляді таблиць дискретних значень [4, 5].

Перша описана стратегія, що отримала в літературі назву методу функціональної апроксимації, пов'язана з необхідністю виробляти титанічні обсяги обчислень, через що вона не набула особливої популярності при проектуванні систем технічного зору для робототехніки. На противагу їй, друга стратегія набула статусу загальноприйнятого стандарту. У професійному середовищі вона фігурує під такими назвами, як градієнтний метод, алгоритм просторового контрастування або метод просторового диференціювання.

Однак перед тим як заглиблюватися в нетрі розбору специфічних алгоритмів пошуку та виділення границь, має сенс дати чіткішу дефініцію самому явищу "границі" або "контур".

3.1 Детальне визначення контуру

Основна мета порогової (бінарizaційної) обробки зводиться до ізоляції таких графічних ділянок, які характеризуються ідентичною (гомогенною) заливкою яскравості. Фіналом порогового перетворення є створення монохромного (бінарного) полотна, на якому чітко окреслені цільові зони. Обчислені геометричні параметри цих зон виступають у ролі наріжних ознак, необхідних машинному інтелекту для класифікації видимих об'єктів та формування цілісного сприйняття сцени [4,5].

У переважній більшості практичних ситуацій максимальну інформаційну цінність несуть у собі властивості границь цих самих зон — тобто їх контури. Як наочно підтверджують дослідження біологів і нейрофізіологів, живі зорові системи при розпізнаванні оточення спираються насамперед на геометрію та контури предметів, а не на просте розбиття видимого поля на різні по яскравості фрагменти.

Таким чином, місія програмного пошуку контурів зводиться до генерування бінарної маски (графічного препарату), яка міститиме в собі виключно лінії цих обрисів. Але перед тим як ми перейдемо до аналізу математичних підходів до реалізації цього завдання, нам слід кристалізувати саме поняття [4,5].

Що ж насправді є контуром? Цей інтуїтивно зрозумілий усім феномен може мати низку наукових трактувань. У рамках нашого дослідження ми спиратимемося на найбільш загальновизнану парадигму. Ми визначатимемо графічний контур зображення як якийсь локалізований у просторі розрив, різкий перепад або стрибкоподібну деформацію рівнів піксельної яскравості. Щоб проілюструвати цю тезу, давайте вивчимо якусь ділянку зображення, прорізану контуром (див. рис. 3.1). Спрощена одновимірна модель цієї ситуації наочно продемонстрована на рис. 3.2.

Відбите на графіку (див. рис. 3.2) відхилення яскравості можна математично охарактеризувати двома ключовими параметрами: загальною висотою стрибка, що відбувся (позначимо її як f_0), а також кутом його нахилу (позначимо як θ), при цьому центральна точка цього похилого перепаду буде мати координату x_0 . З погляду машинної логіки, виявлений перепад інтенсивності визнається справжнім контуром лише тому випадку, якщо значення його висоти і кута нахилу пробивають заздалегідь встановлені порогові ліміти. Абсолютно ідеальний, математично досконалий алгоритм-детектор зобов'язаний сигналізувати про наявність контуру лише одним-єдиним маркером (точкою), який розташовуватиметься рівно у геометричному центрі виявленої похилої ділянки (рис. 3.2 б).

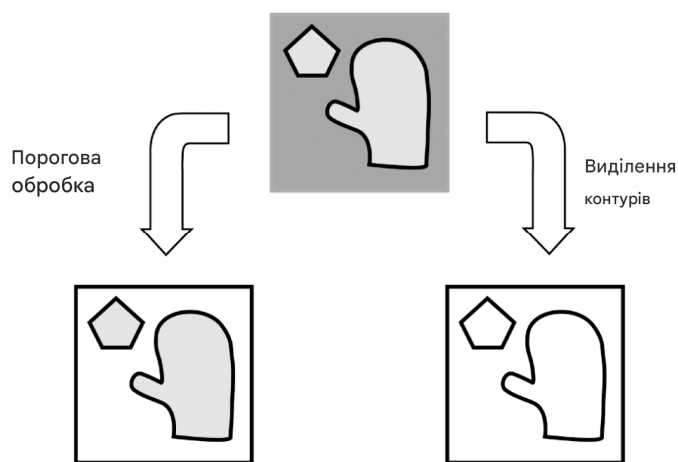


Рисунок 3.1 – Порогова обробка та виділення контурів

Якщо ми переходимо до аналізу повноцінної двовимірної матриці, то в стрибка інтенсивності додається ще один критично важливий параметр - вектор його просторової орієнтації (азимутальний кут на координатній площині). На схемі представлений ізольований локальний фрагмент, у межах якого досліджуваний контур має прямолінійну геометрію. Знову ж таки, ідеальний, позбавлений похибок детектор повинен відмалювати в

результаті своєї роботи нескінченно тонку, не має розривів центральну осьову лінію, що проходить точно по медіані зони градієнтної зміни [4, 5].

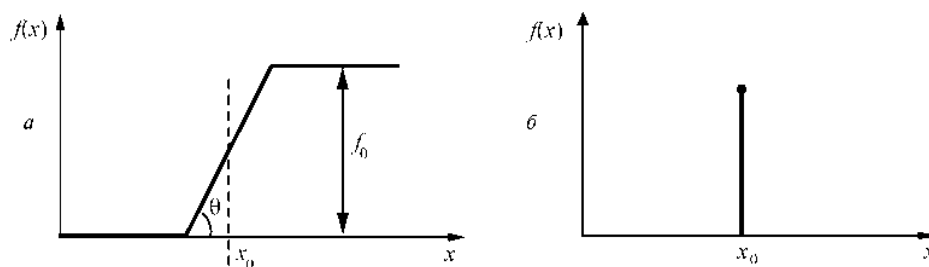


Рисунок 3.2 Контур та його визначення: а -приклад контуру; б - результат ідеального визначення контуру

Важливо загострити увагу на ряді серйозних труднощів, які впливають із прийнятої нами теоретичної дефініції контуру.

Проблема перша: наше базове визначення не гарантує того факту, що отримані контурні криві будуть топологічно замкнутими. Під час процесингу та виділення границь алгоритм може згенерувати зяючі розриви та проломи в тих зонах, де градієнт яскравості змінюється надто слабо або недостатньо контрастно (див. рис. 3.3 та 3.4). Крім цього, присутність у цифровому сигналі паразитних шумів може спровокувати помилкові спрацьовування алгоритму, внаслідок чого детектори "знаходять" неіснуючі контури в тих місцях, де поверхня об'єкта абсолютно однорідна. Всі перераховані вразливості диктують гостру необхідність у впровадженні цілого арсеналу допоміжних алгоритмів постобробки: процедур трекінгу (простежування) граней, математичної інтерполяції розривів, а також евристичних методів пошуку та склеювання зв'язкових векторів з хаотичного набору ізольованих контурних [5].

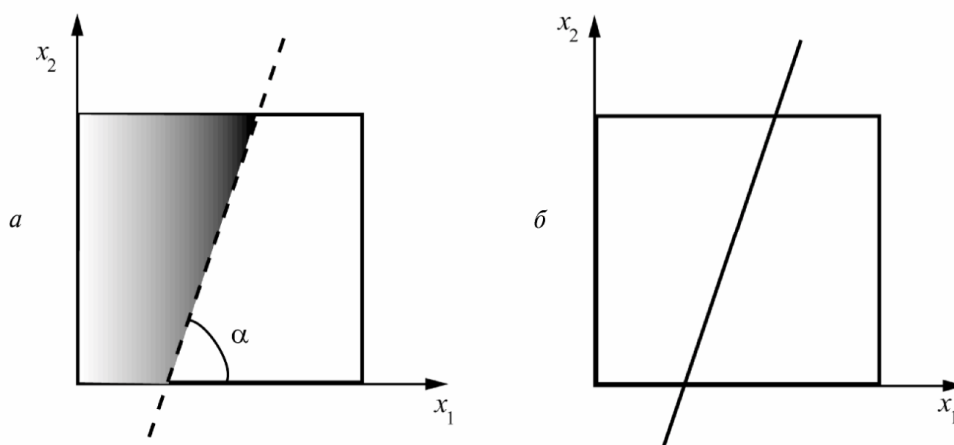


Рисунок 3.3 - Визначення контуру на зображенні: а - зображення з контуром; б - результат ідеального визначення контуру

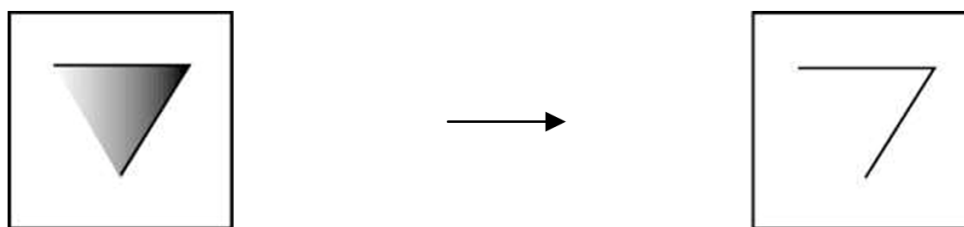


Рисунок 3.4 - Приклад незамінного контуру

Проблема друга: в процесі вилучення контурів, через фізичну розмитість оригінальних фотографій, цифрового шуму матриць або через математичну недосконалість самих фільтрів, на виході можуть генеруватися не просто рвані і переривчасті лінії, але і лінії, що володіють надмірною, "жирної" товщиною (див. рис. 5). У подібних сценаріях розробникам знову доводиться вдаватися до додаткових ітерацій обробки монохромних зображень, щоб примусово "витончити" контури, що виходять, до товщини в один піксель (ця процедура в професійному середовищі носить назву «скелетизації» або створення скелета графічного препарату).



Рисунок 3.5- Приклад широкого контуру та його «скелет»

Проблема третя: в рамках аналізованих сцен періодично зустрічаються (і при цьому обов'язково мають бути детектовані) такі границі об'єктів, фізична структура яких абсолютно не вписується у введене раніше класичне визначення (див. рис. 3.6). Як приклади можна навести об'єкти, візуалізовані у формі надтонких ліній, або ділянки, де яскравість зазнає деформації у вигляді різкого «зламу» або галочки. У випадку з дуже вузькою лінією, вона сама по собі де-факто є своїм власним контуром, і алгоритмічно її без особливих зусиль можна вичленувати за допомогою банальної порогової бінаризації. Що ж до яскравого "зламу", його математичну модель цілком реально штучно «адаптувати» під наші класичні критерії детекції, якщо попередньо прогнати вихідну функцію через операцію математичного диференціювання.

Також у комп'ютерному зорі є фундаментальна невизначеність щодо точного позиціонування контурної лінії з похибкою в плюс-мінус один піксель. З погляду суворої математики було б куди грамотніше і коректніше позиціонувати контур не як фізичний ланцюжок пікселів, пофарбованих у контрастний колір, а як віртуальний кордон (субпіксельний рубіж), що пролягає строго між сусідніми пікселями. Тим не менш, через цілий комплекс програмних та апаратних обмежень, подібний метод субпіксельного подання на практиці застосовується виключно рідко.

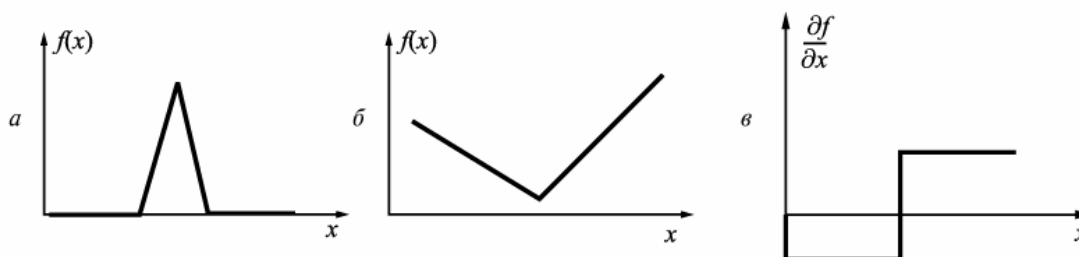


Рисунок 3.6. - Приклади контурів, що не підходять під визначення

Настав час більш предметно розглянути саму механіку вилучення контурних ліній. Найбільш масовий, класичний алгоритмічний підхід до проблематики виявлення перепадів яскравості (що еквівалентно пошуку контурів) в монохромних зображеннях у вигляді структурної схеми представлений нижче (див. рис. 3.7). Згідно з цим алгоритмом, оригінальне вхідне зображення, позначене як f_1 , пропускається через каскад лінійних або нелінійних програмних фільтрів з однією єдиною метою - максимально акцентувати і посилити всі стрибки яскравості, що є. Побічним продуктом цієї математичної фільтрації стає генерація проміжного зображення (маски) f_2 . Рівень яскравості пікселів на цій масці f_2 суттєво відхиляється від нульового (чорного) значення виключно у тих локаціях, де на початковій картинці f_1 було зафіксовано аномально різкі градієнти. Після цього, проміжний шар f_2 піддається процедурі жорсткої порогової відсічки (бінаризації), у результаті система синтезує шуканий фінальний результат - графічний (чи контурний) препарат, що позначається індексом f_3 . Оскільки механізми другого етапу - порогової бінаризації - вже були докладно вивчені раніше, ми сконцентруємо всю нашу увагу на першому, найскладнішому математичному етапі - на алгоритмах вичленування перепадів піксельної інтенсивності. Нижче ми детально розберемо дві найбільш фундаментальні та затребувані в індустрії групи алгоритмів виявлення границь.

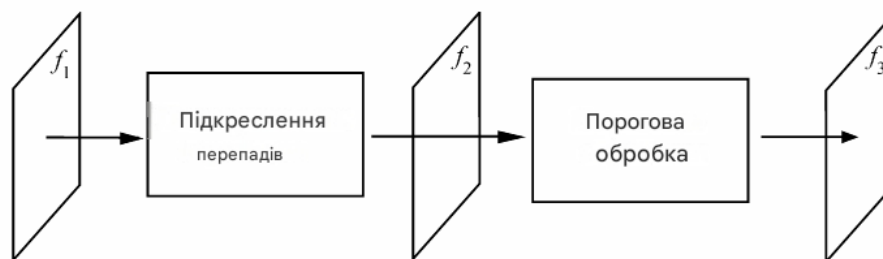


Рисунок 3.7 -Загальний вигляд процедури виділення контурів

3.2 Алгоритм градієнтного виділення

Вектор градієнта для цифрового полотна $f(x, y)$ в заданій координатній точці (x, y) математично задається як вектор у двовимірному просторі [4, 5]:

$$G[f(x, y)] = \begin{pmatrix} G_x \\ G_y \end{pmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}. \quad (3.2)$$

Базуючись на фундаментальних законах векторного аналізу, ми знаємо, що результуючий вектор G завжди спрямований у бік, де спостерігається найвища інтенсивність зростання (стрибка) значення функції f у досліджуваній координаті (x, y) . Даний математичний параметр набуває феноменальної цінності у завданнях, пов'язаних з детекцією границь (тобто зовнішніх контурів) будь-якого фізичного об'єкта, що візуалізується на довільному тлі.

У разі дискретної растрової картини реалізувати це обчислення можна з допомогою цілого арсеналу різних програмних підходів. Один з найбільш примітивних і прямолінійних методів базується на тривіальному обчисленні різниці інтенсивностей між пікселями, що стоять поруч:

$$G_x = \frac{\partial f}{\partial x} = f(x, y) - f(x - 1, y), \quad (3.3)$$

$$G_x = \frac{\partial f}{\partial x} = f(x, y) - f(x, y - 1). \quad (3.4)$$

Існує і дещо більш просунутий у обчислювальному плані алгоритм, який залучає до процесу аналізу не просто сусідні пікселі, а цілу квадратну матрицю (околиця) розміром 3×3 пікселя, центр якої розташовується строго в координаті, що вивчається (x, y) . В арсеналі програмістів є величезна кількість математичних схем для синтезу результуючого (вихідного) графічного шару $g(x, y)$, які повністю спираються на розрахункові дані градієнтів. Найбільш елементарним і ресурсозберігаючим методом є пряме присвоєння функції g в координаті (x, y) того скалярного значення, яке точно дорівнює величині модуля градієнта оригінальної картини f в аналогічній координаті [5].

3.3 Градієнтний оператор Робертса

Давайте уявімо, що певна піксельна матриця 3×3 відображає реальні рівні яскравості в безпосередній близькості (околиці) від якогось конкретного пікселя, що вивчається (див. рис. 3.8). Одним із найтривіальніших і обчислювально легких шляхів для обчислення перших приватних похідних у заданій точці є застосування так званого перехресного градієнтного фільтра (оператора) Робертса [4, 5]. Його математична суть виражається такими формулами:

$$G_x = (z_9 - z_5), \quad (3.5)$$

$$G_y = (z_8 - z_6). \quad (3.6)$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Рисунок 3.8 - Околиця 3x3 всередині зображення

Практична імплементація цих математичних похідних програмний код може бути легко здійснена за допомогою тотального сканування і фільтрації всього растрового масиву із застосуванням спеціального оператора. Цей оператор фізично описується матричними масками згортки, причому сам механізм прогону маски повністю аналогічний стандартній процедурі просторової фільтрації [4, 5].

3.4 Градієнтний оператор Превітта

Математичний оператор Превітта, проводячи аналогію з фільтром Робертса, також здійснює свої маніпуляції в межах піксельної околиці розміром 3x3 (див. рис. 3.8). Однак кардинальна відмінність полягає в тому, що конфігурація маски згортки для цього фільтра регламентується зовсім іншими виразами алгебри [4, 5] :

$$G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_7), \quad (3.7)$$

$$G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7). \quad (3.8)$$

Згідно з цими рівняннями, різниця алгебри між сумами піксельних інтенсивностей верхньої і нижньої горизонтальних рядків нашої матриці 3x3 виступає в ролі грубо апроксимованого значення похідної вздовж координатної осі X . Аналогічним чином, різницю між сумами інтенсивностей першого (лівого) і останнього (правого) вертикальних стовпців цієї матриці інтерпретується як апроксимація похідної вздовж осі

У. Щоб перекласти ці теоретичні викладки в площину практичного коду, інженери використовують оператор згортки, який описується конкретними цифровими масками. Цей алгоритмічний інструмент і має назву оператора Превітта [4, 5] .

3.5 Градієнтний оператор Собеля

Найпопулярніший оператор Собеля у своїх обчисленнях задіює рівно ту саму піксельну матрицю 3x3, показану на рис. 3.8. За своєю архітектурою та логікою роботи він має колосальну схожість з алгоритмом Превітта. Ключова модернізація і відмінна риса фільтра Собеля криється у впровадженні мультиплікативного вагового коефіцієнта, рівного 2, який застосовується виключно до центральних (середніх) елементів згорткової матриці [4, 5] :

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3), \quad (3.9)$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7). \quad (3.10)$$

Використання цього форсованого вагового множника має цілком конкретну мету -потужне придушення негативного ефекту розмиття (згладжування) контурів. Це досягається за рахунок того, що центральні точки піксельної околиці набувають значно більшої математичної «значущості» при розрахунку підсумкового градієнта [4, 5] .

Всі детально розібрані нами вище варіанти матриць згортки активно впроваджуються в код програм для отримання ортогональних складових вектора градієнта по осях X і Y .

4 ЦЕНТР МАС ПЛОСКИХ ФІГУР: ТЕОРЕТИЧНІ ОСНОВИ ТА ПРАКТИЧНІ МЕТОДИ ОБЧИСЛЕННЯ

4.1 Загальна характеристика проблеми визначення цунтра мас

Проблема визначення центру мас (або центру тяжіння) є однією з фундаментальних класичних задач у теоретичній механіці, фізиці, інженерній справі та архітектурі. Центр мас — це геометрична точка, яка характеризує розподіл мас у тілі або системі тіл. Для твердого тіла, що перебуває в однорідному гравітаційному полі, центр мас збігається з центром тяжіння - точкою прикладання рівнодійної всіх сил тяжіння, що діють на частинки цього тіла. Знання координат центру мас є критично важливим при проектуванні будівель, мостів, літальних апаратів, автомобілів та кораблів та в робототехніці. Від правильного розташування цієї точки залежить стійкість, керованість та загальна безпека конструкції. В математичному сенсі, центр мас плоских фігур (двовимірних об'єктів) розглядається як центр ваги нескінченно тонкої однорідної пластини заданої форми [5 – 9].

В даному розділі ми детально розглянемо математичний апарат, що використовується для обчислення координат центру мас як дискретних систем матеріальних точок, так і суцільних плоских фігур. Ми дослідимо інтегральні методи, а також методи розбиття та доповнення.

4.2 Центр мас дискретної системи точок

Для того щоб перейти до суцільних фігур, спершу варто згадати визначення центру мас для системи з n матеріальних точок. Нехай на площині задано дискретну систему точок з масами $m_1, m_2, \dots, m_n$, які

розташовані у точках з координатами $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$. Радіус-вектор центру мас такої системи визначається за класичною формулою:

$$\vec{r}_C = \frac{\sum_{i=1}^n m_i \vec{r}_i}{\sum_{i=1}^n m_i}. \quad (4.1)$$

У проекціях на декартові осі координат Ox та Oy , формули для знаходження координат центру мас (x_C, y_C) набувають вигляду [5 – 9]:

$$x_C = \frac{\sum_{i=1}^n m_i x_i}{M}, \quad y_C = \frac{\sum_{i=1}^n m_i y_i}{M}. \quad (4.2)$$

де $M = \sum_{i=1}^n m_i$ — загальна маса всієї системи. Ці базові співвідношення стануть основою для виведення формул для суцільних фігур шляхом граничного переходу від сум до інтегралів.

4.3 Центр мас суцільної плоскої фігури

Плоска фігура розглядається як матеріальна пластина нескінченно малої товщини, що займає певну область D на координатній площині Oxy . Нехай поверхнева густина цієї фігури (маса одиниці площі) задається функцією $\rho(x, y)$. Якщо розбити область D на нескінченно малі елементи площі $dA = dx dy$, то маса кожного такого елемента становитиме $dm = \rho(x, y) dA$. Застосовуючи принцип суперпозиції та переходячи до границі, коли розміри елементів прямують до нуля, суми перетворюються на подвійні інтеграли по області D . Тоді координати центру мас неоднорідної плоскої фігури обчислюються за формулами: [5 – 9]

$$x_C = \frac{\iint_D x \rho(x, y) dA}{\iint_D \rho(x, y) dA}, \quad y_C = \frac{\iint_D y \rho(x, y) dA}{\iint_D \rho(x, y) dA}. \quad (4.3)$$

4.3.1 Однорідні плоскі фігури

У більшості геометричних задач розглядаються однорідні фігури, тобто такі, густина яких є сталою в усіх точках: $\rho(x, y) = \rho_0 = \text{const}$. У цьому випадку константа ρ_0 виноситься за знак інтеграла і скорочується в чисельнику та знаменнику. Координати центру мас (який у даному випадку збігається з геометричним центром або центроїдом фігури) залежатимуть лише від її форми та розмірів. Знаменник перетворюється на площу фігури A :

$$A = \iint_D dA = \iint_D dx dy. \quad (4.4)$$

Тоді формули для центру мас однорідної плоскої фігури остаточно набувають вигляду:

$$x_C = \frac{1}{A} \iint_D x dx dy, \quad y_C = \frac{1}{A} \iint_D y dx dy. \quad (4.5)$$

4.4 Статичні моменти плоских фігур

Інтеграли, що стоять у чисельниках вищенаведених формул, мають особливе значення в механіці та опорі матеріалів і називаються статичними моментами плоскої фігури відносно координатних осей. Статичний момент фігури відносно осі Ox (позначається S_x) та відносно осі Oy (позначається S_y) визначаються як [5 – 9]:

$$S_x = \iint_D y dx dy, \quad S_y = \iint_D x dx dy. \quad (4.6)$$

Використовуючи поняття статичних моментів, координати центру мас можна записати у компактній формі:

$$x_C = \frac{S_y}{A}, \quad y_C = \frac{S_x}{A}. \quad (4.7)$$

Вкажемо властивості статичних моментів [5 – 9]:

1. Статичний момент може бути додатним, від'ємним або дорівнювати нулю, залежно від розташування фігури відносно осей координат.

2. Якщо статичний момент відносно певної осі дорівнює нулю ($S_x = 0$ або $S_y = 0$), це означає, що центр мас фігури лежить на цій осі (тобто $y_C = 0$ або $x_C = 0$ відповідно).

3. Статичний момент відносно будь-якої центральної осі (осі, що проходить через центр мас) завжди дорівнює нулю.

4.5 Практичні методи визначення центру мас

Обчислення подвійних інтегралів не завжди є найшвидшим шляхом до результату. На практиці для складних фігур використовують ряд спрощених методів.

4.5.1 Метод симетрії

Якщо однорідна плоска фігура має вісь симетрії, то її центр мас обов'язково лежить на цій осі. Якщо ж фігура має дві осі симетрії, то центр мас знаходиться у точці їх перетину. Цей метод дозволяє миттєво визначити центр мас для таких фігур як коло, прямокутник, ромб, правильний багатокутник тощо.

4.5.2 Метод розбиття на прості фігури

Будь-яку складну плоску фігуру можна розбити на k простіших частин (прямокутники, трикутники, кола), площі яких $A_1, A_2, \dots, A_k$, а координати їх центрів мас відповідно $(x_{C1}, y_{C1}), (x_{C2}, y_{C2}), \dots, (x_{Ck}, y_{Ck})$ є відомими. Оскільки статичний момент складної фігури дорівнює алгебраїчній сумі статичних моментів її частин, координати загального центру мас обчислюються за формулами [5 – 9]:

$$x_C = \frac{\sum_{i=1}^k A_i x_{Ci}}{\sum_{i=1}^k A_i}, \quad y_C = \frac{\sum_{i=1}^k A_i y_{Ci}}{\sum_{i=1}^k A_i}. \quad (4.8)$$

4.5.3 Метод від'ємних площ (метод доповнення)

Цей метод є окремим випадком методу розбиття, який використовується для фігур з вирізами (отворами). У цьому випадку фігура розглядається як суцільна (без вирізу), але площа вирізаної частини береться зі знаком мінус (розглядається як "від'ємна маса"). Якщо суцільна фігура має площу A_1 і центр мас (x_1, y_1) , а виріз має площу A_2 і центр мас (x_2, y_2) , то координати центру мас фігури з вирізом [5 – 9]:

$$x_C = \frac{A_1 x_1 - A_2 x_2}{A_1 - A_2}, \quad y_C = \frac{A_1 y_1 - A_2 y_2}{A_1 - A_2}. \quad (4.9)$$

4.5.4 Теореми Паппа-Гюльдена

Для визначення центрів мас ліній і плоских фігур іноді зручно використовувати теореми Паппа-Гюльдена, які пов'язують положення центру мас з площею поверхні або об'ємом тіла обертання.

Перша теорема Паппа-Гюльдена: Площа поверхні, утвореної обертанням плоскої дуги навколо осі, що лежить в її площині і не перетинає її, дорівнює добутку довжини цієї дуги на довжину кола, описаного її центром мас.

Друга теорема Паппа-Гюльдена: Об'єм тіла, утвореного обертанням плоскої фігури навколо осі, що лежить в її площині і не перетинає її, дорівнює добутку площі цієї фігури на довжину кола, описаного її центром мас, $V = A \cdot 2\pi y_C$. Звідси $y_C = \frac{V}{2\pi A}$.

4.5.5 Координати центрів мас основних геометричних фігур

Для успішного застосування методу розбиття необхідно знати положення центрів мас базових геометричних фігур. Наведемо їх:

1. Прямокутник зі сторонами b та h : Центр мас лежить на перетині діагоналей. Відстань від сторін дорівнює $b/2$ та $h/2$.

2. Прямокутний трикутник з катетами b та h : Центр мас (перетин медіан) розташований на відстані $b/3$ та $h/3$ від прямих кутів. Тобто [5 – 9]:

$$x_C = \frac{b}{3}, \quad y_C = \frac{h}{3}.$$

3. Довільний трикутник: Центр мас завжди знаходиться в точці перетину його медіан (ділить кожну медіану у відношенні 2:1, рахуючи від вершини).

4. Півколо радіуса R : Центр мас лежить на осі симетрії. Відстань від прямого краю (діаметра) до центру мас дорівнює:

$$y_C = \frac{4R}{3\pi}.$$

5. Круговий сектор з радіусом R і центральним кутом 2α : Центр мас лежить на осі симетрії на відстані від центру кола:

$$x_C = \frac{2R \sin \alpha}{3\alpha}.$$

(де кут α виражений у радіанах).

4.6 Інтегральний метод: обчислення для фігур, обмежених кривими

Якщо плоска фігура (криволінійна трапеція) обмежена зверху графіком неперервної функції $y = f(x)$, знизу віссю Ox , а з боків вертикальними прямими $x = a$ та $x = b$, то подвійні інтеграли можна звести до визначених (одинарних) інтегралів. Площа такої фігури:

$$A = \int_a^b f(x) dx. \quad (4.10)$$

Координати центру мас обчислюються так. Елементарна вертикальна смужка шириною dx і висотою $y = f(x)$ має площу $dA = f(x)dx$. Її власний центр мас знаходиться в точці з координатами $(x, \frac{f(x)}{2})$. Тоді:

$$x_C = \frac{1}{A} \int_a^b x \cdot f(x) dx, \quad (4.11)$$

$$y_C = \frac{1}{A} \int_a^b \frac{f(x)}{2} \cdot f(x) dx = \frac{1}{2A} \int_a^b [f(x)]^2 dx. \quad (4.12)$$

Якщо фігура обмежена двома функціями $y_2 = f_2(x)$ (зверху) та $y_1 = f_1(x)$ (знизу), де $f_2(x) \geq f_1(x)$ на відрізку $[a, b]$, формули узагальнюються:

$$A = \int_a^b [f_2(x) - f_1(x)] dx, \quad (4.13)$$

$$x_C = \frac{1}{A} \int_a^b x [f_2(x) - f_1(x)] dx, \quad (4.14)$$

$$y_C = \frac{1}{2A} \int_a^b ([f_2(x)]^2 - [f_1(x)]^2) dx. \quad (4.15)$$

4.7 Практичне застосування

Вказані вище співвідношення були використані для визначення центра ваги плоских фігур за їх зображеннями в системах комп'ютерного зору. Для цього на мові програмування Python було написано скрипт із залученням відкритої бібліотеки комп'ютерного зору OpenCV.

Скрипт автоматично визначає границі геометричного об'єкта та обраховує його центр ваги. Положення центра ваги позначається червоною точкою. Результати роботи скрипта містяться на рис. 4.1 і 4.2.

Деталізуємо операції, які виконуються по відношенню до зображення. А саме:

1. Зображення завантажується в оперативну пам'ять.
2. Здійснюється конвертація до відтінків сірого. Оскільки кольорова інформація не потрібна для подальшої бінаризації зображення.
3. Зображення піддається низькочастотній фільтрації шляхом гаусівського розмиття з метою видалити шум з зображення.
4. Відбувається бінаризація зображення (тобто порогова обробка) яка перетворює початкове зображення на чорно-біле.
5. Далі відбувається пошук контурів і морфологічне замикання.

6. Спираючись на отриману інформацію відбувається розрахунок положення точки центру ваги плоскої фігури, обмеженої контуром, за формулою (4.8) у випадку фігур із складною геометрією.

7. Нарешті визначені контури накладаються на зображення для візуалізації результатів роботи скрипта.

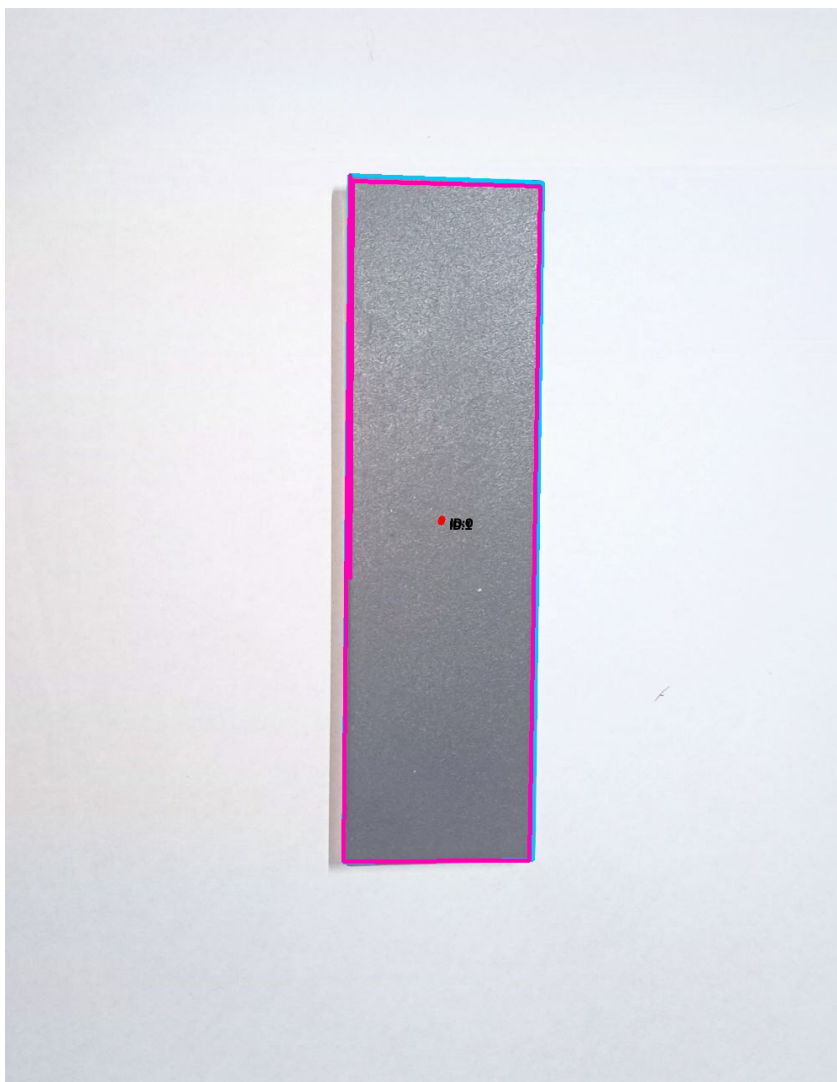


Рисунок 4.1 – Приклад коректного визначення центру ваги

Зауважимо, що виділення границь на зображеннях з низьким контрастом може представляти значну проблему. Наприклад, на рис. 4.3 показано трикутник з білого прозорого пластику на білому тлі. Звичайні алгоритми виділення границь не спрацьовують коректно, тому потрібно

залучати інші, спеціалізовані процедури попередньої обробки зображення. Подібна ситуація виникає, коли ми трохи підвищуємо контраст. Наприклад, на рис. 4.4 зображена канцелярська лінійка із синього напівпрозорого пластику. Як бачимо в порівнянні з рис. 4.3 ситуація покращилась, але недостатньою мірою.

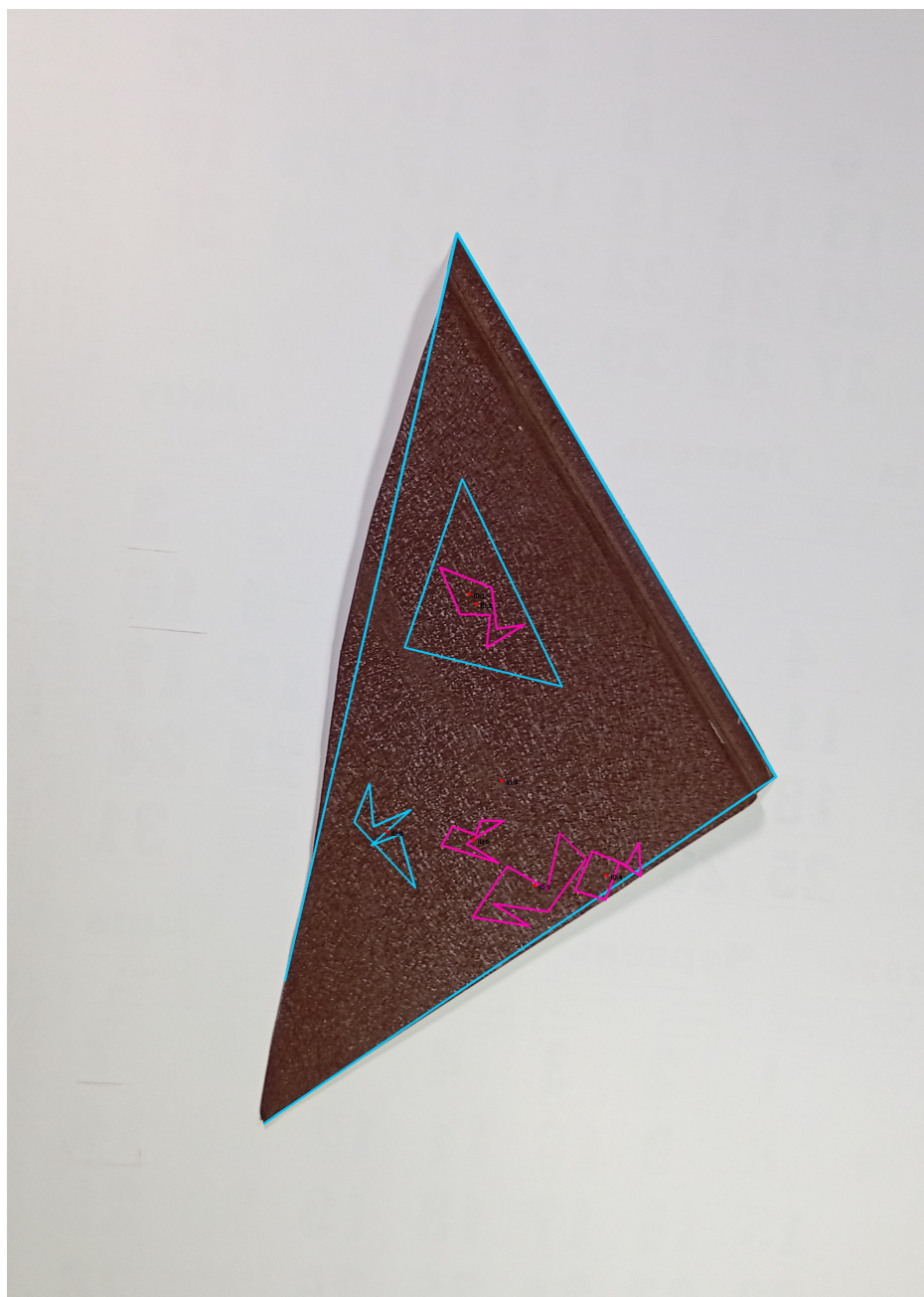


Рисунок 4:2 – Приклад помилкового визначення багатьох центрів ваги фігури внаслідок геометричних спотворень на зображенні

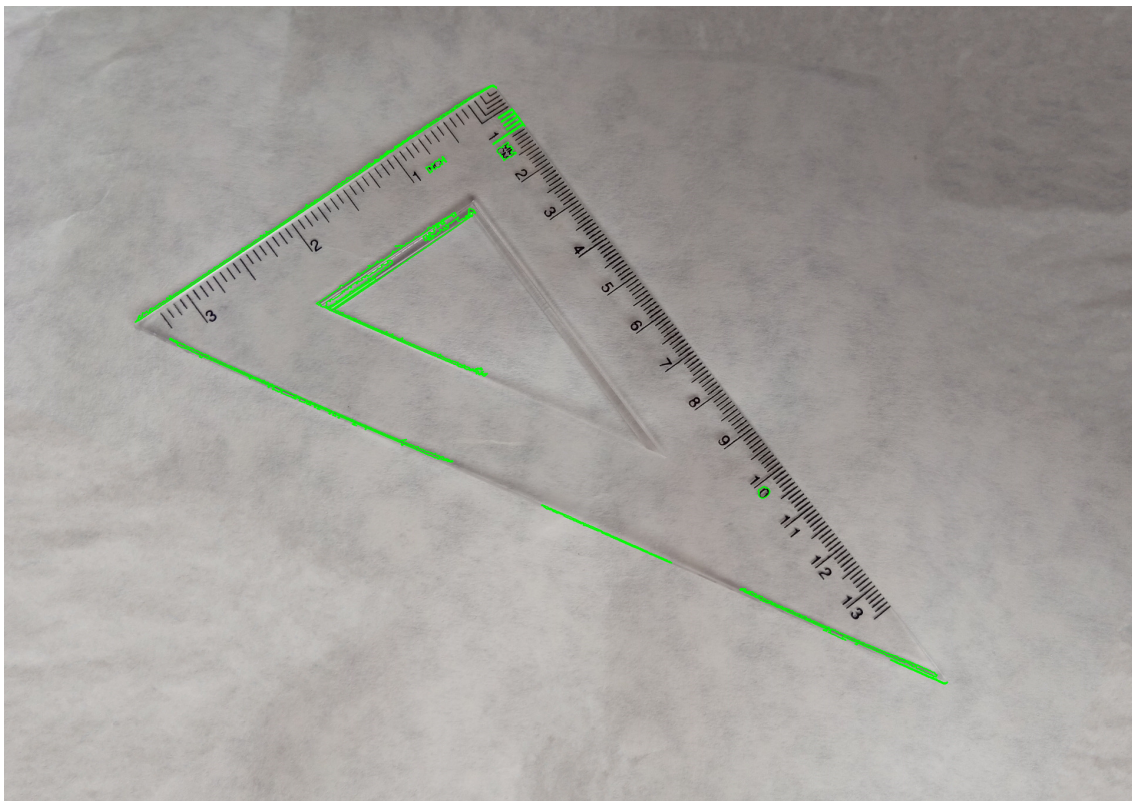


Рисунок 4.3 – Канцелярський трикутник з білого прозорого пластику на білому тлі

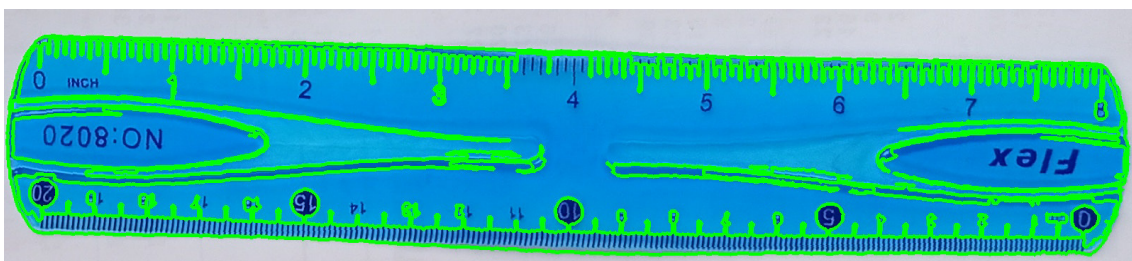


Рисунок 4.4 – Канцелярська лінійка із синього напівпрозорого пластику на білому тлі

ВИСНОВКИ

В рамках кваліфікаційної роботи було всебічно досліджено та проаналізовано фундаментальні питання, що стосуються теорії цифрової обробки графічної інформації, а також математичні алгоритми, які застосовуються для виявлення та виділення границь на растрових зображеннях. Варто особливо підкреслити, що завдання пошуку контурів на сьогодні по праву посідає місце однієї з найбільш пріоритетних і складних проблем у глобальній парадигмі аналізу зображень. Такий формат цифрової обробки візуальних даних має надзвичайну актуальність і попит буквально в кожному секторі сучасної індустрії, де впроваджуються системи штучного (машинного) зору — починаючи від високоточної медичної діагностики та автоматизації виробничих конвеєрів і закінчуючи інструментами аналізу в соціальній сфері.

Детально розглянуто питання визначення центра ваги плоских фігур за їх графічними зображеннями. На мові програмування Python було створено скрипт, який засобами бібліотеки OpenCV розв'язував задачу виділення контурів на зображенні із подальшим обчисленням центру ваги цієї фігури. Такі системи знаходять своє застосування в робототехніці.

ПЕРЕЛІК ПОСИЛАНЬ

1. Horn, B. (1986). Robot vision. MIT press.
2. Fridrich, J. (2009). Steganography in digital media: principles, algorithms, and applications. Cambridge university press.
3. Myler, H. R., & Weeks, A. R. (2009). The pocket handbook of image processing algorithms in C. Prentice Hall Press.
4. Bhuyan, M. K. (2020). Computer vision and image processing: Fundamentals and applications. CRC Press.
5. Soyfer, V. A. (Ed.). (2003). Metody komp'yuternoy obrabotki izobrazheniy [Methods of computer image processing]. Fizmatlit.
6. Hibbeler, R. C. (2015). Engineering mechanics: Statics (14th ed.). Pearson Education.
7. Meriam, J. L., & Kraige, L. G. (2012). Engineering mechanics: Statics (7th ed.). John Wiley & Sons.
8. Beer, F. P., Johnston, E. R., & Mazurek, D. F. (2015). Vector mechanics for engineers: Statics (11th ed.). McGraw-Hill Education.
9. Timoshenko, S., & Young, D. H. (1956). Engineering mechanics. McGraw-Hill.