

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Спосіб оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах

Галузь знань _____ 12 – Інформаційні технології _____
Шифр і назва галузі знань
Спеціальність _____ 122 – Комп'ютерні науки _____
Шифр і назва спеціальності
Освітня програма _____ Комп'ютерні науки _____
Назва освітньої програми


Виконав: _____ студент 4 курсу, група КН-19-1 _____ Н.О. Вусатий _____
Курс, група виконавця Підпис Ініціали, прізвище
Керівник: _____ к.т.н., доцент кафедри КН _____ О.А. Пасічник _____
Науковий ступінь, посада Підпис Ініціали, прізвище
Нормоконтроль: _____ к.т.н., доцент кафедри КН _____ Р.О. Багрії _____
Науковий ступінь, посада Підпис Ініціали, прізвище

До захисту допускаю:
Зав. кафедри КН, д.т.н., професор _____ О.В. Бармак _____
Підпис Ініціали, прізвище
01 06 2023 р.

Хмельницький 2023

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Факультет інформаційних технологій
Кафедра комп'ютерних наук
Освітній ступінь бакалавр
Галузь знань 12 – Інформаційні технології
Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук


(підпис)
д.т.н., професор О.В. Бармак
«06» 03 2023 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

1. Тема кваліфікаційної роботи бакалавра: «Спосіб оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах»
2. Завдання видано студенту Вусатому Нікіті Олександровичу
(прізвище, ім'я, по батькові)
3. Керівник роботи доцент кафедри КН Пасічник Олександр Анатолійович
(посада, прізвище, ім'я, по батькові)
4. Затверджено наказом університету від «01» 03 2023 р. № 5
5. Дата видачі завдання студенту: «03» 03 2023 р.
6. Зміст пояснювальної записки (перелік задач) та вихідні дані: Провести аналіз предметної області, огляд методів аналізу зображень з градієнтом кольору задля визначення положення об'єкту спостереження та його зміщення. Розробити спосіб оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах. Визначити послідовність застосування способу визначення положення об'єкту та його зміщення на зображенні з круговим градієнтом кольору. Спроекувати функціональну структуру інформаційної системи. Створити програмну реалізацію способу оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах.

7. Календарний план виконання кваліфікаційної роботи бакалавра:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи бакалавра з керівником	грудень 2022	виконано
2	Ознайомлення з предметною областю, формулювання мети та задач дослідження, визначення об'єкта та предмета дослідження	січень 2023	виконано
3	Робота над розділом 1 – Характеристика предметної області та постановка задачі	січень 2023	виконано
4	Робота над розділом 2 – Спосіб оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах	березень 2023	виконано
5	Робота над розділом 3 – Програмна реалізація способу оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах	квітень 2023	виконано
6	Оформлення пояснювальної записки згідно вимог	травень 2023	виконано
7	Попередній захист кваліфікаційної роботи бакалавра	травень 2023	виконано
8	Захист кваліфікаційної роботи бакалавра на засіданні Екзаменаційної комісії	червень 2023	виконано

Виконавець: студент 4 курсу, група КН-19-1
Курс, група виконавця


Підпис

Н.О. Вусатий
Ініціали, прізвище

Керівник: к.т.н., доцент кафедри КН
Науковий ступінь, посада


Підпис

О.А. Пасічник
Ініціали, прізвище

Анотація

Тема кваліфікаційної роботи бакалавра: Спосіб оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-19-1 Вусатий Нікіта Олександрович

Керівник кваліфікаційної роботи бакалавра: к.т.н., доцент кафедри КН Пасічник Олександр Анатолійович

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
65	46	0	26	2

Мета кваліфікаційної роботи бакалавра полягає в розробці способу оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах. Для досягнення поставленої мети визначені наступні задачі дослідження: визначити послідовність застосування способу визначення положення об'єкту на зображенні з круговим градієнтом кольору; визначити послідовність оцінки зміщення визначених об'єктів у задачі вимірювання в технічних системах; реалізувати інформаційну технологію методу підвищення точності вимірювання лінійних переміщень за аналізом зображень.

Результатом виконання кваліфікаційної роботи бакалавра є створення інформаційної системи для вимірювання в технічних системах з використанням способу оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору.

Ключові слова: аналіз зображення, круговий градієнт кольору, спосіб оцінки зміщення.

Виконавець: студент 4 курсу, група КН-19-1

Курс, група виконавця



Н.О. Вусатий

Ініціали, прізвище

Зміст

Перелік скорочень	3
Вступ.....	4
Розділ 1 Характеристика предметної області та постановка задачі.....	7
1.1 Аналіз предметної області.....	7
1.2 Аналіз методів визначення положення об'єктів за аналізом зображень з круговим градієнтом кольору	11
1.3 Аналіз методів оцінки зміщення об'єктів за аналізом зображень	13
1.4 Аналіз існуючих підходів оцінки зміщення об'єктів.....	17
1.5 Мета, задачі та вимоги до реалізації інформаційної системи	18
Розділ 2 Спосіб оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах	20
2.1 Спосіб визначення положення об'єктів за аналізом зображень з круговим градієнтом кольору	20
2.2 Спосіб оцінки зміщення об'єктів за аналізом зображень	30
2.3 Функціональна структура інформаційної системи.....	39
2.4 Висновки до розділу 2	40
Розділ 3 Програмна реалізація способу оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах.....	41
3.1 Структура модулів системи, їх взаємозв'язок	41
3.2 Особливості реалізації способу оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах	45
3.3 Опис функціональних можливостей інформаційної системи	66
3.4 Висновки до розділу 3	68
Висновки	69
Перелік посилань.....	70
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
КРБ	Кваліфікаційна робота бакалавра
КН	Комп'ютерні науки

Вступ

Кваліфікаційна робота бакалавра присвячена розробці способу оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах

Актуальність теми. Нинішній етап розвитку людства можна характеризувати як неймовірно великий обсяг інформації у цифровій індустрії. Спостерігаючи за цим, сміливо можна сказати, цифрова індустрія - повністю замінило та автоматизувало життя і життєдіяльність. Як було зазначено, сьогодні все більш поширюється тенденція до повного проникнення інтелектуальних інформаційних технологій у всі сфери людської діяльності, що часто супроводжується використанням різних цифрових технічних пристроїв. Однією зі стійких тенденцій останнього часу є значне поширення використання засобів для отримання зображення навіть у повсякденному житті. У той же час, зростає потреба в визначенні наявності або відсутності зміщення об'єктів спостереження. Ці обставини створюють можливості для розв'язання цієї задачі шляхом порівняння послідовних зображень, що дозволяє не тільки якісно оцінювати наявність зміщення, але й вимірювати його величину. Такий підхід надає можливість вирішувати задачі вимірювання в технічних системах. Ключовим питанням, яке обумовлює потенційну точність в задачі вимірювання, є максимальна точність вимірювання положення об'єкту спостереження на тлі інших. Загалом спостерігач має справу із градієнтною зміною кольорів в умовах як природнього так й штучного освітлення, й завдання визначення положення об'єкту спостереження фактично полягає в аналізі зображень з круговим градієнтом кольору.

Мета і задачі роботи. Мета кваліфікаційної роботи полягає в розробці способу оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах.

Для досягнення поставленої мети визначені наступні задачі дослідження:

- визначити послідовність застосування способу визначення

положення об'єкту на зображенні з круговим градієнтом кольору;

- визначити послідовність оцінки зміщення визначених об'єктів у задачі вимірювання в технічних системах;
- реалізувати інформаційну технологію методу підвищення точності вимірювання лінійних переміщень за аналізом зображень.
- провести експериментальне тестування інформаційної технології.

Об'єкт дослідження - процес обробки зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах.

Предмет дослідження - моделі, методи, алгоритми та засоби для створення способу оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах.

Методи дослідження, застосовані для вирішення поставлених завдань:

- для визначення послідовності застосування методу вимірювання лінійних переміщень за аналізом зображень - методи аналізу та системний підхід, методи обробки зображень;
- для визначення структури інформаційних потоків методу вимірювання лінійних переміщень за аналізом зображень - теорія алгоритмів, методи обробки зображень, системний аналіз, методи оптимізації;
- для формулювання функціональних відмінностей апаратно-програмних та програмно-апаратних вимірювальних комплексів, щодо реалізації методу вимірювання лінійних переміщень за аналізом зображень, - методи системотехніки, метрологія, структурний аналіз, методологія проектування технічних систем, методологія проектування програмних систем;
- для визначення функцій програмної складової методу вимірювання лінійних переміщень за аналізом зображень - методи функціонального аналізу;
- для вирішення задачі встановлення розмірів об'єктів зображення в лінійних метричних одиницях - методи аналітичної геометрії, методи обробки зображень;
- для формулювання математичної залежності для визначення лінійних переміщень за аналізом зображення - методи аналітичної геометрії,

методи обробки зображень;

– для реалізації програмної складової методу вимірювання лінійних переміщень за аналізом зображень - методологія створення програмних систем та об'єктно-орієнтоване програмування.

Структура та обсяг роботи. Кваліфікаційна робота магістра складається з завдання, реферату, змісту, переліку скорочень, вступу, 3 розділів, висновків, переліку посилань із 26 найменувань та 2 додатків. Загальний обсяг дипломної роботи бакалавра становить 85 сторінок, з них 65 сторінки основного тексту та 10 сторінок додатків. У роботі наведено 46 рисунків та 0 таблиці.

Ключові слова: спосіб зміщення, аналіз зображень, метод k-means.

Розділ 1 Характеристика предметної області та постановка задачі

1.1 Аналіз предметної області

Сучасний етап еволюції людства відзначений величезною кількістю та плинною інформацією, яка пронизує всі аспекти його діяльності. Особливим аспектом розвитку нових технологій і обладнання є інформаційний вибух. Наприклад, якщо в середині 19 століття за 50 років було вдвічі більше інформації, то на початку 20 століття вона подвоїлася за 20 років, а до кінця 20 століття цей період скоротився до 3-х років [1].

Серед різноманітних видів інформації, що повсякденно використовуються людиною у своїй науковій, виробничій та побутовій діяльності, першочергове значення має вимірювальна інформація, яка містить кількісну оцінку результатів наукових досліджень і спостережень, стан виробництва та технологічні процеси, характеристики продукції, фактори зовнішнього середовища, фізіологічні та психоемоційні стани тощо [2].

Розвиток наукових досліджень і дотримання фундаментальних і прикладних понять, здійснення технологічних процесів і регулювання їх параметрів викликають необхідність вимірювання значної кількості фізичних величин різного типу. Про існування та різноманітність фізичних величин свідчить той факт, що система одиниць СІ містить 7 основних, 2 додаткові та 113 похідних одиниць, у тому числі магнітних і електричних 40, світлових 15, механічних 14, акустичних 14, теплових 11, молекулярно-фізичних і фізична хімія 11, часові та просторові 6, іонізуюче випромінювання 2. Що об'єми, виміряні наприкінці 20-го століття в промисловості, включали такі частки вимірювання різних фізичних величин, зокрема вимірювання об'єму становило 50% загального, масового та вимірювання об'ємної витрати становило 15%, вимірювання тиску становило 10% від загального, вимірювання рівня становило 5% від загального, вимірювання маси та об'єму становило 5%, вимірювання часу становило 4% від загального, визначення властивостей матеріалу становило 4% , вимірювання маси та об'ємної витрати склали 5%, вимірювання електрики та

магніту склали 5%. Як наслідок, найбільшу конкретну частину фізичних вимірювань становлять неелектричні вимірювання, цей процес є обширним, і намічена тенденція продовжуватиметься та зростатиме в майбутньому [3].

Загальносвітова тенденція вдосконалення обладнання та технологій полягає у покращенні майже кожного продукту. Фактором, який об'єктивно впливає на підвищення їх якості, надійності та довговічності, є дослідження відповідних властивостей, параметрів і характеристик матеріалів, деталей, вузлів, агрегатів і систем з деталізованим рівнем інформації щодо їх збереження та накопичення, їх структурування, систематизація, класифікація тощо.

Вченим і дослідникам завжди доводилося з'ясувати максимально повну і точну інформацію про процеси і явища, що відбуваються під час кожного експерименту. Проблема підвищення інформативності спостережень і достовірності отриманих експериментальних даних може бути вирішена за рахунок збільшення обсягу отриманих даних, зокрема за рахунок скорочення часу між дискретними вимірюваннями, збереження безперервного вимірювання параметрів процесу без зупинки і без розбирання, використання відповідних методів реєстрації, які не спотворюють спостережувані закономірності, використання різноманітних методів візуалізації, використання найновіших методів і технік вимірювання, які базуються на принципово інших фізичних процесах чи явищах [4].

Одним з перспективних підходів до вирішення цих завдань і проблем є впровадження різноманітних систем автоматичної реєстрації параметрів процесу.

Методи візуалізації та документування, які використовують фотографію, популярні в наукових дослідженнях протягом майже двох століть. Широке використання фотографії пояснюється здатністю документувати процеси та явища як якісно, так і кількісно [5].

Особливості фотографічних методів фіксації інформації в результаті досліджень і спостережень мають значну перевагу і тому вони так широко використовуються в багатьох сферах життя людини [6].

Значну перевагу в інформативності мають кількісні фотоматеріали. Однак інформація може бути об'єднана з якісними даними для отримання кількісної інформації про об'єкти або процеси. З теоретичної точки зору, основний принцип фотографічного методу вимірювання лінійного зміщення полягає у визначенні зміни положення об'єкта на серії фотографій. У типовому випадку складність і практичність питання виключають установку вимірювального обладнання таким чином, щоб лінійний рух об'єкта обмежувався однією заздалегідь визначеною віссю. Як наслідок, у типовому випадку необхідно враховувати рух об'єкта як у горизонтальному, так і у вертикальному напрямках, а також відстань або близькість об'єкта до вимірювального обладнання. Вигідно мінімізувати потенційну відстань або наближення маркера, можливим прикладом цього є правильне встановлення вимірювального обладнання на маркері [7].

Зрештою, сучасний стан техніки і технологій, зокрема наукових досліджень, характеризується широкомасштабним використанням інформаційних технологій із різноманітним застосуванням [8].

Сучасні тенденції інформатизації завдань на основі обчислювальної техніки набувають особливого значення в різних сферах життя людини. Цифрові фото та відео-технології відіграють визначну роль. Значні досягнення в цій галузі з технічної, технологічної та економічної точки зору призвели до широкого використання цифрових камер у науці та промисловості. Перевагами цифрової фотографії сьогодні є прості, невеликі розміри, недорогі пристрої, які використовуються, результати можна отримати швидко, перегляд простий, зв'язок швидкий та ефективний, необхідні та зручні форми представлені у бажаному вигляді. Особливо широкі можливості існують для використання технологій цифрової фотографії завдяки високотехнологічним попередньо налаштованим рішенням, які оснащені вбудованими апаратними та програмними інтерфейсами, наявність яких полегшує інтеграцію та адаптацію систем до будь-яких цілей. на основі технології.

Кількісна фотографія насправді є непрямим методом вимірювання. Метод вимірювання лінійного руху на основі аналізу зображення є непрямим, оскільки

дослідник починає з фіксованого набору об'єктів, які складають зону зйомки, ці об'єкти потім відтворюються як фотодрук або зображення на моніторі.

З теоретичної точки зору, розміри цифрового зображення задаються в пікселях. Спосіб вирішення задачі визначення розмірного розміру окремого пікселя в лінійних одиницях полягає у використанні інформації про умови зйомки, розмірний розмір ділянки або еталонні об'єкти з апріорно відомими розмірами [9].

Цифрова фотографія насправді є кількісною за своєю суттю, оскільки складається з певної кількості пікселів, це число зручно для використання методів вимірювання лінійного руху на основі результатів аналізу зображення. Широке застосування методу вимірювання лінійних переміщень на основі аналізу зображень безпосередньо пов'язане з двома конкретними явищами:

- по-перше, цьому методу сприяє кількісний і саме цифровий підхід до фотографії;
- для визначення розміру об'єктів на цифровому зображенні необхідна невелика кількість додаткової інформації щодо умов отримання, ця інформація включає вимірювання лінійних рухів за допомогою аналізу зображення, яке можна виконати пізніше, після достатнього, можливо, тривалого періоду часу.

Інформатизація наукових досліджень і промисловості, включаючи фотографію, дозволяє використовувати сучасні комп'ютерні технології для обробки і автоматизації процесів спостереження і вимірювання в найширшому сенс термінів. Крім того, при реалізації методу вимірювання лінійних рухів на основі аналізу зображення, цей метод в основному реалізується за допомогою аналізу зображення.

Зазначені практичні зручності обумовлені відсутністю додаткових кроків щодо керування пристроєм та отримання первинної інформації, ця інформація одразу представлена у багатоваріантній, доступній та практичній формі. Це означає, що цифрова фотографія та цифрові технології, пов'язані з фотографією, є доступними та практичними для встановлення методу вимірювання лінійного руху за допомогою аналізу зображення.

Аналіз предметної області демонструє поширеність і об'єктивну необхідність різноманітних вимірювань, більшість з яких є непрямими. Впровадження інформаційних технологій є однією з найбільш значущих тенденцій підвищення точності та надійності, розширення сфери застосування різноманітних методів вимірювання з урахуванням сучасних і перспективних потреб виробництва і науки [10].

1.2 Аналіз методів визначення положення об'єктів за аналізом зображень з круговим градієнтом кольору

Суперечки навколо вимірювання лінійних переміщень мають давню історію, але питання залишається актуальним, насамперед через його важливість щодо точності, засобів і методів.

У результаті обговорюються основні методи і засоби вимірювання електричних, магнітних і немеханічних величин, а також процедури оцінки точності вимірювань. Як приклад різноманітності та широти вимірювань у загальному розумінні цього терміну було розпочато проект, присвячений тригонометричному та геометричному нівелюванню, а також розробці методів та рекомендацій для підвищення їх точності.

Однією з найпопулярніших тенденцій сучасного вимірювання є значна роль інформаційних технологій. Одним з життєздатних рішень проблеми вимірювання за допомогою цього підходу є непрямі методи, які базуються на обробці зображень.

Наприклад, науково-технічна мета підвищення ефективності методів вимірювання може бути реалізована шляхом оптико-електронного позиціонування та пасивних стереоскопічних вимірювань дальності завдяки розробці та впровадженню перешкодостійких методів обробки оптичного потоку.

У проекті параметри переміщень об'єктів розраховуються за алгоритмічною обробкою їх зображень, зокрема запропоновано два способи

визначення розмірів геометричної натури - шляхом виділення контурів, що переходять у символічний опис зображення, або шляхом сегментації зображення за рівнем яскравості та визначенням координат точки на контурі об'єкта. У результаті прогресування цих зображень дозволяє ідентифікувати лінійні та кутові рухи [9].

Інша важлива сфера, яка потребує, з одного боку, результатів вимірювань, а з іншого - первинної інформації, а саме зображень, представлена в системах технічного зору.

У підсумку в статті описані галузі техніки і науки, які активно використовують різні системи технічного зору. У статті викладено основні підходи до використання таких систем, запропоновано нову формулу для розрахунку відстані між об'єктами та наведено її аналіз [11].

Реалізація вимірювальних систем, які передбачають використання зображень, потребує використання певних алгоритмів цифрової обробки зображень і комп'ютерного зору.

У результаті проект містить дослідницькі матеріали щодо створення структурних методів розпізнавання спостережуваних об'єктів на відповідних зображеннях у контексті геометричних перетворень за наявності інтерференції. У цій статті описано сучасні підходи до створення структурного опису зображення, створення структур і асоціацій ознак, обчислення мір подібності між описами, використання методів обробки зображень для прискорення процесу або підвищення його точності при вирішенні практичних завдань, які включають аналіз зображення. .

Дослідження сучасних наукових публікацій на тему вимірювання лінійних переміщень за допомогою аналізу зображень підтверджує можливість розробки багатьох методів вимірювання, зазначений підхід є ефективним і має високу ступінь застосовності.

1.3 Аналіз методів оцінки зміщення об'єктів за аналізом зображень

Методики оцінки зміщення об'єктів за аналізом зображень є досить різноманітними, оскільки вони можуть використовувати різні алгоритми та техніки обробки зображень.

Оцінка зміщення зображень за допомогою крос-кореляції. Метод використовується для визначення різниці між двома зображеннями, які були зсунуті одне відносно іншого. Цей метод базується на математичній операції, відомій як крос-кореляція, яка використовується для порівняння двох сигналів [12].

Крос-кореляція зображень полягає в тому, що спочатку одне зображення зсувається на певну відстань вліво або вправо відносно другого зображення. Потім виконується крос-кореляція між зсунутим і незсунутим зображеннями, що дає значення, що характеризує подібність між зображеннями на кожній позиції.

Цей метод може бути корисним для багатьох задач обробки зображень, таких як реєстрація зображень, розпізнавання об'єктів на зображеннях, визначення руху та інших.

У випадку з зображеннями, крос-кореляція виконується піксель за пікселем, якщо зображення має 1-канальну (чорно-білу) градацію кольорів, або за кожним каналом RGB, якщо зображення має кольорову гаму.

Крос-кореляцію між двома зображеннями можна виконати за допомогою вбудованих функцій обробки зображень, наприклад, функції `cv2.matchTemplate` в бібліотеці OpenCV.

Операція крос-кореляції відбувається в декілька етапів.

1. Створення шаблону: на початку обирається одне з зображень як шаблон, а інше зображення вважається вхідним зображенням, для якого ми шукаємо зсув.

2. Створення матриці крос-кореляції: матриця розміром $(W - w + 1, H - h + 1)$, де W та H - розміри вхідного зображення, а w та h - розміри шаблону. Кожен елемент матриці - це значення крос-кореляції між вхідним зображенням і

шаблоном для конкретного зсуву.

3. Обчислення крос-кореляції: застосовується фільтр крос-кореляції на вхідному зображенні та шаблоні для всіх можливих зсувів.

4. Пошук позиції з максимальною крос-кореляцією: знайти позицію в матриці крос-кореляції з найбільшим значенням, що вказує на те, що відповідний зсув є найбільш ймовірним зсувом між двома зображеннями.

5. Оцінка зсуву: розрахування зсуву між двома зображеннями, що відповідає позиції з максимальною крос-кореляцією в матриці крос-кореляції.

Цей метод застосовується у багатьох областях, таких як комп'ютерний зір, обробка зображень, медична діагностика тощо. Наприклад, він може бути використаний для автоматичного знаходження об'єктів на зображеннях, визначення руху на відео, виправлення перспективи та інших завдань обробки зображень.

Оцінка зміщення зображень за допомогою крос-кореляції може бути особливо корисною в ситуаціях, коли необхідно порівняти два зображення, що містять один і той же об'єкт, але зняті зі зсувом або з різних кутів. Знайдений зсув може бути використаний для виправлення зображення, таким чином, що об'єкт буде розміщений на одній і тій же позиції на обох зображеннях.

Проте, метод оцінки зміщення зображень за допомогою крос-кореляції має свої обмеження. Наприклад, він може не працювати ефективно в разі, коли зображення містять значні різноманітності або мають складну текстурну структуру. Крім того, метод може бути відчутним до шумів або змін в освітленні на зображенні.

У цілому, оцінка зміщення зображень за допомогою крос-кореляції є потужним і простим методом для визначення зсуву між двома зображеннями, і вона знаходить широке застосування в багатьох галузях, пов'язаних з обробкою зображень.

Оцінка зміщення зображень за допомогою блочної обробки. Метод використовується для визначення відносного зміщення двох зображень з однаковими об'єктами, отриманих з різних джерел або під різними умовами. Цей

метод є ефективним для знаходження піксельного зміщення в дослідженнях з обробкою зображень, включаючи відстеження руху об'єктів, визначення різниці між зображеннями, розпізнавання образів тощо [13].

Оцінка зміщення зображень за допомогою блочної обробки передбачає поділ кожного зображення на блоки з однаковими розмірами, наприклад, 8 на 8 пікселів. Для кожного блоку зображення здійснюється перехід до зображення з іншого джерела, і знаходиться блок з найбільшою кореляцією пікселів. Кореляція визначається як коефіцієнт Пірсона між значеннями пікселів в двох блоках.

Потім для кожного блоку знаходиться вектор зміщення, що вказує на різницю між центром блоку у першому зображенні та блоком з найвищою кореляцією в другому зображенні. Ці вектори зміщення можуть бути використані для створення візуальної карти зміщення, яка показує рівень зміщення між двома зображеннями у різних частинах.

Цей метод дозволяє визначити навіть малі зміщення між зображеннями, і може бути ефективним для різноманітних завдань обробки зображень. Однак, метод може бути обмеженим у випадку, коли об'єкти на зображеннях занадто відмінні або коли на зображеннях присутні великі затемнення.

Крім того, метод оцінки зміщення зображень за допомогою блочної обробки може бути чутливим до вибору розміру блоків та методу обчислення кореляції. Наприклад, неправильний вибір розміру блоків може призвести до великих похибок в знаходженні векторів зміщення, тоді як невірний вибір методу обчислення кореляції може призвести до погіршення якості оцінки.

Для покращення точності оцінки зміщення за допомогою блочної обробки можна використовувати різні підходи, такі як вибір оптимального розміру блоків, використання більш складних методів кореляції, врахування геометричних перетворень та ін. Також, існують альтернативні методи оцінки зміщення зображень, такі як методи зіставлення функцій, методи оптимізації та інші, які можуть бути більш ефективними в певних ситуаціях.

Застосування методу оцінки зміщення зображень за допомогою блочної обробки є широким, він використовується в різних областях, таких як медична діагностика, астрономія, відео-нагляд, комп'ютерне зорове спостереження та інші.

Оцінка зміщення зображень за допомогою аналізу оптичного потоку. Метод є технікою обробки зображень, яка використовується для визначення руху об'єктів на відео [14].

Оптичний потік - це швидкість руху пікселів на зображенні, який можна визначити шляхом вимірювання зміни яскравості пікселів у відповідь на зміну часу. Зміна яскравості пікселів може бути викликана зміною положення об'єкта на зображенні або зміною освітлення.

Метод оцінки зміщення зображень на основі аналізу оптичного потоку складається з таких кроків:

1. Визначення оптичного потоку: Для кожної пари кадрів відео вимірюється зміна яскравості пікселів, щоб визначити швидкість руху об'єктів на зображенні
2. Розрахунок відхилень: Використовуючи отримані значення оптичного потоку, розраховуються відхилення об'єктів на зображенні відносно попереднього кадру.
3. Обробка відхилень: Отримані відхилення піддаються фільтрації для видалення шуму та інших випадкових артефактів.
4. Оцінка зміщення: За допомогою оброблених відхилень визначається зміщення об'єктів на зображенні.

Метод оцінки зміщення зображень за допомогою аналізу оптичного потоку застосовується в багатьох галузях, таких як відеоспостереження, робототехніка, комп'ютерний зір, медична діагностика та інші.

1.4 Аналіз існуючих підходів оцінки зміщення об'єктів

Існує кілька реальних прикладів використання методів оцінки зміщення об'єктів за аналізом зображень.

1. Візуальне слідкування джойстика для дистанційного управління: цей метод використовує кореляційні методи для відстеження позиції джойстика на зображенні та визначення зміщення. Він застосовується в системах дистанційного управління, таких як квадрокоптери, щоб дозволити користувачеві контролювати політ зі свого мобільного пристрою.

2. Визначення швидкості руху об'єктів на зображеннях моніторингу трафіку: цей метод використовує оптичний потік для визначення швидкості руху автомобілів на зображенні моніторингу трафіку. Він застосовується для збору даних про рух транспорту на дорогах та для розрахунку часу подорожі

3. Визначення розміру та форми раковин на зображеннях раків: цей метод використовує алгоритми глибинного навчання для визначення розміру та форми раковин на зображеннях. Він застосовується в морській біології для дослідження популяцій раків та визначення їх розміру та стану.

4. Визначення руху камери для стабілізації відео: цей метод використовує інерціальні датчики для визначення зміщення та руху камери на зображенні та стабілізації відео. Він застосовується в галузі кіно та відеозйомки для зменшення ризику руху камери та забезпечення якості зображення.

5. Відстеження м'яча в тенісі: цей метод використовує алгоритми комп'ютерного зору та машинного навчання для відстеження м'яча на зображенні та визначення його траєкторії. Він застосовується в тенісі для аналізу гри, оцінки техніки гравців та для покращення навичок тренувань.

6. Визначення зміщення об'єктів в медичних зображеннях: цей метод використовує алгоритми обробки зображень для визначення зміщення об'єктів в медичних зображеннях, таких як зображення МРТ або КТ. Він застосовується

для діагностики та лікування захворювань, таких як рак, деменція та інші захворювання.

7. Відстеження об'єктів на складах та в логістичних центрах: цей метод використовує комп'ютерний зір для відстеження руху товарів та обладнання на складах та в логістичних центрах. Він застосовується для автоматизації та управління логістичними процесами та для підвищення ефективності логістичних операцій.

8. Визначення зміщення об'єктів на зображеннях відеоспостереження: цей метод використовує алгоритми комп'ютерного зору для відстеження руху та зміщення об'єктів на зображеннях відеоспостереження. Він застосовується для забезпечення безпеки та захисту в громадських місцях, таких як аеропорти, стадіони, торгові центри та інші.

1.5 Мета, задачі та вимоги до реалізації інформаційної системи

Аналіз, який було проведено в предметній області, є великим і демонструє необхідність різноманітних непрямих вимірювань неелектричних величин. Значна частина обсягу приписується лінійним рухам, які є непрямыми. Впровадження інформаційних технологій є однією з найбільш значущих тенденцій підвищення точності та надійності, розширення сфери застосування різноманітних методів вимірювання з урахуванням сучасних і перспективних потреб виробництва і науки.

Дослідження сучасних наукових публікацій на тему вимірювання лінійних переміщень за допомогою аналізу зображень підтверджує можливість розробки багатьох методів вимірювання, зазначений підхід є ефективним і має високу ступінь застосовності.

Дослідження, які використовують інформацію для підтримки вимірювання лінійних рухів за допомогою аналізу зображень, демонструють специфічний характер технології та критичну потребу в адаптації для вирішення

конкретних проблем і задоволення існуючих потреб у різних сферах людської діяльності.

Враховуючи вищесказане, слід розробити метод вимірювання лінійного руху за допомогою аналізу зображення.

Для досягнення наміченої мети необхідно виконати наступні завдання:

- визначити порядок, у якому використовується метод аналізу лінійного руху через аналіз зображення;
- визначити склад інформаційних потоків, пов'язаних із методом вимірювання лінійного руху шляхом аналізу зображення;
- визначити чіткі функціональні відмінності між програмно-апаратними та програмно-апаратними комплексами, пов'язаними з реалізацією методу вимірювання лінійного руху за допомогою аналізу зображення;
- визначити призначення програмної складової методу вимірювання лінійного руху за допомогою аналізу зображення;
- вирішити питання визначення розмірів візуальних об'єктів у лінійних метричних одиницях;
- створити математичну формулу, яка співвідносить лінійні рухи з аналізом зображення;
- Завершити програмну складову методу вимірювання лінійного руху за допомогою аналізу зображення.

Розділ 2 Спосіб оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах

2.1 Спосіб визначення положення об'єктів за аналізом зображень з круговим градієнтом кольору

Комплексна методологія вимірювання лінійних переміщень на основі аналізу зображень дозволяє отримувати інформацію про досліджуваний процес або об'єкт спостереження безпосередньо, зберігати, обробляти та аналізувати її (рис. 2.1).

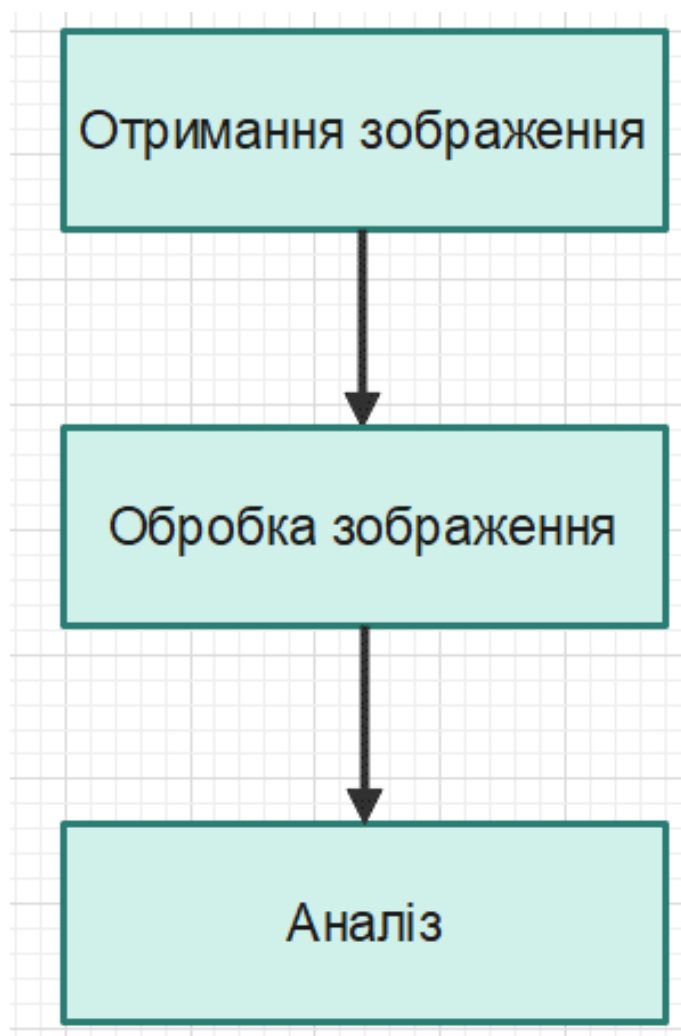


Рисунок 2.1 - Опис загальної схеми використання методу аналізу зображень для вимірювання лінійних переміщень

Отримати зображення можна різними способами (рис 2.2).

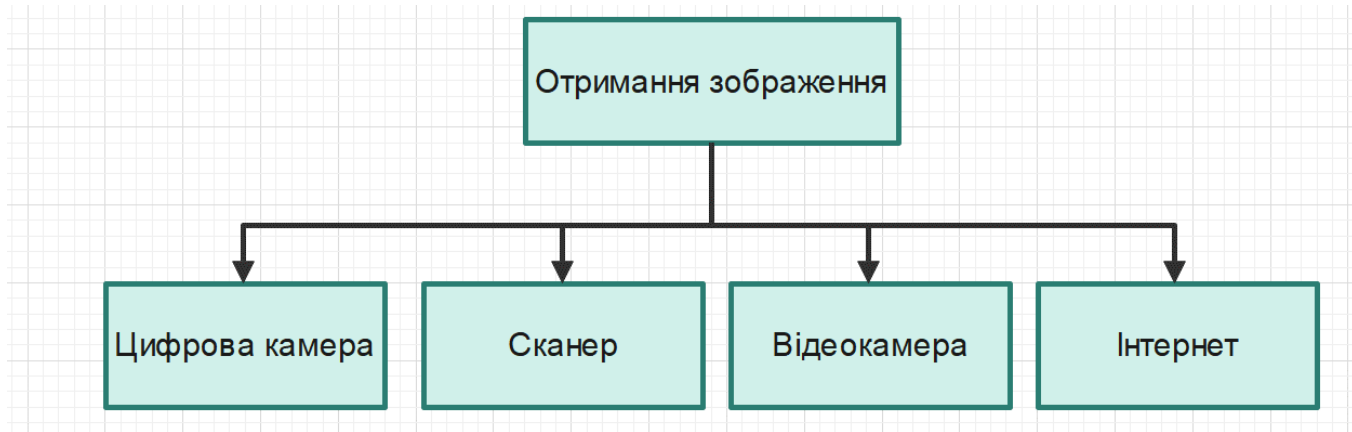


Рисунок 2.2 - Способи отримання зображення

Зокрема за допомогою цифровою відеокамери, яка знімає зображення використовуючи світлочутливий сенсор та перетворює його на цифрові дані, що можуть бути збережені на пам'яті камери або передані на комп'ютер. Ще одним чудовим способом отримання зображення є сканер. Сканер також, перетворює зображення з фізичного виду до цифрового виду, тобто до цифрових даних за допомогою сканування. Він використовує для цього спеціальне світло та детектор. Зображення також можна отримати за допомогою відеокамери, що знімає та фіксує зображення в режимі реального часу і знову ж таки перетворює його у цифрові дані. Ці дані зберігаються у пам'яті відеокамери і надалі передаються комп'ютеру. І останні, популярний спосіб отримання зображення є Інтернет. Зображення можна завантажити з Інтернету або бути передані через мережу за допомогою різних програм, таких як пошта або меседжери.

Після отримання зображення, вони обробляються за допомогою спеціального програмного забезпечення, яке має відповідне призначення та може редагувати ці зображення. Наприклад збільшити їх роздільну здатність або зовсім змінити формат.

Наступним етапом, відповідно до рис. 2.1 це є обробка зображень.

Відповідно, обробка зображення складається з двох етапів (рис. 2.3).

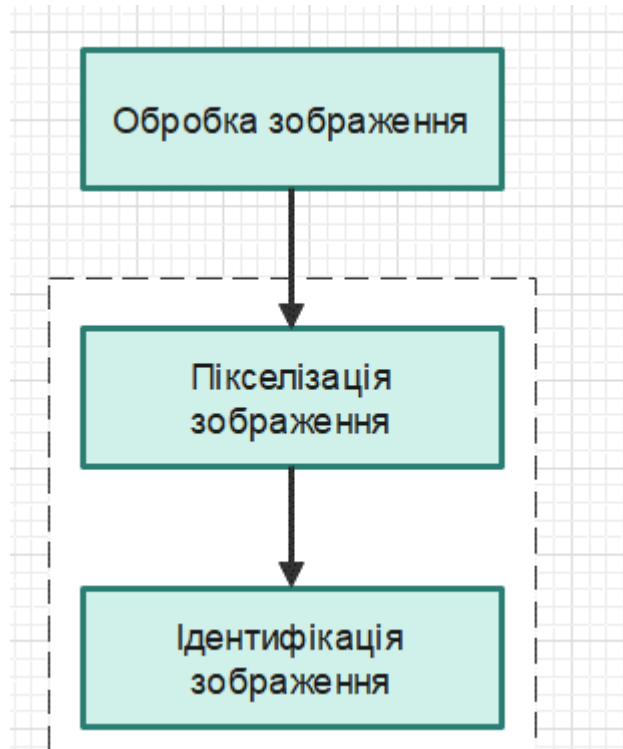


Рисунок 2.3 - Алгоритм обробки зображення

Власне, пікселізація зображення є процесом заміни реального зображення його дискретним аналогом. Зменшення роздільної здатності зображення шляхом заміни певних пікселів самого зображення більшими блоками пікселів. Даний процес використовується для зменшення розміру файлу зображення та для створення ефекту старовинного зображення.

Для того, аби провести пікселізацію зображення, необхідно спочатку визначити розмір нових блоків пікселів, на які буде поділене зображення. Це робиться за допомогою конкретного параметра, що має назву розмір ядра. Власне, ядро у нашому випадку - це квадратна матриця, розмір якої задається як $n \times n$, де n - є довжиною сторони ядра.

Тож, далі для кожного блоку пікселів, що визначається розміром ядра, необхідно змінити кожен піксель у блоку на середнє значення яскравості усіх пікселів в цьому блоці. При умові, коли блок має розмір 3×3 , тоді для кожного блоку пікселів буде виконуватися наступний алгоритм (рис. 2.4).

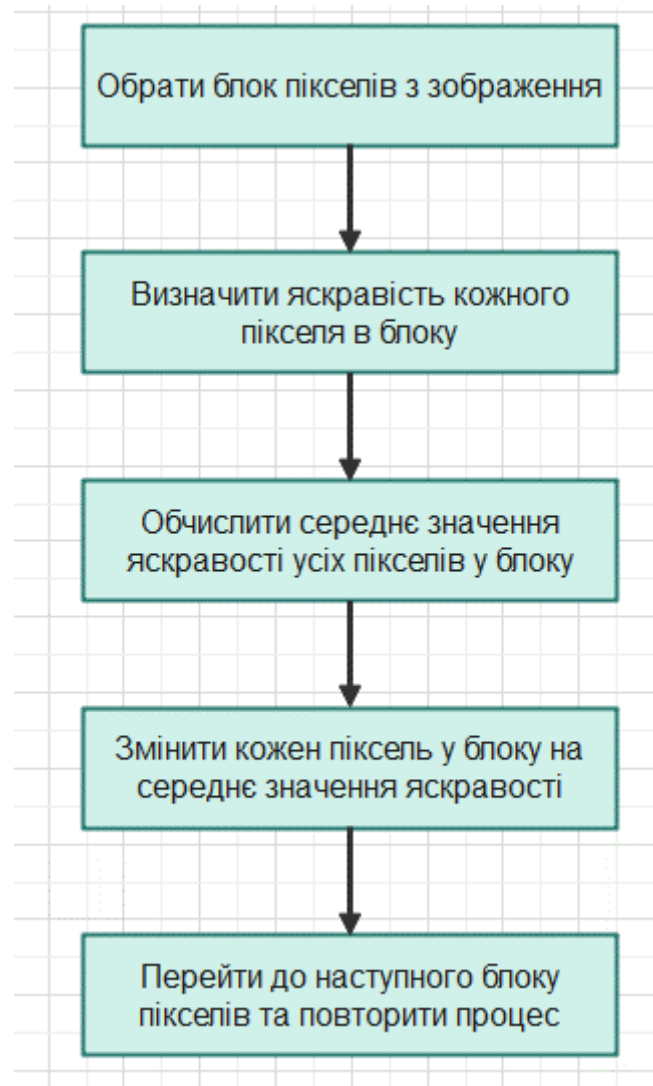


Рисунок 2.4 - Алгоритм роботи блоку при розмірі ядра 3x3

Також, математично це буде виглядати наступним чином:

$$Y(i, j) = \frac{1}{n^2} * \sum \sum X(k, l) \quad (2.1)$$

де, $Y(i, j)$ - яскравість пікселя в пікселізованому зображенні з координатами i та j , $X(k, l)$ - яскравість пікселя в вихідному зображенні з координатами k та l , а n це є розмір ядра. Власне, сумування проводиться по всіх пікселях блоку з координатами від $(\frac{i-n}{2}, \frac{j-n}{2})$ до $(\frac{i+n}{2}, \frac{j+n}{2})$. Необхідно звернути увагу, що значення n зазвичай є непарним, для того, аби забезпечити центрування ядра навколо

кожного пікселя у зображенні. Дана формула розраховує середнє значення яскравості усіх пікселів в блоку, розмір ядра якого становить $N \times N$ і змінює кожен піксель блоку на дане середнє значення. Цей процес буде повторюватися для кожного блоку пікселів у зображенні, яке призводить до пікселізованої версії вихідного зображення.

Ідентифікація зображення на основі пікселізації полягає у порівняння пікселізованих версій двох зображень. Для цього спочатку необхідно пікселізувати кожне з зображень.

Далі, для порівняння двох пікселізованих зображень можна використовувати такі методи як вимір різниці між кожними пікселями в двох зображеннях, обчислення середнього та максимального значення різниці.

Якщо значення різниці між двома пікселізованими зображеннями є незначним, тоді можна вважати, що зображення ідентичні. Але якщо різниця значень є великою, це може свідчити про те, що зображення є досить різними.

Отже, переходимо до етапу аналізу зображень. Власне, аналіз зображення складається з двох етапів (рис. 2.5).

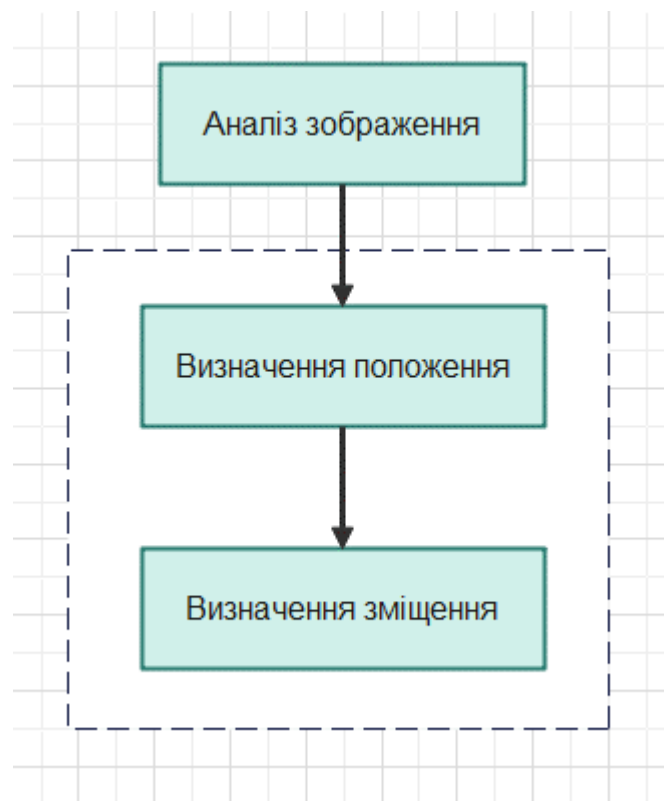


Рисунок 2.5 - Послідовність роботи аналізу зображення

Відповідно до схеми на рисунку 2.5, для початку необхідно визначити положення.

Положення об'єкту на зображенні визначається як множина положень пікселів, що його складають (рис. 2.6).

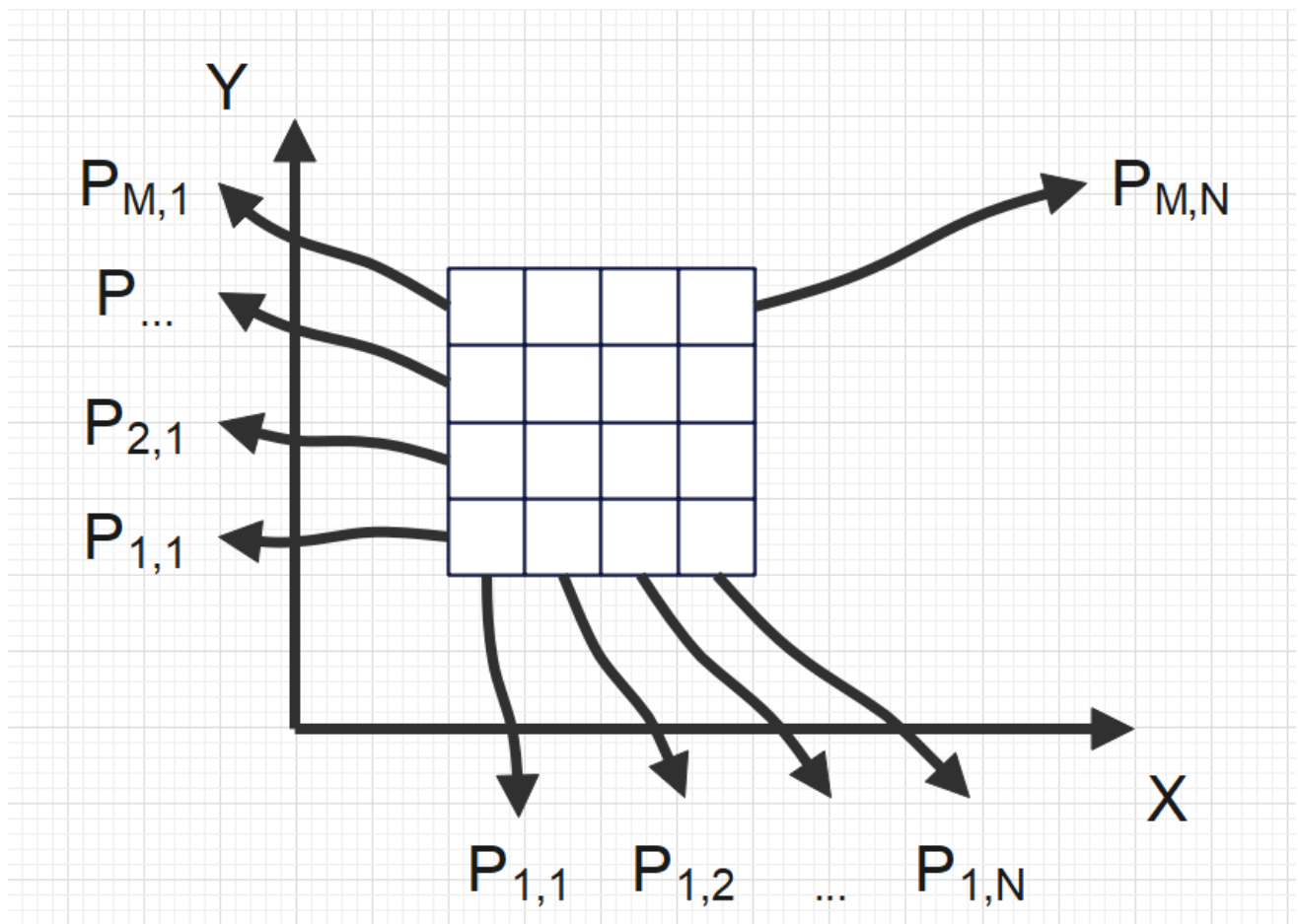


Рисунок 2.6 - Положення об'єкту спостереження, як множина пікселів

Отже, положення об'єкту спостереження визначається, як положення його геометричного центру, координати якого визначається за наступною формулою:

$$P_{i,j} = f(X_{i,j}, Y_{i,j}), \quad (2.2)$$

Положення геометричного центру об'єкта спостереження визначається як середнє арифметичне координат окремих пікселів, що складають це зображення.

У двовимірному просторі слід визначити дві координати об'єкту за відповідними осями:

Координата по осі x визначається так:

$$X_{oc} = \frac{I}{N} \sum_{i=1}^N X_{ij} \quad (2.3)$$

Координата по осі y визначається так:

$$Y_{oc} = \frac{I}{M} \sum_{j=1}^M Y_{ij} \quad (2.4)$$

Власне, як працює визначення положення. Спочатку маємо певний об'єкт (рис. 2.7).

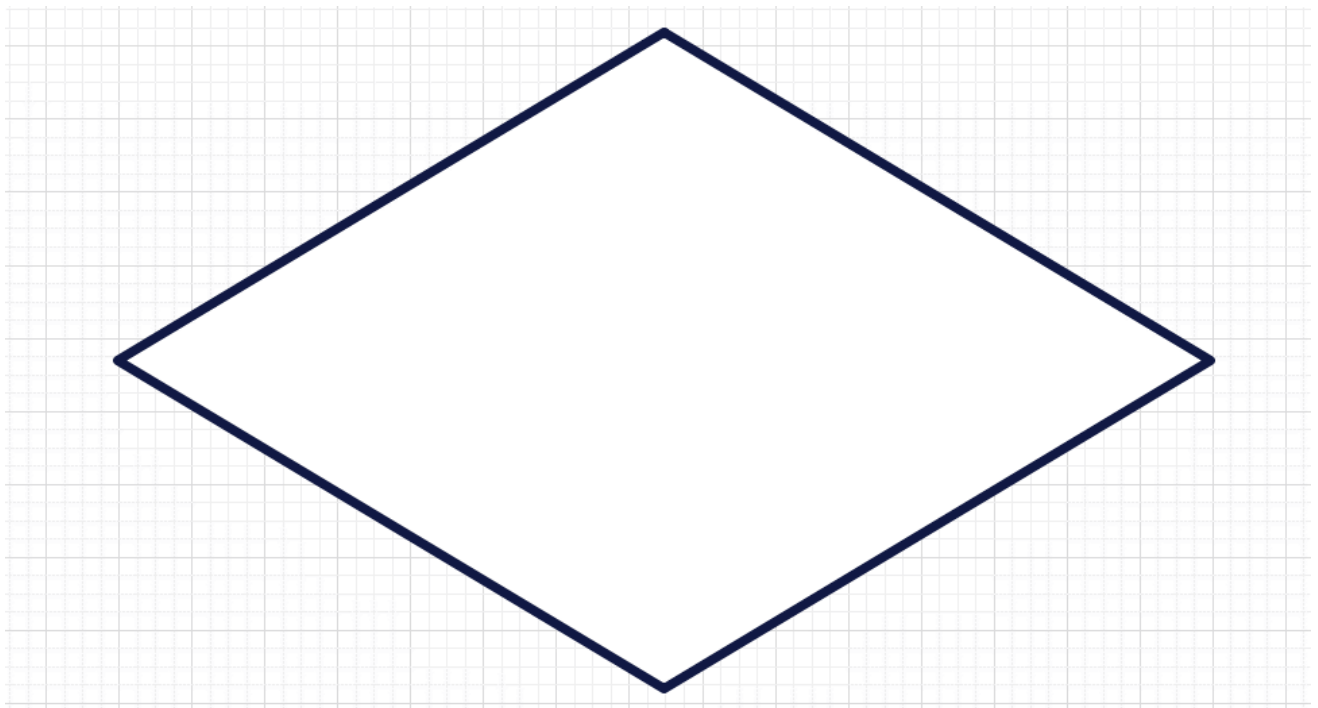


Рисунок 2.7 - Певний об'єкт спостереження

Розбиваємо певний об'єкт спостереження на пікселі, як продемонстровано на рис. 2.8.

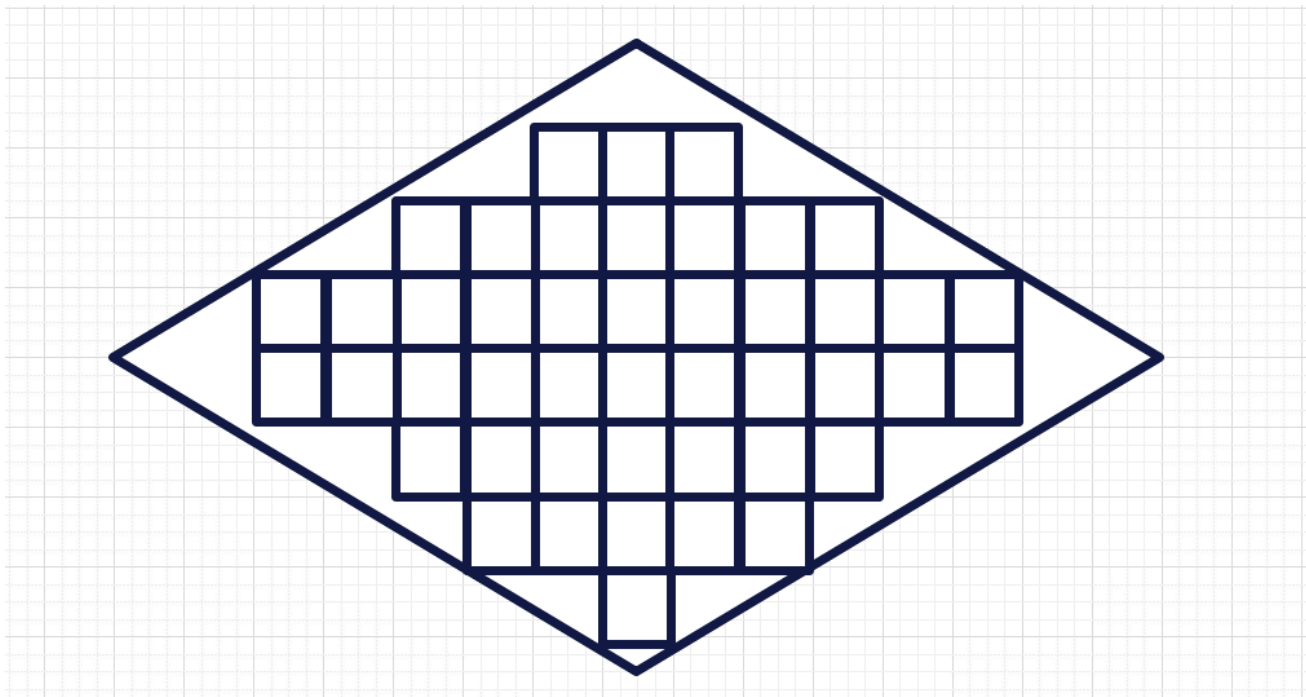


Рисунок 2.8 - Розбиття об'єкта спостереження на пікселі

Після розбиття об'єкта спостереження на пікселі, необхідно прибрати кордони об'єкта спостереження, як продемонстровано на рис. 2.9.

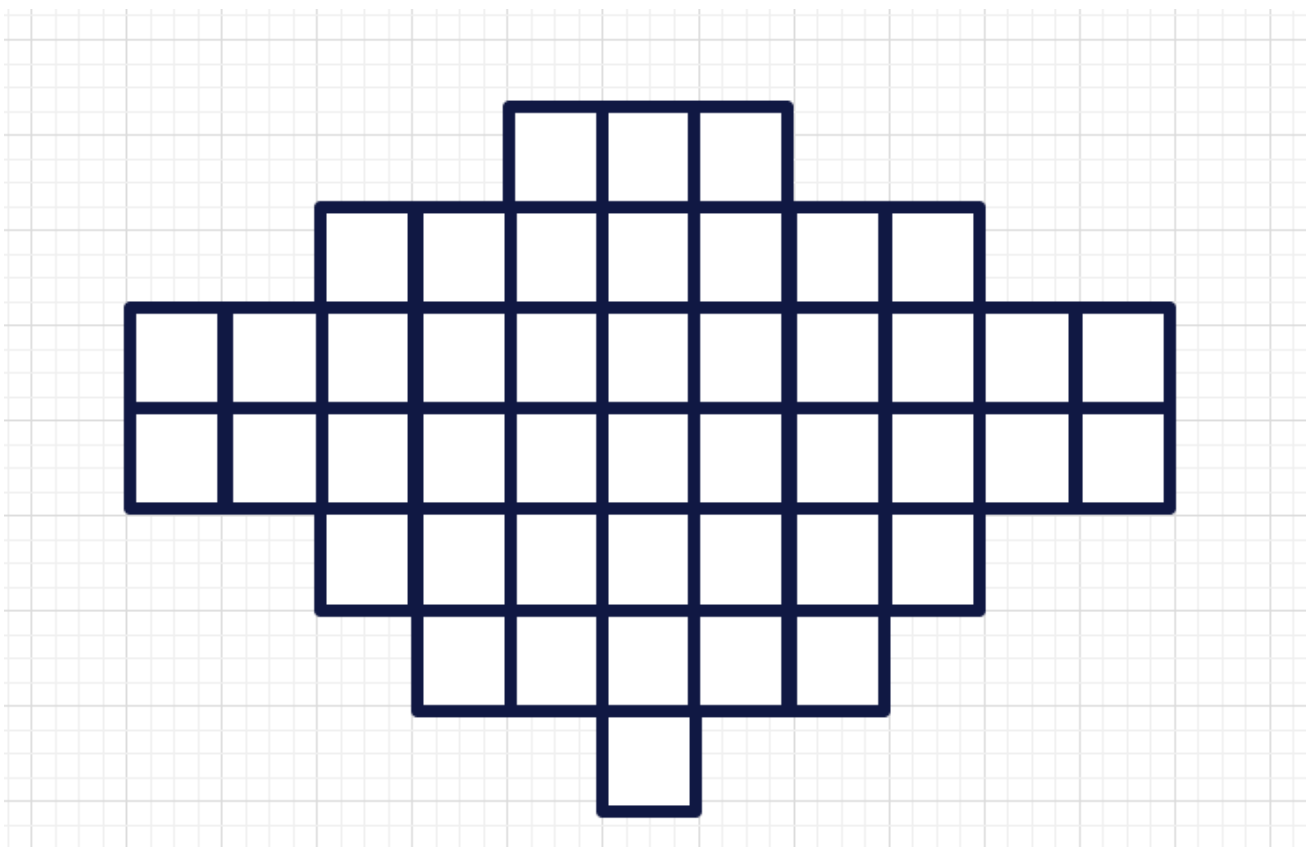


Рисунок 2.9 - Об'єкт спостереження без кордонів

Операція визначення одиничного зміщення одиничного об'єкту спостереження полягає в опрацюванні пари зображень (рис. 2.10).

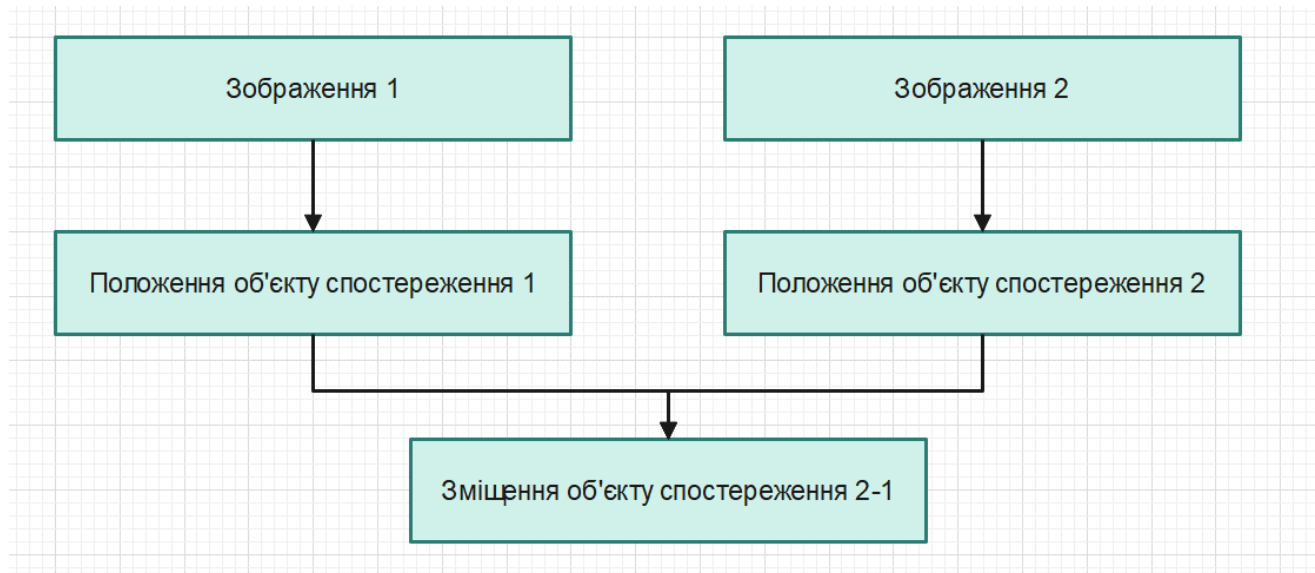


Рисунок 2.10 - Операція визначення одиничного зміщення об'єкту

В загальному випадку ця операція може бути застосована для довільної кількості зображень (рис. 2.11).

Операція визначення зміщення одиничного об'єкту для довільної кількості зображень складається з трьох етапів. Власне, першим етапом необхідно отримати зображення, спочатку виконується фіксація зображення, після чого необхідно виконати дискретизацію зображення. У другому етапі визначається визначення положення об'єкту спостереження. Необхідно визначити положення всіх об'єктів спостереження. Після визначення положення всіх об'єктів переходимо до третього етапу, де розраховується зміщення об'єкту спостереження. Коли у третьому зображенні отримується два зображення, визначається зміщення за допомогою порівняння координат визначених об'єктів. Таким чином визначається зміщення для довільної кількості об'єктів.

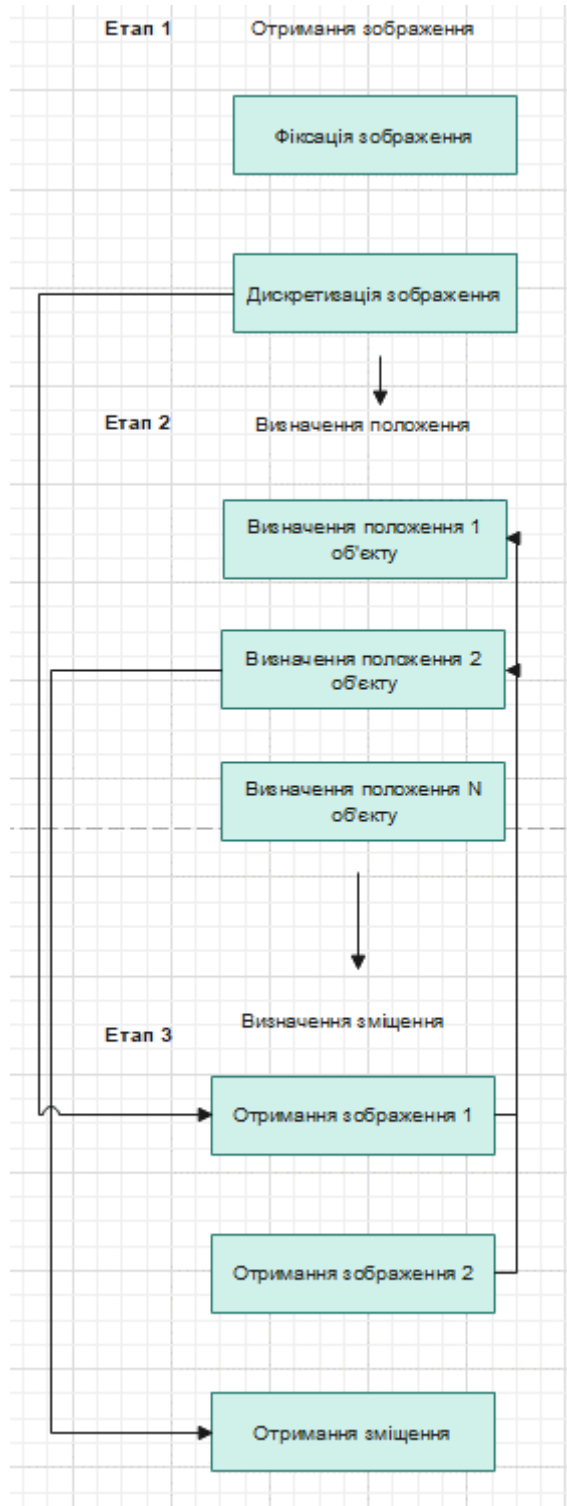


Рисунок 2.11 - Операція визначення зміщення одиничного об'єкту для довільної кількості зображень

Таким чином, загальна послідовність визначення зміщення одиничного об'єкту спостереження представлено на рис. 2.12.

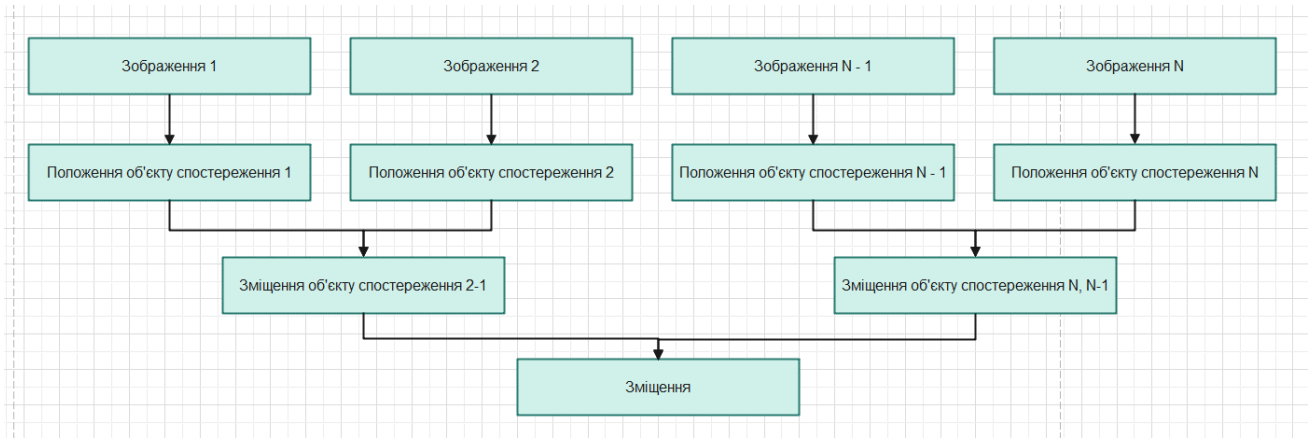


Рисунок 2.12 - Операція визначення зміщення для довільної кількості парних зображень

В загальному випадку при наявності на зображенні кількох об'єктів спостереження з не синхронізованими зміщеннями, зміна положення кожного з них визначається шляхом застосування наведеною вище послідовності до кожного з цих об'єктів окремо.

2.2 Спосіб оцінки зміщення об'єктів за аналізом зображень

Зміщення об'єкту спостереження визначається як зміна розташування цього об'єкта на серій послідовних зображень. Першочерговим завданням є визначення положення об'єкту на окремому зображенні. Ця процедура може бути масштабована на кілька об'єктів спостереження. Визначення положення одразу кількох об'єктів спостереження на одному зображенні може бути реалізовано за допомогою методів кластеризації.

Алгоритм кластеризації k-means (k-means clustering) - це техніка машинного навчання, яка використовується для кластеризації подібних об'єктів у класи на основі їхніх властивостей і характеристик. В алгоритмі k-середніх кожен кластер представлений своїм центроїдом, який є точкою, яка є середнім значенням об'єктів у кластері.

Мета кластеризації k-середніх полягає в тому, щоб розділити набір даних на підмножини (кластери), кожна з яких містить об'єкти, схожі один на одного, і

об'єкти, які не схожі один на одного. Кластеризація k-середніх використовується в різних галузях, включаючи медицину, біологію, маркетинг тощо.

При використанні методу k-середніх для реалізації способу оцінки зміщення об'єктів за аналізом зображень кластером є окремий об'єкт спостереження на цьому зображенні. Застосування кластеризації дозволяє визначити положення та, в подальшому, зміщення одразу кількох об'єктів спостереження з не синхронізованим рухом.

Розглянемо блок-схему на рис. 2.13 що демонструє роботу кластеризації k-середніх

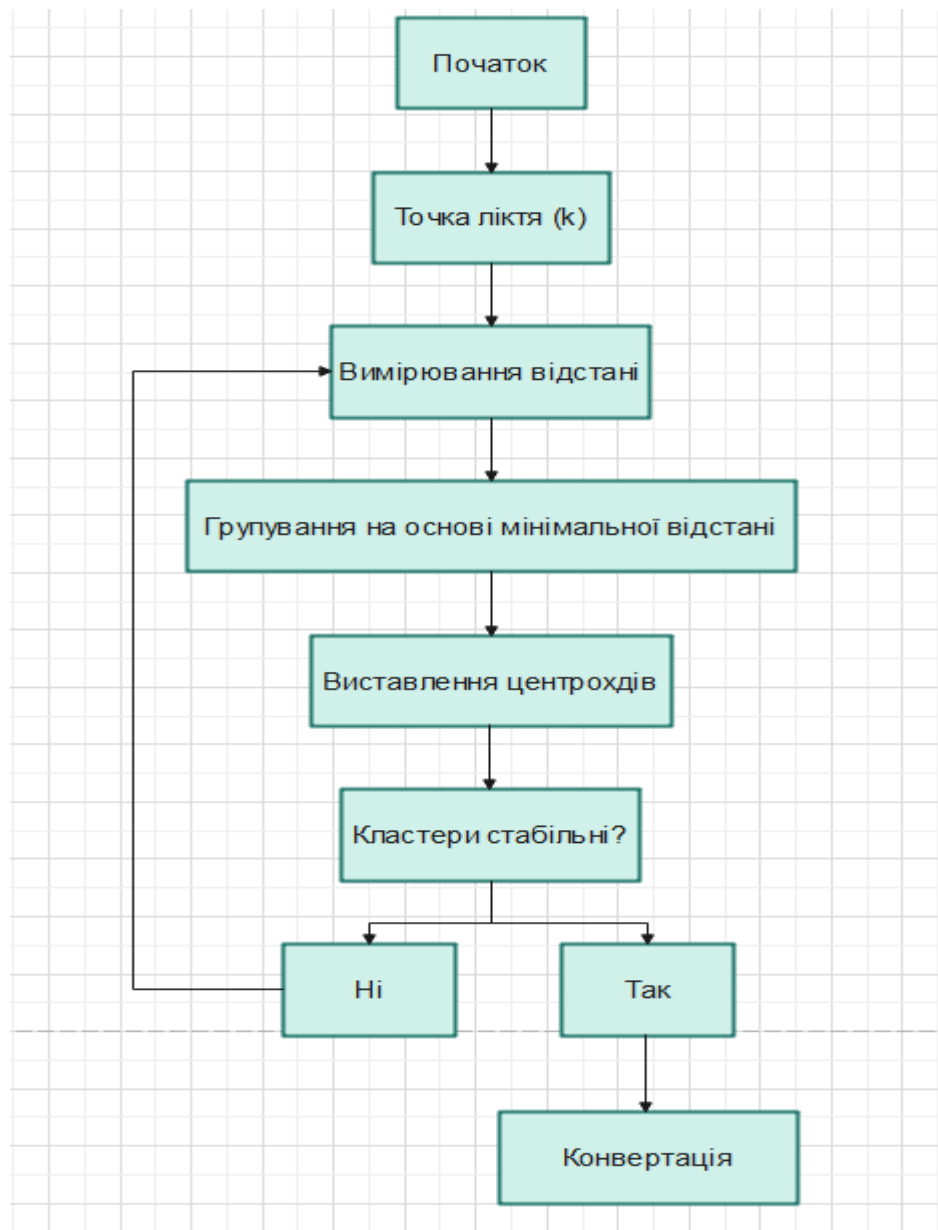


Рисунок 2.13 - Послідовність кластеризації за методом k-середніх

Головною метою кластеризації k -середніх це пошук кластерів у заданих вхідних параметрах. Існує кілька способів цього досягти. Можна використовувати метод через випробування, та отримувати безліч помилок і саме на цих помилках вчитися, вказавши значення K , наприклад, як 5, 6 або 7. І по мірі просування постійно змінювати відповідний параметр до того моменту, поки не буде отримано найкращий результат, а саме, найкращі, так звані, кластери.

Наступний метод використовує техніку Elbow. Цей метод полягає у визначенні значення K . Тож, коли система отримує значення K , воно випадковим чином призначає відповідну кількість центроїдів та визначає відстань кожної точки даних один від одних центроїдів. Таким чином, у кожного центроїда є свої точки. Відстань між точками та цим одним центроїдом є мінімальною. В результаті, усі точки призначені конкретного та відповідному центроїду, і призначені з мінімальною відстанню. На цьому етапі, отримано K кількість початкових, так званих, кластерів.

Для новоутворених кластерів визначається нове центральне положення. Положення центру тяжіння зміщується порівняно з випадковим розподілом.

Знову ж таки, відстань кожної точки обчислюється від нової точки центроїда. Якщо необхідно, точки даних переміщуються на нові центроїди, а середнє положення або новий центроїд знову обчислюється.

Якщо центроїд зміщується, ітерація продовжується, що вказує на те, що конвергенції не відбулося. Однак, як тільки центроїд перестане дрейфувати (що означає, що процес кластеризації завершено), результат буде відображено.

Щоб знайти оптимальну кількість кластерів, необхідно скористатися наступними кроками:

Крок 1.

Використовуємо метод Elbow, як спосіб знаходження кількості кластерів. Виконання кластеризації k -середніх на наборі даних відбувається за допомогою методу ліктя. Далі ми обчислюємо суму квадратів як метрику, щоб знайти

оптимальну кількість кластерів для певного набору даних. Загальні квадрати суми (WSS) - це сума квадратів відстаней між кожним членом кластера та центроїдом його кластера.

$$WSS = \sum_{i=1}^m (x_i - c_i)^2 \quad (2.5)$$

де, x = точка даних та c = найближча точка до центроїда.

Сума квадратів (WSS), вимірюється для кожного значення K . А саме значення K , що має найменшу кількість WSS, приймається як найкраще значення.

Тепер намалюємо певну криву між суми квадратів та кількістю кластерів(рис. 2.14).

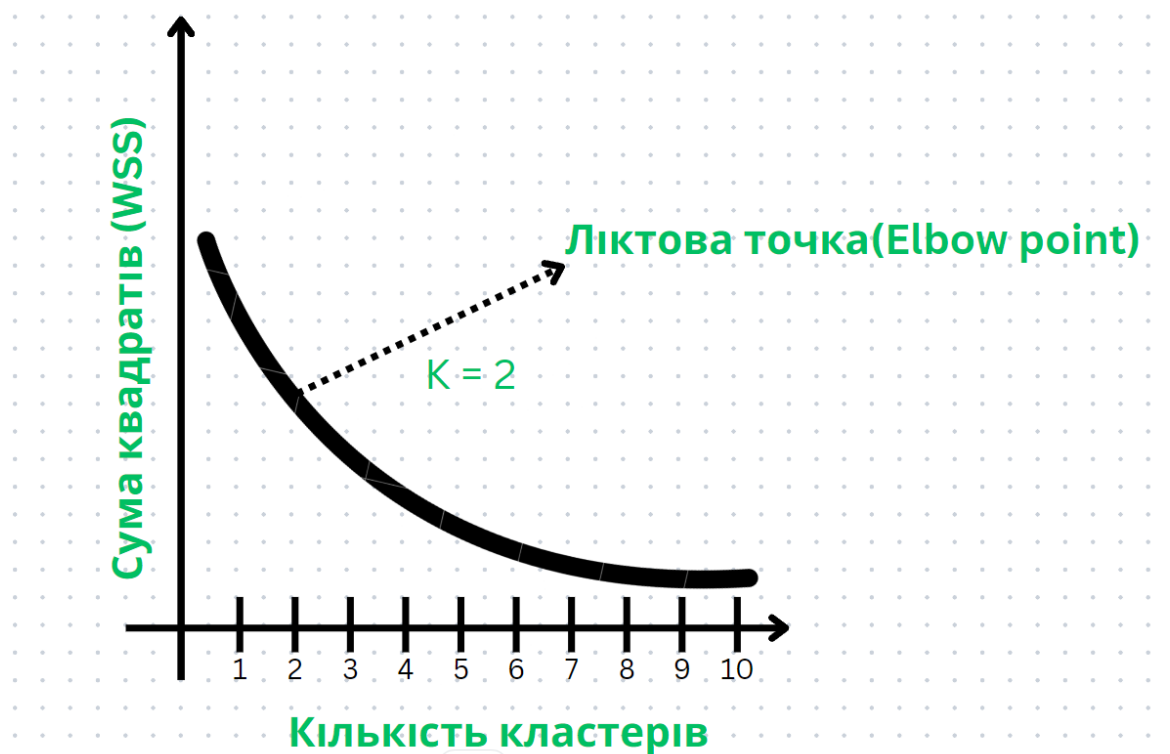


Рисунок 2.14 - Крива між суми квадратів(WSS) та кількістю кластерів

Як видно на рисунку, сума квадратів знаходиться на осі y , а кількість кластерів на осі x .

Можна помітити, що відбувається дуже по-етапна зміна відповідного значення суми квадратів у тому випадку, якщо значення K збільшується з 2.

У результаті значення точки ліктя можна вважати оптимальним значенням K . Це може бути одне з двох, трьох або не більше чотирьох значень. Проте збільшення кількості кластерів мінімально впливає на значення WSS , воно лише стабілізується.

Крок 2.

Намалюємо точки у випадковому порядку та улюбих місцях (рис. 2.15).

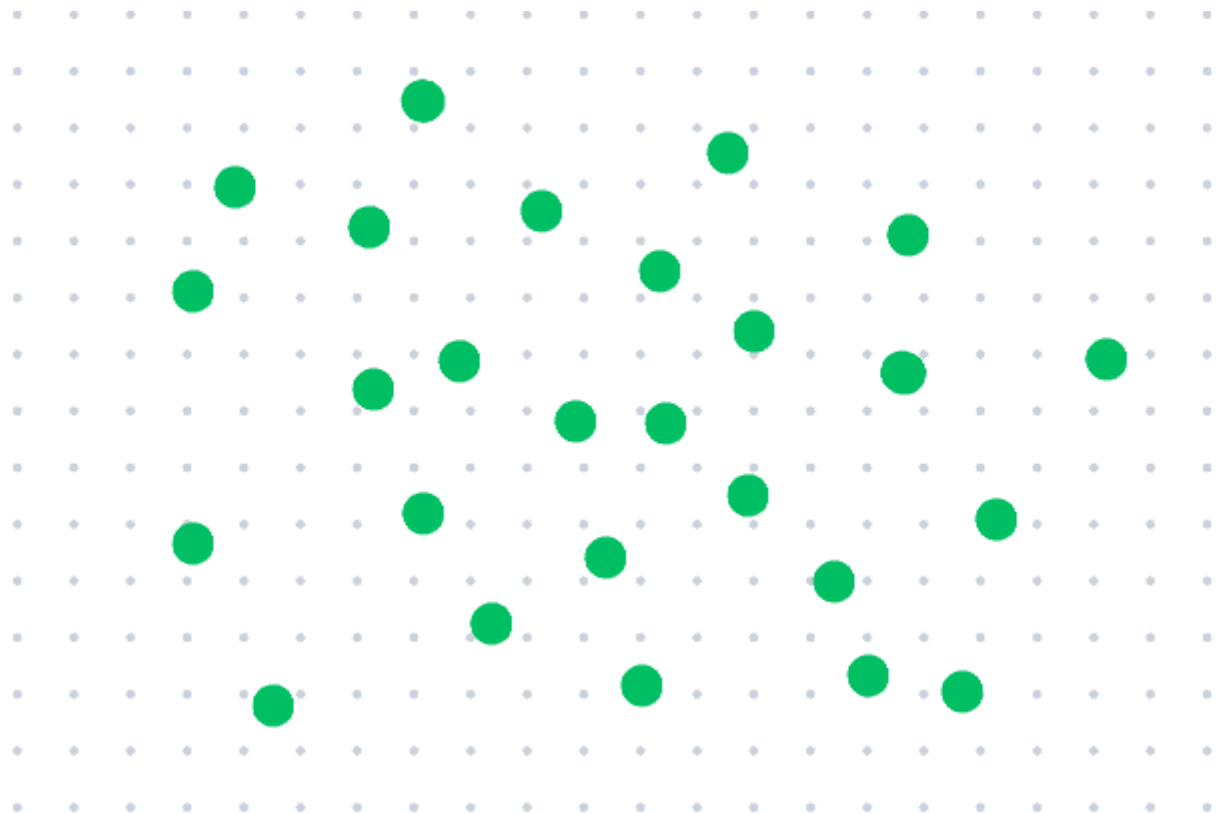


Рисунок 2.15 - Випадкові точки

Тепер намалюємо два центроїди, $C1$ та $C2$ у випадкових місцях посеред точок (рис. 2.16).

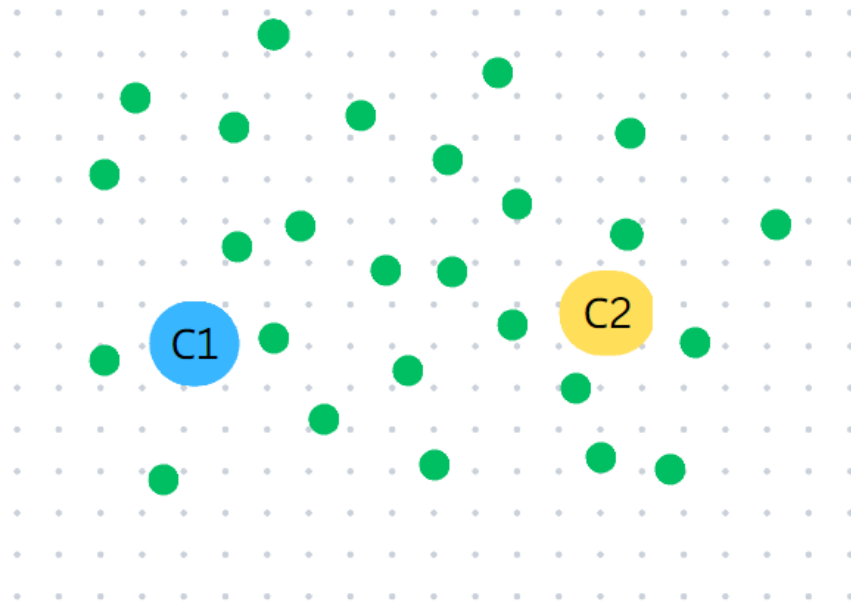


Рисунок 2.16 - Розміщення центроїдів у випадкових місцях

Крок 3.

На цьому кроці необхідно виміряти відстань кожної точки до кожного центроїда. Відстань до центроїда, яка буде найменшою, до такого ж центроїду точка буде призначена.

Отже, порахуємо кожну відстань та призначимо їй найближчий центроїд (рис. 2.17).

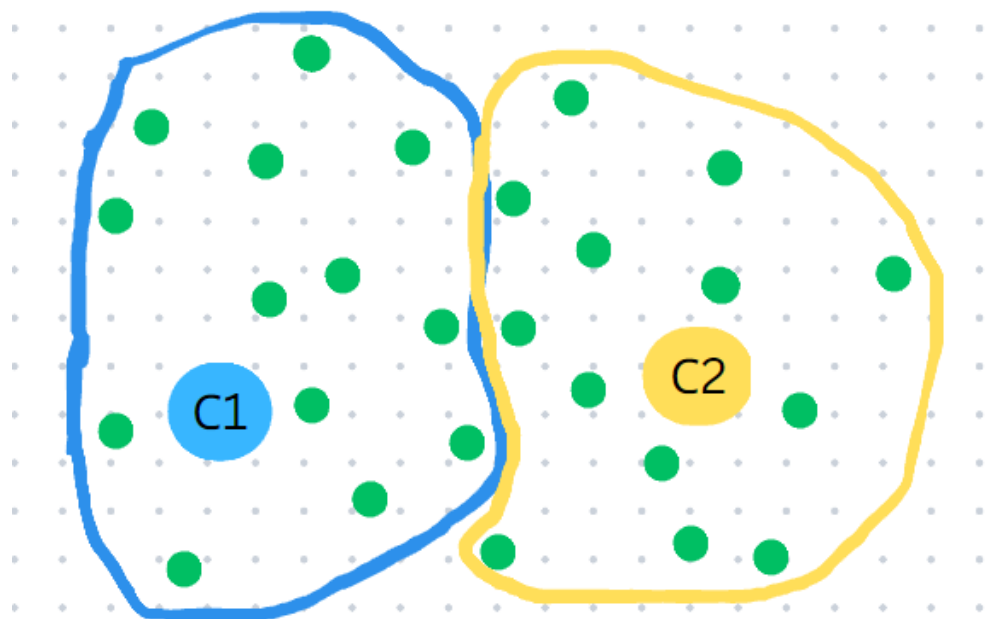


Рисунок 2.17 - Визначення центроїда для кожної точки

Крок 4.

Необхідно та обчислюємо фактичний центроїд точок даних для першої групи.

Крок 5.

Перемістимо випадковий центроїд до фактичного центроїда (рис. 2.18).

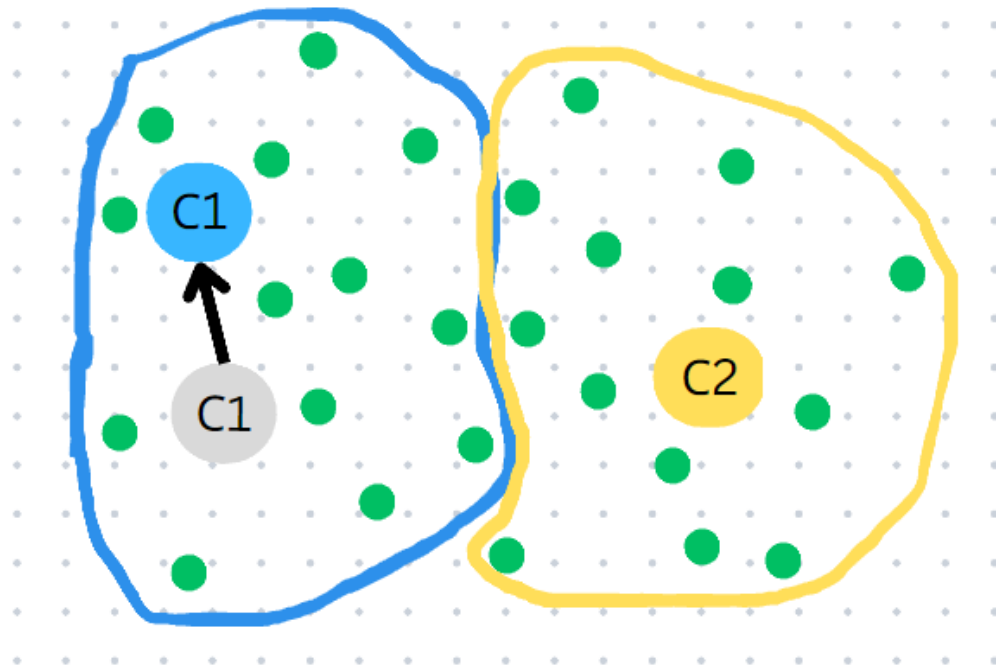


Рисунок 2.18 - Переміщення випадкового центроїда до фактичного центроїда

Крок 6.

Обчислимо фактичний центроїд точок даних для другої групи.

Крок 7.

Наступний крок схожий на крок 5, необхідно перемістити випадковий центроїд до фактичного центроїду, як продемонстровано на рис. 2.19.

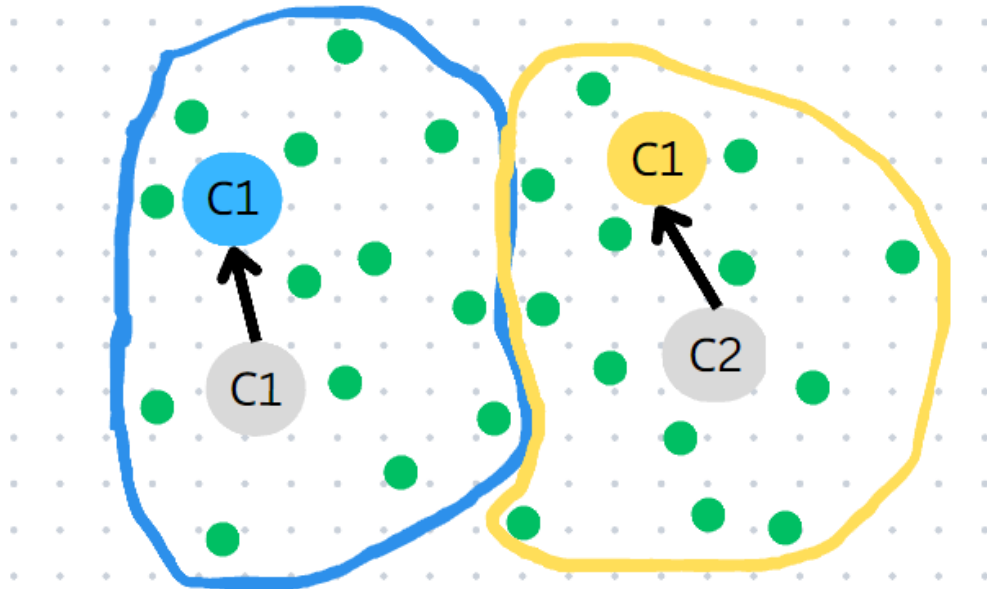


Рисунок 2.19 - Переміщення випадкового центроїда до фактичного центроїда

Крок 8.

Як результат, коли кластер досягає статичності, алгоритм К-середніх набуває та стає конвергентивним.

Отже, в результаті кластер з центроїдами як C1 та C2 буде виглядати наступним чином (рис. 2.20).

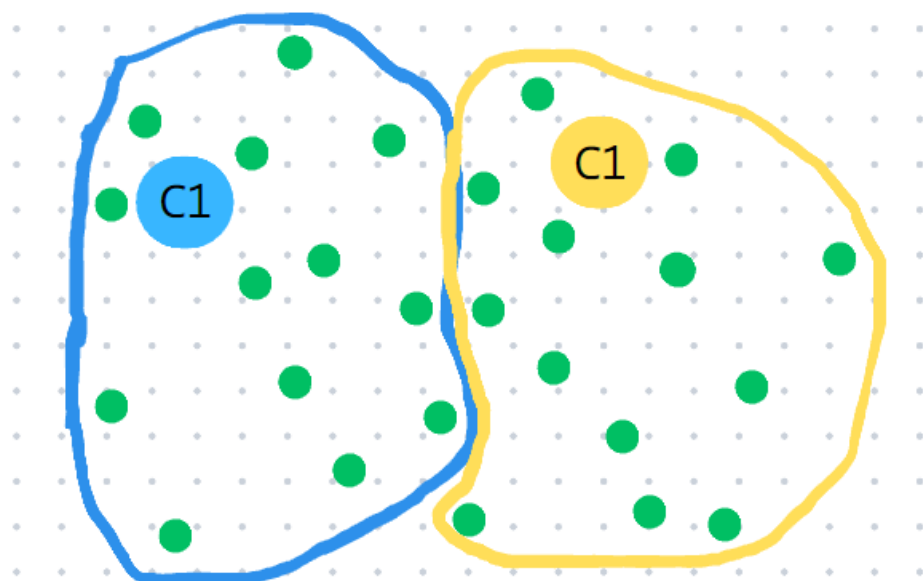


Рисунок 2.20 - Остаточний варіант вигляду кластерів з центроїдами C1 та C2

Поговоримо про алгоритм кластеризації K-середніх.

Скажімо, є $x_1, x_2, x_3, \dots, x_n$ як вхідні дані і необхідно розділити це на K кластери.

В такому випадку маємо наступні кроки для цього.

Крок 1.

Випадково обрати K (центроїди), тобто $c_1, c_2, c_3, \dots, c_k$, тоді можна сформулювати таку формулу:

$$C = (c_1, c_2, c_3, \dots, c_k), \quad (2.6)$$

де, C - множина всіх центроїдів.

Крок 2.

Позначимо кожену точку даних її найближчому центру, що може отримуватися за допомогою обрахування евклідової відстані.

$$\arg \min_{c_k \in C} \text{dist}(c_k, x)^2, \quad (2.7)$$

де, $\text{dist}()$ - евклідова відстань.

Іншими словами, обчислюється відстань кожного значення x від кожного значення c . Це виглядає приблизно ось так: $x_1-c_1, x_1-c_2, x_1-c_3, \dots, x_n-c_k$. Надалі знаходиться найменше значення та призначається x_n цьому конкретному центроїду.

Крок 3.

У наступному кроці необхідно визначити фактичний центроїд, візьмемо середнє значення усіх точок, які були призначені цьому кластеру.

$$C_i = \sum_{x_j \in S_i}^{x_i} \frac{l}{|S_i|} \quad (2.8)$$

де, S_i - множина всіх точок, які були призначені, скажімо, i -тому кластеру. Іншими словами, вихідна точка, яка була центроїдом, змінила своє положення, що являє собою фактичним центроїдом для кожної з цих груп.

Крок 4.

У цьому кроці, необхідно повторювати кроки 2 та 3 до того моменту, доки не буде досягнуто саме конвергенції.

2.3 Функціональна структура інформаційної системи

Послідовність визначення траєкторії об'єктів представлено на рис. 2.21.

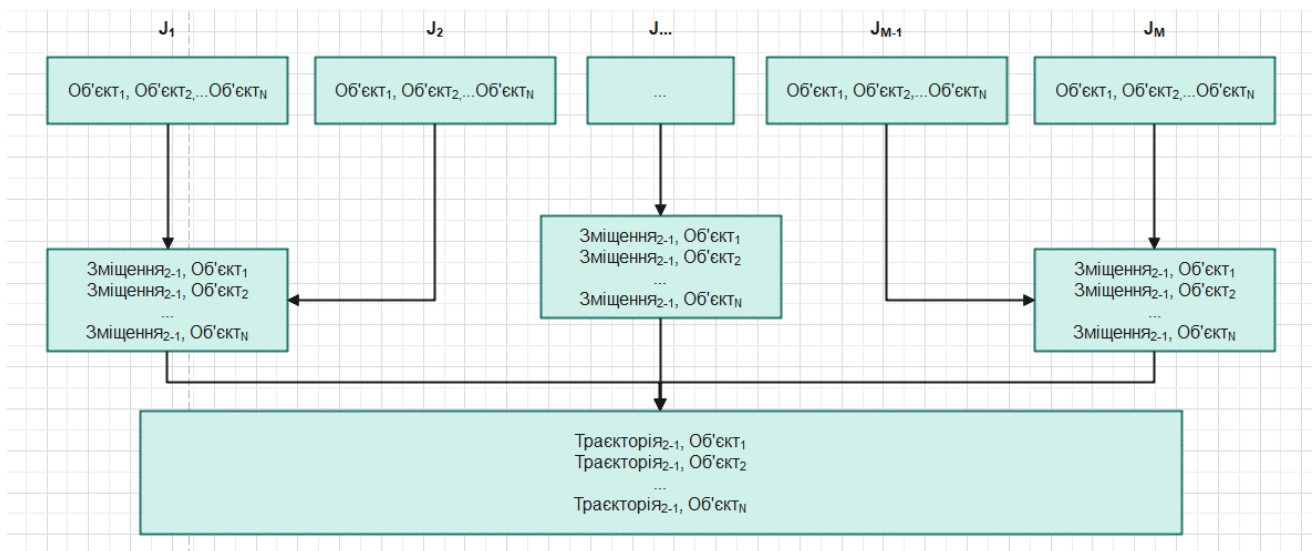


Рисунок 2.21 - Послідовність визначення траєкторії об'єктів

На рисунку 2.21, де J - є зображенням.

У першому блоці визначаються усі об'єкти, що є на зображенні. Необхідно визначити об'єкти, як мінімум, на двох зображеннях.

У другому блоці знаходяться зміщення об'єктів з двох зображень.

У третьому блоці поєднуються усі зміщенням об'єктів та будується траєкторія зміщених об'єктів.

2.4 Висновки до розділу 2

Реалізовано спосіб оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору. Суть способу полягає в опрацюванні серій послідовних зображень із визначенням на кожному з них положення об'єкту. Для визначення положення об'єкту використовується метод кластеризації, а саме метод k-means. Застосування методу кластеризації дозволяє визначення положення одразу кількох об'єктів на одиничному зображенні. Це дозволяє визначати величину зміщення кількох об'єктів одночасно за один цикл обробки однієї серії послідовної зображень, тобто оцінювати зміщення кількох об'єктів спостереження з несинхронізованим рухом без застосування ітераційної по об'єктової процедури. Сформована інформаційна структура способу оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах

Розділ 3 Програмна реалізація способу оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах

3.1 Структура модулів системи, їх взаємозв'язок

На рисунках 3.1 та 3.2 представлена послідовність роботи програми за допомогою модулів.

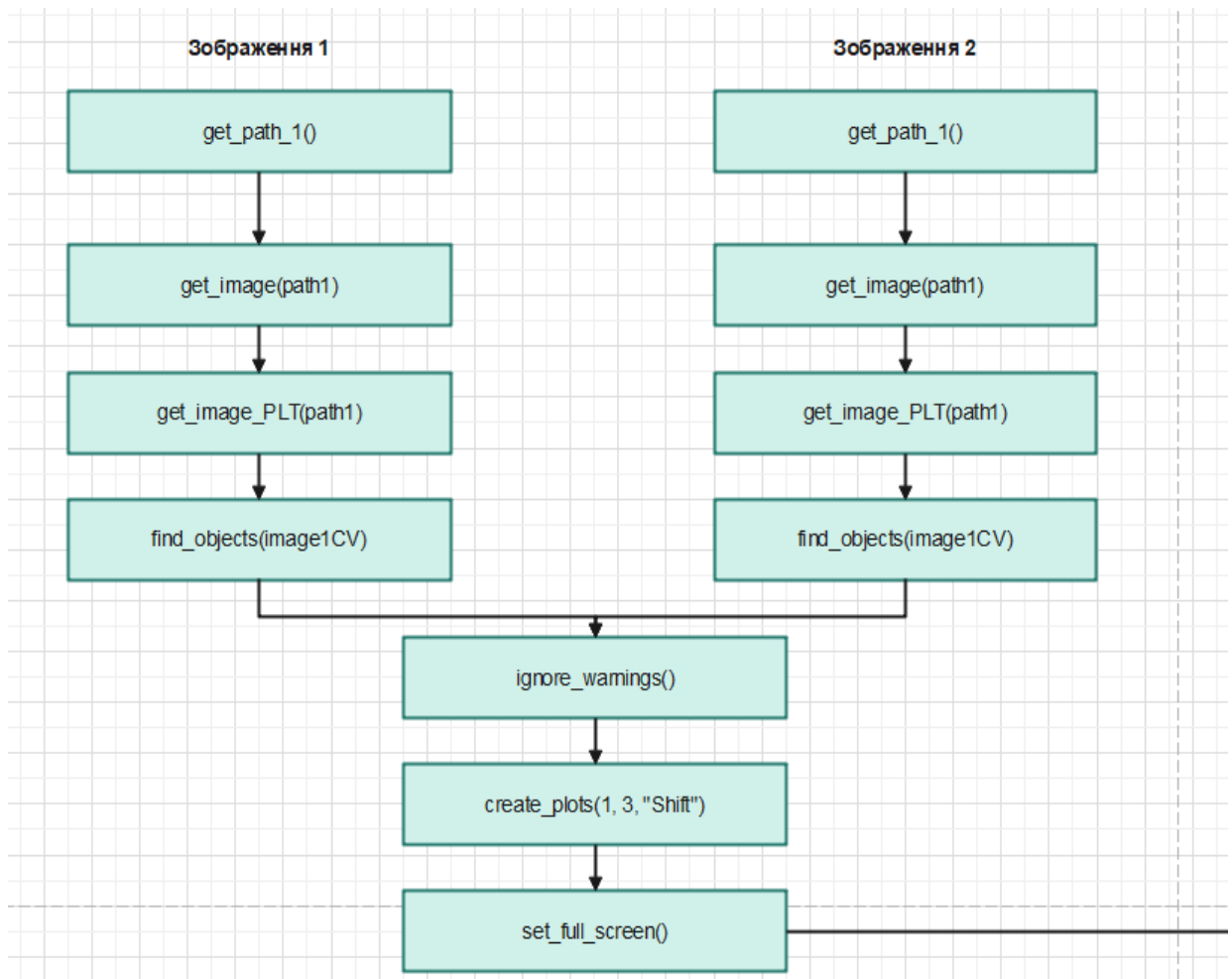


Рисунок 3.1 - Послідовність роботи модулів програми (Перша частина)

У першому модулі, що мають назву **get_path_1()** та **get_path_2()**, відповідають та виконують роботу, аби передати у глобальну змінну необхідний шлях до першого та другого зображення. Модуль не вимагає передачі ніяких параметрів. Модуль повертає шлях до зображення у вигляді стрічки. Далі ця

змінна буде використовуватися у багатьох інших модулях, таких як: `get_image()`, `get_image_PLT()`, `get_image_CV`.

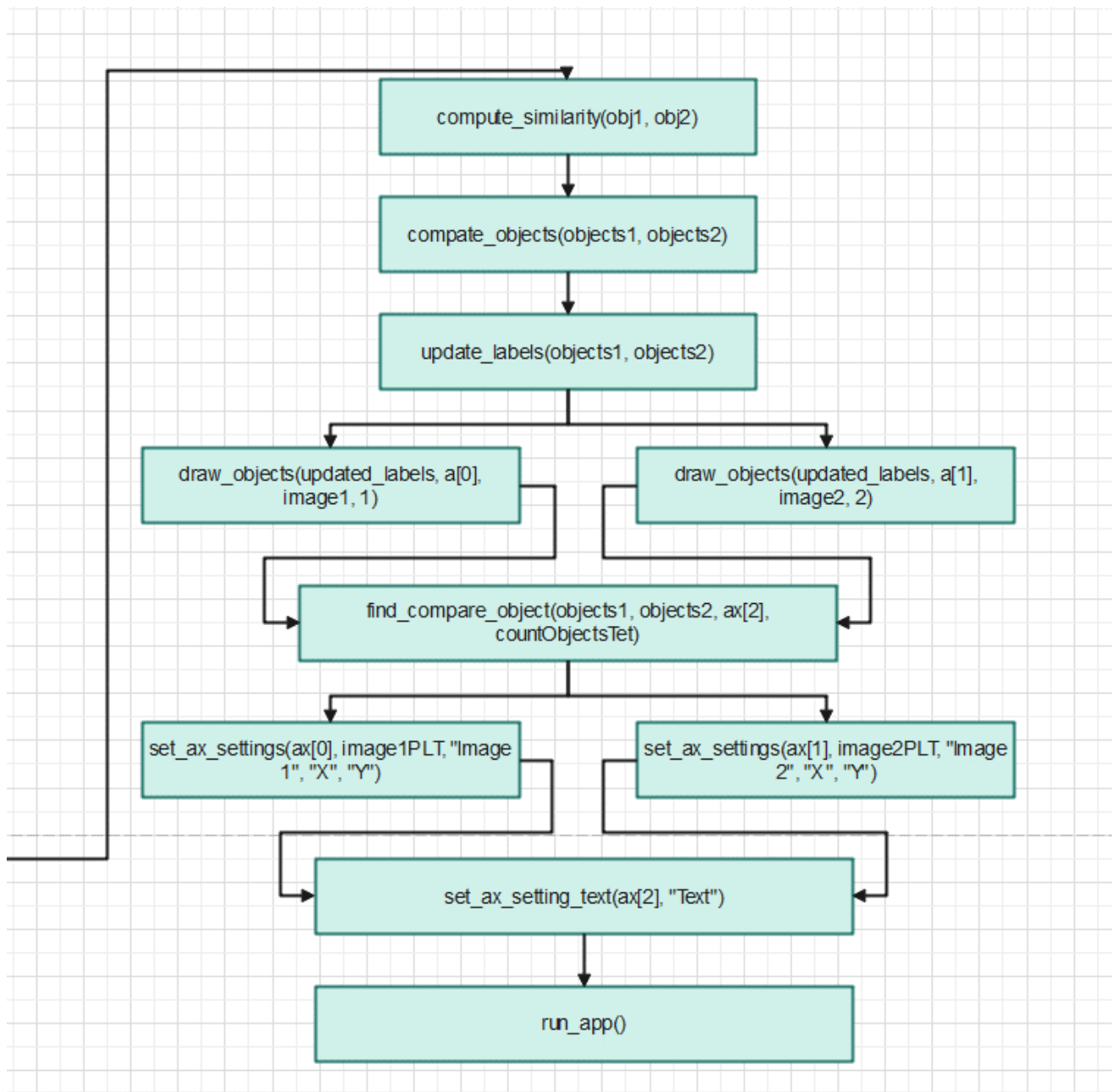


Рисунок 3.2 - Послідовність роботи модулів програми (Друга частина)

Другий модуль програми приймає передані програмі зображення. Модуль, що називається `get_image(path1)` та `get_image(path2)`, отримує у якості параметру шлях до зображення у вигляді стрічки, змінну `path1` та `path2`, що був отриманий з модуля `get_path_1()` та `get_path_2()`. Модуль обробляє його, перетворюючи в необхідний для роботи формат.

Наступний модуль програми **get_image_PLT(path1)** та **get_image_PLT(path2)** виконує схожу функцію, що і модуль **get_image()**, даний модуль отримує у якості параметра стрічку, шлях до зображення, змінну **path1** або **path2**, які були отриманні раніше. Модуль відповідає за перетворення зображення у формат для роботи з графіками далі, адже тільки такий формат приймається у графік.

Четвертий модуль програми з назвою **get_image_CV()** приймає у якості параметру змінну **path1** або **path2**, що є стрічкою та шляхом до зображення. Модуль перетворює зображення у формат **CV**.

Наступний модуль програми займається знаходженням об'єктів на зображенні, під назвою **find_objects()**. Модуль приймає оброблене зображення від попередніх модулів. Модуль, використовує для роботи кластеризацію **k-means**, після чого повертає масив об'єктів унікальних об'єктів на зображенні вказуючи їхні координати та унікальну назву.

Шостий модуль, **ignore_warnings()** відповідає за не показ і видалення спаму, що затримує процес запуску програми.

Сьомий модуль застосунку, **create_plots()** відповідає за створення та ініціалізацією графіків. У модуль передаються три параметри. Перший параметр є числом та відповідає за кількість графіків у горизонталі, цей параметр у застосунку дорівнює 1. Другий параметр відповідає за створення графіків у вертикалі і він дорівнює 3. Оскільки нам необхідно два графіка, та третє місце для виведення тексту, результату програми. Третій параметр модуля є стрічка та відповідає за назву програми, він дорівнює "Shift".

Восьмий модуль програми, **set_full_screen()**, відповідає за візуальне бачення графіків, а саме за повний екран відкриття програми під час запуску. Таким чином економить час та робить застосунок зручним у користуванні.

Дев'ятий модуль **compute_similarity()** займається пошуком погрішності об'єктів спостереження. Наприклад, якщо десь зображення маленьке, а десь більше.

Використовуючи десятий модуль **compare_objects()** порівнюються два зображення та знаходяться однакові об'єкти на них. Для роботи його головного призначення використовується унікальне значення об'єкту, що буде встановлено під час роботи модуля `updated_labels`. Модуль приймає два параметри, де перший - об'єкти першого зображення, а другий параметр - об'єкти другого зображення. Власне, при знаходженні однакових об'єктів, модуль повертає масив, де вказано координати об'єкта на першому зображенні, координати об'єкта на другому зображенні та координати зміщення об'єкта.

Одинадцятий модуль, **updated_label()** відповідає за знаходження однакових об'єктів та встановлення однакових для них значень, а саме "label". Модуль приймає два параметри, які мають об'єкти 1 зображення та об'єкти другого зображення. Проводить пошук однакових об'єктів та встановлює їм однаковий "label", який надалі виводиться, як результат роботи програми.

Наступний дванадцятий модуль, відповідно до його назви, **draw_objects()**, має за мету намалювати на графіку зображення та прямокутники, що показують його координати та назву. Він приймає об'єкти зображення, сам графік на якому необхідно намалювати та власне зображення для отримання його ширини та висоти.

Тринадцятий модуль програми, **draw_compare_object()** займається виведенням тексту, що показує користувачеві початкові координати, кінцеві координати та зміщення однакових об'єктів на двох зображень. Для його роботи необхідно передати об'єкти першого зображення, об'єкти другого зображення, місце, де потрібно вивести текст про зміщення та змінну `countObjectsText` для коректного виведення тексту.

Чотирнадцятий модуль програми, **set_ax_setting()** використовується для налаштування графіків. Модуль приймає п'ять параметрів. Перший параметр вказується сам графік, налаштування якого необхідно виконати. Другий параметр відповідає за зображення, яке буде виведено на графіку. Третій параметр використовується для назви графіку, яка буде виведена зверху над графіком. Четвертий параметр використовується теж як назва, але по осі X, що

буде виведена знизу, під графіком. І останній параметр приймає стрічку, що буде виведена, як назва по осі Y, зліва від графіку.

І останній, п'ятнадцятий модуль, пусковий модуль, де виконується запуск графіків та починаються їх будова.

3.2 Особливості реалізації способу оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах

Для роботи програми використовується шість бібліотек.

Перша з них, є **PIL**, що імпортує Image. Вона використовується для обробки та аналізу вхідного зображеннями. Бібліотека використовується у модулі `get_image`, власне у Image викликається метод `open()`, що відкриває зображення та аналізує його. Він приймає стрічку, тобто шлях зображення на комп'ютері.

Друга, має назву **numpy**, що в даній програмі використовується у якості ведення математичних операцій. У програмі бібліотека використовуються в багатьох випадках, власне, для переведення зображення у масив за допомогою методу `.array()`, що приймає зображення. Наступний випадок, використовує метод `.unique()`. Метод приймає масив елементів, любых елементів. А повертає лише унікальні елементи цього масиву, тобто, видаляючи однакові значення і залишаючи тільки унікальні. Ще, бібліотека використовує метод `.where()`, який займається пошуком конкретного елемента в масиві. Метод приймає масив елементів як перший параметр та сам елемент пошуку у другому параметрі. Метод повертає елементи, вибрані з `x` або `y` у залежності від умови. Також використовується метод `.min()`, що приймає масив елементів та повертає мінімальне значення елемента від даного масиву. Використовується метод `.max()`, що приймає масив елементів та повертає максимальне значення елемента від даного масиву.

Третя бібліотека **CV2** використовується для кластеризації зображень та для сегментації зображень. За допомогою неї знаходять об'єкти на зображеннях та їхні координати. Метод `.KMeans()` приймає значення `n_clusters` типу `int`, де вказується кількість бажаних кластерів, проте дефолтне значення є вісім, ще метод приймає значення `algorithm`, де вказується бажаний алгоритм роботи методу `k-means`, алгоритм може бути «lloyd», «elkan», «auto», «full». Далі, у цього ж методу існує ще свої методи, деякі з них використовується у програмі, власне метод `.fit()`, що приймає масив зображення. Метод проводить над зображенням кластеризацію. Ще існує значення `.labels_` у метода `.KMeans()`, яке використовується у програмі для вилучення унікальних значень знайдених об'єктів.

Четверта бібліотека **matplotlib** використовується для роботи з графіками. За допомогою неї виводяться, створюються та налаштовуються графіки зображень. Для самого початку, як й необхідно, бібліотека використовує метод `.subplots()` з трьома параметрами. Даний метод призначений для створення графіку з налаштуваннями. Перший та другий параметр методу є число і вони відповідають за створення кількості графіків по вертикалі та по горизонталі. Власне нам необхідно лише три місця, де перші два графіка, а третє для виведення інформації про зміщення. Третій параметр має назву `label` та використовується для назви самої програми, він приймається у вигляді стрічки. Не менш важливий метод `.imshow()`, що відповідає за відображення власне зображення на графіку. Метод `.set_title()` відповідає за назву графіку, приймається стрічка. Метод `.set_xlabel()` відповідає за назву осі X, приймає стрічку. Метод `.set_ylabel()` відповідає за назву осі Y, приймає стрічку. Метод `.axis()` відповідає за значення графіку, в нашому випадку необхідно для зникнення третього графіку, з значенням «off». Метод `.annotate()` необхідний для відображення тексту на графіку, де перший параметр вказується стрічка, назва тексту, третій параметр `xy` вказується координати розміщення тексту по осям X та Y. Метод `.add_path()` відповідає за додавання різних форм фігур на графік, в даному випадку додавання прямокутника.

П'ята бібліотека **matplotlib.image** використовується для читання зображення у форматі plt для того, щоб вивести його у графік. Використовується метод `.imread()`, що приймає стрічку, як шлях до зображення.

Шоста бібліотека **warnings** використовується для ігнорування зайвих повідомлень від бібліотек. Це пришвидшує роботу та запуск програми. Використовується метод `.filterwarnings()`, в даному випадку необхідний без параметрів.

На рисунку 3.3 показана послідовність роботи модуля **get_path()**

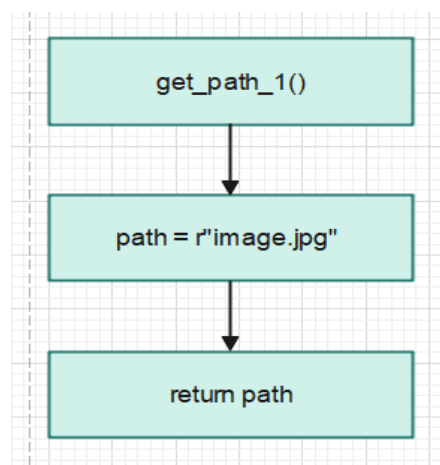


Рисунок 3.3 - послідовність роботи модуля `get_path()`

На рисунку 3.4 показана послідовність роботи модуля **get_image()**.

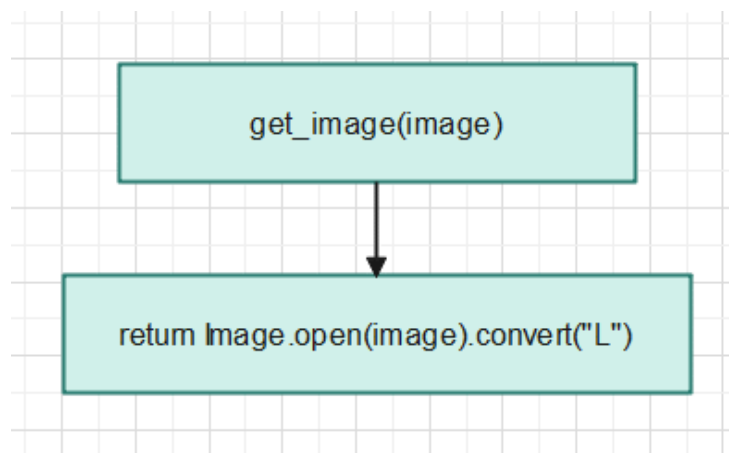


Рисунок 3.5 - Послідовність роботи модуля `get_image()`

На рисунку 3.5 показана послідовність роботи модуля **get_image_PLT()**.

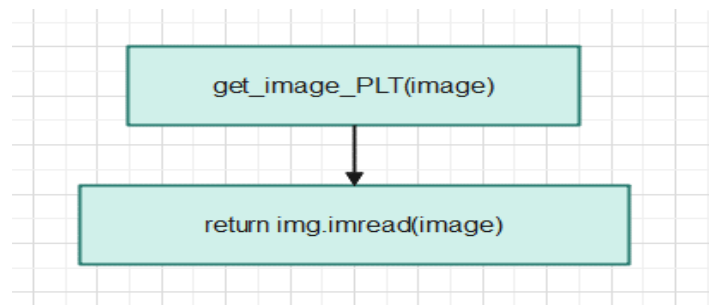


Рисунок 3.5 - Послідовність роботи модуля `get_image_PLT()`

На рисунку 3.6 та рисунку 3.7 показана послідовність роботи модуля `find_objects()` з різними бібліотеками та конкретними значеннями.

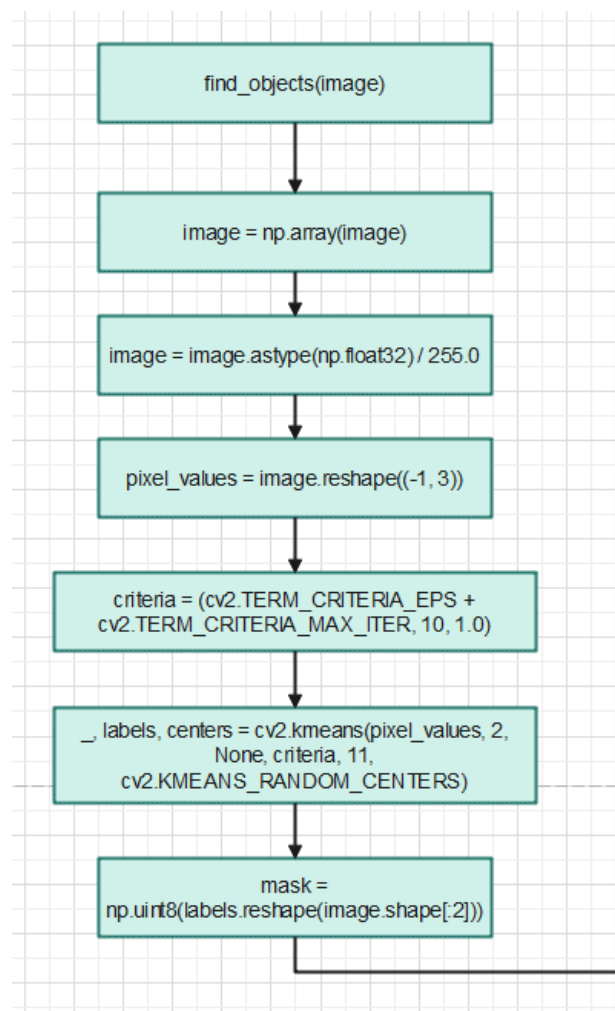


Рисунок 3.6 - Послідовність роботи модуля `find_objects()` (Перша частина)

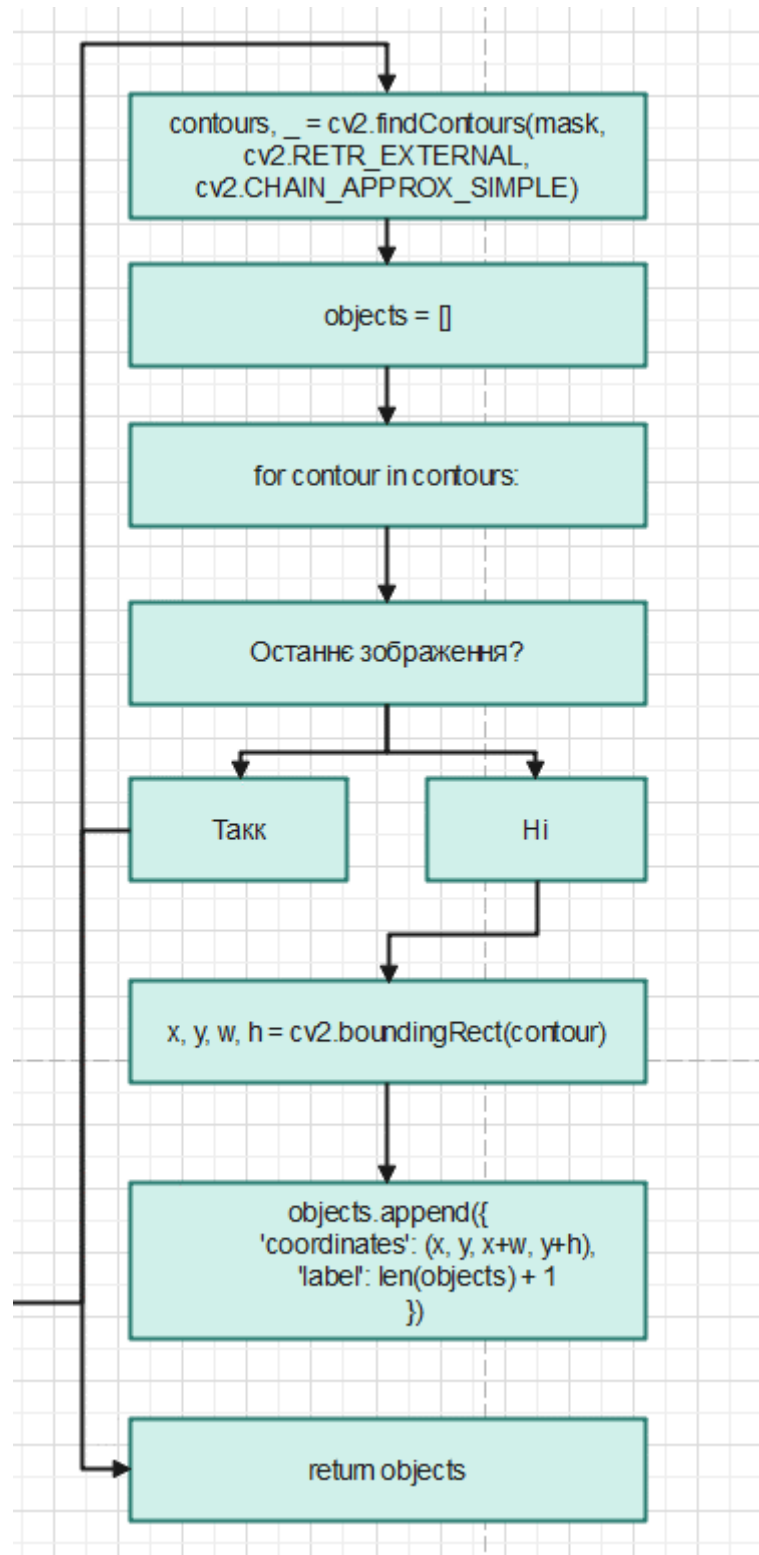


Рисунок 3.7 - Послідовність роботи модуля `find_objects()` (Друга частина)

На рисунку 3.8 показана послідовність роботи модуля `ignore_warnings()`.

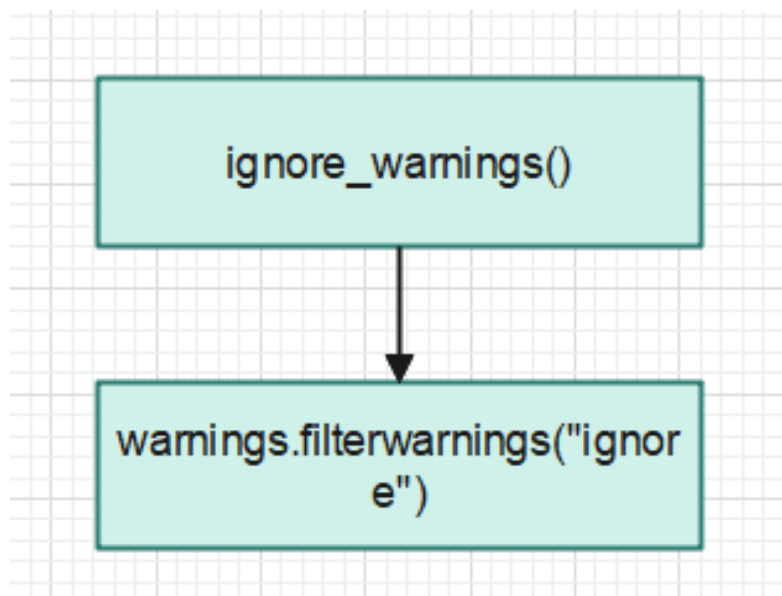


Рисунок 3.8 - Послідовність роботи модуля `ignore_warnings()`

На рисунку 3.9 показана послідовність роботи модуля `create_plots()`.

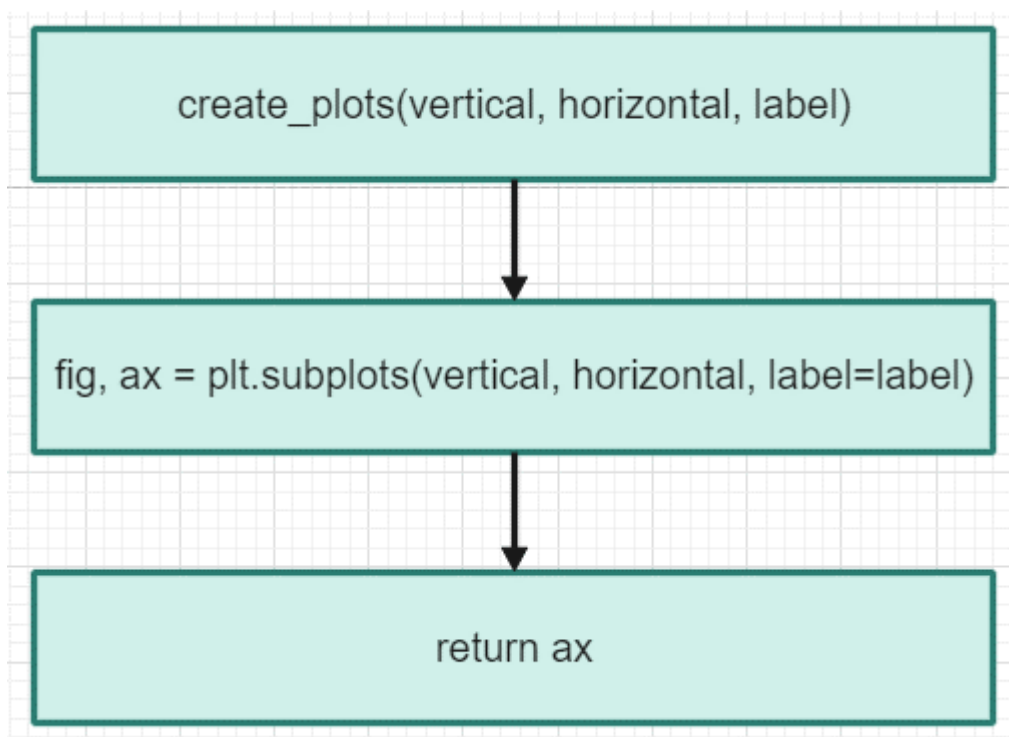


Рисунок 3.9 - Послідовність роботи модуля `create_plots()`

На рисунку 3.10 показана послідовність роботи модуля `set_full_screen()`.

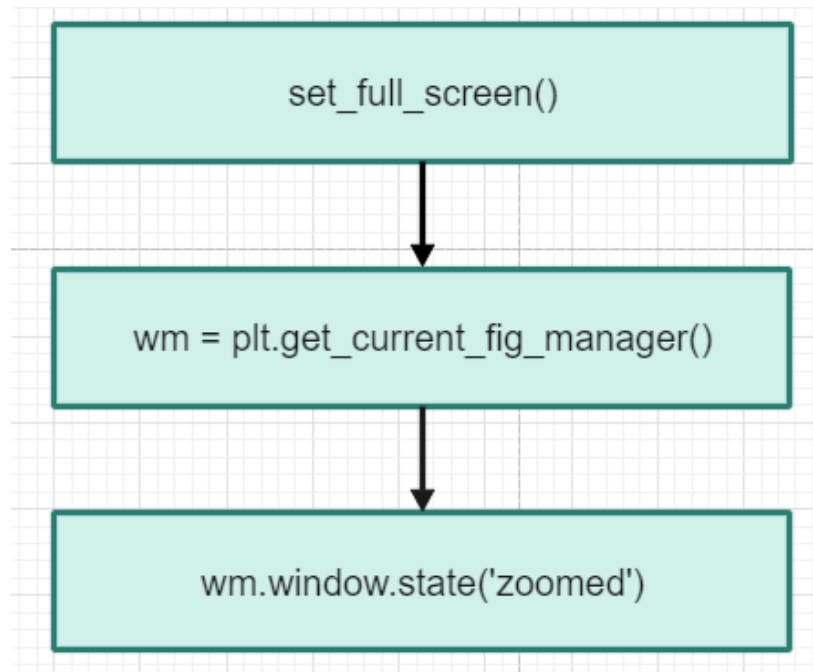


Рисунок 3.10 - Послідовність роботи модуля `set_full_screen()`

На рисунку 3.11 показана послідовність роботи модуля `compute_similarity()`.

Модуль `compute_similarity()` виконує обчислення схожості між двома об'єктами на основі принципу перетину злиття. Основною метою модуля є обчислення показника схожості між двома об'єктами, щоб визначити, наскільки сильно вони перекриваються. Перший параметр, представлений у вигляді словника з ключами “coordinates” (координати обмежувальної рамки) і “label” (мітка об'єкта). Другий параметр, представлений у вигляді словника з ключами “coordinates” і “label”. Координати обмежувальної рамки об'єкта вказуються у форматі (x1, y1, x2, y2), де (x1, y1) - верхній лівий кут рамки, а (x2, y2) - нижній правий кут рамки. Модуль спочатку витягує координати обмежувальних рамок з об'єктів `obj1` і `obj2`. Далі виконується обчислення перетину, площі кожного об'єкта і об'єднання їх площ. Перетин обчислюється як площа прямокутника, утвореного перетином координат обмежувальних рамок. Площі об'єктів `area1` і `area2` обчислюються як площі прямокутників, утворених координатами обмежувальних рамок. Об'єднання обчислюється шляхом додавання площ об'єктів і віднімання перетину.

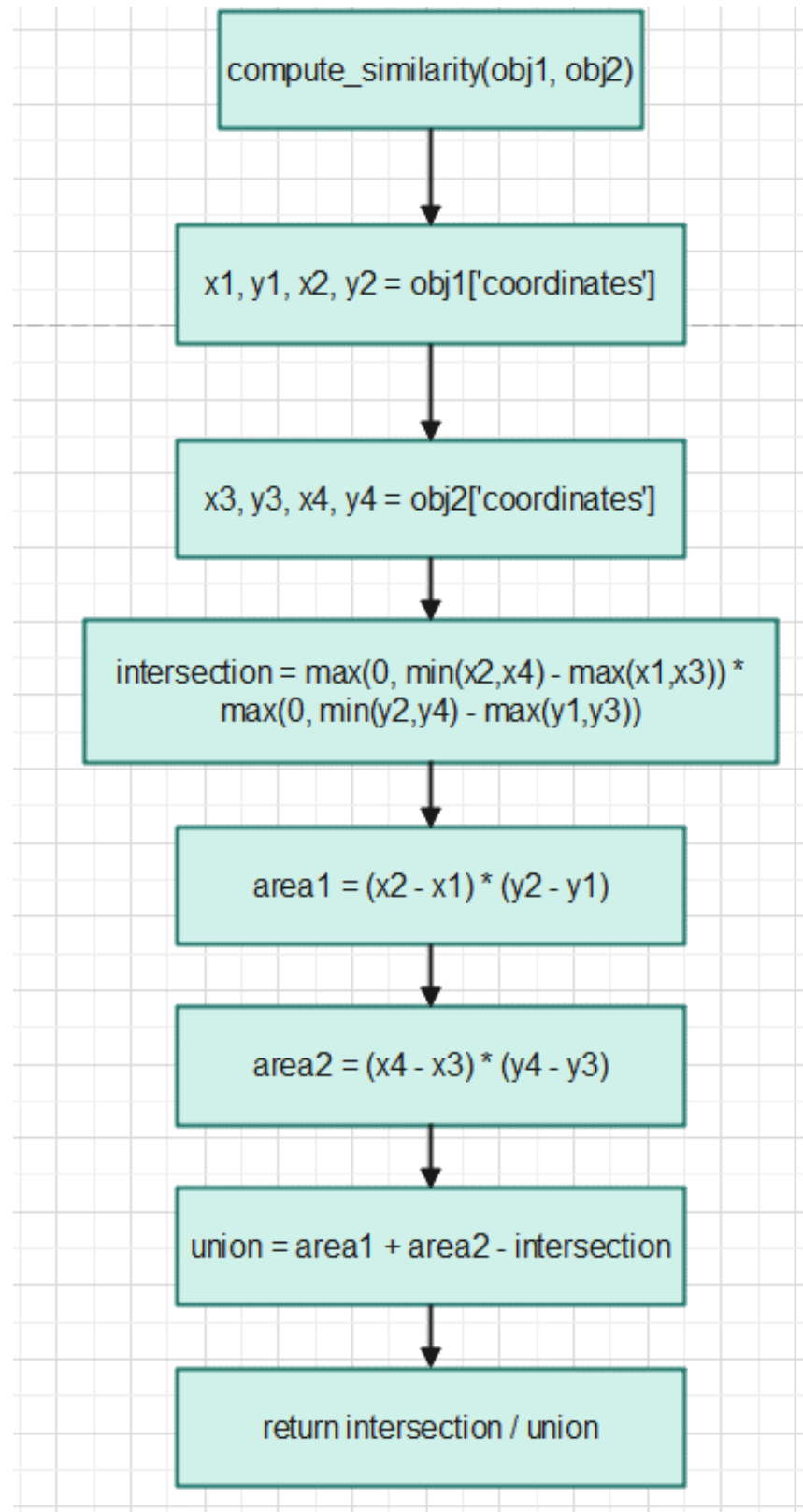


Рисунок 3.11 - Послідовність роботи модуля `compute_similarity()`

На рисунку 3.12 та 3.13 показана послідовність роботи модуля `compare_objects()`, що порівнює об'єкти двох зображень.

Модуль `compare_objects()` призначений для порівняння об'єктів, знайдених на двох зображеннях. Він визначає спільні об'єкти, які присутні на обох зображеннях, і обчислює їх схожість. Перший параметр відповідає за список об'єктів на першому зображенні. Другий параметр відповідає за список об'єктів на другому зображенні. Кожен об'єкт у списку представлений у вигляді словника з ключами “`coordinates`” (координати обмежувальної рамки) і “`label`” (мітка об'єкта). Модуль спочатку створює порожній список `common_objects`, який буде містити спільні об'єкти з обох зображень, а також створює порожню множину `used_indices`, що використовується для відстеження використаних індексів об'єктів на другому зображенні. Потім виконується подвійний цикл, який проходить по кожному об'єкту на першому зображенні і спробує знайти йому найкраще співставлення на другому зображенні. Для цього обчислюється показник схожості між об'єктом на першому зображенні і всіма об'єктами на другому зображенні. Об'єкт на другому зображенні з найменшим показником схожості вважається найкращим співставленням. У циклі для кожного об'єкту на першому зображенні порівнюються його показники схожості з усіма об'єктами на другому зображенні. Якщо показник схожості є кращим (меншим) за поточний найкращий показник, то зберігається об'єкт на другому зображенні як найкраще співставлення. Після знаходження найкращого співставлення об'єкту з першого зображення, індекс цього об'єкту на другому зображенні додається до множини `used_indices`, щоб уникнути повторного використання об'єктів. Крім того, створюється словник з інформацією про спільний об'єкт і додається до списку `common_objects`. Цей словник містить мітки об'єктів, координати обмежувальних рамок об'єктів на обох зображеннях, зсув між ними і показник схожості. На виході модуль повертає список `common_objects`, що містить інформацію про спільні об'єкти на обох зображеннях.

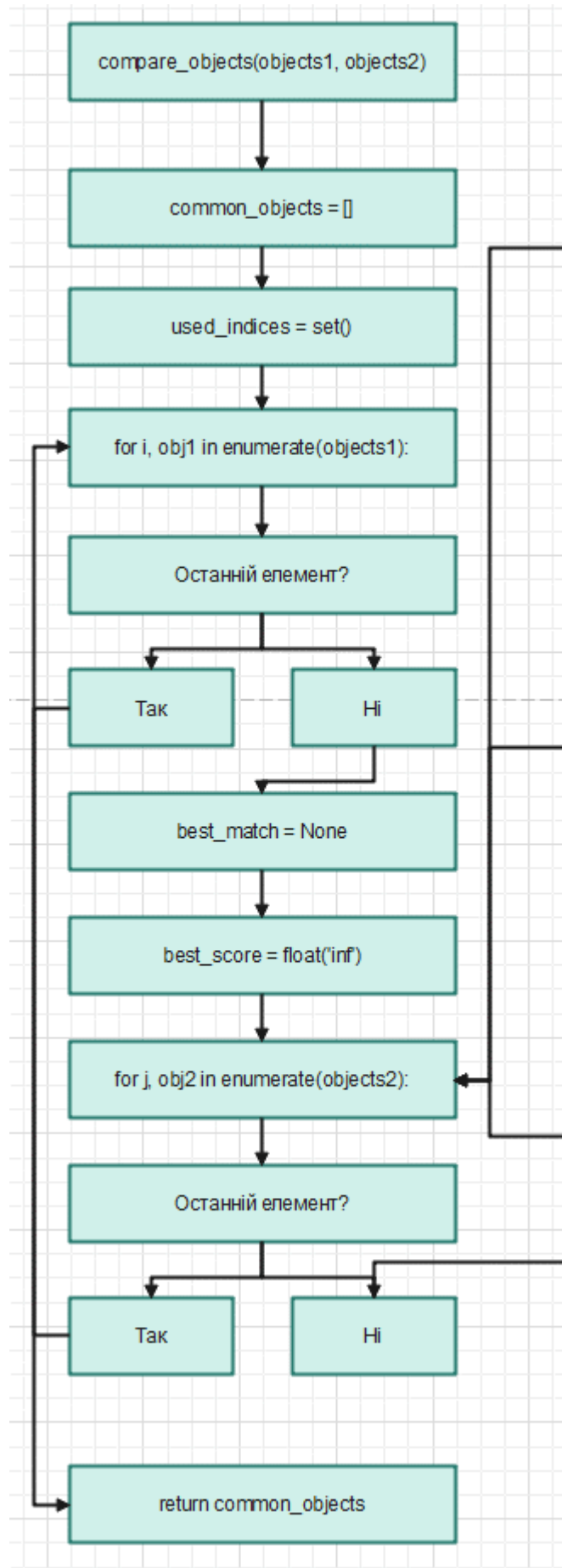


Рисунок 3.12 - Послідовність роботи модуля `compare_objects()` (Перша частина)

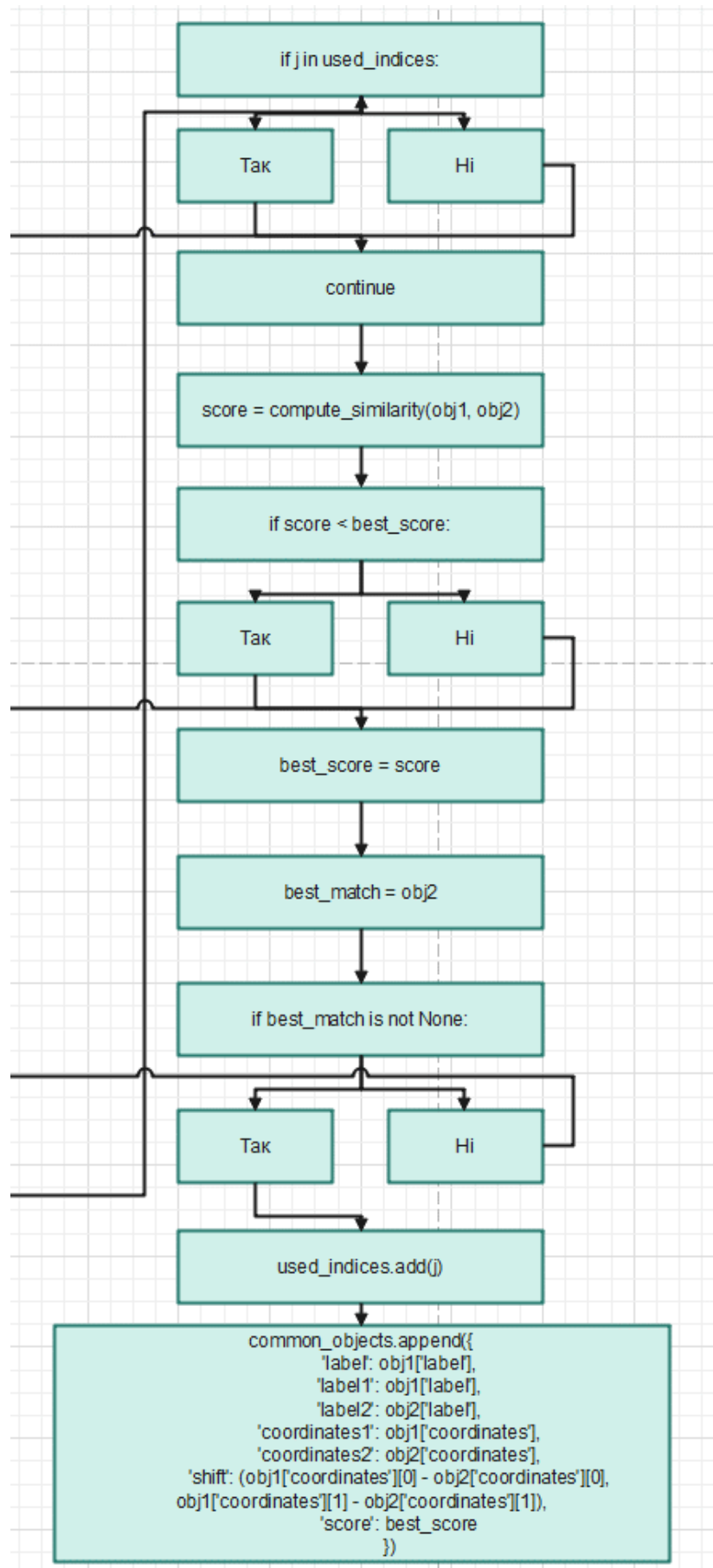


Рисунок 3.13 - Послідовність роботи модуля `compare_objects()` (Друга частина)

На рисунку 3.14 показана послідовність роботи модуля `update_labels()`.

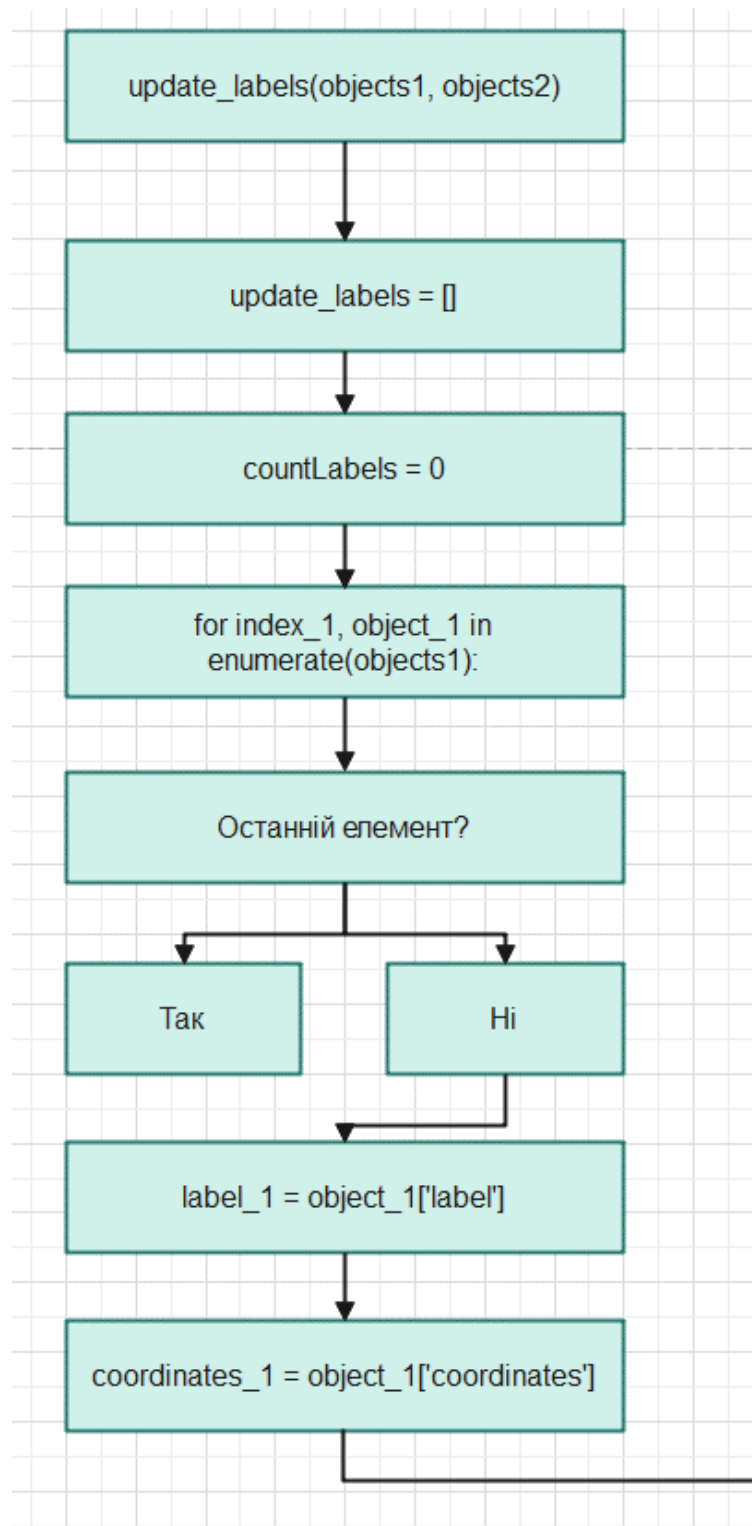


Рисунок 3.14 - Послідовність роботи модуля `update_labels()` (Перша частина)

На рисунку 3.15, 3.16 та 3.17 показано послідовність роботи модуля `draw_objects()`.

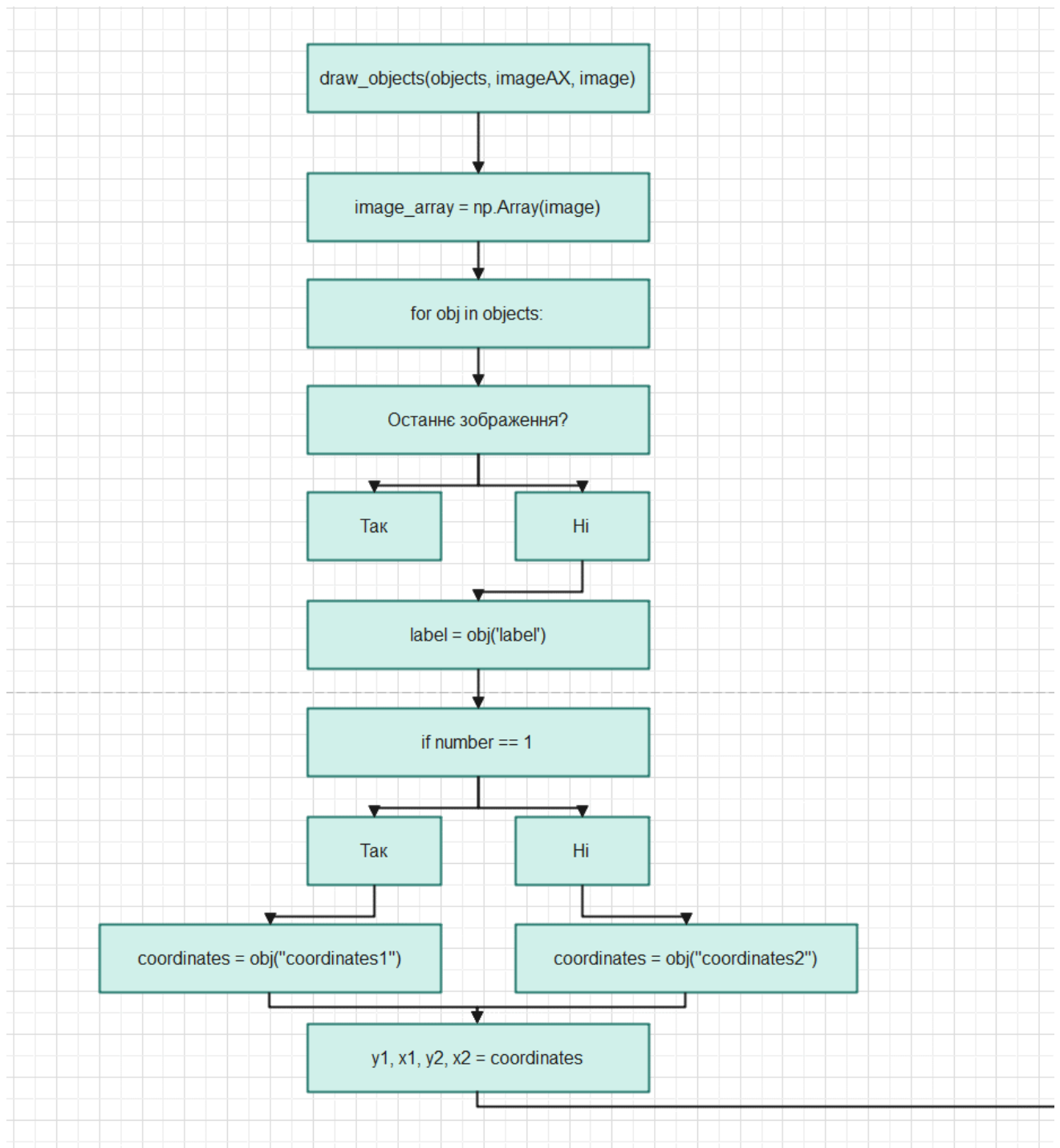


Рисунок 3.16 - Послідовності роботи модуля `draw_objects()` (Перша частина)

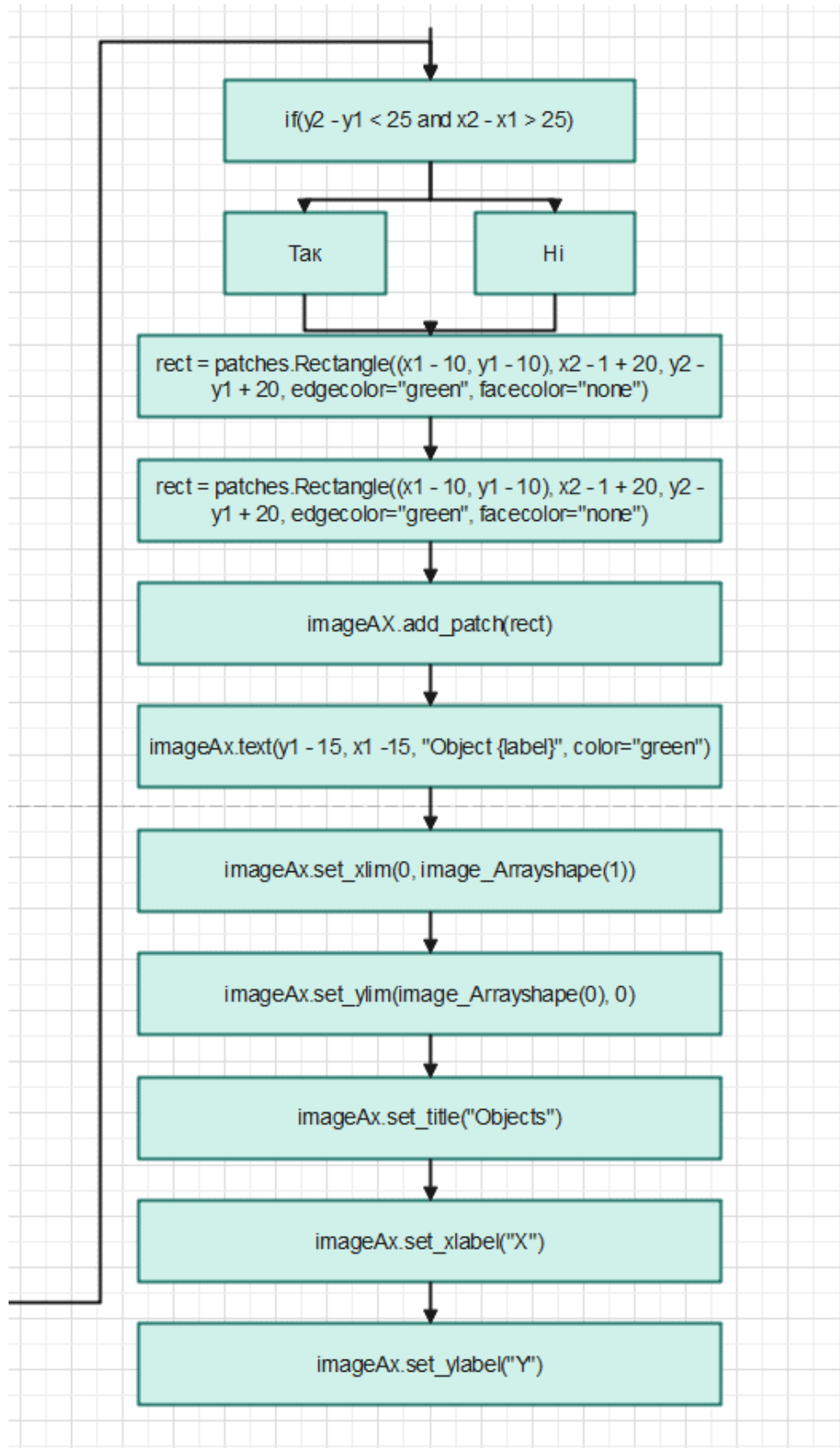


Рисунок 3.17 - Послідовності роботи модуля `draw_objects()` (Друга частина)

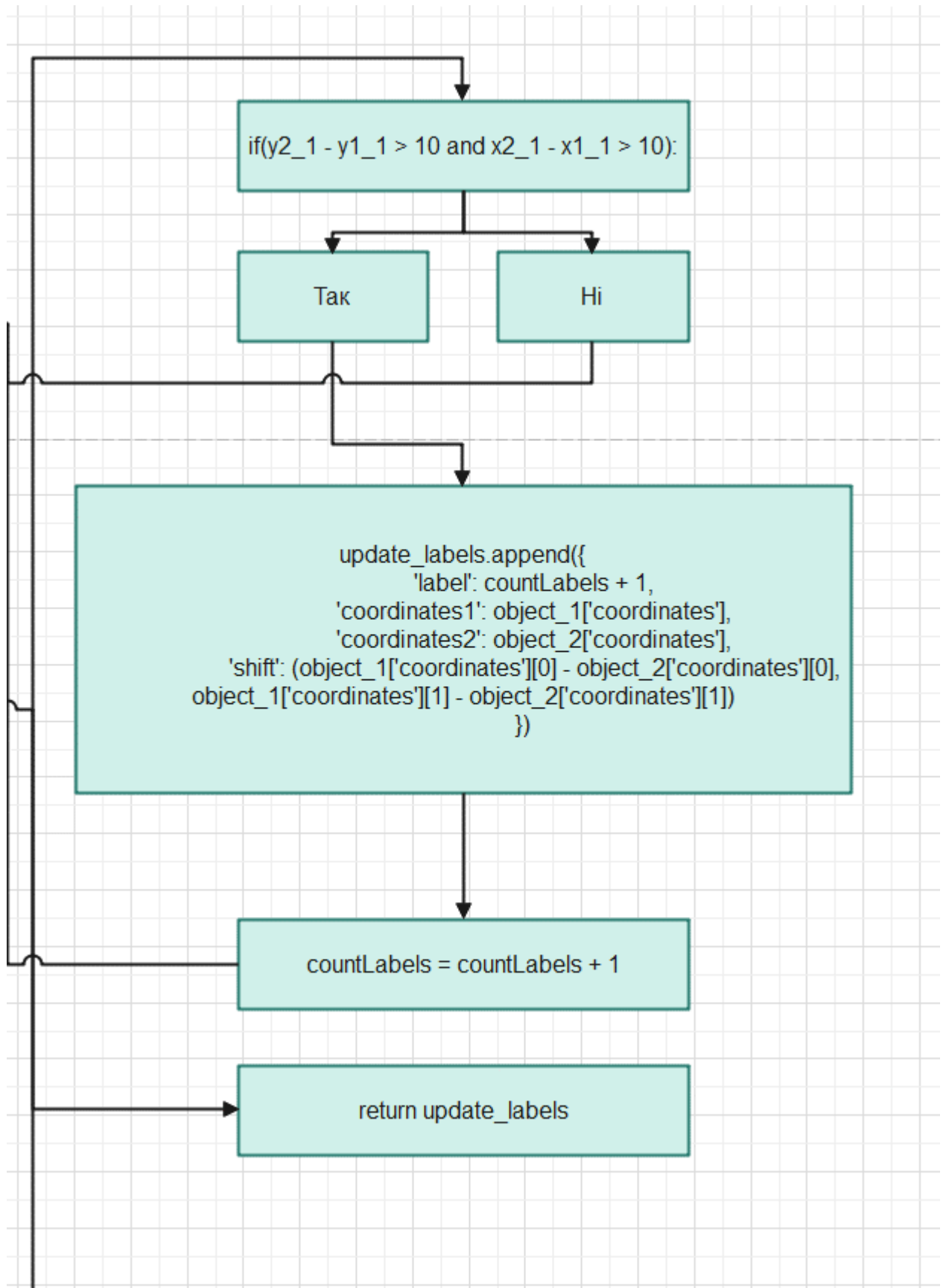


Рисунок 3.18 - Послідовності роботи модуля `draw_objects()` (Третя частина)

На рисунку 3.19 та рисунку 3.20 показана послідовність роботи модуля **find_compare_objects()**.

Модуль `find_common_objects()` перевіряє, чи схожі об'єкти на обох зображеннях, і знаходить спільні об'єкти. Основна мета цього модуля - розпізнавати об'єкти, які відображаються з розбіжністю або зсувом між двома зображеннями.

Основні параметри модуля, яка знаходить спільні об'єкти між `objects1` і `objects2`:

- `objects1`: список об'єктів на першому зображенні.
- `objects2`: список об'єктів, отриманих з другої фотографії.

Модуль починається зі створення порожнього списку спільних об'єктів, цей список міститиме інформацію про спільні об'єкти. Крім того, створюється порожній набір індексів, які вже використовуються, це використовується для відстеження індексів, які наразі використовуються для об'єктів із другого зображення.

Далі процедура обходить кожен об'єкт на першому зображенні. Для кожного об'єкта на першому зображенні об'єкт із найбільшою схожістю визначається на другому зображенні за допомогою модуля `compute_similarity()` для вимірювання подібності. Об'єкт, який найбільше схожий на найкраще, має найменшу відстань або найбільшу подібність.

Якщо зустрічається відповідний об'єкт, його індекс включається до набору `current_used_indices()`, щоб вказати, що він використовується на даний момент. Інформація про загальний об'єкт включена до списку `common_objects()`, включаючи мітку об'єкта з першого та другого зображень, їхні відповідні координати, різницю між ними та обчислену подібність.

На виході модуль створює список `common_objects()`, який містить інформацію про загальні об'єкти, які спостерігаються між двома зображеннями.

Загальна мета модуля `find_common_objects()` - знайти об'єкти, розташовані на обох зображеннях, незважаючи на те, що вони зберігаються в різних місцях або мають різні другорядні характеристики. Це може бути

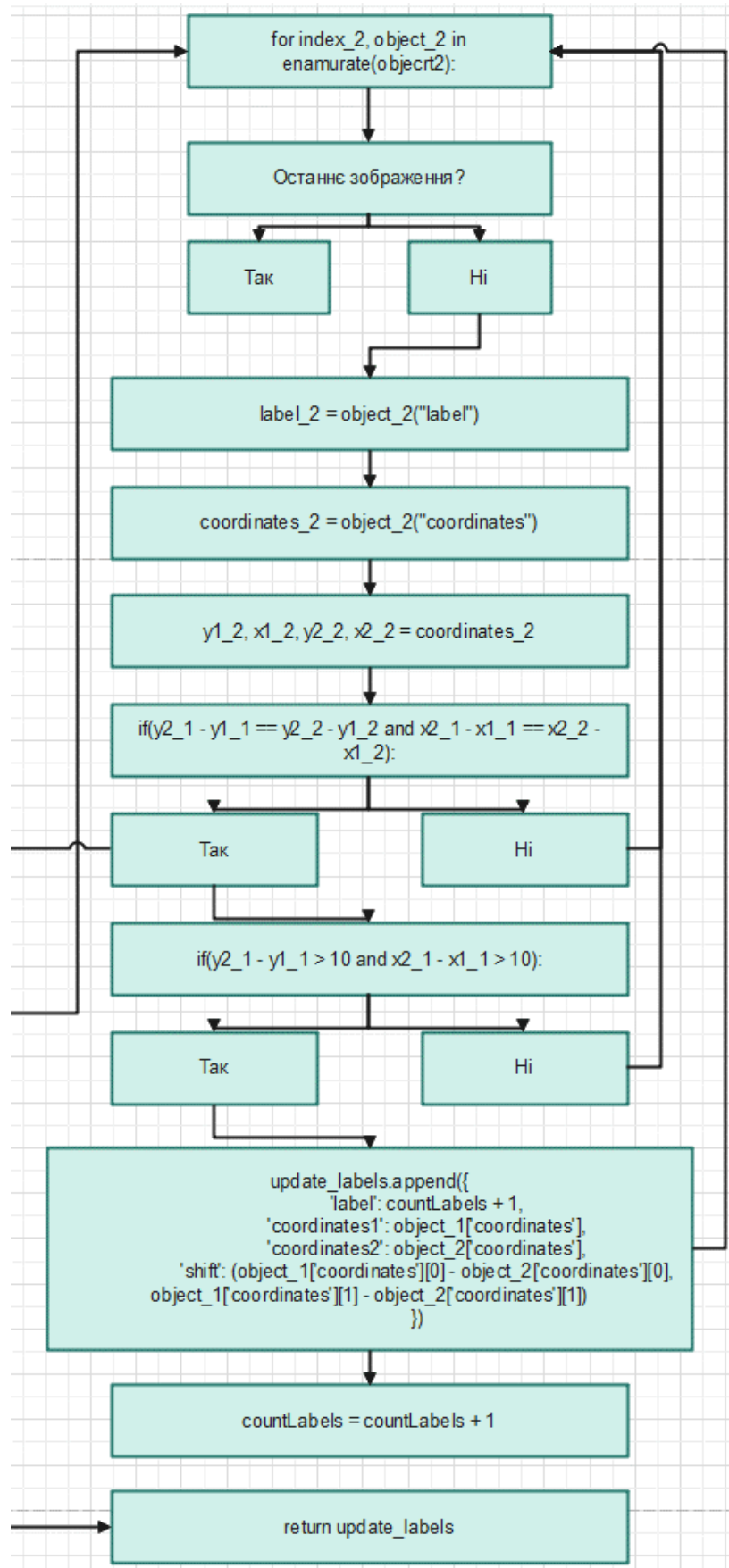


Рисунок 3.20 - Послідовності роботи модуля `find_compare_objects()` (Друга частина)

На рисунку 3.21 показана послідовність роботи модуля `set_ax_setting_text()`.

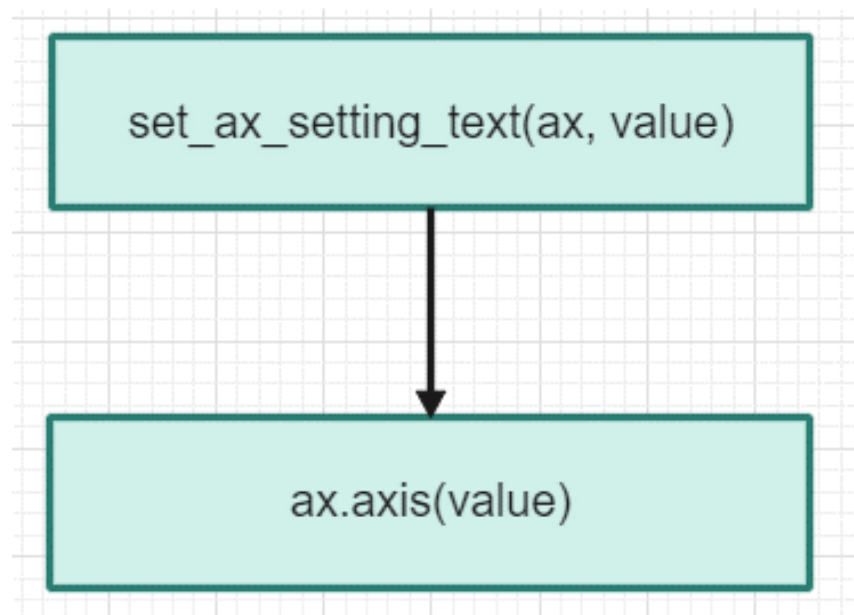


Рисунок 3.21 - Послідовність роботи модуля `set_ax_setting_text()`

На рисунку 3.22 показана послідовність роботи модуля `set_ax_setting_text()`.

Модуль `set_ax_setting_text(ax, title)` виконує налаштування графічного об'єкту `ax` типу `Axes` для відображення тексту.

Модуль має два параметра. Перший параметр відповідає за графічний об'єкт типу `Axes`, на якому виконуються налаштування. Другий параметр відповідає за заголовок, що встановлюється для графічного об'єкту `ax`.

Модуль встановлює заголовок `title` для графічного об'єкту `ax` за допомогою методу `set_title`. Вимикає сітку на графіку, встановлюючи параметр `grid` в значення `False`. Встановлює межі графіку, використовуючи метод `set`, з параметрами `xlim` (межі по осі X) і `ylim` (межі по осі Y), що встановлюються на значення від 0 до 100. Вимикає відображення позначок по осі X за допомогою методу `get_xaxis().set_visible(False)`. Вимикає відображення позначок по осі Y за допомогою методу `get_yaxis().set_visible(False)`.

Цей модуль дозволяє налаштувати графічний об'єкт типу `Axes` для відображення тексту, вимкнути сітку і позначки по осях X та Y, та встановити

межі графіку для зручного відображення текстової інформації. Він забезпечує зручне налаштування графічного об'єкту для відображення тексту на графіку.

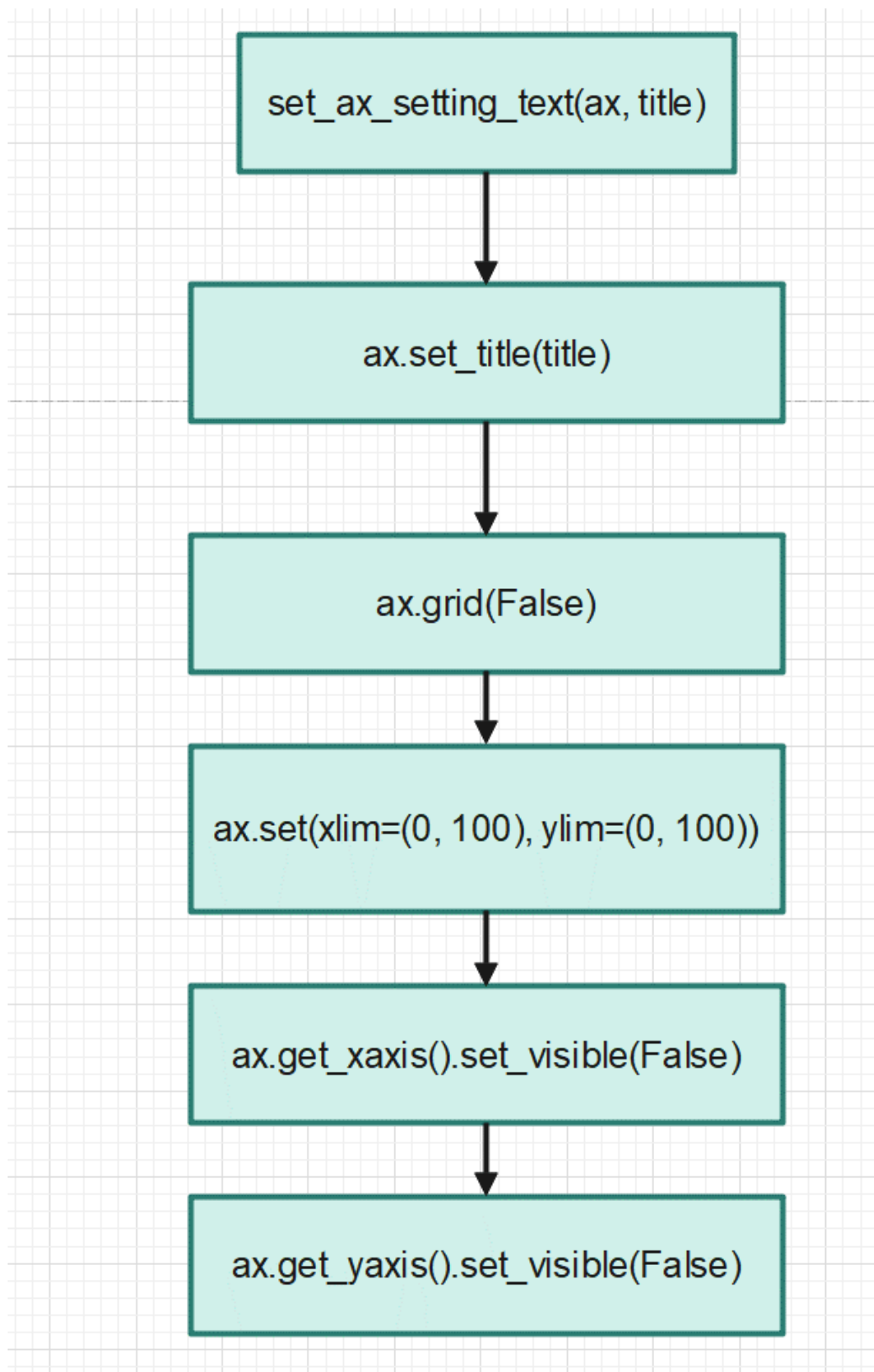


Рисунок 3.22 - Послідовність роботи модуля `set_ax_setting_text()`

На рисунку 3.23 показана послідовність роботи модуля `run_app()`. Відповідно до назви модуля, модуль відповідає за запуск застосунку.

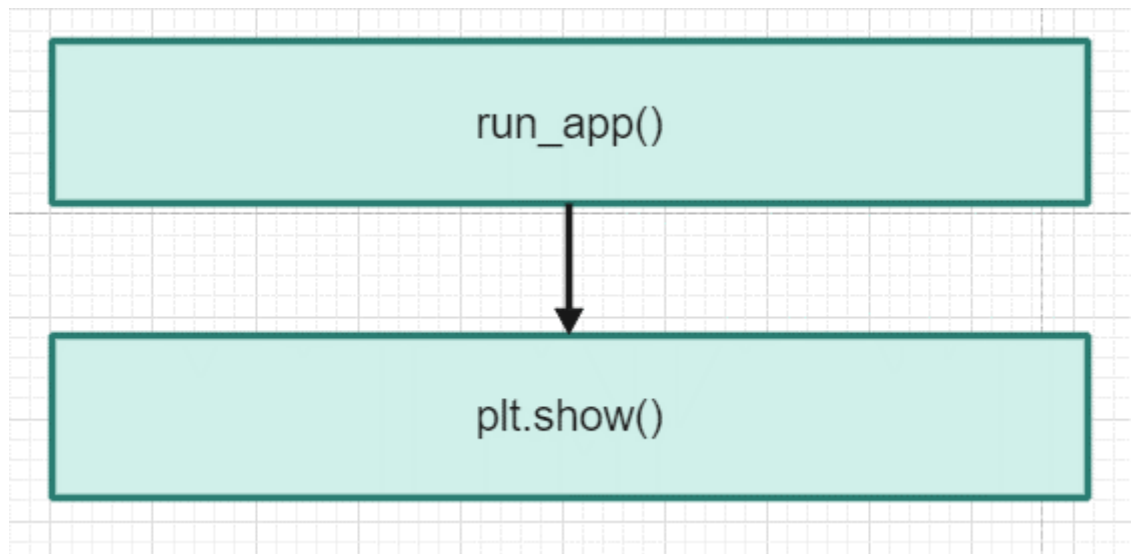


Рисунок 3.23 - Послідовність роботи модуля `run_app()`

Модуль `run_app()` відповідає за запуск відображення графіків і виведення їх на екран.

Модуль викликає функцію `plt.show()`. Ця функція виводить всі графічні об'єкти, що були побудовані за допомогою бібліотеки `matplotlib`, на екран. Вона блокує подальше виконання програми до тих пір, поки не буде закрито вікно з графіками або не буде виконано відповідну команду.

Отже, модуль `run_app()` виконує необхідні дії для запуску відображення графіків та забезпечує їх виведення на екран. Вона дозволяє користувачу переглянути результати обробки зображень та порівняння об'єктів на графічних візуалізаціях.

3.3 Опис функціональних можливостей інформаційної системи

При вхідних даних, де є білий фон та зображення з двох об'єктів, а саме, апельсин та машину, програма працює наступним чином (рисунок 3.24).

Відповідно до рисунку 3.24, видно два графіка, з двома зображеннями, де знайшлося по два об'єкта з назвою "Object 1" та "Object 2" на першому зображенні

та два об'єкти з назвою “Object 1” та “Object 2” на другому зображенні відповідно. Однакові назви свідчать про те, що об'єкти є однакові та рівні між собою. Навколо всіх об'єктів з'явився зелений прямокутник, що виділяє їх.

Також, знизу існують кнопки роботи та функціонування з графіками, такі як зміна положення, відміна дії та інші. Справа зверху на рисунку 3.24 у відповідній колонці можна побачити текст, що інформує про координати об'єкту на першому зображенні та координати зображення на другому зображенні, що змістилося, а також одиниці, на скільки по осям змістився об'єкт.

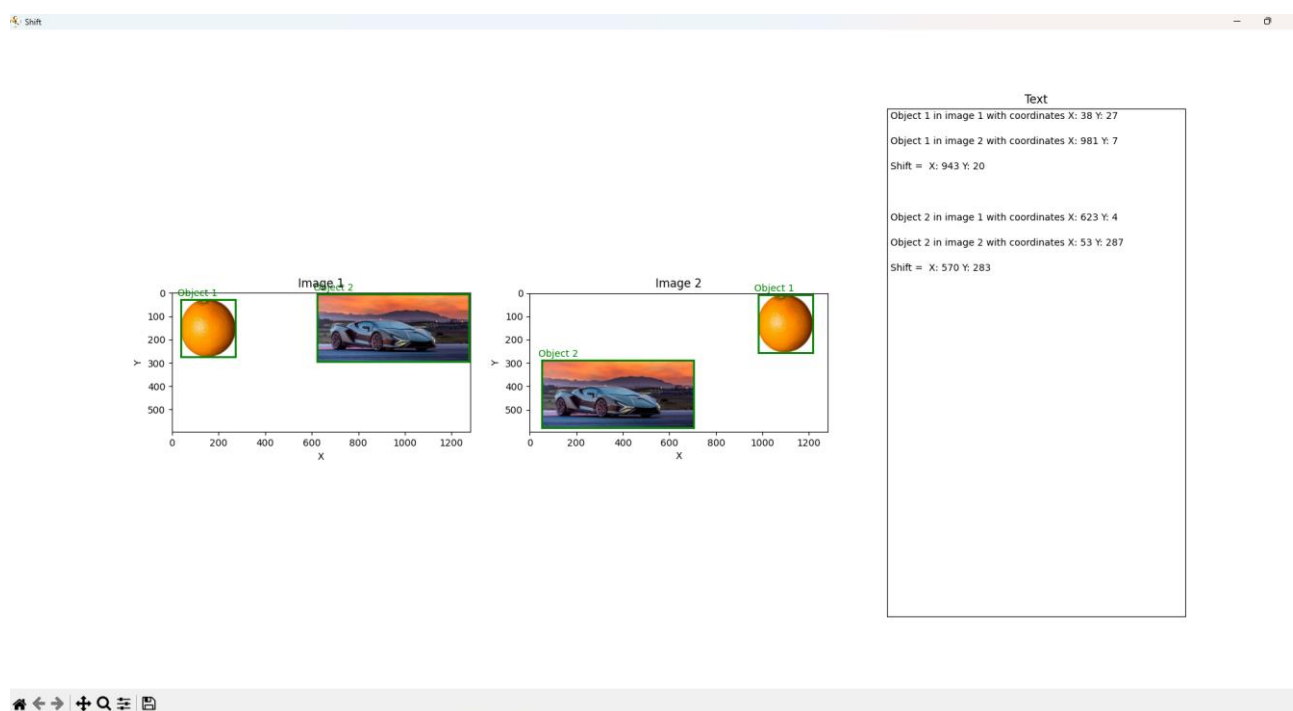


Рисунок 3.24 - Робота програми при вхідних даних з двома об'єктам

Приклад роботи програми з двома іншими об'єктами за вхідними даними (рис. 3.25)

На рисунку 3.25 представлено роботу програми з вхідними даними з двома об'єктами у різних положеннях. Програма знаходить ці об'єкти, позначає їх прямокутниками та називає. Справа у колонці з текстом, виведено які об'єкти були зміщені, на яку позицію були зміщені та на скільки одиниць були зміщені.

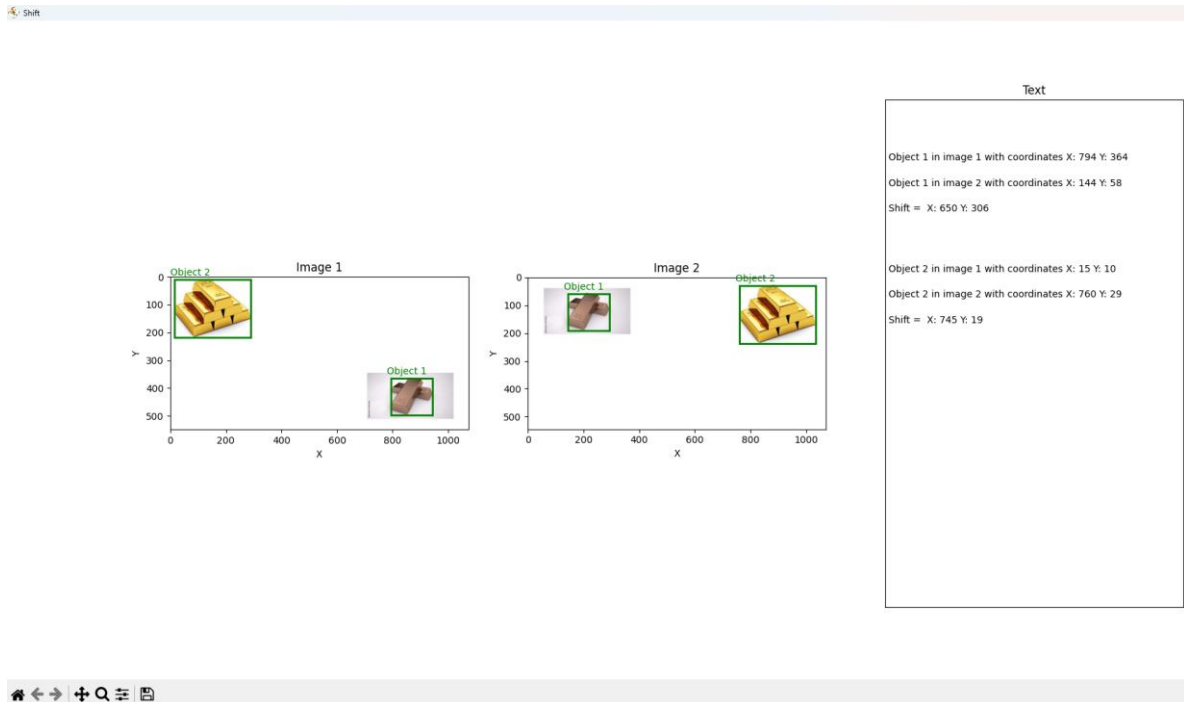


Рисунок 3.25 - Роботи програми при вхідних даних з двома об'єктами

3.4 Висновки до розділу 3

Визначена структура модулів інформаційної системи способу оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах і їх взаємозв'язок. Програмний застосунок реалізований як система з 15 модулів із використанням мови програмування Python на основі семи її бібліотек.

Визначені функціональні можливості інформаційної системи.

Проведене експериментальне тестування на кількох серіях послідовних зображень з кількома об'єктами спостереження на кожному із них підтвердило ефективність запропонованого способу оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору та коректність роботи реалізованого програмного застосунку.

Висновки

За результатами виконання кваліфікаційної роботи було розроблено спосіб оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах.

Реалізовано спосіб оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору. Суть способу полягає в опрацюванні серій послідовних зображень із визначенням на кожному з них положення об'єкту. Для визначення положення об'єкту використовується метод кластеризації, а саме метод k-means. Застосування методу кластеризації дозволяє визначення положення одразу кількох об'єктів на одиничному зображенні. Це дозволяє визначати величину зміщення кількох об'єктів одночасно за один цикл обробки однієї серії послідовної зображень, тобто оцінювати зміщення кількох об'єктів спостереження з несинхронізованим рухом без застосування ітераційної по об'єктової процедури.

Було визначена послідовність застосування способу визначення положення об'єкту на зображенні з круговим градієнтом кольору. Визначено послідовність оцінки зміщення визначених об'єктів у задачі вимірювання в технічних системах.

Визначена структура модулів системи і їх взаємозв'язок. Програмний застосунок реалізований як система з 17 модулів із використанням мови програмування Python на основі семи її бібліотек.

Визначені функціональні можливості інформаційної системи.

Проведене експериментальне тестування на кількох серіях послідовних зображень з кількома об'єктами спостереження на кожному із них підтвердило ефективність запропонованого способу оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору та коректність роботи реалізованого програмного застосунку.

Перелік посилань

1. Big data will become a key basis of competition, underpinning new waves of productivity growth, innovation, and consumer surplus - as long as the right policies and enablers are in place. [Електронний ресурс] - Режим доступу до ресурсу: <https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/big-data-the-next-frontier-for-innovation>

2. The Necessity of Measurement in Everyday Life. [Електронний ресурс] - Режим доступу до ресурсу: <https://www.bmdshapi.com/the-necessity-of-measurement-in-everyday-life-fred-b-bryant-ph-d-loyola-university-chicago/>

3. Measurement in metrology, psychology and social sciences: data generation traceability and numerical traceability as basic methodological principles applicable across sciences. [Електронний ресурс] - Режим доступу до ресурсу: <https://link.springer.com/article/10.1007/s11135-020-00970-2>

4. Recent advances and applications of deep learning methods in materials science. [Електронний ресурс] - Режим доступу до ресурсу: <https://www.nature.com/articles/s41524-022-00734-6>

5. Heritage Recording and 3D Modeling with Photogrammetry and 3D Scanning. [Електронний ресурс] - Режим доступу до ресурсу: <https://www.mdpi.com/2072-4292/3/6/1104>

6. Visual Methodologies in Qualitative Research: Autophotography and Photo Elicitation Applied to Mental Health Research. [Електронний ресурс] - Режим доступу до ресурсу: <https://journals.sagepub.com/doi/10.1177/1609406917748215>

7. Використання інформаційних технологій в навчальному процесі. [Електронний ресурс] - Режим доступу до ресурсу: http://kovtunove-nvk.edukit.sumy.ua/metodichna_robota_nv/vikoristannya_informacijnih_tehnologij_v_navchaljnomu_procesi/

8. Qualitative Research from Start to Finish. [Електронний ресурс] - Режим доступу до ресурсу: <https://eli.johogo.com/Class/Qualitative%20Research.pdf>

9. BASICS OF DIGITAL CAMERA PIXELS. [Электронный ресурс] - Режим доступа до ресурсу: <https://www.cambridgeincolour.com/tutorials/digital-camera-pixel.htm>

10. Analysis of Digital Photography Technology in the Era of Big Data. [Электронный ресурс] - Режим доступа до ресурсу: https://www.researchgate.net/publication/361335495_Analysis_of_Digital_Photography_Technology_in_the_Era_of_Big_Data

11. Optical methods for distance and displacement measurements. [Электронный ресурс] - Режим доступа до ресурсу: <https://opg.optica.org/aop/fulltext.cfm?uri=aop-4-4-441&id=241332>

12. Displacement estimation in ultrasound images using pseudo-phase. [Электронный ресурс] - Режим доступа до ресурсу: https://www.researchgate.net/publication/224466523_Displacement_estimation_in_ultrasound_images_using_pseudo-phase

13. Displacement field estimation and image segmentation using block matching enhanced by a neural network [Электронный ресурс] - Режим доступа до ресурсу: <https://www.csie.ntu.edu.tw/~fuh/personal/DisplacementFieldEstimationandImageSegmentation.pdf>

14. An evaluation of optical flow algorithms for background oriented schlieren imaging. [Электронный ресурс] - Режим доступа до ресурсу: https://www.cs.ubc.ca/labs/imager/tr/2008/Atcheson_BOS_EiF/eif08.pdf

15. Improved surface displacement estimation through stacking cross-correlation spectra from multi-channel imagery. [Электронный ресурс] - Режим доступа до ресурсу: <https://www.sciencedirect.com/science/article/pii/S2666017222000323>

16. Department of Evolutional Neural Networks for Displacement Estimation in ARFI Imaging. [Электронный ресурс] - Режим доступа до ресурсу: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8363049/>

17. Computer Vision-Based Structural Displacement Measurement Robust to Light-Induced Image Degradation for In-Service Bridges. [Електронний ресурс] - Режим доступу до ресурсу: <https://www.mdpi.com/1424-8220/17/10/2317>
18. Image downscaling via tso-occurrence learning. [Електронний ресурс] - Режим доступу до ресурсу: <https://www.sciencedirect.com/science/article/abs/pii/S1047320323000160>
19. Обробка зображень (зображення як масив пікселів). [Електронний ресурс] - Режим доступу до ресурсу: <https://dystosvita.org.ua/mod/page/view.php?id=1113>
20. Кластерний аналіз. [Електронний ресурс] - Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Кластерний_аналіз#Методи_кластеризації
21. Кластеризація методом к–середніх. [Електронний ресурс] - Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Кластеризація_методом_к–середніх
22. K-Means Clustering Algorithm. [Електронний ресурс] - Режим доступу до ресурсу: <https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning>
23. The Ultimate Guide to K-Means Clustering: Definition, Methods and Applications. [Електронний ресурс] - Режим доступу до ресурсу: <https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/>
24. Кластерний аналіз. [Електронний ресурс] - Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Кластерний_аналіз#Методи_кластеризації
25. Elbow method (clustering). [Електронний ресурс] - Режим доступу до ресурсу: [https://en.wikipedia.org/wiki/Elbow_method_\(clustering\)](https://en.wikipedia.org/wiki/Elbow_method_(clustering))
26. Центроїд. [Електронний ресурс] - Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Центроїд>

ДОДАТКИ

Додаток А

main.py

```

from PIL import Image
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as img
import cv2
import warnings

def ignore_warnings():
    warnings.filterwarnings("ignore")

def get_path_1():
    # path = r"C:\Users\Ya\Desktop\garden_1.jpg" # TRUE Gargen
    path = r"C:\Users\Ya\Desktop\slitok_1.jpg" # TRUE Slitok
    # path = r"C:\Users\Ya\Desktop\apel_car_1.jpg" # TRUE Apel / Car
    return path

def get_path_2():
    # path = r"C:\Users\Ya\Desktop\garden_2.jpg" # TRUE Gargen
    path = r"C:\Users\Ya\Desktop\slitok_2.jpg" # TRUE Slitok
    # path = r"C:\Users\Ya\Desktop\apel_car_2.jpg" # TRUE Apel / Car
    return path

def get_image(image):
    return Image.open(image).convert("L")

def get_image_PLT(image):
    return img.imread(image)

def get_image_CV(image):
    return cv2.imread(image)

def create_plots(vertical, horizontal, label):
    fig, ax = plt.subplots(vertical, horizontal, label=label)
    return ax

def find_objects(image):
    image = np.array(image)
    image = image.astype(np.float32) / 255.0

    pixel_values = image.reshape((-1, 3))

    criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 10, 1.0)
    _, labels, centers = cv2.kmeans(pixel_values, 2, None, criteria, 11,
cv2.KMEANS_RANDOM_CENTERS)

    mask = np.uint8(labels.reshape(image.shape[:2]))

    contours, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

    objects = []

    for contour in contours:
        x, y, w, h = cv2.boundingRect(contour)
        objects.append({
            'coordinates': (x, y, x+w, y+h),
            'label': len(objects) + 1
        })

```

```

    return objects

def set_full_screen():
    wm = plt.get_current_fig_manager()
    wm.window.state('zoomed')

def compute_similarity(obj1, obj2):
    x1, y1, x2, y2 = obj1['coordinates']
    x3, y3, x4, y4 = obj2['coordinates']
    intersection = max(0, min(x2,x4) - max(x1,x3)) * max(0, min(y2,y4) - max(y1,y3))
    area1 = (x2 - x1) * (y2 - y1)
    area2 = (x4 - x3) * (y4 - y3)
    union = area1 + area2 - intersection
    return intersection / union

def compare_objects(objects1, objects2):
    common_objects = []
    used_indices = set()

    for i, obj1 in enumerate(objects1):
        best_match = None
        best_score = float('inf')

        for j, obj2 in enumerate(objects2):
            if j in used_indices:
                continue

            score = compute_similarity(obj1, obj2)

            if score < best_score:
                best_score = score
                best_match = obj2

        if best_match is not None:
            used_indices.add(j)

            common_objects.append({
                'label': obj1['label'],
                'label1': obj1['label'],
                'label2': obj2['label'],
                'coordinates1': obj1['coordinates'],
                'coordinates2': obj2['coordinates'],
                'shift': (obj1['coordinates'][0] - obj2['coordinates'][0], obj1['coordinates'][1]
- obj2['coordinates'][1]),
                'score': best_score
            })

    return common_objects

def update_labels(objects1, objects2):
    update_labels = []
    countLabels = 0

    for index_1, object_1 in enumerate(objects1):
        label_1 = object_1['label']
        coordinates_1 = object_1['coordinates']

        y1_1, x1_1, y2_1, x2_1 = coordinates_1
        for index_2, object_2 in enumerate(objects2):
            label_2 = object_2['label']
            coordinates_2 = object_2['coordinates']

```

```

y1_2, x1_2, y2_2, x2_2 = coordinates_2
if(y2_1 - y1_1 == y2_2 - y1_2 and x2_1 - x1_1 == x2_2 - x1_2):
    if(y2_1 - y1_1 > 10 and x2_1 - x1_1 > 10):
        update_labels.append({
            'label': countLabels + 1,
            'coordinates1': object_1['coordinates'],
            'coordinates2': object_2['coordinates'],
            'shift': (object_1['coordinates'][0] - object_2['coordinates'][0],
object_1['coordinates'][1] - object_2['coordinates'][1])
        })
        countLabels = countLabels + 1

return update_labels

def draw_objects(objects, imageAX, image, number):
    image_array = np.array(image)

    for obj in objects:
        label = obj['label']
        coordinates = ''
        if number == 1:
            coordinates = obj['coordinates1']
        else:
            coordinates = obj['coordinates2']
        y1, x1, y2, x2 = coordinates

        if(y2 - y1 > 25 and x2 - x1 > 25):
            rect = plt.Rectangle((y1, x1), y2 - y1, x2 - x1, linewidth=2, edgecolor='green',
facecolor='none')
            imageAX.add_patch(rect)

            imageAX.text(y1 - 15, x1 - 15, f'Object {label}', color='green')

    imageAX.set_xlim(0, image_array.shape[1])
    imageAX.set_ylim(image_array.shape[0], 0)

    imageAX.set_title('Objects')
    imageAX.set_xlabel('X')
    imageAX.set_ylabel('Y')

def find_compare_object(objects1, objects2, imageText, countObjects):
    position = 0
    for index_1, object_1 in enumerate(objects1):
        label_1 = object_1['label']
        coordinates_1 = object_1['coordinates']

        y1_1, x1_1, y2_1, x2_1 = coordinates_1
        for index_2, object_2 in enumerate(objects2):
            label_2 = object_2['label']
            coordinates_2 = object_2['coordinates']

            y1_2, x1_2, y2_2, x2_2 = coordinates_2
            if(y2_1 - y1_1 == y2_2 - y1_2 and x2_1 - x1_1 == x2_2 - x1_2):
                if(y2_1 - y1_1 > 10 and x2_1 - x1_1 > 10):
                    countObjectsLocal = countObjects + 1
                    countObjects = countObjects + 1
                    imageText.text(1, (100 - index_1 - position - index_2), f'Object
{countObjectsLocal} in image 1 with coordinates X: {y1_1} Y: {x1_1}')
                    imageText.text(1, (100 - index_1 - position - index_2 - 5), f'Object
{countObjectsLocal} in image 2 with coordinates X: {y1_2} Y: {x1_2}')

```

```

        imageText.text(1, (100 - index_1 - position - index_2 - 10), f'Shift = X:
{abs(y1_1 - y1_2)} Y: {abs(x1_1 - x1_2)}')
        position = position + 20

```

```

def set_ax_setting(ax, image, title, xlabel, ylabel):
    ax.imshow(image)
    ax.set_title(title)
    ax.set_xlabel(xlabel)
    ax.set_ylabel(ylabel)

```

```

def set_ax_setting_text(ax, title):
    ax.set_title(title)
    ax.grid(False)
    ax.set(xlim=(0, 100), ylim=(0, 100))
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)

```

```

def run_app():
    plt.show()

```

```

ignore_warnings()

```

```

path1 = get_path_1()
image1 = get_image(path1)
image1PLT = get_image_PLT(path1)
image1CV = get_image_CV(path1)

```

```

path2 = get_path_2()
image2 = get_image(path2)
image2PLT = get_image_PLT(path2)
image2CV = get_image_CV(path2)
objects2 = find_objects(image2CV)
objects1 = find_objects(image1CV)

```

```

ax = create_plots(1, 3, "Shift")
set_full_screen()

```

```

common_objects = compare_objects(objects1, objects2)

```

```

updated_labels = update_labels(objects1, objects2)

```

```

countObjectsText = 0

```

```

draw_objects(updated_labels, ax[0], image1, 1)
draw_objects(updated_labels, ax[1], image2, 2)
find_compare_object(objects1, objects2, ax[2], countObjectsText)

```

```

set_ax_setting(ax[0], image1PLT, "Image 1", "X", "Y")
set_ax_setting(ax[1], image2PLT, "Image 2", "X", "Y")
set_ax_setting_text(ax[2], "Text")

```

```

run_app()

```

Додаток Б

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

СПОСІБ ОЦІНКИ ЗМІЩЕННЯ ОБ'ЄКТІВ ЗА
АНАЛІЗОМ ЗОБРАЖЕНЬ З КРУГОВИМ ГРАДІЄНТОМ
КОЛЬОРУ У ЗАДАЧІ ВИМІРЮВАННЯ В ТЕХНІЧНИХ
СИСТЕМАХ

ВИКОНАВЕЦЬ КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА:

СТУДЕНТ ГРУПИ КН-19-1 ВУСАТИЙ НІКІТА ОЛЕКСАНДРОВИЧ

КЕРІВНИК КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА:

К.Т.Н., ДОЦЕНТ КАФЕДРИ КН ПАСІЧНИК ОЛЕКСАНДР АНАТОЛІЙОВИЧ



Рисунок 1 - Слайд 1

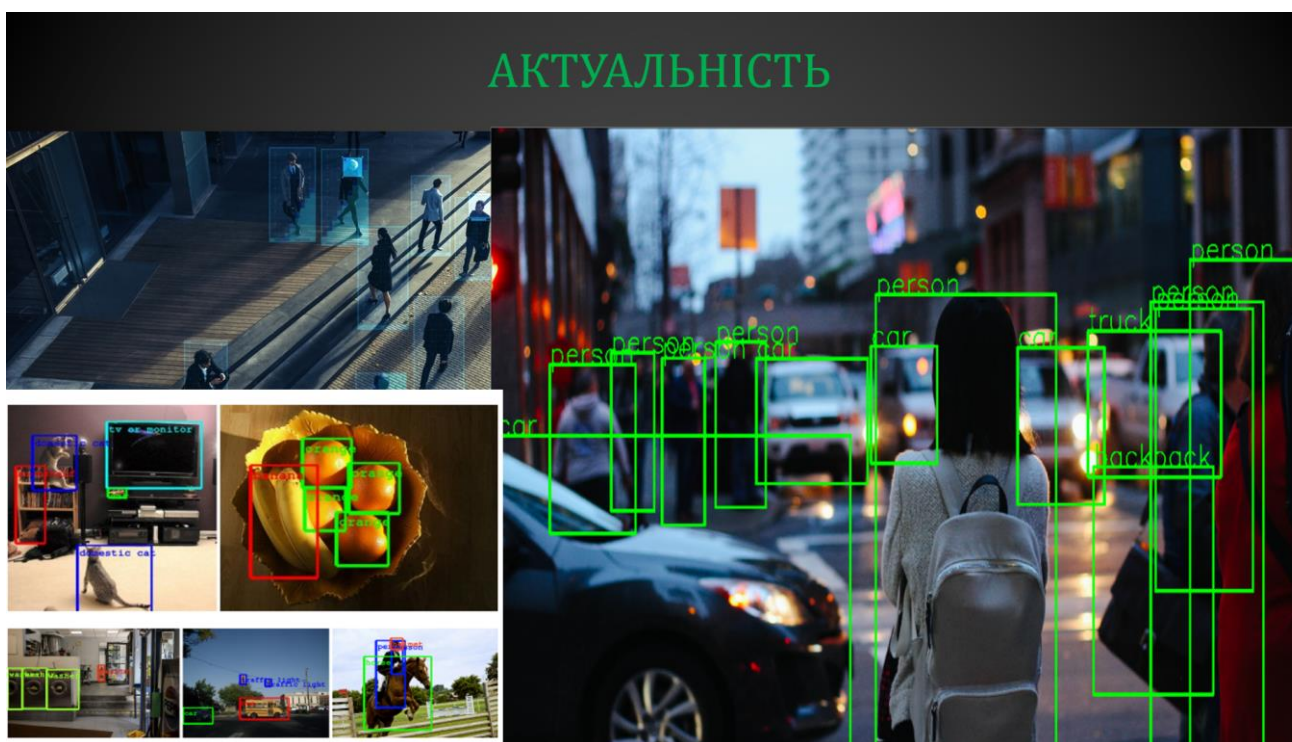


Рисунок 2 - Слайд 2

МЕТА І ЗАДАЧІ

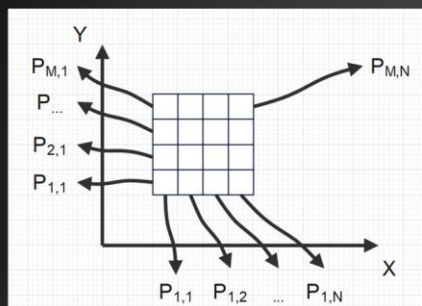
Мета кваліфікаційної роботи полягає в розробці способу оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах.

Для досягнення поставленої мети визначені наступні задачі дослідження:

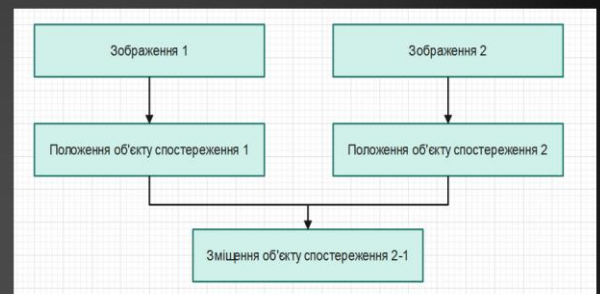
- визначити послідовність застосування способу визначення положення об'єкту на зображенні з круговим градієнтом кольору;
- визначити послідовність оцінки зміщення визначених об'єктів у задачі вимірювання в технічних системах;
- реалізувати інформаційну технологію методу підвищення точності вимірювання лінійних переміщень за аналізом зображень.
- провести експериментальне тестування інформаційної технології.

Рисунок 3 - Слайд 3

ВИЗНАЧЕННЯ ПОЛОЖЕННЯ ТА ЗМІЩЕННЯ



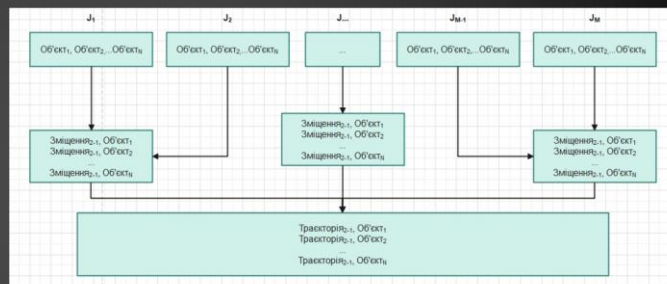
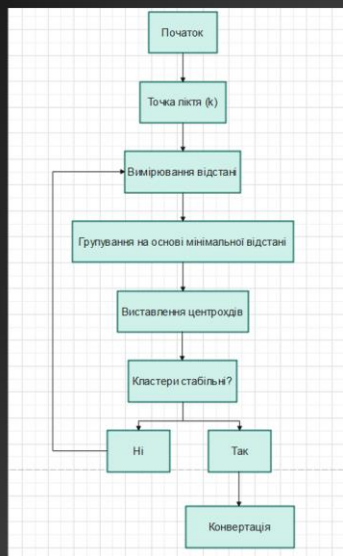
Положення об'єкту на зображенні визначається як множина положень пікселів, що його складають



Операція визначення одиничного зміщення одиничного об'єкту спостереження полягає в опрацюванні пари зображень

Рисунок 4 - Слайд 4

МЕТОД K-MEANS

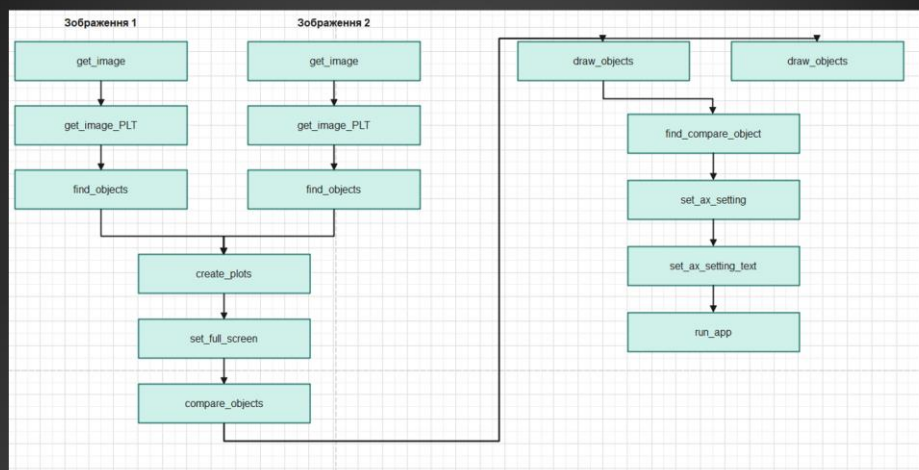


Послідовність визначення траєкторії об'єктів

Послідовність кластеризації за методом k-середніх

Рисунок 5 - Слайд 5

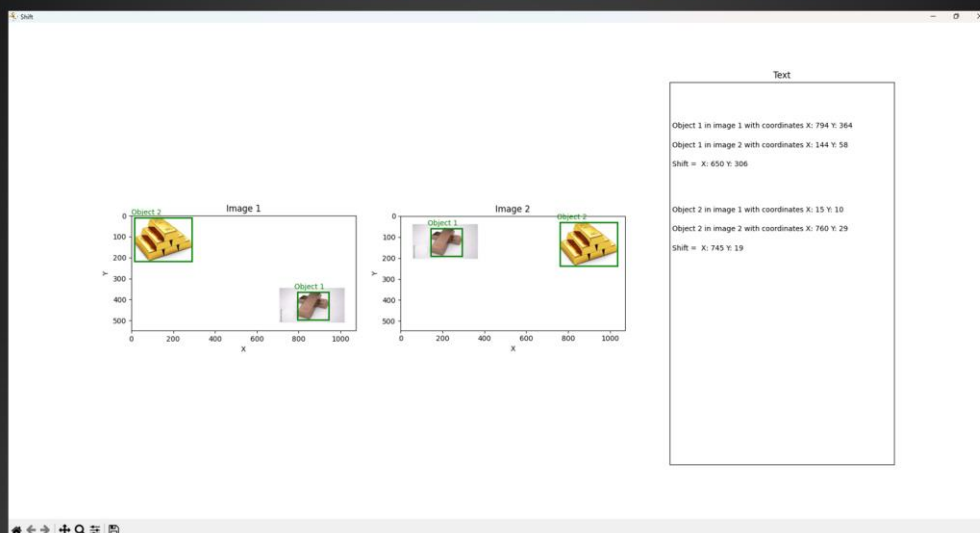
МОДУЛІ ПРОГРАМИ



Послідовність роботи модулів програми

Рисунок 6 - Слайд 6

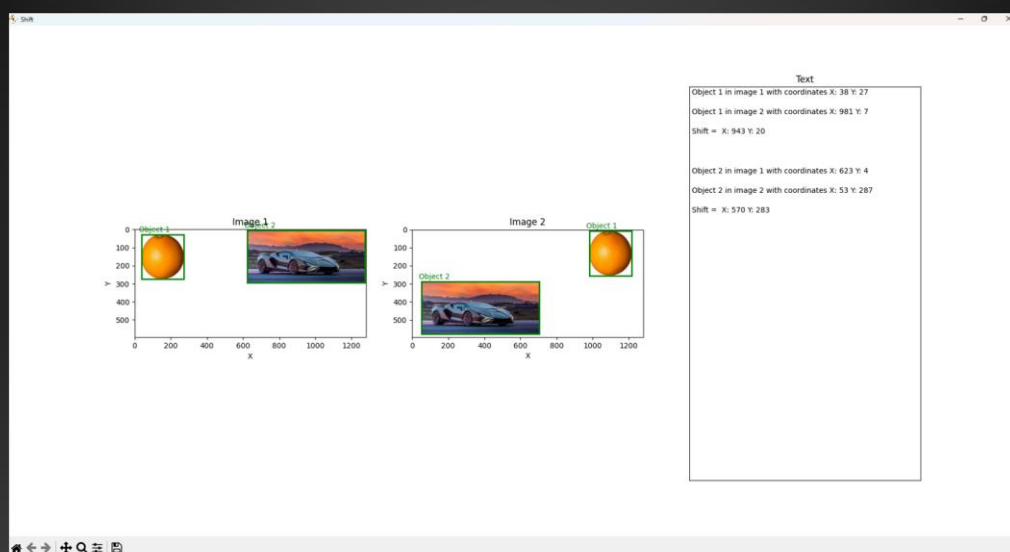
ПЕРШИЙ ЗАПУСК ПРОГРАМИ



Робота програми при вхідних даних з двома об'єктами

Рисунок 7 - Слайд 7

ДРУГИЙ ЗАПУСК ПРОГРАМИ



Робота програми при вхідних даних з двома об'єктами

Рисунок 8 - Слайд 8

ВИСНОВКИ

За результатами виконання кваліфікаційної роботи було розроблено спосіб оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачі вимірювання в технічних системах.

Реалізовано спосіб оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору. Суть способу полягає в опрацюванні серій послідовних зображень із визначенням на кожному з них положення об'єкту. Для визначення положення об'єкту використовується метод кластеризації, а саме метод k-means. Застосування методу кластеризації дозволяє визначення положення одразу кількох об'єктів на одиничному зображенні. Це дозволяє визначити величину зміщення кількох об'єктів одночасно за один цикл обробки однієї серії послідовної зображень, тобто оцінювати зміщення кількох об'єктів спостереження з несинхронізованим рухом без застосування ітераційної по об'єктовій процедури.

Було визначена послідовність застосування способу визначення положення об'єкту на зображенні з круговим градієнтом кольору. Визначено послідовність оцінки зміщення визначених об'єктів у задачі вимірювання в технічних системах.

Визначена структура модулів системи і їх взаємозв'язок. Програмний застосунок реалізований як система з 17 модулів із використанням мови програмування Python на основі семи її бібліотек.

Визначені функціональні можливості інформаційної системи.

Проведене експериментальне тестування на кількох серіях послідовних зображень з кількома об'єктами спостереження на кожному із них підтвердило ефективність запропонованого способу оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору та коректність роботи реалізованого програмного застосунку.

Рисунок 9 - Слайд 9

ДЯКУЮ ЗА УВАГУ!

СЛАВА УКРАЇНІ!
ГЕРОЯМ СЛАВА!

Рисунок 10 - Слайд 10

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 9.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилки в документах: 8%**

ID: 114324 Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА Додано в БД: 2023-05-30 Автора: Н.О. Вусатий Керівники: О.А. Пасічник Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	62450	905	7783 (12%)	103 (11%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Ім'я користувача:
Кафедра КН

ID перевірки:
1015324325

Дата перевірки:
30.05.2023 16:03:15 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
30.05.2023 16:12:58 EEST

ID користувача:
100005671

Назва документа: КН-19-1 Вусатий

Кількість сторінок: 71 Кількість слів: 9676 Кількість символів: 76175 Розмір файлу: 1.73 MB ID файлу: 1014994504

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

10.3% Схожість

Найбільша схожість: 2.96% з джерелом з Бібліотеки (ID файлу: 1009334714)

9.93% Джерела з Інтернету

836

Сторінка 73

5.91% Джерела з Бібліотеки

99

Сторінка 80

0% Цитат

Не знайдено жодних цитат

Не знайдено жодних посилань

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

8

Підозріле форматування

12
сторінок

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНИХ НАУК
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи генерованою системою виявлення текстових збігів/ідентичності/схожості:

Назва: Спосіб оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачах вимірювання в технічних системах

Автор: Вусатий Нікіта Олександрович

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: к.т.н., доцент Пасічник Олександр Анатолійович

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) за програмою Anti-Plagiarism виявлені 9 %.

2) за програмою UNICHECK виявлені 10.3%.; Найбільша схожість: 2.96% з джерелом з Бібліотеки (ID файлу: 1009334714), яке містить матеріали огляду предметної області; інші схожості є фрагментарними – містять поширені конструкції, загальновідомі терміни, скорочення та визначення.

збігів/ідентичності/схожості, складає 9 % і 10.3% відповідно, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КН

Олександр ПАСІЧНИК

Олександр МАЗУРЕЦЬ

Олександр БАРМАК



ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ



Кафедра комп'ютерних наук

**ВІДГУК НАУКОВОГО КЕРІВНИКА
на кваліфікаційну роботу бакалавра**

студента *гр. КН-19-1 Вусатого Нікіти Олександровича*

за темою *Спосіб оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачах вимірювання в технічних системах*

1. Актуальність теми

Глобальною тенденцією сьогодення є тотальне насичення всіх сфер життєдіяльності людини інтелектуальними інформаційними технологіями часто у поєднанні із найрізноманітнішими цифровими технічними пристроями. Стійкою тенденцією останнього часу є суттєве насичення, навіть повсякденного життя, пристроями отримання зображення. З іншого боку існує та постійно зростає потреба у визначенні зміщення або відсутності його для певних об'єктів спостереження. Зазначені вище обставини формують можливості вирішення згаданої задачі шляхом порівняння послідовного ряду зображень на предмет наявності відмінностей. Такий підхід визначає потенційні можливості не лише якісної оцінки наявності чи відсутності зміщення об'єкту спостереження, але й отримання кількісних значень величини цього зміщення, тобто його вимірювання та, як наслідок, можливість вирішення у такий спосіб задачі вимірювання в технічних системах. Ключовим питанням, яке обумовлює потенційну точність в задачі вимірювання, є максимально точне вимірювання положення об'єкту спостереження на тлі інших. Загалом спостерігач має справу із градієнтною зміною кольорів в умовах як природнього так й штучного освітлення, й завдання визначення положення об'єкту спостереження фактично полягає в аналізі зображень з круговим градієнтом кольору. Розробка такого способу є актуальною задачею комп'ютерних наук.

**2. Відповідність роботи предметній області Стандарту спеціальності
122 Комп'ютерні науки**

За стандартом, а саме описом предметної області, об'єктами вивчення та діяльності є математичні, інформаційні, імітаційні моделі реальних явищ, об'єктів, систем і процесів та методи і технології отримання, зберігання, обробки, передачі та використання інформації. Метою роботи є розробка способу оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачах вимірювання в технічних системах та розробка програмної системи реалізації вказаного методу. При вирішенні поставленої задачі використано математичні моделі, методи та алгоритми розв'язання теоретичних і

прикладних задач, що виникають при розробці інформаційних технологій, а саме метод кластеризації k-середніх. Тому результати виконання кваліфікаційної роботи бакалавра відповідають стандарту бакалавра спеціальності 122 – Комп'ютерні науки.

3. Професійні та особистісні якості бакалавра

При роботі над кваліфікаційною роботою бакалавра Вусатий Нікіта Олександрович проявив себе кваліфікованим фахівцем та дисциплінованим студентом, який із дотриманням визначених термінів виконував усі етапи дослідження. При написанні пояснювальної записки та при розробці прикладного програмного забезпечення виявив достатні для одержання успішного результату загальні та фахові компетентності, досягнув усіх програмних результатів навчання за напрямком «Комп'ютерні науки».

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Одержані в роботі результати є наслідком особистої діяльності студента, який самостійно виконував всі поставлені задачі.

5. Ступінь оволодіння методами дослідження

При реалізації кваліфікаційної роботи показав достатній рівень компетентностей та володіння необхідними інструментарієм, методами, методиками та технологіями предметної області комп'ютерних наук.

6. Повнота та якість розкриття теми роботи

Тема роботи в повній мірі обґрунтована й розкрита, проведено аналіз актуальності та відомих досліджень в межах обраної теми, поставлені завдання, які у роботі виконані, та розроблено програмне забезпечення для валідації та верифікації запропонованого метода.

7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу

Структура роботи та послідовність викладення логічні та відповідають поставленій меті. Викладення матеріалу послідовне, аргументоване, літературно грамотне.

8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин

Розроблений у роботі спосіб та його програмна реалізація може бути використана в технічних системах широкого спектру при оцінці зміщення в задачах вимірювання для різноманітних об'єктів спостереження.

9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота

Враховуючи високий рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Керівник



к.т.н., доцент каф. КН Олександр ПАСІЧНИК



РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента гр. КН-19-1 Вусатого Никіти Олександровича
за темою: Спосіб оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачах вимірювання в технічних системах

1. Актуальність обраної теми

Суцільна цифровізація є характерною ознакою та панівною тенденцією сьогодення. Широке розповсюдження цифрових засобів фото та відео фіксації дозволяє їх використання для різноманітних задач, однією яких є оцінка зміщення об'єктів спостереження. Посідання цифрових пристроїв із відповідними технологіями дозволяють у багатьох випадках відмовитися від раніше традиційних способів вирішення цієї задачі за допомогою прямих інструментальних методів, а сам процес зробити швидким та ефективним

2. Повнота розкриття мети та завдань роботи

Мета роботи розкрита повністю та у повному обсязі вирішено усі завдання

3. Зміст кожного розділу роботи

В першому розділі аналіз сучасних методів визначення положення об'єктів за аналізом зображень та інформаційного забезпечення цієї задачі. Сформульовано мету та визначено задачі.

В другому розділі реалізовано спосіб оцінки зміщення об'єктів за аналізом зображень з круговим градієнтом кольору у задачах вимірювання в технічних системах з використанням методу k-середніх та визначенням функціональної структури системи.

В третьому розділі виконано програмну реалізацію способу.

4. Оцінка розробленої інформаційної системи, її практична цінність

Розроблений у роботі спосіб та його програмна реалізація може бути використана в технічних системах широкого спектру при оцінці зміщення в задачах вимірювання для подіб'єктних різнокласових спостереження.

5. Якість оформлення кваліфікаційної роботи бакалавра

Структура роботи та послідовність викладення логічні та відповідають поставленій меті. Викладення матеріалу послідовне, аргументоване, літературно грамотне. Дотримано вимоги що до структури та оформлення.

6. Недоліки кваліфікаційної роботи бакалавра

В роботі не зазначено, яким чином виконується перехід від умовних одиниць зміщення до величин регламентованих міжнародною системою СІ.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота.

Враховуючи рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Рецензент

[Signature]

*к.т.н., доцент каф. АКТ та Р
Л. Корсунька*