

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Радіотехнічний факультет  
Кафедра радіотехнічних пристроїв**

«На правах рукопису»  
УДК 621.396.621:004.896.72

До захисту допущено:  
Завідувач кафедри  
\_\_\_\_\_ Сергій ЖУК  
«\_\_» \_\_\_\_\_ 2022р.

**Магістерська дисертація  
на здобуття ступеня магістра  
за освітньо-професійною програмою  
«Радіотехнічні комп'ютеризовані системи»  
зі спеціальності 172 «Телекомунікації та радіотехніка»  
на тему: «Алгоритм виявлення об'єктів в системах технічного зору»**

Виконав (-ла):  
студент (-ка) II курсу, групи РС-11мп  
Перцов Вадим Миколайович \_\_\_\_\_

Керівник:  
Ст.викладач, к.т.н.  
Вишневий Сергій Валерійович \_\_\_\_\_

Рецензент:  
Доцент, к.т.н.  
Перегудов Сергій Миколайович \_\_\_\_\_

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць  
інших авторів без відповідних  
посилань.  
Студент (-ка) \_\_\_\_\_

Київ – 2022 року

**Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»**

**Радіотехнічний факультет**

**Кафедра радіотехнічних систем**

Рівень вищої освіти – другий (магістерський)

Спеціальність – 172 «Телекомунікації та радіотехніка»

Освітньо-професійна програма «Радіотехнічні комп'ютеризовані системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Сергій ЖУК

«\_5\_»\_вересня\_2022р.

**ЗАВДАННЯ**

**на магістерську дисертацію студенту**

**Перцову Вадиму Миколайовичу**

1. Тема дисертації «Алгоритм виявлення об'єктів в системах технічного зору» \_\_\_\_\_,

науковий керівник дисертації Вишневий Сергій Валерійович, к.т.н., доц.,

затверджені наказом по університету від «\_09\_»\_листопада\_2022 р.  
№\_4137-с\_\_\_\_\_

2. Термін подання студентом дисертації: 20 грудня 2022р.

3. Об'єкт дослідження: особливості виявлення об'єктів в системах технічного зору

4. Вихідні дані: зображення шуканої цілі, напрямки руху шуканої цілі\_\_\_\_\_

5. Перелік завдань, які потрібно розробити: розглянути поняття технічного зору; виконати аналіз алгоритмів виявлення об'єктів; розглянути існуючі алгоритми виявлення об'єктів в системах технічного зору; виконати аналіз методу Лукаса-Канаде; виконати модернізацію методу Лукаса-Канаде; виконати аналіз методу виявлення за допомогою кольорових фільтрів; провести порівняння алгоритмів виявлення; розглянути можливості практичного використання розглянутих алгоритмів в сучасних реаліях\_\_\_\_\_

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: презентація 19 слайдів

7. Орієнтовний перелік публікацій: одна публікація.

8. Дата видачі завдання 5 вересня 2022р.

#### Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз існуючого матеріалу	21.10.2022	
2	Аналіз актуальності обраної теми	25.10.2022	
3	Аналіз структури і принципів поняття технічного зору	28.10.2022	
4	Аналіз основних методів виявлення об'єктів в системах технічного зору	3.11.2022	
5	Аналіз поняття оптичний потік	9.11.2022	
6	Аналіз методів обчислення оптичного потоку	15.11.2022	
7	Метод Лукаса-Канаде	24.11.2022	
8	Метод кольорових фільтрів	1.12.2022	
9	Висновки та стартап-проект	10.12.2022	

Студент

Вадим ПЕРЦОВ

Науковий керівник

Сергій ВИШНЕВИЙ

## РЕФЕРАТ

Магістерська дисертація на тему «Алгоритм виявлення об'єктів в системах технічного зору» складається з 59 сторінок, що включають в себе 6 розділів, 2 додатки, 21 ілюстрація, 8 таблиць, 12 бібліографічних найменування за переліком джерел посилань.

В роботі розглянуті алгоритми виявлення об'єктів в системах технічного зору, а саме алгоритм Лукаса-Канаде та алгоритм виявлення за допомогою кольорових фільтрів. Метод Лукаса-Канаде було модифіковано для більш швидкої та якіснішої його роботи.

**Метою дослідження** являється підвищення ефективності виявлення об'єктів, шляхом знаходження оптимального методу виявлення та модифікації його.

**Об'єкт дослідження** – алгоритм Лукаса-Канаде та алгоритм кольорових фільтрів.

**Предмет дослідження** – алгоритми виявлення об'єктів в системах технічного зору.

Для виконання заданої задачі потрібно виконати такі завдання:

- Провести аналіз існуючих алгоритмів виявлення об'єктів;
- Знайти оптимальний алгоритм виявлення об'єктів;
- Оцінити ефективність алгоритмів;
- Оцінити можливості практичної реалізації алгоритмів.

**Ключові слова:** технічний зір, виявлення об'єктів, особливі точки, оптичний потік, Лукас-Канаде, HSV колірна модель.

## ABSTRACT

The master's thesis on the topic "Object detection algorithm in technical vision systems" consists of 59 pages, which include 6 sections, 2 appendices, 21 illustrations, 8 tables, 12 bibliographic names according to the list of reference sources.

The work deals with object detection algorithms in technical vision systems, namely the Lucas-Canada algorithm and the detection algorithm using color filters. The Lucas-Kanady method has been modified to make it work faster and better.

**The purpose of the research** is to increase the efficiency of object detection by finding the optimal method of detection and modifying it.

**The object of the study** is the Lucas-Canady algorithm and the algorithm of color filters.

**The subject of research** is object detection algorithms in technical vision systems.

To complete the given task, you need to perform the following tasks:

- Analyze existing object detection algorithms;
- Find the optimal object detection algorithm;
- Evaluate the effectiveness of algorithms;
- Evaluate the possibilities of practical implementation of algorithms.

**Key words:** technical vision, object detection, special points, optical flow, Lucas-Kanade, HSV color model.

## ЗМІСТ

ВСТУП .....	8
1 ТЕОРИТИЧНІ ОСНОВИ МЕТОДІВ ВИЯВЛЕННЯ ТА СТЕЖЕННЯ ЗА РУХОМИМ ОБ'ЄКТОМ У ВІДЕОПОСЛІДОВНОСТІ .....	9
1.1 Системи технічного зору .....	10
1.1.1 Предметні області .....	11
1.1.2 Завдання технічного зору .....	12
1.2 Основні методи виявлення рухомих об'єктів .....	15
2 МЕТОД ВИЯВЛЕННЯ ОБ'ЄКТІВ ІЗ ВИКОРИСТАННЯМ ОПТИЧНОГО ПОТОКУ .....	17
2.1 Оптичний потік .....	17
2.2 Алгоритми визначення оптичного потоку та його застосування ..	19
2.3 Обчислення дійсного оптичного потоку методом Лукаса-Канаде	21
2.4 Обчислення штучного оптичного потоку .....	23
2.5 Ідентифікація динамічних точок та їх фільтрація .....	23
2.6 Модифікація метод .....	26
2.7 Недоліки методу оптичного потоку .....	28
3 МЕТОД ВИЯВЛЕННЯ ОБ'ЄКТІВ ЗА ДОПОМОГОЮ КОЛЬОРОВИХ ФІЛЬТРІВ .....	30
3.1 Колірна модель HSV .....	31
3.2 Конвертація RGB у HSV .....	32
3.3 Недоліки методу кольорових фільтрів .....	33
4 РЕАЛІЗАЦІЯ ТА ПОРІВНЯННЯ МЕТОДІВ .....	35
4.1 Вибір інструмента .....	35
4.2 Ілюстрація роботи методу із використанням оптичного потоку ...	35

4.2.1 Тестування методу із використанням оптичного потоку .....	37
4.3 Висновки до методу Лукаса-Канаде .....	40
4.4 Ілюстрація роботи методу за допомогою кольорових фільтрів .....	40
4.5 Висновки до методу кольорових фільтрів .....	46
4.6 Порівняння методів виявлення рухомих об'єктів в системах технічного зору .....	46
5 СТАРТАП .....	49
5.1 Маркетинговий аналіз стартап-проєкту .....	49
5.2 Розроблення маркетингової програми стартап-проєкту .....	51
5.3 Висновки до стартап-проєкту .....	52
ВИСНОВКИ .....	53
СПИСОК ЛІТЕРАТУРИ .....	54
ДОДАТОК А .....	56
ДОДАТОК Б .....	57

## ВСТУП

Детектування об'єктів, що рухаються, стало важливою частиною вирішення безлічі прикладних завдань у різних сферах. Прикладом може бути завдання стеження за технічним станом і функціонуванням об'єктів, спостереженням за потоком руху транспорту. Безпілотні літальні апарати разом з методами детектування об'єктів, що рухаються, можуть широко застосовувати у військових цілях. Причини очевидні – отримання інформації про противника без ризику втрат людини. Існує ще велика кількість завдань та сфер застосування літальних апаратів та алгоритмів детектування об'єктів.

У цій роботі розглядається метод визначення об'єктів, що рухаються, і напрями їх руху на відеозображенні. Існує кілька підходів для детектування об'єктів: методи, що виключають фон, методи спостереження за особливими точками, методи формування фону, моделі з автоматичним переміщенням. Алгоритми, що використовують стаціонарні камери для зйомки, використовують підхід відокремлення динамічних об'єктів на кадрі від фону кадру. На основі статичних точок, які не змінюють свого положення на зображенні, можна визначити контури, що рухаються.



# 1 ТЕОРИТИЧНІ ОСНОВИ МЕТОДІВ ВИЯВЛЕННЯ ТА СТЕЖЕННЯ ЗА РУХОМИМ ОБ'ЄКТОМ У ВІДЕОПОСЛІДОВНОСТІ

У цьому розділі йтиметься про алгоритми, що застосовуються для виявлення об'єктів та стеження за ними. Буде проаналізовано можливі варіанти їх побудови, а також розглянуті конкретні алгоритми, реалізовані в пакеті прикладних програм MATLAB, які були використані при реалізації підсумкового алгоритму виявлення об'єкта з відеопослідовності.

Основним завданням, яке вирішують системи відеоспостереження, є виділення «об'єктів інтересу» та відстеження їх руху у наступних кадрах. В рамках даної роботи як «об'єкт інтересу» виступатимуть транспортні засоби, рух яких буде відстежуватися і на основі отриманих даних визначатиметься необхідна інформація. Кадри з відеопотоку можуть аналізуватися та оброблятися з допомогою різних методів або алгоритмів виявлення об'єктів, застосування яких залежить від конкретних ситуацій та умов. Усі ці алгоритми спрямовані на вирішення задачі виділення областей руху на відео, яка є одним із класичних завдань комп'ютерного зору. На вході є послідовність кадрів  $I_1, I_2, \dots, I_N$  деякого відео  $V$ :

$$I_k = \{I_k(x, y), 0 \leq x < width, 0 \leq y < height\} \quad (1.1)$$

$$k = \overline{1, N}$$

де  $width$  — ширина кадру,  $height$  — висота кадру,  $I_k(x, y)$  — вектор фіксованої розмірності.

У разі напівтонового зображення інтенсивність пікселя у точці  $(x, y)$  —  $I_k(x, y)$  скалярна величина. Якщо вхідне зображення встановлено у форматі RGB, то  $I_k(x, y)$  буде вектором з розмірністю, що дорівнює трьом, кожна компонента якого визначає інтенсивність пікселя відповідним каналом.

Вирішенням цього завдання є сукупність областей зображення для кожного кадру відео, в яких відбувається рух одного чи кількох об'єктів. Таким чином, в результаті обробки відео необхідно сформувати набір бінарних

зображень, у яких білі пікселі (інтенсивність 255) відповідають пікселям, що належать рухомим об'єктам, а чорні (інтенсивність 0) – пікселям фону:

$$M_k(x, y) = \begin{cases} 255, (x, y) \\ 0, (x, y) \end{cases}, k = \overline{1, N} \quad (1.2)$$

Для того щоб під час отримання набору бінарних зображень відстежувати положення об'єкта, застосовуються алгоритми, що відстежують траєкторію руху.

Таким чином, можна побудувати узагальнений алгоритм виявлення об'єктів з відео та стеження за ними, який матиме такий вигляд:

Крок 1. Виявлення об'єктів, що рухаються на відео.

Крок 2. Виділення з об'єктів, що рухаються, об'єктів інтересу.

Крок 3. Відстеження траєкторії руху об'єктів

### 1.1 Системи технічного зору

Термін «технічний зір» має безліч пов'язаних областей: машинний зір, розпізнавання зорових образів, аналіз зображень тощо. Однак, сенс, що ховається за всіма цими визначеннями - це спроба навчити комп'ютер бачити світ очима людини, сприймати його як людина і виконувати у зв'язку з цим різні дії так само, як робила б це людина, тим самим підмінюючи або повністю виключаючи її. Необхідність у цьому виникає у ситуаціях, пов'язаних із ризиком для життя і буває обумовлена особливостями людини як живого організму, якому властиво швидко втомлюватися, пропускати через себе обмежений обсяг інформації, а також обробляти дані із відносно низькою швидкістю [1]. Процес комп'ютерного зору є складним технологічним ланцюжком, включає отримання цифрового зображення, обробку зображення з метою виділення значної інформації на зображенні та аналіз цього попередньо обробленого зображення для вирішення певного завдання. Звичайно, ідеальним буде створення універсальної самонавчаючої системию. Керуючись такими високими цілями, розробники в галузі комп'ютерного зору сьогодні вирішують непрості завдання. Можна сказати, що область

комп'ютерного зору має недовгу за мірками фундаментальних наук, але дуже бурхливу історію зародження та розвитку.

Актуальність роботи обумовлюється тим, що автоматична обробка та аналіз візуальної інформації стають дедалі актуальнішими питаннями з допомогою зростання складності розв'язуваних науково-технічних завдань. Технології технічного зору використовуються у дуже затребуваних галузях науки і техніки, таких як автоматизація технологічних процесів, підвищення продуктивності праці, підвищення якості виробів, контроль виробничого обладнання, інтелектуальні робототехнічні комплекси, системи управління рухомими апаратами, біомедичні дослідження та багато інших.

Практична значимість роботи полягає в тому, що за її підсумками сформульовано рекомендації про можливість та особливості застосування досліджуваних технологій виявлення об'єктів у розв'язанні задач технічного зору.

### **1.1.1 Предметні області**

Комп'ютерний зір, обробка зображень, аналіз зображень, зір робота та технічний зір є тісно пов'язаними предметними областями. Якщо поглибитися в ці області, то ми побачимо значні збіги в їх застосуванні та техніках, які вони використовують. Це має на увазі, що основні технічні прийоми, які використовуються та розробляються, є більш-менш ідентичними. Можна зробити висновок, що існує лише одна область знань, яка має різні назви.

З іншого боку, здається необхідним представляти себе виключно до однієї з цих областей. Тому представлені різні напрямки, що відрізняють одну область від іншої. Наступні напрямки є найважливішими:

1. Обробка зображень та аналіз зображень в основному зосереджені на роботі з двовимірними зображеннями, тобто на перетворенні одного зображення на інше. Наприклад, попиксельні операції збільшення контрастності, операції по виділення країв, усунення шумів або геометричні перетворення, такі як обертання зображення. Дані

операції припускають, що обробка/аналіз зображення діють незалежно від змісту самих зображень.

2. Комп'ютерний зір зосереджується на обробці тривимірних сцен, наприклад, відновлення структури або іншої інформації про тривимірну сцену по одному або кільком зображенням. Комп'ютерний зір часто залежить від більш-менш складних припущень щодо того, що подано на зображеннях.
3. Технічний зір зосереджується переважно в промисловості. Так, в автономних роботах можуть використовуватися системи візуальної перевірки та вимірів.
4. Також існує область знань, названа візуалізацією, яка спочатку була пов'язана з процесом створення зображень, але іноді мала справу з обробкою та аналізом. Наприклад, рентгенографія працює з аналізом відеоданих медичного застосування.
5. Розпізнавання образів є областю, яка використовує різні методи для отримання інформації з відеоданих, в основному, засновані на статистичному підході. Значна частина цієї галузі присвячена практичному застосуванню цих методів.

### **1.1.2 Завдання технічного зору**

В цілому, до завдань систем технічного зору входить отримання цифрового зображення, обробка зображення з метою виділення значимої інформації на зображенні та математичний аналіз отриманих даних для вирішення поставлених задач. Безліч завдань, які вирішує технічний зір, умовно можна поділити на чотири групи:

1. Розпізнавання положення. Ціль технічного зору в даному застосуванні - визначення просторового розташування (місце розташування об'єкта щодо зовнішньої системи координат) або статичного положення об'єкта (у якому положенні знаходиться об'єкт щодо системи координат з початком відліку в межах самого об'єкта) та передача

інформації про положення та орієнтацію об'єкта в систему управління чи контролер. Прикладом такої програми може бути вантажно-розвантажувальний робот, перед яким стоїть завдання переміщення об'єктів різної форми. Інтелектуальне завдання технічного зору полягає, наприклад, у визначенні оптимальної базової системи координат та її центру для локалізації центру тяжіння деталей. Отримана інформація дозволяє роботу захопити деталь належним чином та перемістити її у належне місце.

2. Інспекція. У програмах, пов'язаних з інспекцією, мета технічного зору - підтвердити певні властивості, наприклад, наявність або відсутність етикетки на пляшці, болтів для проведення операції складання, шоколадних цукерок у коробці або наявність різних дефектів.
3. Вимірювання. У програмах даного типу основне завдання полягає у вимірі різних фізичних параметрів об'єкта. Прикладом фізичних параметрів може бути лінійний розмір, діаметр, кривизна, площа, висота та кількість. Приклад реалізації цієї задачі - вимірювання різних діаметрів скляної пляшки.
4. Ідентифікація. У задачах ідентифікації основне призначення відеокамери - зчитування різних кодів (штрих-кодів, 2D-кодів тощо) з метою їх розпізнавання засобами камери або системним контролером, а також визначення різних буквено-цифрових позначень. Крім того, до завдань цієї групи можна віднести системи, що виконують завдання безпеки, такі як ідентифікація особистості та техніки, детектори руху.

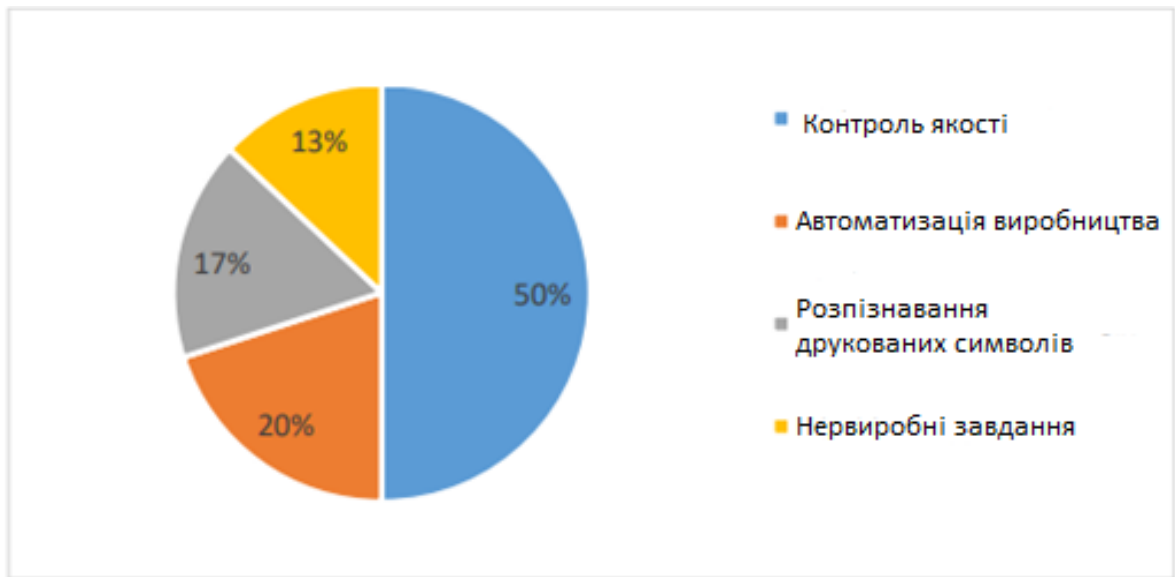


Рисунок 1.1 — Співвідношення задач, які вирішуються системами технічного зору

50% всіх систем технічного зору експлуатуються у завданнях контролю якості, тобто. вирішують інспекційні завдання технічного зору (рисунок 1.1). Це перш за все візуальний контроль за процесом збирання, кольором та якістю поверхні продукції, зовнішнім виглядом та чистотою упаковки, правильністю та розбірливістю етикеток, рівнем рідини у всілякій тарі тощо. Приблизно 10% цих завдань виконуються системами тривимірного зору. Окрема область використання систем технічного зору на виробництві - проведення різноманітних візуальних вимірювань параметрів технологічних процесів та, зокрема, визначення розмірів предметів, тобто розв'язання задач виміру.

20% попиту посідає на системи технічного зору для проектів автоматизації виробництва та впровадження промислових роботів. Такі системи технічного зору спрощують різні види високоточної діяльності (складання та розбирання, фасування, фарбування, зварювання, утилізація), полегшують транспортування вантажів, застосовуються в системах обліку, маркування, реєстрації та сортування продукції. Також інспекційні завдання та завдання розташування для правильної роботи робота.

17% всіх продажів систем технічного зору складають широко відомі та добре працюючі OCR (Системи оптичного розпізнавання символів (Optical

Character) Recognition - OCR) які призначені для автоматичного набору друкованих документів в комп'ютер з системою розпізнавання друкованих символів та штрих-кодів. Рішення задачі ідентифікації.

Ринок систем технічного зору для невиробничих (розважальних, побутових, дослідницьких) роботів становить 13%.

Невеликий відсоток ринку посідає на системи віртуальної реальності, що пропонують якісно новий інтерфейс "людина - комп'ютер", заснований на розпізнаванні обличчя і жестів, на системи, що забезпечують виконання завдань безпеки, смисловий аналіз мультимедійних даних тощо.

## **1.2 Основні методи виявлення рухомих об'єктів**

Виявлення об'єктів є не від'ємною задачею в галузі комп'ютерного зору. Остатнім часом на неї покладають велику надію. Багато фахівців комп'ютерного зору намагаються розробити універсальний алгоритм, який би дав би можливість виявляти будь-які об'єкти на відеозображенні. Але як показала практика, це не цілком можливо. Причина в тому, що різні умови задачі вимагають різні підходи для її розв'язку. Дві головні причини які можуть виникати при реалізації виявлення об'єктів наступні:

1. Продуктивність апаратури на якій реалізовується задача виявлення об'єктів відеозображення;
2. Умови навколишнього середовища в якому здійснюється процес виявлення об'єктів з відео-камери;

Основними методами виявлення об'єктів є наступні методи, а саме:

1. Виявлення об'єктів за допомогою кольорових фільтрів;
2. Виявлення рухомих об'єктів за допомогою між-кадрової різниці;
3. Виявлення об'єктів за допомогою їхнього еталону через особливі точки.

В даній роботі, буде розглянуто два з них, а саме метод виявлення за допомогою кольорових фільтрів та виявлення за допомогою еталону через

особливі точки. На основі цих методів, буде проведено порівняльний аналіз та моделювання.



## 2 МЕТОД ВИЯВЛЕННЯ ОБ'ЄКТІВ ІЗ ВИКОРИСТАННЯМ ОПТИЧНОГО ПОТОКУ

Один із підходів по виявленню в кадрі об'єкта інтересу засноване на виявленні особливостей статичних точок, та порівняння оптичного потоку для виявлення точок, що належать динамічним об'єктам. Головна ідея аналізованого методу полягає у порівнянні штучного оптичного потоку, заснованого на русі камери, з дійсним оптичним потоком та відстеження відмінностей у них.

### 2.1 Оптичний потік

Оптичний потік — це представлення видимого сліду руху об'єктів, поверхонь, і граней візуальної сцени[2]. Він виникає внаслідок відносного руху об'єктів на зображенні та спостерігача. Відповідно, з оптичного потоку можна отримати важливу інформацію про розташування видимих об'єктів у просторі та швидкість зміни цього розташування. На її основі можна виконати сегментування зображення на частини, які відповідають різним об'єктам.

Оптичний потік ґрунтується на твердженні про те, що для кожної точки вихідного зображення  $I_1(x, y)$ , де  $x$  та  $y$  — координати точки, можна знайти такий зсув  $(\Delta x, \Delta y)$ , що початкова точка відповідає точці  $I_2(x + \Delta x, y + \Delta y)$  на другому зображенні. Для того, щоб визначити відповідність між ними використовується певна функція точки, яка не змінюється в результаті зсуву[3]. Для цього можуть бути застосовані величина градієнта, гессіан, лапласіан та інші. На практиці найчастіше використовується яскравість точки або колір для кольорових зображень, хоча при різкій зміні освітлення можуть виникати проблеми з встановленням відповідності.

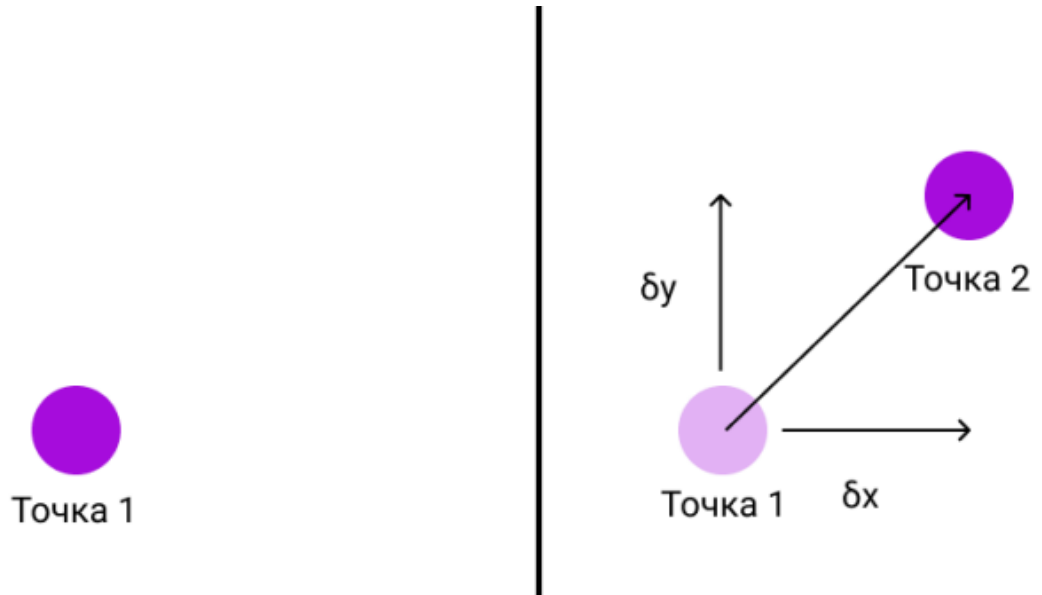


Рисунок 2.1 — Переміщення Точки 1 у Точку 2 та відповідні вектори зсуву на послідовних кадрах

Це завдання розглядається для двовимірних зображень з урахуванням часу, тому піксель з координатами  $(x, y)$  має інтенсивність  $I(x, y, t)$ , де  $t$  - змінна часу. За один кадр точка буде переміщена на  $\Delta x$ ,  $\Delta y$ ,  $\Delta t$  і виходить наступне рівняння:

$$I(x, y, t) \approx I(x + \Delta x, y + \Delta y, t + \Delta t) \quad (1.1)$$

Вважаючи переміщення малим, за формулою Тейлора отримаємо:

$$I(x + \Delta x, y + \Delta y, t + \Delta t) \approx I(x, y, t) + \frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t \quad (1.2)$$

Отже:

$$\frac{\partial I}{\partial x} \frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y} \frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t} \frac{\Delta t}{\Delta t} = 0 \quad (1.3)$$

та

$$\frac{\partial I}{\partial x} V_x + \frac{\partial I}{\partial y} V_y + \frac{\partial I}{\partial t} = 0 \quad (1.4)$$

де  $(V_x, V_y)$  — вектор швидкості оптичного потоку у  $I(x, y, t)$ .

Маємо:

$$I_x V_x + I_y V_y = -I_t \quad (1.5)$$

або

$$\nabla I^T \cdot \vec{V} = -I_t \quad (1.6)$$

Отримане рівняння має назву рівняння обмеження градієнта (gradient constraint equation) або рівняння оптичного потоку [4] і не може бути розв'язане однозначно, оскільки містить дві невідомі змінні. Ця неоднозначність має назву проблема апертури і полягає в тому, що у деяких випадках неможливо визначити напрям руху певного об'єкта, спостерігаючи лише за невеликою частиною зображення. Для розв'язання даної проблеми використовується регуляризація – накладання додаткових обмежень [3].

## 2.2 Алгоритми визначення оптичного потоку та його застосування

- Фазова кореляція - обернена до нормованого спектру потужності.
- Блокові методи – використовують мінімізацію суми різниці квадратів, суму абсолютних різниць чи максимізацію нормалізованих взаємкореляцій.
- Диференціальні методи оцінки оптичного потоку, засновані на часткових похідних сигналу зображення та/або шуканого поля потоку та часткових похідних вищого порядку, таких як:
  1. Метод Лукаса-Канаде – базується на припущенні про те, що напрямок руху сусідніх пікселів приблизно однаковий;
  2. Метод Хорна – Шанка – оптимізація функціоналу з урахуванням проблем з обмеження сталості яскравості та припущення про те, що яскравість будь-якої частини зображення змінюється дуже повільно;
  3. Метод Бакстона - Бакстона - заснований на моделі руху країв у послідовностях зображень;

4. Метод Блека – Джемсона – грубий оптичний потік через кореляцію;
  5. Загальні варіаційні методи – модифікації та розширення методу Хорна-Шанка, які використовують інші обмеження на дані та швидкість зміни яскравості.
- Дискретні методи оптимізації - простір пошуку квантується, а потім виконується зіставлення зображень шляхом присвоєння міток кожному пікселю, тому відповідна деформація мінімізує відстань між джерелом і цільовим зображенням. Оптимальне рішення визначається за допомогою алгоритмів знаходження максимального потоку в графі (алгоритм Форда-Фалкерсона) або алгоритму поширення довіри[5].

В даній роботі будемо використовувати метод Лукаса-Канаде оскільки він являється найбільш популярним серед них.

Оптичний потік часто використовується для різних цілей. У цьому розділі мова йде про деякі з них.

Оптичний потік широко застосовується в візуальній одометрії. Візуальна одометрія – це процес визначення положення та орієнтації пристрою за допомогою аналізу послідовності кадрів. Використовується в комп'ютерних оптичних мишах, квадрокоптерах та робототехніці. Найчастіше використовуваний алгоритм візуальної одометрії, який ще називається прямим алгоритмом[6], складається з таких кроків:

1. Отримання вхідного зображення — зазвичай пристрої, що використовують візуальну одометрію, оснащені однією або кількома камерами, які і створюють ці зображення.

2. Редагування вхідного зображення — використовуються методи обробки зображень для зменшення перспективних викривлень та шуму.

3. Визначення та відстеження ключових точок в різних кадрах — спочатку відбувається співставлення послідовних кадрів для встановлення відповідностей між ними. Далі на зображеннях виділяються ключові точки.

Після цього відбувається побудова поля оптичного потоку(часто за допомогою метода Лукаса-Канаде).

4. Виявлення та корекція значень векторів оптичного потоку, які виділяються — значення векторів перевіряються на потенційні помилки та відкидаються значення.

5. Оцінка руху камери за оптичним потоком — мінімізація функції втрат(часто з використанням фільтра Калмана). Використовуються дані про попереднє розташування пристрою для передбачення зміни їх розташування. Отримане передбачення порівнюється з поточними даними.

6. Періодичне перевизначення ключових точок для відстеження — використовується для захоплення всіх областей зображення.

Існує також інший підхід до візуальної одометрії, який базується на визначенні та відстеженні характерних ознак(feature-based). Для більшої ефективності часто використовується поєднання прямого та заснованого на ознаках підходів.

Ще одне поширене використання оптичного потоку – співставлення зображень, тобто, обчислення кута повороту у тривимірному або двовимірному просторі одного зображення або його частин відносно іншого. Для цього спочатку визначаються ключові точки на обох зображеннях. Для визначених точок обчислюються значення оптичного потоку та встановлюється відповідність між ними.

Інші застосування оптичного потоку також включають аналізування за його допомогою руху в візуальних системах, таких як спостереження за погодою або рухом транспорту.

### **2.3 Обчислення дійсного оптичного потоку методом Лукаса-Канаде**

У попередньому розділі було виведено рівняння оптичного потоку або рівняння обмеження градієнта у загальному вигляді:

$$\nabla I^T \cdot \vec{V} = -I_t$$

Отримане рівняння не має однозначного розв'язку, оскільки містить дві невідомі -  $V_x$  та  $V_y$ . Алгоритм Лукаса-Канаде використовує інформацію про сусідні пікселі, щоб розв'язати цю неоднозначність. Визначення дійсного оптичного потоку буде реалізовано алгоритмом Лукаса - Канаде [7]. Цей метод є диференціальним локальним алгоритмом знаходження оптичного потоку. Неоднозначність рівняння 1.6 вирішується за рахунок інформації про сусідні пікселі кожної точки. Вводиться припущення, що оптичний потік однаковий на околиці кожної точки.

Розглянемо вікно із центром у пікселі  $p$ . Оптичний потік має бути однаковий для всіх точок вікна. А вектор  $(V_x, V_y)$  оптичного потоку в точці  $p$  має бути рішенням системи:

$$\begin{cases} I_x(q_1) \cdot V_x + I_y(q_1) \cdot V_y = -I_t(q_1), \\ I_x(q_2) \cdot V_x + I_y(q_2) \cdot V_y = -I_t(q_2), \\ \dots \\ I_x(q_n) \cdot V_x + I_y(q_n) \cdot V_y = -I_t(q_n), \end{cases} \quad (1.7)$$

де  $q_1, \dots, q_n$  — пікселі у вікні,  $I_x(q_i), I_y(q_i), I_t(q_i)$  — часткові похідні  $I$  в точці  $q_i$

Переписавши систему (1.7) в матричній формі, отримаємо:

$$Av = b \quad (1.8)$$

де

$$A = \begin{bmatrix} I_x(q_1) & I_y(q_1) \\ I_x(q_2) & I_y(q_2) \\ \dots & \dots \\ I_x(q_n) & I_y(q_n) \end{bmatrix}, \quad v = \begin{bmatrix} V_x \\ V_y \end{bmatrix}, \quad b = \begin{bmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \dots \\ -I_t(q_n) \end{bmatrix}$$

Порахувавши систему (1.8) за допомогою метода найменших квадратів, отримаємо систему  $A^T Av = A^T b$ , звідки  $v = (A^T A)^{-1} A^T b$ .

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_i I_x(q_i)^2 & \sum_i I_x(q_i)I_y(q_i) \\ \sum_i I_x(q_i)I_y(q_i) & \sum_i I_y(q_i)^2 \end{bmatrix}^{-1} = \begin{bmatrix} -\sum_i I_x(q_i)I_t(q_i) \\ -\sum_i I_y(q_i)I_t(q_i) \end{bmatrix} \quad (1.9)$$

Порахувавши систему (1.9) для кожної точки зображення, знайдемо оптичний потік.

## **2.4 Обчислення штучного оптичного потоку**

Штучний потік є положенням усіх точок, отриманих для дійсного оптичного потоку, математично спроектованих на наступний кадр з урахуванням руху камери. Іншими словами, вирахувавши штучний оптичний потік, можна визначити вектор напрямку руху камери. Рух камери є поворотом і зміщенням, які одержують у результаті роботи алгоритму РТАМ[8].

Для отримання штучного потоку використовується гомографічна проекція. Зазвичай гомографія проектує положення точки із площини однієї камери до іншої. У нашому випадку використовуються деякі припущення для спрощення та прискорення процесу проектування. Головне припущення полягає в тому, що поверхня землі не має суттєвого нахилу, тобто середній нахил вважатимуться близьким до нуля. Невеликі зміни рельєфу близько 40-50 см не вплинуть на працездатність алгоритму.

## **2.5 Ідентифікація динамічних точок та їх фільтрація**

У розглянутій задачі в якості представлення оптичних потоків використовуються набори векторів. Варто відзначити, що переважна кількість векторів штучного оптичного потоку взаємопаралельна. Підсумувавши їх, можна отримати результуючий вектор, який показуватиме напрямок руху камери в сцені.

Динамічний характер точки визначається в порівнянні перерахованих вище оптичних потоків. Динамічні об'єкти ідентифікуються набором векторів дійсного оптичного потоку, що мають відмінне від вектора руху камери напрямком та/або величиною. Для продовження обчислення необхідно групувати вектори зображення. Може вийти, що деякі вектори виявляться ізольованими щодо інших. Часто вони можуть бути викликані неконтрольованими коливаннями чи іншими помилками. У загальному випадку ці вектори не

беруться до уваги. Група векторів, що знаходяться в околиці один від одного, береться за динамічний об'єкт.

Внаслідок роботи алгоритму Лукаса-Канаде виходить оптичний потік зображення. Після віднімання сумарного вектора штучного з векторів дійсного потоку виходить дійсною оптичний потік з урахуванням штучного, який буде представлений двома матрицями швидкостей  $V_x$  та  $V_y$  для кожної точки зображення з яких, у свою чергу, можна обчислити матриці напрямку швидкості та величини швидкості  $M$  в кожній точці. Кожен елемент матриці  $M$  обчислюється за формулою:

$$M_{i,j} = \sqrt{V_{x_{i,j}}^2 + V_{y_{i,j}}^2} \quad (1.10)$$

За значеннями матриці швидкості можна реалізувати виділення рухомих об'єктів на зображенні за принципом пошуку груп поруч стоячих ненульових значень матриці. Накопичення таких значень можна знайти шляхом визначення контуру. Контуром, що обмежує це скупчення, буде прямокутник, обмежений паралельними прямими. Так як пошук відбуватиметься у матриці, паралельними прямими тут будуть рядки та стовпці матриці з координатами  $j = M_{j_{\min}}, M_{j_{\max}}$  і  $i = M_{i_{\min}}, M_{i_{\max}}$ .

Отримати ці значення можна за таким методом:

1. Знаходиться елемент матриці  $M$  зі значенням  $M_{i,j} > 0$ . Пошук здійснюється у порядку з першого до останнього рядка матриці  $M$ .
2. Доки  $j > 1$  і  $M_{i,j} > 0$  переміщуємось по рядку матриці вліво ( $j = j - 1$ ).
3. Точка  $M_{i,j}$  в даний момент буде належати стовпцю  $M_{j_{\min}}$ , що обмежує контур зліва.
4. Доки  $M_{i+1,j} \neq 0$  переміщуємось по рядку вправо ( $j = j + 1$ ).
5. Переходимо до наступного рядка  $i = i + 1$ .



6. Переходимо до кроку 2 і виконуємо доти, доки не досягнемо краю матриці або стану, коли праворуч і знизу від поточної точки будуть тільки нулі.
7. Досягши граничного значення, отримаємо крайню ліву точку обмежуючого контуру об'єкта, що рухається. Так само знайдемо рядок  $M_{i_{\max}}$  – нижня границя та стовпець  $M_{j_{\max}}$  – права границя. Рядок  $M_{i_{\min}}$ , що є межею контуру зверху, буде знайдено на кроці 1 так як пошук ненульових значень проводився зверху-вниз.

З отриманих границь можна отримати ліву верхню та праву нижню точку прямокутника:  $(M_{i_{\min}}, M_{j_{\min}})$  та  $(M_{i_{\max}}, M_{j_{\max}})$  відповідно.

Для запобігання випадку повторного розгляду того самого контуру алгоритмом, потрібно застосувати наступний підхід: після знаходження чергового контуру необхідно обнулити всі значення всередині нього та на його границі. У цьому випадку, за подальших ітерацій, алгоритм нічого не буде знати про попередні значення і не виділятиме їх знову.

Розглянемо приклад. Нехай дана матриця  $M$  швидкостей зображення розміром 7 на 5 пікселів:

$$M = \begin{bmatrix} 0 & 0.7 & 0.9 & 0.8 & 0 & 0 & 0 \\ 0 & 0 & 0.8 & 0.7 & 0.4 & 0 & 0 \\ 0 & 0.7 & 0.9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.3 & 0.4 \end{bmatrix}$$

Матриця  $M$  містить виділені кольором елементи. Один колір характеризує об'єкт. Таким чином, у цьому прикладі на зображенні присутні два об'єкти, що рухаються, які будуть виділені двома прямокутниками з межами:  $M_{i_{\min}} = 1, M_{j_{\min}} = 2, M_{i_{\max}} = 3, M_{j_{\max}} = 5$  і  $M_{i_{\min}} = 4, M_{j_{\min}} = 6, M_{i_{\max}} = 5, M_{j_{\max}} = 7$  або з координатами лівого верхнього та правого нижнього кутів (1, 2), (3, 5) та (4, 6), (5, 7) відповідно.

$$M' = \begin{bmatrix} 0 & 0.7 & 0.9 & 0.8 & 0 & 0 & 0 \\ 0 & 0 & 0.8 & 0.7 & 0.4 & 0 & 0 \\ 0 & 0.7 & 0.9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.3 & 0.4 \end{bmatrix}$$

Варто зазначити, що для фільтрації помилкових векторів можна використовувати нижнє граничне значення швидкості, відмінне від нуля. Іноді коректне граничне значення буде змінюватися. Воно залежить від висоти між камерою та об'єктами, швидкістю їхнього руху та швидкістю руху камери.

Крім того, можна ввести обмеження на мінімальну та максимальну площу просторів, обмежених контурами. Якщо площа поточного контуру більше максимально заданої або менше мінімальної, то даний контур просто не враховується.

## 2.6 Модифікація метод

У первинній реалізації рішення задачі виявлення рухомих об'єктів використовувався пірамідальний алгоритм Лукаса-Канаде. Швидкість роботи алгоритму залежить від розміру оброблюваних кадрів: чим більше кадр, тим довше він обробляється. З іншого боку, чим якісніше кадр і чим більше його розмір, то точніше буде визначено оптичний потік. Отже, залежно від конкретної задачі та наявних обчислювальних потужностей потрібно вибирати, з якою якістю та швидкістю будуть виконані обчислення.

Причиною неточної роботи алгоритму Лукаса – Канаде можуть бути різні шуми на зображенні. Поява шумів багато в чому залежить від камери, що робить зйомку. В алгоритмі Лукаса-Канаде перед тим, як обробити зображення, для оптимальності часто його знебарвлюють.

Через особливості алгоритму, перед тим, як працювати із зображенням, для кращої продуктивності до нього застосовують згладжування. Було випробувано застосування згладжування Гауссом і медіанного фільтра. Однак дані інструменти помітно не вплинули на якість результату.

Замість пірамідального алгоритму Лукаса-Канаде ефективнішим є використання ітеративного алгоритму Лукаса-Канаде. Основна його відмінність у тому, що при роботі з поточним кадром у ньому використовується інформація, накопичена у період роботи з попередніми кадрами. Даний різновид алгоритму працює у кілька разів швидше та якісніше.

Робота пірамідального алгоритму Лукаса-Канаде представлена на рисунку 2.2, а ітеративного – на рисунку 2.3. Червоні стрілки – це візуалізація оптичного потоку. Результат роботи пірамідального алгоритму містить велику кількість зайвих векторів. Результат роботи ітеративного алгоритму немає цього недоліку. Таким чином, ітеративний алгоритм працює якісніше.



Рисунок 2.2 — Приклад роботи пірамідального алгоритму Лукаса-Канаде



Рисунок 2.3 — Приклад роботи ітеративного алгоритму Лукаса-Канаде

Для демонстрації роботи алгоритмів для обох прикладів було обрано відеофайл з розміром кадрів  $640 \times 480$  пікселів. Пірамідальний алгоритм має швидкість роботи 6 кадрів за секунду. Ітеративний алгоритм, у свою чергу, обробляє 24 кадри за секунду. Звідси випливає, що пірамідальний алгоритм працює повільніше за ітеративний.

## 2.7 Недоліки методу оптичного потоку

Метод має декілька недоліків. По-перше, для деяких текстур зображення матриця  $M$  може бути виродженою і тому для неї не можливо знайти обернену матрицю. У цьому випадку не можна однозначно визначити зсув пікселя та визначити напрям руху об'єктів на зображенні. Ця проблема також відома як проблема апертури або *barberpole problem*[9]. По-друге, метод не може визначити рух всередині однорідних областей, більших за розмір вікна,

оскільки є строго локальним. По-третє, алгоритм добре працює лише для невеликих зсувів, що зазвичай не є проблемою, оскільки відмінності між двома послідовними кадрами мінімальні та не перевищують значення у кілька пікселів, але в окремих випадках призводить до некоректного обчислення оптичного потоку.

### 3 МЕТОД ВИЯВЛЕННЯ ОБ'ЄКТІВ ЗА ДОПОМОГОЮ КОЛЬОРОВИХ ФІЛЬТРІВ

Один із підходів, що може бути застосований при виявленні об'єктів за певною ознакою, це метод, що передбачає застосування фільтрації за кольором. Колір виступає в якості основної ознаки об'єкта виявлення. Подібний підхід може бути застосований при виділенні об'єктів, які за кольором суттєво відрізняються від інших об'єктів та від фону. Враховуючи, що часто для представлення кольору в цифрових зображеннях використовується модель RGB, то для зменшення ймовірності помилкових спрацьовувань можна застосувати порогову фільтрацію в колірних просторах HSL або HSV. Відображення даних колірних простористів наведено на рисунку 3.1

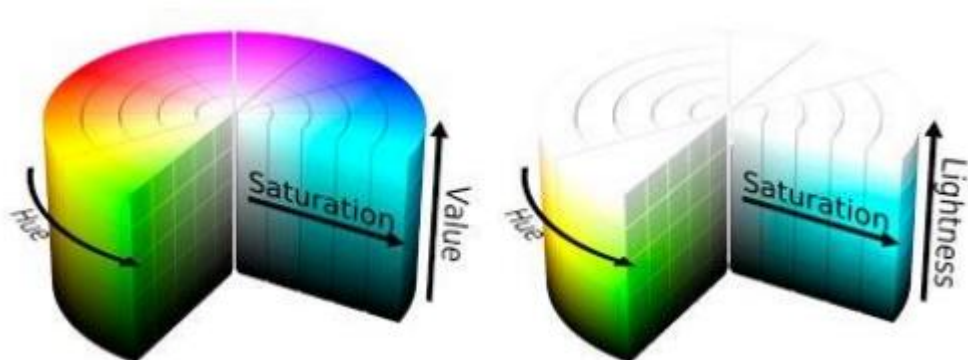


Рисунок 3.1 — Представлення кольорових просторів HSV і HSL у формі циліндра

Наприклад, для виділення області жовтого кольору можна обмежити тільки значення показника тону, що зробить даний пороговий фільтр менш чутливим до освітлення зображення. А при пошуку областей білого або чорного кольору необхідно обмежити рівень світлоти зображення.

Загалом, етапи обробки по виділенню кольорових об'єктів на зображенні або відеопослідовності можуть бути представлені у вигляді наступних кроків:

1. Отримання кадру;
2. Конвертація отриманого RGB-кадру в HSV-кадр;

HSV (hue, saturation, value) – це колірна модель представлення зображення. В цій моделі є 3 головні параметри [10]:

a) Hue — колірний тон, (наприклад, червоний, зелений або синьо-блакитний). Варіюється в межах 0-360°, але іноді приводиться до діапазону 0-100.

b) Saturation — насиченість. Варіюється в межах 0-100.

c) Value — Brightness — яскравість. Також задається в межах 0-100.

Як видно з наведених вище параметрів, мета даної конвертації – це комфортне людське представлення параметрів кольорів, за допомогою яких можна в зрозумілому вигляді вибрати необхідний діапазон.

3. Встановлення необхідного порогу кольорів за допомогою HSV моделі;

4. Вибраний діапазон кольорів застосовується до кадру (відповідні пікселі які попадають в дану межу, зберігаються на матриці, інакше – позначаються чорними). Тут відповідно й застосовується “бінаризація” по кольору;

5. Отриманий кадр з виділеними об'єктами – бінаризується ще раз, з метою зменшення інформації. Але перед тим необхідно конвертувати RGB-кадр в півтонне зображення, а вже потім здійснювати бінаризацію.

6. Видалення шумів за допомогою методу математичної морфології (наприклад операцією erode а потім dilate);

7. Здійснення пошуку об'єкта за контурами.

### 3.1 Колірна модель HSV

В системах технічного зору існує кілька кольірних моделей для опису специфікації кольорів, таких як RGB, CMY і HSV. В даному методі будемо використовувати модель HSV (hue, saturation, value), оскільки вона значно зменшує розмір кольорової інформації та інформації про відтінки сірого на зображенні. Набір ізольованих точок кластеризується в об'єкти шляхом виділення кольору. Для сегментації виділеного кольору використовується фільтрація за відтінком. При встановленні значення відтінку на зображення накладається маска. У бінарному зображенні, коли значення пікселів



задовольняють заданому критерію, наприклад, відтінку, значення трансформується шляхом накладання маски. Функція встановлюється в нуль, що відображається білим кольором, в іншому випадку всі пікселі в призначеному діапазоні відтінків позначаються як передній план і відображаються білим кольором, а інші пікселі позначаються як задній план і відображаються чорним кольором.

У багатьох алгоритмах технічного зору та обробки зображень робляться спрощені припущення про рівномірність інтенсивності в локальних областях зображення. Однак, зображення реальних об'єктів часто не мають областей з однорідною інтенсивністю. Наприклад, зображення дерев'яної поверхні не є однорідним, а містить варіації інтенсивності, які утворюють певні повторювані патерни, що називаються візуальною текстурою.

Текстурні об'єкти - це такі об'єкти, для яких, на відміну від об'єктів на основі форми, не існує очевидної покомпонентної відповідності. Такі об'єкти краще описуються їх текстурою, ніж геометричною будовою.

Текстури забезпечують важливі характеристики для ідентифікації поверхонь, супутникових фотографій, біомедичних зображень та багатьох інших типів зображень.

### **3.2 Конвертація RGB у HSV**

Колірну модель HSV можна розглядати як інший вигляд куба RGB. Таким чином, значення HSV можна розглядати як перетворення від RGB з використанням геометричних методів. Діагональ куба RGB від чорного (початок координат) до білого відповідає осі V шестигранного конуса в моделі HSV. Для будь-якого набору значень RGB вісь V дорівнює максимальному значенню в цьому наборі. Точка HSV, що відповідає набору значень RGB, лежить на перерізі шестикутника при значенні V. Тоді параметр S визначається як відносна відстань цієї точки від осі V. Параметр H визначається шляхом розрахунку відносного положення точки в межах кожного секстанта шестикутника. Значення RGB визначаються в діапазоні [0,



1], тому ж діапазоні значень, що і HSV. Значення Н - це відношення, перераховане від 0 до 360 градусів.

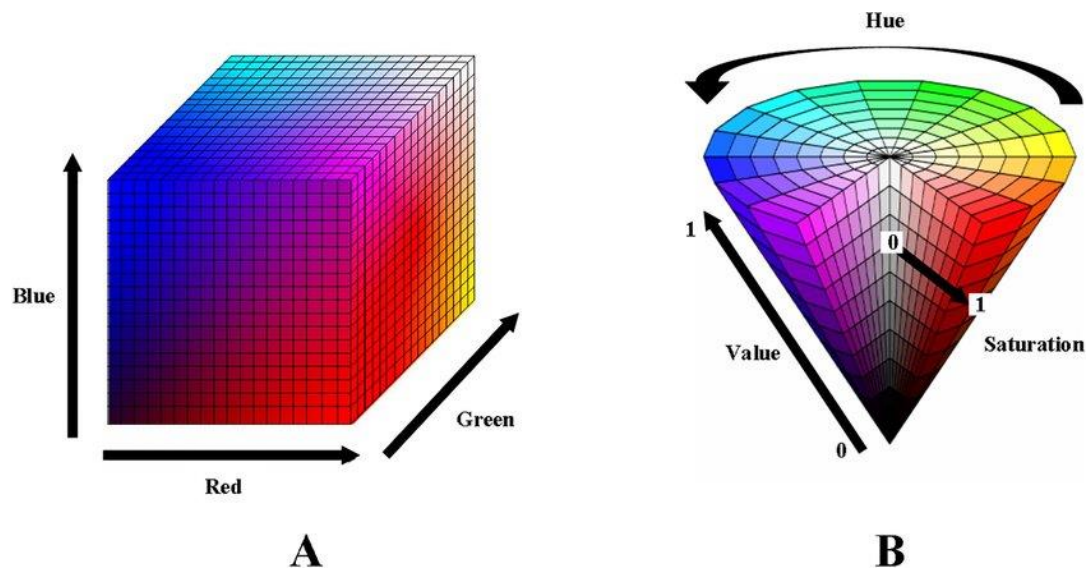


Рисунок 3.2 — Конвертація RGB у HSV

Процес RGB конвертації в HSV, проходить за наступною формулою:

$$Max = \max(R, G, B);$$

$$Min = \min(R, G, B);$$

$$H = \begin{cases} \text{undefined}, & Max = Min \\ 60 \times (G - B) / (Max - Min), & Max = R, \quad G \geq B, \\ 60 \times (G - B) / (Max - Min) + 360, & Max = R, \quad G < B, \\ 60 \times (B - R) / (Max - Min) + 120, & Max = G, \\ 60 \times (R - G) / (Max - Min) + 240, & Max = B, \end{cases} \quad (3.1)$$

$$S = \begin{cases} 0, & Max = 0, \\ 1 - \frac{Min}{Max}, & \text{otherwise,} \end{cases}$$

$$V = Max.$$

### 3.3 Неліки методу кольорових фільтрів

Одним з головних недоліків даного методу полягає в тому, що потрібно ретельно підбирати зображення, щоб точно детектувати ціль. Зображення або відео-файли які містять велику кількість відтінків кольорів які співпадають з

відтінками кольору цілі — є поганими для цього методу. Прості зображення з птахом у синьому небі або червоний квадрат на білому фоні, такі приклади гарно підходять під суть методу.

Вибір кольору в ручному режимі – це також є недоліком, такий метод без додаткових модернізацій складно використовувати у промислових масштабах.

## 4 РЕАЛІЗАЦІЯ ТА ПОРІВНЯННЯ МЕТОДІВ

### 4.1 Вибір інструмента

Вибір інструменту реалізації складався з трьох мов програмування: C++, Python та MATLAB. C++ використовується через його високу продуктивність. Мова Python прискорює процес написання програм через його простоту синтаксису. MATLAB використовує бібліотеки, написані для більшості популярних процесорів. Бібліотеки використовують процесорні інструкції, наприклад MMX та SSE2, та інші інструменти процесора для ефективної роботи з матричними обчисленнями.

Разом з кожною мовою передбачається використовувати інструмент для роботи із графікою. Для C++ та Python це бібліотека OpenCV [11]. Для MATLAB - Computer Vision System Toolbox [12].

Незважаючи на те, що на C++ при належному підході виходять ефективні програми, у цієї мови є недоліки, одна з яких – складність написання якісного коду та складність пошуку помилок. Мова Python лаконічний і простий, але розробники платять швидкістю його роботи.

Середовище MATLAB було обрано як інструмент реалізації алгоритму. MATLAB не має перелічених вище недоліків інших мов програмування.

### 4.2 Ілюстрація роботи методу із використанням оптичного потоку

1. Отримання вхідних зображень — алгоритм отримує на вхід два зображення, між якими відбувся рух об'єктів на зображенні або рух камери. Для зменшення шуму на початкових зображеннях використовується згладжування за допомогою фільтра Гаусса з бібліотеки OpenCV.
2. Обрахування часткових похідних за допомогою згортки — часткові похідні за напрямками зсуву обчислюються за допомогою застосування згортки.

3. Обрахування матриці  $A$  та її зворотної матриці  $A^{-1}$ , обрахування вектора  $b$ . Для виконання матричних операцій було створено окремий модуль. Він включає такі функції як множення матриці на вектор, обчислення зворотної матриці, обчислення союзної матриці та обчислення визначника матриці для матриць розміру  $2 \times 2$ , які використовуються при виконанні алгоритму. Для проходження усіх пікселів зображення використовується вікно заданого розміру, який визначає локальний окіл точки.
4. Обрахування векторів зсуву точки — для визначення векторів зсуву виконується множення отриманих на попередньому кроці матриці  $A^{-1}$  та вектора  $b$ . В результаті отримується вектор  $v$ , який містить обраховані значення зсуву для точки по осях  $x$  та  $y$ .
5. Побудова зображення поля оптичного потоку — отримані значення зсуву відображаються як вектори або як точки, колір яких позначає напрямок вектора зсуву.

#### 4.2.1 Тестування методу із використанням оптичного потоку

Для тестування роботи алгоритму використовувалась відеопослідовність, на якій відбувається рух автомобіля вниз.



Рисунок 4.1 — Перше зображення для тестування



Рисунок 4.2 — Друге зображення для тестування

Для цих зображень отримали наступний результат роботи алгоритму:

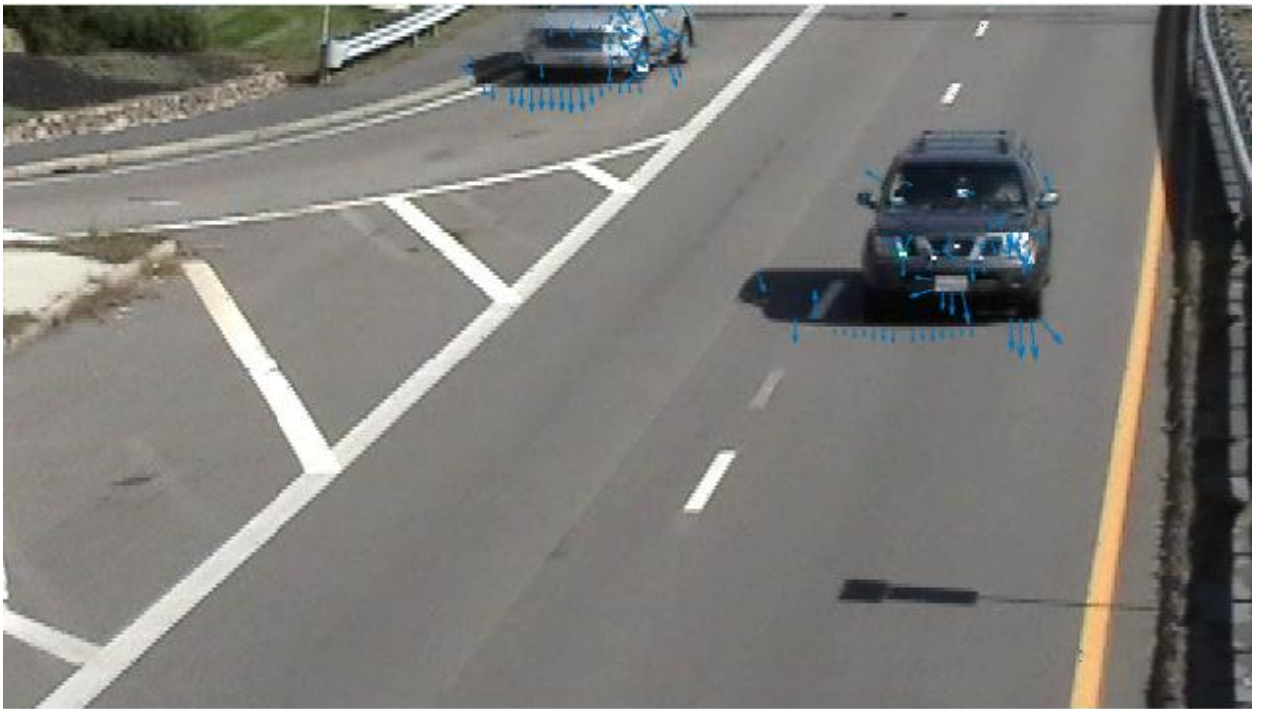


Рисунок 4.3 — Результат роботи алгоритму

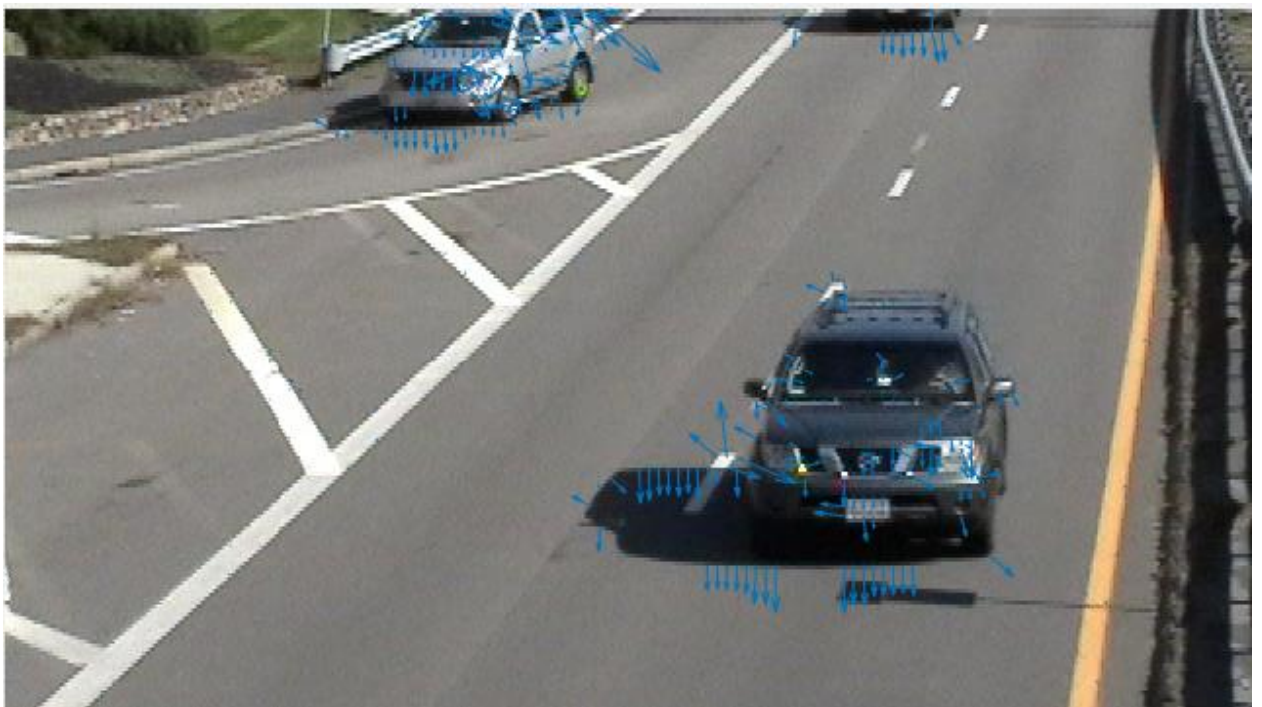


Рисунок 4.4 — Результат роботи алгоритму



Бачимо вектори напрямків руху автомобілів на зображенні. Користуючись отриманими даними можна також виділити межі об'єкта та знайти його обмежуючий прямокутник.

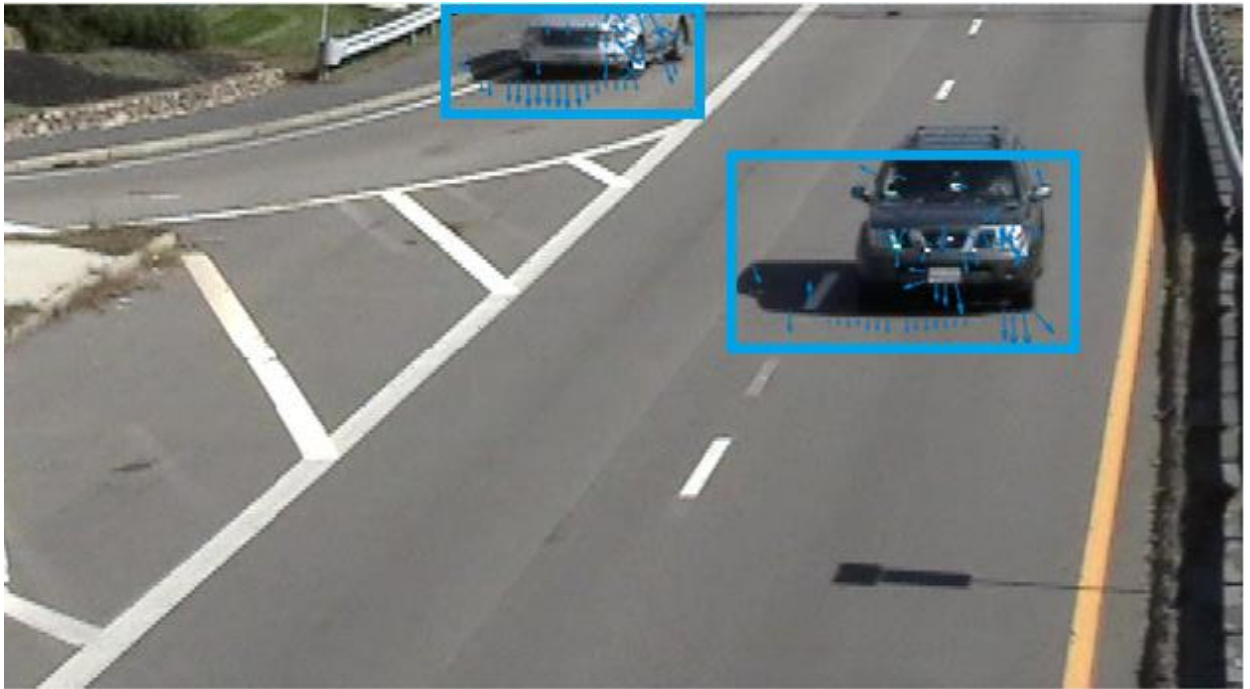


Рисунок 4.5 — Результати роботи алгоритму з обмежуючим прямокутником

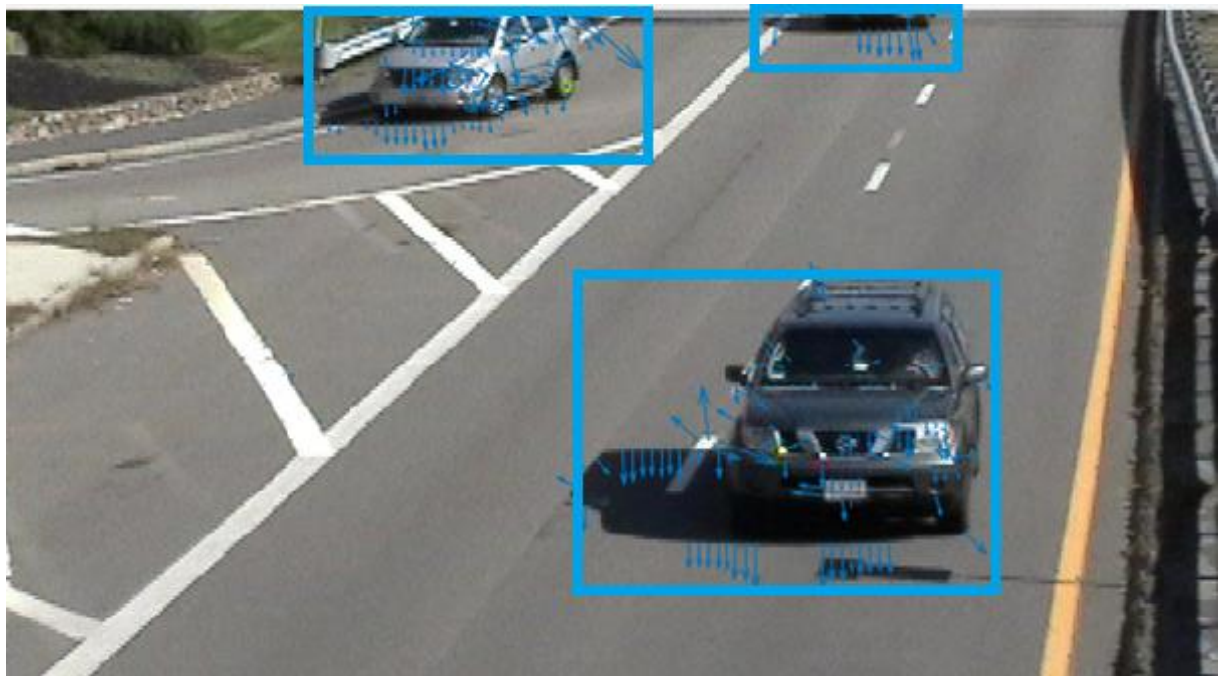


Рисунок 4.6 — Результати роботи алгоритму з обмежуючим прямокутником

### 4.3 Висновки до методу Лукаса-Канаде

Для ефективної реалізації алгоритма Лукаса-Канаде необхідно було його модифікувати, а саме замінити пірамідальний алгоритм Лукаса-Канаде на ітеративний алгоритм Лукаса-Канаде.

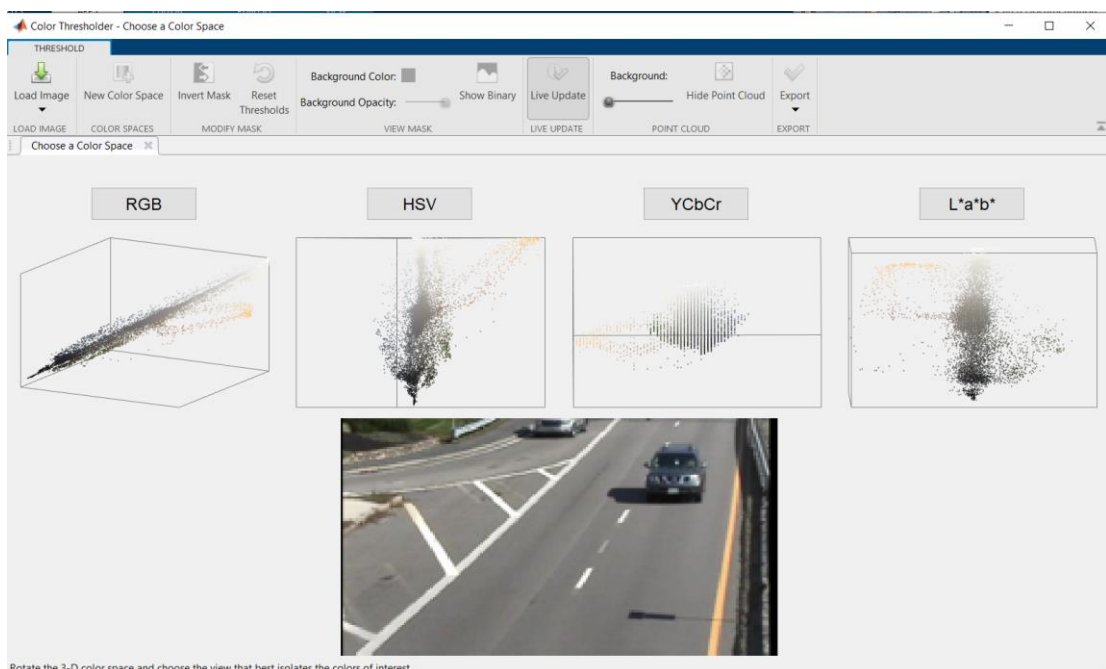
Якщо зображення містить досить великі однорідні області є шанс, що виникнуть проблеми у визначенні напрямків руху. Для досить деталізованих сцен такі проблеми не виникають. Отримані результати можна використовувати для визначення меж об'єктів, які рухаються, на зображенні, а також для відображення напрямків їх руху.

### 4.4 Ілюстрація роботи методу за допомогою кольорових фільтрів

Нехай дано відеофайл, де об'єктом виявлення буде автомобіль темно-сірого кольору. Будемо стежити за певним порогом кольорів(за допомогою HSV моделі), які можуть з'являтися в полі зору камери.

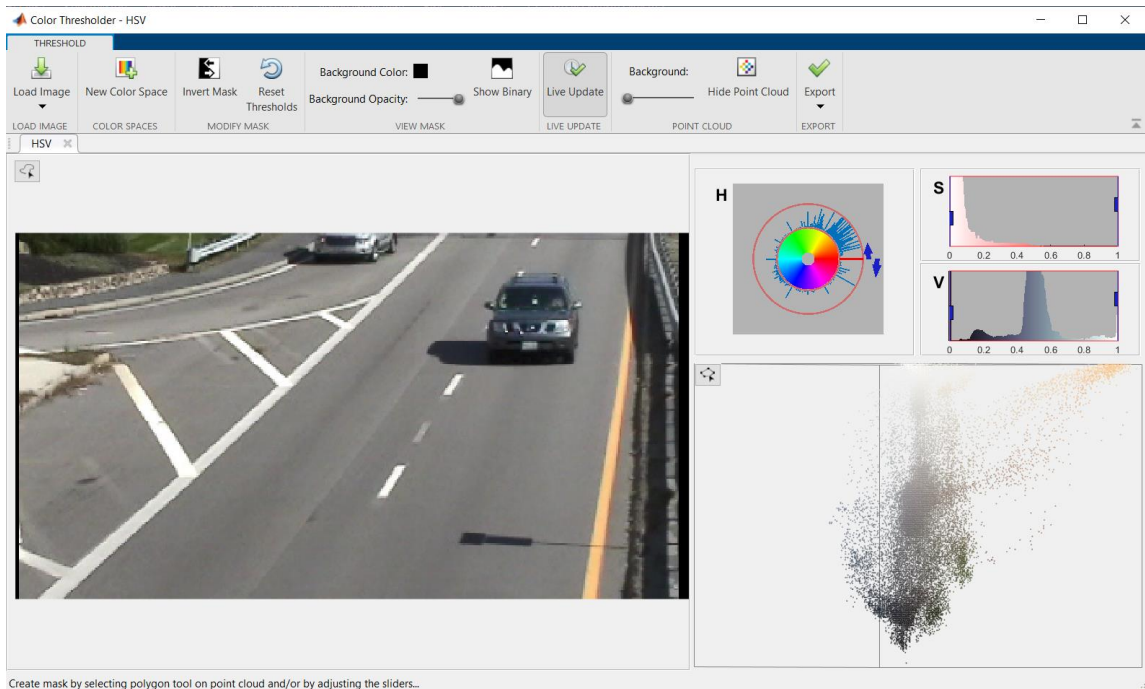
Щоб визначити кольоровий поріг для HSV моделі буду використовувати пакет «Color Threshold» який є стандартним пакетом для MATLAB R2021b.

Обравши зображення з «Workspace» та кольорову модель, можемо бачити HSV діаграми та кольори які використовуються на певному кадрі відеоряду.



a)





б)

Рисунок 4.7 — а) Вибір кольорової моделі; б) Пакет «Color Threshold»

Далі спеціальним інструментом виділяємо певну область цього об'єкту для аналізу кольорів які буде відстежувати алгоритм у ході своєї роботи.

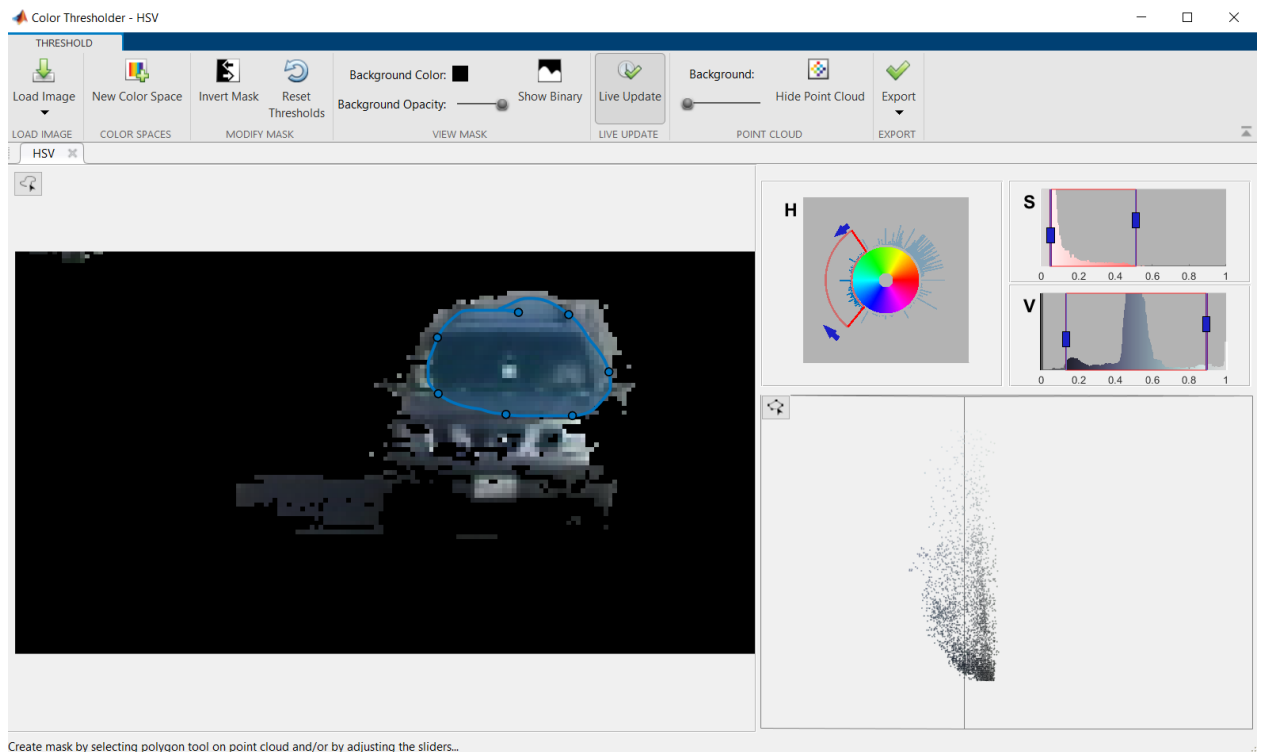


Рисунок 4.8 — Пороги кольорів

У результаті роботи пакету «Color Threshold» отримали наступні параметри:

$$H = [0.346, 0.641];$$

$$V = [0.050, 0.511];$$

$$S = [0.133, 0.894];$$

Які відповідають кольору об'єкту, у нашому випадку кольору автомобіля — темно-сірий.

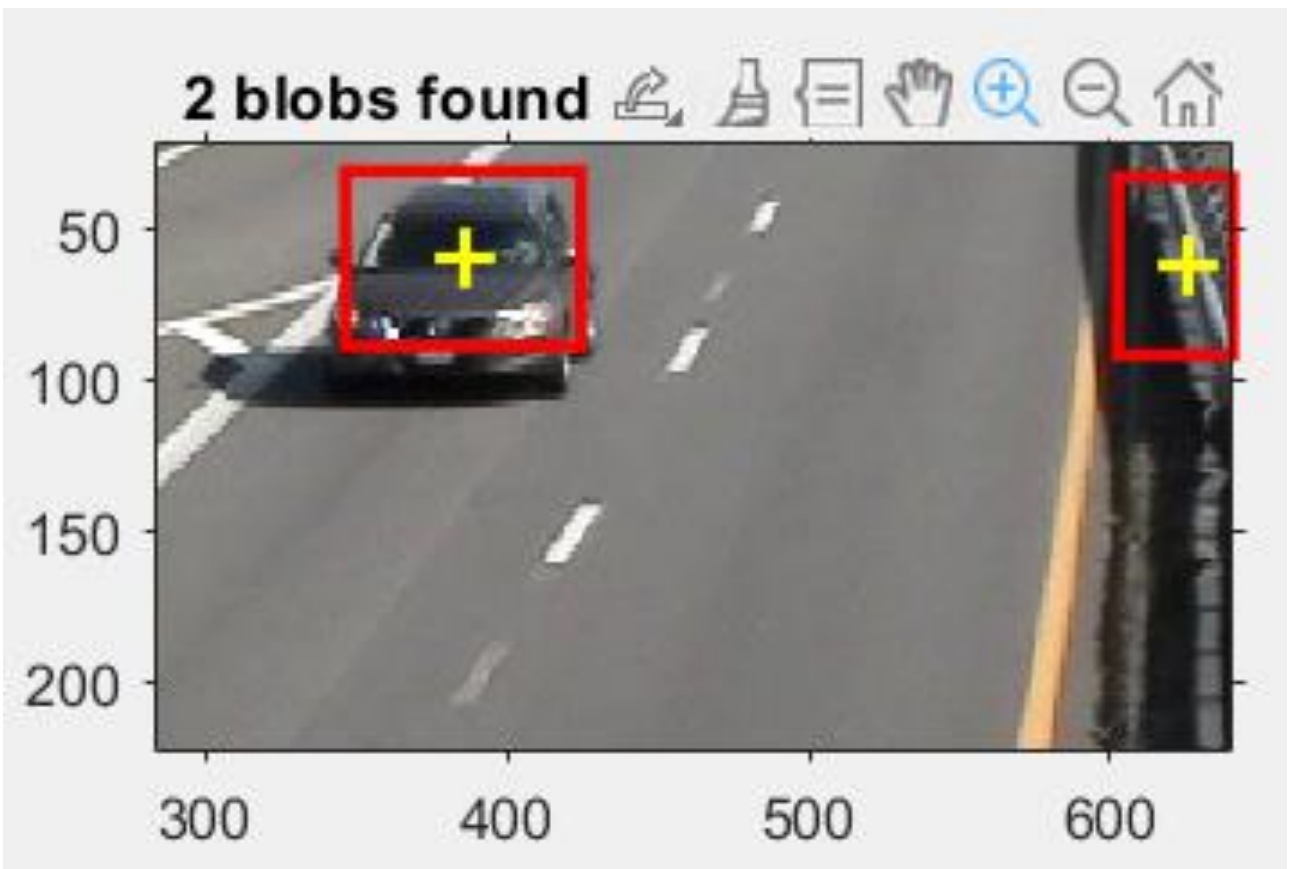
Програма обробляє кожен кадр відео послідовно з певним інтервалом та знаходить ціль та її координати за заданими кольорами.



Рисунок 4.9 — Перший кадр відео



а)



б)

Рисунок 4.10 — а) Ціль знайдена б) Збільшений варіант

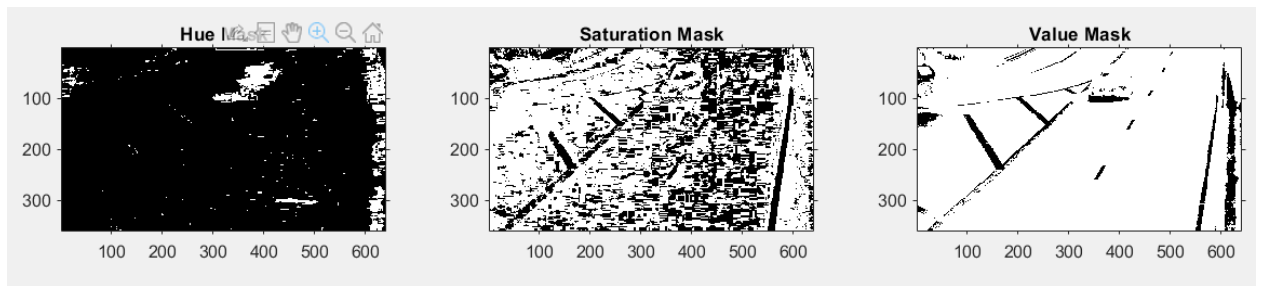


Рисунок 4.11 — HSV mask для 1-го кадру

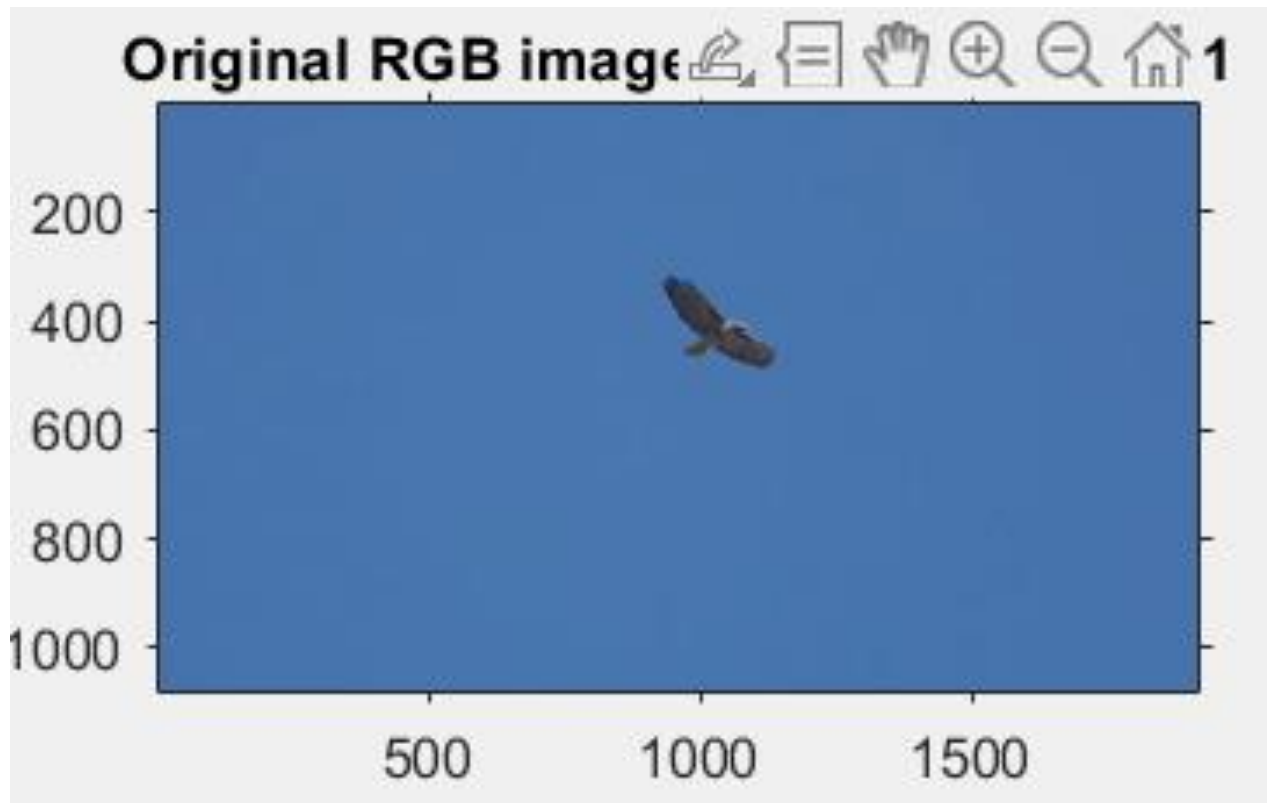
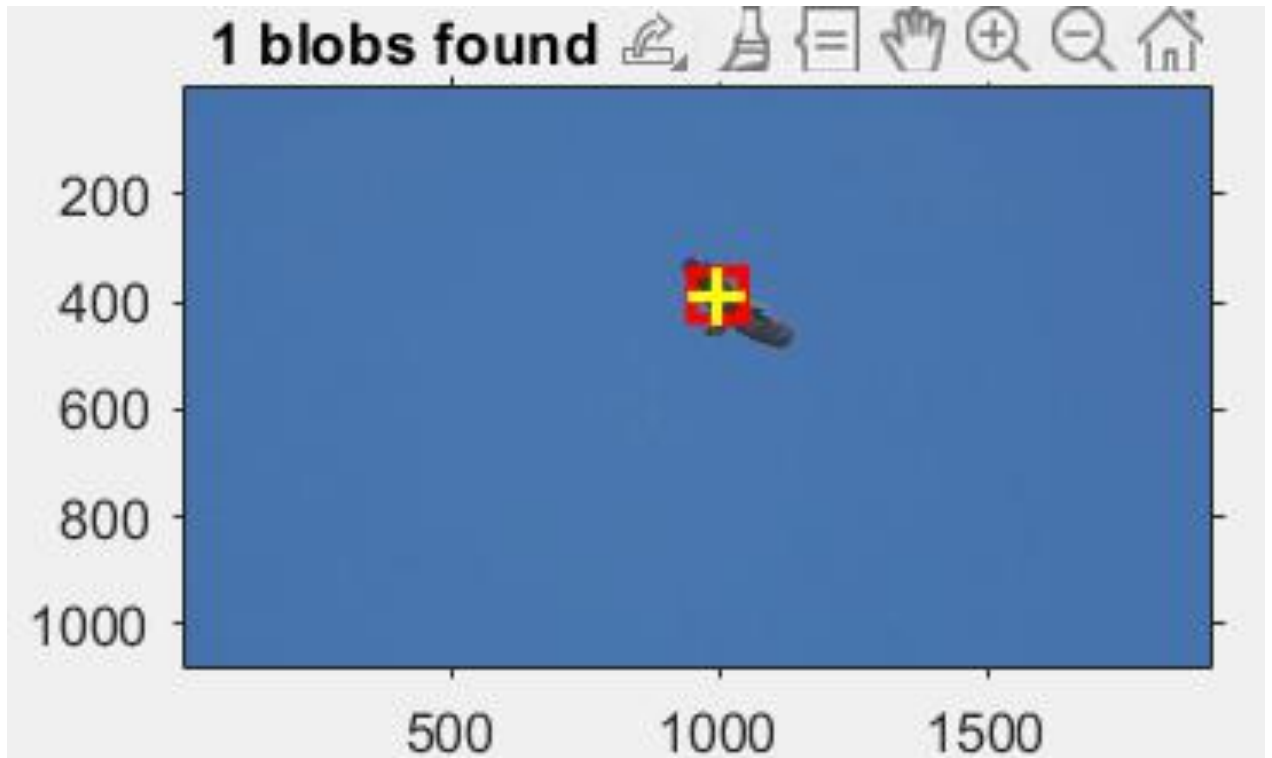
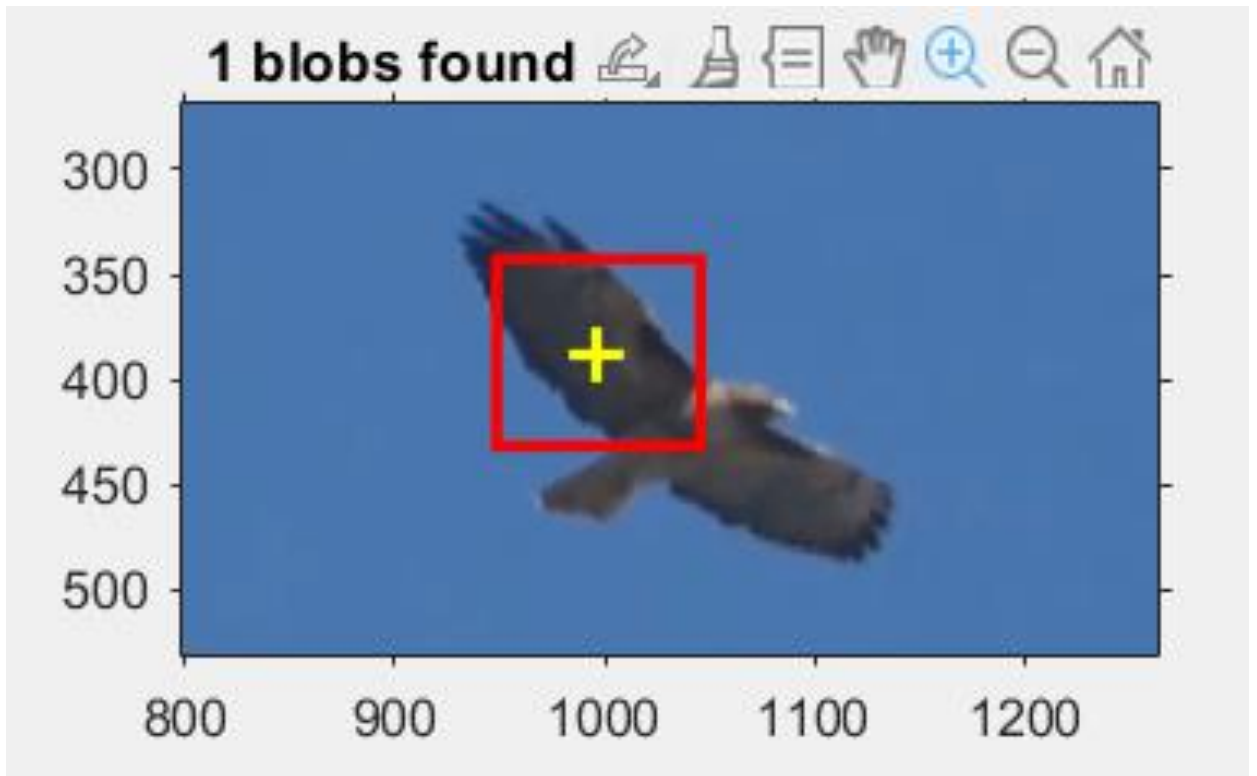


Рисунок 4.12 — Ідельне зображення для методу кольорових фільтрів



а)



б)

Рисунок 4.13 — а) Результат роботи; б) Збільшений варіант

#### 4.5 Висновки до методу кольорових фільтрів

Було реалізовано та протестовано метод на основі кольорових фільтрів. Якщо зображення містить багато відтінків кольорів, то виникають проблеми у визначенні об'єкту, як можемо бачити на рисунку 4.10, де окрім автомобіля алгоритм виділив область взагалі за межами дороги. Такий метод більш підходить для зображень на яких явно виражений об'єкт інтересу — наприклад сіра птаха у синьому небі, як на рисунку 4.13, а такому випадку ніякі проблеми не виникають. Отримані результати можна використовувати для виявлення об'єктів, які рухаються на відеопослідовності.

#### 4.6 Порівняння методів виявлення рухомих об'єктів в системах технічного зору

В минулих розділах ми проаналізували та реалізували два різні методи виявлення об'єктів на відеопослідовностях та можемо зробити висновки.

Для порівняння будемо брати однаковий відеофайл та аналізувати: швидкість роботи, якість виявлення, технічні вимоги.

Тип файлу	AVI
Розмір	24,2 МБ
Тривалість	17 сек
Ширина кадру	640 px
Висота кадру	360 px
Частота кадрів	29,97 к/с

Таблиця 4.1 — Характеристики відеофайлу

Технічні характеристики ноутбуку на якому проводяться досліді:

Процесор	Intel Core i5-10300H
Графіка	NVIDIA GF GTX 1650 4гб
Оперативна пам'ять	16 гб

Таблиця 4.2 — Характеристики ноутбука



Нижче наведені результати роботи 2 методів виявлення рухомих об'єктів в системах технічного зору, які ми й будемо порівнювати.



Рисунок 4.14 — Кольорові фільтри

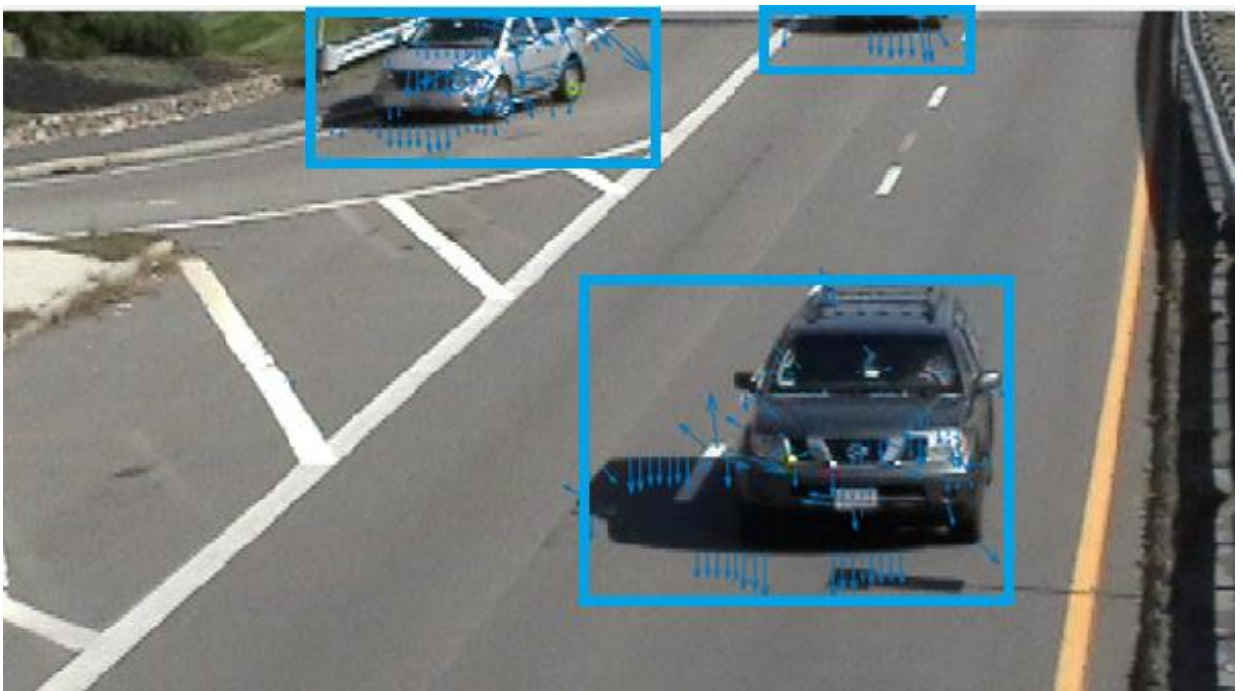


Рисунок 4.15 — Оптичний потік

Метод / хар-ка	Швидкість	Якість
Оптичний потік	~5 секунд	Гарна
Кольорові фільтри	~3 секунд	Погана

Таблиця 4.3 — Порівняння методів

З отриманих даних можемо зробити висновки:

1. При наявності сучасної апаратури час виконання алгоритму майже не відрізняється;
2. Якість роботи суттєво відрізняється, тому доцільніше використовувати метод з використанням оптичного потоку;
3. Метод оптичного потоку детектував всі рухомі цілі на відеопослідовності, що не сказати про метод кольорових фільтрів.



## 5 СТАРТАП

### 5.1 Маркетинговий аналіз стартап-проєкту

Ідея проєкту полягає у використанні розглянутих методів виявлення об'єктів для таких сфер як: навігація, розвідка, наука, тощо.

Таблиця 5.1 — Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Знайти оптимальний метод для виявлення об'єктів в системах технічного зору	1. Навігація	Стеження за технічним станом і функціонуванням об'єктів, спостереженням за потоком руху транспорту
	2. Розвідка	Отримання інформації про противника без ризику втрат людини

Розглянуті алгоритми аналізують фото або відео зображення виявляючи на них ціль, яка рухається або обраному кольору. Прикладом може бути завдання стеження за технічним станом і функціонуванням об'єктів, спостереженням за потоком руху транспорту. Безпілотні літальні апарати разом з методами детектування об'єктів, що рухаються, можуть широко застосовувати у військових цілях.

Конкуренцію можуть представити тільки спеціалізовані алгоритми, які налаштовують за допомогою штучного інтелекту або на конкретну ціль.

Таблиця 5.2 — Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживача до товару
1	Необхідність виявлення об'єкту з фото чи відео	Військові	Швидкість	Можливість швидко отримати координати шуканого об'єкту
2	Необхідність виявлення напрямку руху об'єкту з відеополідовності	Правоохоронні органи, військові, наука, тощо	-	Можливість дізнатися напрямок руху об'єкту

Для подальшого впровадження проекту потрібно складання SWOT-аналізу, це матриця аналізу сильних та слабкий сторін, загроз та можливостей.

Ринкові загрози та ринкові можливості складаються на основі аналізу факторів загроз та можливостей маркетингового середовища. Загрози та можливості є наслідками впливу факторів, але відмінність в тому, що вони не є реалізованими, а мають певну ймовірність здійснення.

Таблиця 5.3 — SWOT- аналіз стартап-проекту

Сильні сторони: широка направленість системи	Слабкі сторони: присутня конкуренція
Можливості: модернізація методу, впровадження штучного інтелекту	Загрози: поява нових потреб споживачів

## 5.2 Розроблення маркетингової програми стартап-проекту

Формування маркетингової концепції товару, який отримає споживач.

Таблиця 5.4 — Визначення ключових переваг концепції потенційного товару

Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами(існуючі або такі, які потрібно створити)
Виявлення цілі на відео або фото, для подальшої роботи з ним	Аналіз даних які були зібрані під час роботи. Отримання координат та напрямку руху цілі.	Можливість роботи з багатьма форматами та розширенням зображення, точність виявлення.

Наступний крок є визначення цінових меж, яке передбачає аналіз цін на товари-аналоги, а також на рівень доходів цільової групи споживачів.

Таблиця 5.5 — Визначення меж встановлення ціни

Рівень цін на товари- замінники	Рівень на цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціна на товар/послугу
-	-	-	-

### **5.3 Висновки до стартап-проєкту**

Аналізуючи стартап-проєкт можна зробити висновок. Попит на продукцію існує, буде існувати та буде вдосконалюватися, це підтверджується позитивною динамікою ринку та сферами застосування, особливо у військових цілях - такі системи є незамінними.

## ВИСНОВКИ

У результаті написання магістерської дисертації було досліджено методи аналізу руху та виявлення об'єктів за допомогою технічного зору, зокрема досліджено використання для цього методу Лукаса-Канаде та метод виявлення за допомогою кольорових фільтрів, створено їх програмну реалізацію та протестовано на відеопослідовності.

Був запропонований спосіб модифікації методу Лукаса-Канаде, що полягає у використанні ітеративного алгоритма Лукаса-Канаде, замість пірамідального.

Отримані результати тестування показують, що метод Лукаса-Канаде працює коректно для відеопослідовностей, що містять деталізовані сцени, без великих однорідних областей. За його результатами також можна встановити напрямки руху об'єкту.

Метод кольорових фільтрів працює коректно для відеопослідовностей, що містять явно виражений об'єкт інтересу, з відеопослідовностями які містять багато відтінків кольорів - виникають проблеми.

В майбутньому результати цієї роботи можуть використовуватись для продовження дослідження теми. Зокрема для реалізації на базі алгоритму оптимізованих версій, збільшення швидкості роботи алгоритму за допомогою використання багатопоточності, впровадження штучного інтелекту.

## СПИСОК ЛІТЕРАТУРИ

1. Gonzalez R.C. Digital Image Processing / R.C. Gozalez, R.E. Woods. — Pearson — 2018 — 1168 p.
2. Warren D.H. Electronic Spatial Sensing for the Blind: Contributions from Perception, Rehabilitation, and Computer Vision / D.H. Warren, Edward R. Strelow, — Springer, 1985. — 521 c.
3. Horn B.K.P. Determining Optical Flow / B.K.P. Horn and B.G. Rhunck // Artificial Intelligence. - 1981. - Volume 17, Issues 1–3.
4. Paragios N. Handbook of Mathematical Models in Computer Vision / N. Paragios, Y. Chen, O. Faugeras., — Springer, 2005. — 612 c.
5. Beauchemin S.S. The Computation of Optical Flow / S.S. Beauchemin, J.L. Barron // ACV Computing Surveys. — 1995. — No 27. — c.433-466.
6. Aqel M.O.A. Review of visual odometry: types, approaches, challenges, and applications / M.O.A. Aqel, M.H. Marhaban, M.I Saripan // SpringerPlus. — 2016. — No 5.
7. Lucas B.D. An Iterative Image Registration Technique with an Application to Stereo Vision / B.D. Lucas, T.Kanade // Proceedings of Imaging Understanding Workshop. — 1981. — c. 121-130.
8. Klein G., Murray D. Parallel tracking and mapping for small AR workspaces // Proceedings of 6th IEEE and ACM International Symposium on Mixed and Augmented Reality. 2007. P. 225–234.
9. Sun P. Two mechanisms that determine the Barber-Pole Illusion / P. Sun, C. Chubb, G. Sperling // Vision Research. — 2015. — Volume 111, Part A. — c.43- 54.
10. Al-Saif K.I. Color Image Enhancement Based on Contourlet Transform Coefficients / K.I. Al-Saif, A.S. Abdullah // Australian Journal of Basic and Applied Sciences. — 2013. — vol. 7. — P. 207—213.
11. OpenCV Library <http://opencv.org/>

12.Computer Vision System Toolbox <http://www.mathworks.com/products/computer-vision/>

## ДОДАТОК А

```

vidReader = VideoReader('video.mp4','CurrentTime', 5);
opticFlow = opticalFlowLK('NoiseThreshold',0.01);
h = figure;
movegui(h);
hViewPanel = uipanel(h,'Position',[0 0 1 1],'Title','Plot of Optical Flow Vectors');
hPlot = axes(hViewPanel);
while hasFrame(vidReader)
    frameRGB = readFrame(vidReader);
    frameGray = im2gray(frameRGB);
    flow = estimateFlow(opticFlow,frameGray);
    imshow(frameRGB)
    hold on
    plot(flow,'DecimationFactor',[5 5],'ScaleFactor',10,'Parent',hPlot);
    hold off
    pause(10^-3)
end

```



## ДОДАТОК Б

```

clc;
close all;
imtool close all;
clear;
workspace;
format long g;
format compact;
fontSize = 0;
folder = pwd;
baseFileName = 'video.mp4';
fullFileName = fullfile(folder, baseFileName);

if ~exist(fullFileName, 'file')
    fullFileNameOnSearchPath = baseFileName;
    if ~exist(fullFileNameOnSearchPath, 'file')
        errorMessage = sprintf('Error: %s does not exist in the search path
folders.', fullFileName);
        uiwait(warndlg(errorMessage));
        return;
    end
end

videoObject = VideoReader(fullFileName);
numberOfFrames = videoObject.NumberOfFrame;

hThresholds = [0.563, 0.833];
sThresholds = [0.037, 0.306];
vThresholds = [47, 67];

for k = 200 : numberOfFrames
    thisFrame=read(videoObject,k);
    hImage=subplot(3, 4, 1);
    imshow(thisFrame);
    axis on;
    caption = sprintf('Original RGB image, frame #%%d Of %%d', k,
numberOfFrames);
    title(caption);
    drawnow;

    hsv = rgb2hsv(double(thisFrame));
    hue=hsv(:,:,1);
    sat=hsv(:,:,2);

    val=hsv(:,:,3);
    subplot(3, 4, 2);
    imshow(hue, []);
    impixelinfo();
    axis on;
    title('Hue');
    subplot(3, 4, 3);
    imshow(sat, []);
    axis on;

```

```

title('Saturation');
subplot(3, 4, 4);
imshow(val, []);
axis on;
title('Value');
if k == 1
    set(gcf, 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]);
    set(gcf, 'Name', 'Demo by ImageAnalyst', 'NumberTitle', 'Off')
    hCheckbox = uicontrol('Style','checkbox',...
        'Units', 'Normalized',...
        'String', 'Finish Now',...
        'Value',0,'Position', [.2 .96 .4 .05], ...
        'FontSize', 14);
End

binaryH = hue >= hThresholds(1) & hue <= hThresholds(2);
binaryS = sat >= sThresholds(1) & sat <= sThresholds(2);
binaryV = val >= vThresholds(1) & val <= vThresholds(2);
subplot(3, 4, 10);
imshow(binaryH, []);
axis on;
title('Hue Mask');
subplot(3, 4, 11);
imshow(binaryS, []);
axis on;
title('Saturation Mask');
subplot(3, 4, 12);
imshow(binaryV, []);
axis on;
title('Value Mask');

coloredMask = binaryH & binaryS & binaryV;
coloredMask = bwareaopen(coloredMask, 500);
coloredMask = imfill(coloredMask, 'holes');
subplot(3, 4, 9);
imshow(coloredMask, []);
axis on;
title('Colored Blob Mask');
drawnow;

[labeledImage, numberOfRegions] = bwlabel(coloredMask);
if numberOfRegions >= 1
    stats = regionprops(labeledImage, 'BoundingBox', 'Centroid');
    if exist('hRect', 'var')
delete(hRect);
    end
    if exist('hText', 'var')
        delete(hText);
    end
    end

subplot(3, 4, 5);
hImage=subplot(3, 4, 5);
imshow(thisFrame);
axis on;

```

```

        hold on;
        caption = sprintf('%d blobs found in frame #%d Of %d',
numberOfRegions, k, numberOfFrames);
        title(caption);
        drawnow;

        for r = 1 : numberOfRegions
            thisBB = stats(r).BoundingBox;
            thisCentroid = stats(r).Centroid;
            hRect(r) = rectangle('Position', thisBB, 'EdgeColor', 'r',
'LineWidth', 2);
            hSpot = plot(thisCentroid(1), thisCentroid(2), 'y+', 'MarkerSize',
10, 'LineWidth', 2)
        end
        hold off
        drawnow;
    end
    if get(hCheckbox, 'Value')
        msgbox('Done with demo. ');
        return;
    end
end
msgbox('Done with demo. ');

```