МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ В. Н. КАРАЗІНА
ФІЗИКО-ЕНЕРГЕТИЧНИЙ ФАКУЛЬТЕТ
Кафедра інформаційних технологій в фізико-енергетичних системах

Пояснювальна записка

до дипломної роботи
за освітньо-кваліфікаційним рівнем магістр

«Обробка відеозображень із трекінгом рухомих об’єктів »

Виконав: студент 2 курсу, групи НФ-63
Юзюк Д.С.
Спеціальність:
105 Прикладна фізика та наноматеріали
Освітня програма: Прикладна фізика енергетичних систем
Керівник: к.т.н., доц. Лісін Д.О.

Рецензент:

Харків – 2018

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет імені В.Н. Каразіна

Факультет фізико-енергетичний
Кафедра інформаційних технологій в фізико-енергетичних системах
Рівень вищої освіти (освітньо-кваліфікаційний рівень) магістр
Спеціальність 105 Прикладна фізика та наноматеріали

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ проф. Нємченко К.Е.

“____” _____20___року

З А В Д А Н Н Я
НА ДИПЛОМНУ РОБОТУ (ПРОЕКТ)

Юзюка Дениса Сергійовича

1. Тема роботи «Обробка відео зображень із трекінгом рухомих об’єктів»
керівник роботи к.т.н. Лісін Д.О.
затверджені наказом по університету від “24” жовтня 2018 року № 0210-05/1991
2. Строк подання студентом роботи 30.11.2018
3. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити). Основна частина дипломної роботи представлена чотирма розділами. Перший розділ дипломної роботи містить загальну інформацію про існуючі методи і рішення для визначення та супроводження рухомих об’єктів. Другий розділ представляє детальну інформацію про алгоритм роботи та структуру розробленої програми. Третій розділ присвячений опису розробленої системи реалізації програми супроводження об'єктів в відео послідовності та огляду роботи програми .

4. План роботи

№ з/п	Назви етапів роботи
1	Вибір теми, узгодження, закріплення за кафедрою
2	Робота над літературою та іншими джерелами
3	Затвердження теми на кафедрі
4	Написання вступу до роботи та першого розділу
5	Написання другого розділу
6	Написання третього розділу
7	Написання четвертого розділу
8	Завершення написання дипломної роботи (висновки та література)
9	Подача науковому керівнику, доробка зауважень
10	Відгук, рецензія, підготовка до захисту
11	Попередній захист на кафедрі
12	Захист

5. Дата видачі завдання 12 вересня 2018 року.

Студент _____ **Юзюк Д.С.**

Керівник роботи _____ **Лісін Д.О.**

РЕФЕРАТ

Дипломна робота Юзюка Дениса Сергійовича на тему «Обробка відео зображень із трекінгом рухомих об’єктів»

Дипломна робота представлена на 56 сторінках; складається з вступу, трьох розділів та висновків; містить 28 рисунків, список використаної літератури з 20 найменувань.

Мета роботи - реалізувати програму трекінгу рухомих об’єктів. Для цього розглянемо та проаналізуємо існуючі рішення та засоби в області визначення та відстеження рухомих об’єктів. Надамо інформацію про платформу, яка має готовий функціонал для обробки зображення. Самостійно розробимо алгоритм та архітектуру власної програми.

Об’єкт дослідження – дослідити сучасні методи та засоби обробки зображень. Розглянути готові реалізації та функції засобів в області комп’ютерного бачення.

Предмет дослідження – програмний продукт обробки відео зображень із трекінгом рухомих об’єктів.

Дана дипломна робота присвячена розробці програми визначення об’єктів та їх трекінгу у відео послідовності. Приведено огляд алгоритмів та технологій для розпізнавання об’єктів і їх супроводження.

На основі вивченого розроблено алгоритм та архітектуру програми, наведено основні переваги і недоліки. Реалізовано програму відстеження та супроводження об’єктів у відео послідовності. Проведено тестування для детальної демонстрації роботи програми, а також надані рекомендації по покращенню алгоритму та програми в цілому.

КЛЮЧОВІ СЛОВА: КОМПЮТЕРНИЙ ЗІР,МОДЕЛЬ ЗАДНЬОГО ФОНУ,ГАУССОВА МОДЕЛЬ, ДЕТЕКТОР КАНІ, ФІЛЬТР КАЛМАНА.

РЕФЕРАТ

Дипломная работа Юзюка Дениса Сергеевича на тему «Обработка видео изображений трекингом подвижных объектов».

Дипломная работа представлена на 56 страницах; состоит из введения, трех глав и выводов; содержит 28 рисунков, список использованной литературы из 20 наименований.

Цель работы - реализовать программу трекинга движущихся объектов. Для этого рассмотрим и проанализируем существующие решения и средства в области определения и отслеживания движущихся объектов. Предоставим информацию о платформе, которая имеет готовый функционал для обработки изображения. Самостоятельно разработаем алгоритм и архитектуру собственной программы.

Объект исследования - исследовать современные методы и средства обработки изображений. Рассмотреть готовы реализации и функции средств в области компьютерного видения.

Предмет исследования - программный продукт обработки видео изображений трекингом движущихся объектов.

Данная дипломная работа посвящена разработке программы определения объектов и их трекинга в видео последовательности. Приведены обзор алгоритмов и технологий для распознавания объектов и их сопровождение.

На основе изученного разработан алгоритм и архитектуру программы, приведены основные преимущества и недостатки. Реализована программа отслеживания и сопровождения объектов в видео последовательности. Проведено тестирование для детальной демонстрации работы программы, а также даны рекомендации по улучшению алгоритма и программы в целом.

КЛЮЧЕВЫЕ СЛОВА: КОМПЬЮТЕРНОЕ ЗРЕНИЕ, МОДЕЛЬ ФОНА, ГАУССОВА МОДЕЛЬ, ДЕТЕКТОР КАНИ, ФИЛЬТР КАЛМАНА.

ABSTRACT

Degree work of Yusyuk Denis Sergeevich on the topic "Processing of video images with tracking of moving objects"

Thesis is presented on 56 pages; consists of an introduction, three chapters and conclusions; contains 28 drawing, a list of used literature of 20 names.

The purpose of the work is to implement a tracking program for moving objects. To do this, consider and analyze existing solutions and tools in the area of determining and tracking moving objects. We will provide information about a platform that has a ready-made function for image processing. We will independently develop the algorithm and the architecture of our own program.

The object of the research is to investigate modern methods and means of image processing. Consider ready-made implementations and features of tools in the field of computer vision.

Subject of research - software product for processing video images with tracking moving objects.

This thesis is devoted to the development of a program for defining objects and their tracking in video sequences. An overview of algorithms and technologies for object recognition and their support is presented.

Based on the developed algorithm and architecture of the program, the main advantages and disadvantages are presented. The tracking and tracking software for video sequences is implemented. A test has been conducted for a detailed demonstration of the program's work, as well as recommendations for improving the algorithm and the program as a whole.

KEYWORDS: COMPUTER ZIR, MODEL OF THE BACKGROUND, HOUSE MODEL, CHANNEL DETECTOR, KALMAN FILTER.

ЗМІСТ

1 ЛІТЕРАТУРНИЙ ОГЛЯД 10

1.1 Проблематика та особливості 10

1.2 Загальна послідовність обробки відеосигналу 11

1.3. Аналіз існуючих рішень 12

1.4 Методи зображення об'єктів 13

1.5 Виділення основних рис відстеження 15

1.6 Методи визначення об'єктів 15

1.7 Адаптивні порогова обробка зображень 25

1.8 Методи відстеження об'єктів 27

2 РОЗРОБКА ПРОГРАМИ ТРЕКІНГУ ОБ'ЄКТІВ НА ВІДЕОПОСЛІДОВНОСТІ 31

2.1.Розробка алгоритму роботи програми 31

2.2 Фільтр Калмана 34

2.3 Висновки 36

3 ТЕСТУВАННЯ ПРОГРАМИ 37

3.1 Оцінка роботи програми 37

3.2 Система « Object Tracking » 38

3.3 Висновки 48

ВИСНОВКИ 50

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 51

ВСТУП

Визначення рухомих об’єктів є важливою задачею у сфері комп’ютерного зору. Популяризація потужних комп’ютерів, поширення високоякісних дешевих камер та зростання потреб для автоматизованого аналізу відео сприяло зростанню великого інтересу в області алгоритмів супроводження руху. Існує три основні ключові кроки аналізу відео: розпізнавання об’єкта, який рухається, спостереження за об’єктом кадр за кадром, і аналіз об’єктів для визначення їх поведінки.

Отже, використання алгоритму відстеження об’єкту є доречним у вирішенні таких задачах:

- Взаємодію людини і комп’ютера, наприклад, розпізнавання жестів,
- спостереження за шосе і залізницями для запобігання аварій.
- Контроль доступу на об'єкти, що охороняються,виявлення підозрілої активності.
- Автоматичне розпізнавання об’єктів,основане на особливостях їх руху.
- Виявлення перешкод, що виникають по ходу руху автомобіля.

Програма відстеження повинна вирішувати проблему оцінки траєкторії руху об’єкту на зображенні, що рухається навколо сцени. Інакше кажучи , спостерігач виділяє рухомий об’єкт контуром ,або точками чи іншим способом. В залежності від поставлених задач програма спостереження надає додаткову інформацію про рухомий об’єкту,наприклад: площу,розмір,форму.

Виділення об’єктів являється складною задачею через низку факторів:

- Шум на зображені
- Складна траєкторія руху об’єктів
- Часткове або повне перекриття об’єктів
- Зміну освітлення
- Необхідність визначати об’єкти у реальному часі

Актуальність теми. Питанню створення математичних моделей цифрової обробки відеозображень у теперішній час приділяють велику увагу багато дослідників, оскільки кількість практичних задач, у яких

використовуються трекінг рухомих об’єктів та їх аналіз, весь час зростає. На сьогоднішній момент обробка й аналіз відеозображень широко використовуються в охоронних системах стеження, військовій справі і ще дуже багатьох галузях .

Розробка нових методів трекінгу рухомих об’єктів залишається актуальним напрямком досліджень для спрощення аналізу та скорочення обчислювальних витрат, особливо при спостережені у реальному часі.

Мета і задачі дослідження**.** Метою дипломної роботи є визначення рухомих об’єктів на відео зображеннях за допомогою технічних систем зору. Необхідно розробити математичну і комп'ютерну модель, вибрати методи і створити алгоритми трекінгу для впровадження в реальні системи обробки зображень. Для досягнення цієї мети в дипломної роботи поставлені такі основні задачі:

- провести аналіз існуючих методів та алгоритмів виділення та відстеження рухомих об’єктів на відео,
- розробити алгоритм власного системного продукту трекінгу рухомих об’єктів на відео зображеннях ,на основі вивчених методів,
- реалізувати комп’ютерну модель запропонованого алгоритму для оцінки якості роботи, дослідження впливу параметрів і виявлення особливостей практичної реалізації.

1 ЛІТЕРАТУРНИЙ ОГЛЯД

1.1 Проблематика та особливості

В останні роки істотно зріс інтерес до цифрової обробки відеозображень, що пов'язано з різким зниженням цін на цифрові телекамери, завдяки чому вони стали доступними широкому класу користувачів і почали впроваджуватися в багатьох сферах людської діяльності для вирішення завдань автоматичного контролю і відеоспостереження. І якщо спочатку телекамери знайшли застосування в системах охоронного телебачення, то незабаром їх стали використовувати для вирішення найрізноманітніших завдань, таких як:

• спостереження за шосе і залізницями для запобігання аварій,

• вимірювання швидкості автомобілів,

• фіксування проїзду автомобілями перехрестя на заборонний сигнал світлофора ,зупинки в недозволеному місці,неправильних паркувань,

• моніторинг банків, гіпермаркетів, аеропортів, музеїв, вокзалів, парковок ,

• здійснення автоматичного контролю повітряної обстановки,

• контроль доступу на об'єкти, що охороняються,

• вимір інтенсивності дорожнього руху і підрахунок числа транспортних засобів різного класу (легкові автомобілі, вантажівки, фури),

• проведення статистичних досліджень в торгових і розважальних центрах,

• безконтактне спостереження за тваринами,

• виявлення перешкод, що виникають по ходу руху автомобіля.

Однією з найбільш складних і актуальних завдань обробки відеозображення є проблема виділення і розпізнавання рухомих об'єктів при наявності різного роду перешкод і створення на цій основі системи моніторингу. Головне завдання таких систем - інформувати людину про ситуацію, що склалася в полі зору камери, і по можливості заздалегідь передбачити наступні дії помічених об’єктів.

Спочатку завдання зводилася до простого детектування рухомих об'єктів. Уже в такому вигляді це важливо для багатьох систем безпеки, коли в зоні, що охороняється виключено будь-який рух. Більш досконалі системи мають на увазі наявність у своєму складі інтелектуальних відео детекторів руху, здатних відрізнити рухи людини від собаки, машини або дерево, що розгойдується на вітрі.

На даний момент лише деякі системи безпеки можуть похвалитися такими можливостями. Привертаючи увагу до певного монітора і автоматично реєструючи події, що відбуваються, такі системи значно підвищують рівень безпеки об'єкту, що охороняється. Крім того, відеодетектори руху часто використовуються для інтелектуальної компресії, що дозволяє значно економити дисковий простір при архівації відеозображень.

Процес побудови зазначених систем є складним технологічний ланцюжок, що включає отримання цифрового зображення, його обробку з метою виділення важливої інформації та аналіз цієї інформації для вирішення певної задачі. Звичайно, ідеальним є створення універсальної системи, яка в умовах можливих обмежень забезпечувала б ефективний моніторинг рухомих об’єктів.

1.2 Загальна послідовність обробки відеосигналу

Незважаючи на те, що в світі створюється велика кількість систем, що мають різноманітне призначення, послідовність обробки відеосигналу в них приблизно однакова.[20] Більш того, аналіз різних систем показав, що в основі кожної з них лежать практично одні і ті ж модулі. Умовно обробку відеозображень в таких системах можна розділити на наступні етапи:

• виділення переднього плану.
• виділення і класифікація об'єктів, що рухаються.
• відстеження траєкторії руху знайдених об'єктів.
• розпізнавання і опис дій об'єктів, що становлять інтерес.

Процес виділення переднього плану полягає в відділенні рухомих фрагментів зображення від нерухомих, які називають фоновими або належать

до заднього плану. Від того, наскільки акуратно і коректно вирішена ця задача, залежать всі наступні етапи обробки інформації, а також необхідні обчислювальні ресурси. Саме тому етапу виділення переднього плану зображення і застосованим для цього методам приділено особливу увагу розробників.

На другому етапі виділення і класифікації спочатку проводиться сегментація зображення переднього плану. Знаходяться компактні області, що рухаються з однаковою швидкістю, які вважаються рухомими об'єктами.
Після цього можна переходити до наступного етапу - відстеження траєкторії кожного об'єкта, що рухається(трекінгу). Для здійснення трекінгу потрібно встановити взаємно однозначну відповідність між виявленими об'єктами на послідовних кадрах.

При цьому забезпечується тимчасова ідентифікація виділених областей зображення і видається відповідна інформація про об'єкти в спостережуваній зоні, а саме: траєкторія, швидкість і напрямок руху. На заключному етапі обробки проводиться розпізнавання і опис дій виділених об'єктів.

1.3. Аналіз існуючих рішень

Основні проблеми, що виникають при створенні та роботі трекінгу рухомих зображень, пов’язані з виглядом об’єкта та схожими іншими об’єктами на зображенні. Вигляд інших об'єктів може бути накладений на задній фон. В такому випадку, важко знайти на зображенні потрібний рухомий об’єкт. На рисунку 1.1 зображено невизначеність кольору, що привела до неправильного визначення об’єктів стеження.[3]



Рис.1.1 – Помилкове виділення об’єктів

Також важко розпізнати об'єкти, вигляд яких в площині об’єктива змінився через такі фактори:

• Зміна позицій, під час руху вигляд об’єкту на площині зображення змінюється, наприклад якщо повертає чи крутиться;

• Зміна освітлення, інтенсивність, колір може вплинути на вигляд об'єкта;

• Зміна освітлення в основному плані призводить до проблем у сусідніх сценах**.** Наприклад, хмари закриваючи сонце змінюють навколишнє світло, а також кут між поверхнею об’єкта і напрямком світла. Це впливає на наше сприйняття зображення через лінзу камери;

• Шум. В залежності від якості матриці камери отримані зображення мають певну частину шуму;

• Перекриття. Спостерігати за об’єктом , коли він частково або повністю перекрита іншим об’єктом на зображенні. Перекриття виникають коли:

а) Об’єкт рухається за нерухомими предметами , наприклад, двері чи колона;

б) Інші об’єкти, які рухаються, затуляють вид на нього.

1.4 Методи зображення об’єктів

Об’єктом спостереження являється будь-що важливе для подальшого аналізу. Наприклад, людина, машина на дорозі, риба в акваріуму. Об’єкти представлені їх формою та виглядом. Способи представлення, які широко використовуються для трекінгу, зображено на Рис. 1.2 [11]:

• Точки. Об’єкти представляють собою точку, або набір точок (Рис. 1.2(b)). Підходить для спостереження за невеликими регіонами зображення.

• Прості геометричні форми. Форма об’єкта представлена прямокутником чи еліпсом і т.п. (Рис 1.2(c,d)). Рух об’єктів моделюється як переміщення, афінні чи проективні перетворення. Підходить щоб стежити за твердими об’єктами.

• З’єднані між собою частини. З’єднані між собою об’єкти складаються з частин тіла, що тримаються разом за рахунок суглобів. Наприклад, людське тіло - це об’єкт, який складається з голови,торсу, ніг, рук сполучених

суглобами (Рис. 1.2(e)). Співвідношення частин регулюється моделями кінематичного руху.

• Контур об’єкту чи силует. Силует знаходиться всередині контуру (Рис. 1.2(i)). Підходить щоб стежити за нетвердими об’єктами. Контур представляє собою границю об’єкта (Рисунок 1.2(g, h)).

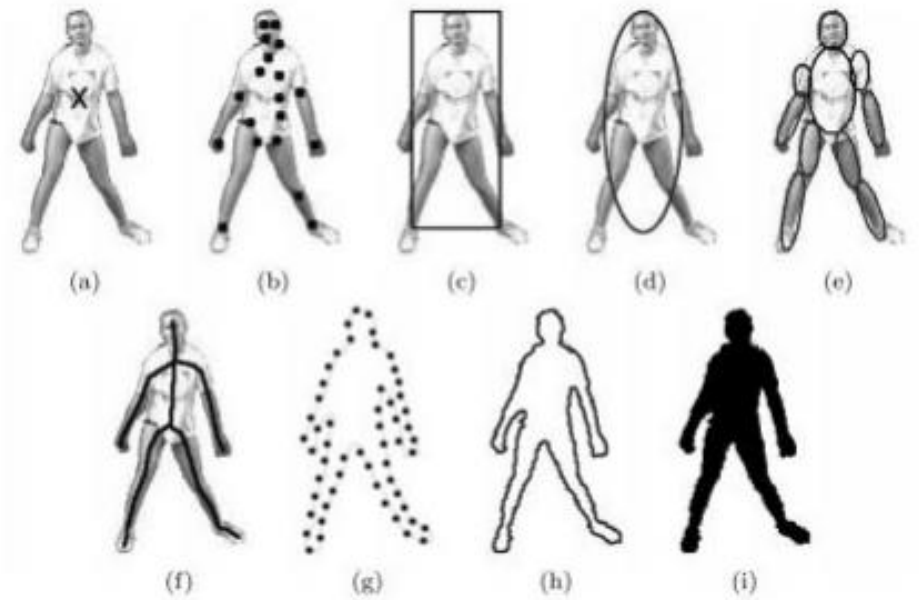


Рис. 1.2 – Варіанти представлення об’єкта: a - центр об’єкта, b- особливі точки, c - об’єкт у вигляді прямокутника, d - об’єкт у вигляді еліпса, e – з’єднання форм, f - скелетна модель, g та h - контур об’єкта, i - силует.

Також існує ще декілька способів представлення вигляду об’єкта. Найбільш широко використовуються такі:

• Багато-ракурсна модель. Ця модель з різних ракурсів описує об’єкт .

• Шаблони. Формуються з простих форм і силуетів. Перевага заключається в тому, що шаблони несуть не тільки просторову інформацію, а і про зовнішній вигляд. Використовуються, для нерухомих об’єктів .

• Активна модель вигляду. Генерує одночасно форму і вигляд об’єкта. Об’єкт представлений набором орієнтирів кожен з яких зберігає вектор вигляду такі як колір, текстура, градієнт.

1.5 Виділення основних рис відстеження

Для визначення об’єкту відстеження ,мають бути задані унікальні характерні риси визначення порівняно з іншими. Виділення характерних рис тісно пов’язано з представленням об’єкта. Так границі об’єкта є характерними рисами для представлення контуром. Основні візуальні риси:

• Границі. Значні зміни в інтенсивності зображення створюють кордони об’єкта. Важливою властивістю границь являється менша чутливість до змін світла порівняно з рисами кольору. Розпізнавання границь використовуються для знаходження цих змін.

• Оптичний потік - представлення видимого сліду руху об'єкта, поверхонь, і граней візуальної сцени, які прослідковуються під час відносного руху між сценою і спостерігачем . Найвідоміші алгоритми знаходження оптичного потоку: Лукас-Канаде, Хорна.

1.6 Методи визначення об’єктів

Кожен метод відстеження об’єктів потребує механізм котрий визначає об’єкт на кадрах чи коли об’єкт вперше з’явиться на відео. Загальний підхід базується на використанні інформації з одиничного кадру. Хоча деякі алгоритми можуть використовувати інформацію підраховану з послідовності кадрів.

Технічно нескладним і надійним способом є спосіб опису об'єкта набором його особливих точок. Ці точки повинні мати наступні властивості: їх не повинно бути багато і їх розташування повинне слабо змінюватися при невеликій зміні розміру і орієнтації об'єкта і не залежати від зміни освітлення. Повною мірою цими властивостями володіють кутові точки на зображенні об'єкта. Далі ми також будемо називати особливі точки ключовими точками, а

алгоритми, які використовуються для виявлення на зображені кутових точок, для стислості називати детекторами кутів.

1.6.1 Різновидності детекторів точок

Приміняється для знаходження особливих точок , які мають чіткі текстури в місцях їх локалізації, які не залежать від змін навколишнього світла і розташування камери.

1. Один з перших детекторів кутів, запропонований в [9], заснований на порівняні інтенсивностей зображень в невеликому локальному вікні, що оточує кожен піксель вихідного зображення. Якщо розглянутий піксель відповідного кута, то при зсуві вікна різницю інтенсивностей повинна значно змінитися, в іншому випадку ця різниця повинна майже не змінюватися.

Позначимо інтенсивність зображення в т очку (x, y) як $I(x, y)$, тоді при зсуві вікна на $(\Delta x, \Delta y)$ відповідно уздовж осей абсцис і ординат зміна суми квадратичних різниць в кожній точці зображення (x_0, y_0) запишеться наступним чином:

$$E(\Delta x, \Delta y) = \sum_{i=x_0-w}^{x_0 \to w} \sum_{j=y_0}^{y_0 \to h} \left| I(i - \Delta x, j + \Delta y) - I(i, j) \right|^2 \qquad (1.1),$$

де w і h відповідно ширина і висота локального вікна.

Детектор працює наступним чином. Для всіх допустимих зрушень знаходиться мінімальне значення суми квадратичних різниць (1.1). Якщо це значення є найбільшим в заданій околиці, то вважається, що дана точка відповідає куту.

Зазвичай використовуються одиничні зсуви по горизонталі, вертикалі і по діагоналі. Ясно, що якщо напрями сторін кута не збігаються з напрямками зсуву, то такі кути не будуть виявлятися.

2. Оператор Моравека заснований на обчисленні середнього значення квадратів різниць значень пікселів зображення у вертикальному, горизонтальному і діагональних напрямках всередині вікна заданого розміру. Сила кута рахується як мінімальна різниця квадратів між полем і його

оточенням(у всіх напрямках). Зміна інтенсивності та точки представлені на Рис 1.3:

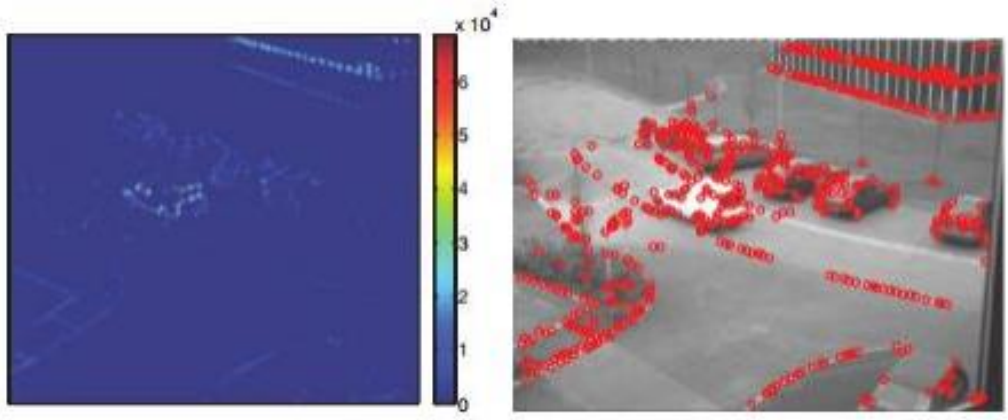


Рис 1.3. Визначення особливих точок оператором Моравека

Фрагмент зображення, з великими змінами яскравості дає більше значення оператора Моравека. Знайдені фрагменти мають максимальні значення в заданих областях. Для того щоб розподілити знайдені фрагменти рівномірно по області пошуку використовується наступний алгоритм:

- для обраної області розраховується значення оператора Моравека;
- знаходиться фрагмент з найбільшим значенням. Його центральна точка вибирається в якості опорної точки.
- поле навколо знайденої проміжної точки затінюється у відповідності з наступною затіненою функцією:

$$I_{i,j} = I_{i,j} \times \left(1 - e^{\frac{\sqrt{(i-x)^2 + (j-y)^2}}{dist}}\right) \qquad (1.2)$$

де $dist$ – задана мінімальна відстань між проміжними точками.

- перехід до першого етапу до тих пір, поки не буде вибрано задану кількість проміжних точок**.**

Даний алгоритм гарантує рівномірний розподіл проміжних точок, але не повне покриття ними області пошуку. Для вирішення цієї проблеми зона пошуку ділиться на сектори, після чого алгоритм застосовується для кожного сектора, використовуючи засоби паралельних обчислень. Таким чином, ми

отримаємо повний і рівномірний розподіл проміжних точок при максимально ефективному використанні обчислювальних ресурсів.

3. Детектор Харріса [4] дозволяє визначити кути без використання операції зсуву вікон, яка вимагає значних обчислювальних ресурсів.

Розглянемо наступну матрицю M – матриця зміни інтенсивності для кожного пікселя.

$$M = \sum_{(u,v)} \in w(u,v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{1.3}$$

Де I_x і I_y відповідно відповідають похідним зображення по x і y, які можуть бути знайдені, наприклад, за допомогою операторів Превітта або Собеля, маски яких зображені на Рис. 1.4.

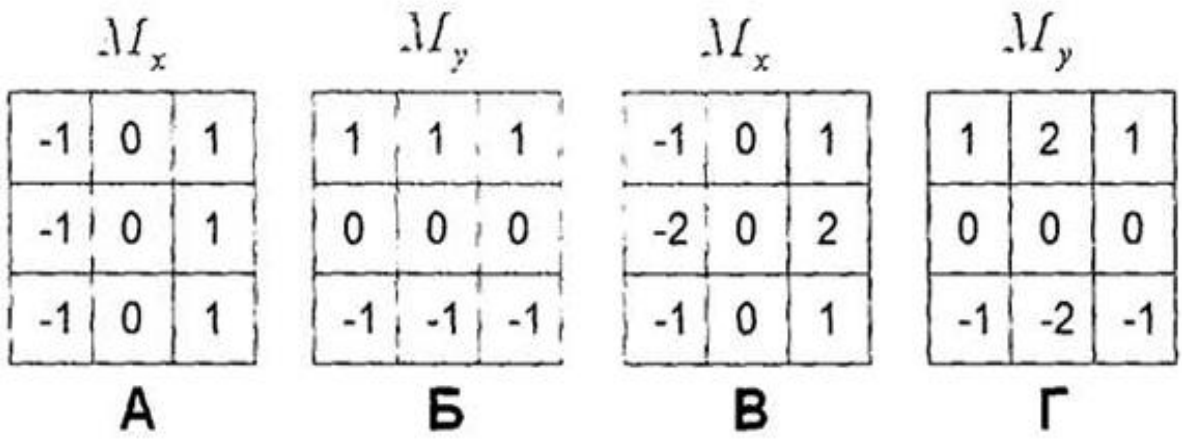


Рис. 1.4. А,Б-маски Превіта ,В,Г – маски Собеля.

Детектор Харріса заснований на аналізі значення наступної величини:

$$R = \det M - k(trM)^2 > k \tag{1.4}$$

R – міра кута, де k – емпірична константа**.** Зазвичай для нього рекомендується вибирати значення з діапазону 0.05-0.15. Значення величини

R позитивно в області, що містить кут, негативно для області, що містить ребро, і приймає невеликі по модулю значення для інших областей.

Для кутових точок значення $R \geq 0$. Потім відсікаються точки за граничним значенням**.** Визначаються локальні максимуми в заданому радіусу і приміняються в якості особливих точок.

4. Кутовий детектор Ши-Томасі визначає мінімальні власні значення матриці напряму M,наведеній у формулі (1.4) і багато в чому збігається з детектором Харріса, але різниться в обчисленні міри відгуку: алгоритм безпосередньо обчислює значення $\min(\lambda_1,\lambda_2)$, оскільки робиться припущення, що пошук кутів буде більш стабільним. Той же принцип використовують для аналізу оптичного потоку Лукаса-Канаде . Подробиці алгоритму дані в статтях [12] і [15]

5. SIFT-визначає екстремуми піраміди Гаусіан. Потім відфільтровуються особливі точки і визначається напрямок ключових точок ,наприклад ,розташованих на границі.

Метод SIFT (Scale Invariant Feature Transform) [10] є найбільш стійким до всіляких перекручень зображення. Він складається з наступних кроків [5].
На першому етапі здійснюється згортання зображення з гауссовими фільтрами

$$\eta(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 - y^2}{2\sigma^2}\right) \tag{1.5}$$

котрі мають різні значення параметра σ. З їх допомогою обчислюється оператор різниці гауссіанів:

$$D(x, y, \sigma) = (\eta(x, y, k\sigma) - \eta(x, y, \sigma)) * I(x, y), \tag{1.6}$$

де k - параметр зміни масштабу, а символом $*$ позначена операція згортання. Точки , в яких $D(x, y, \sigma)$ досягає екстремуму, відбираються в якості ключових точок.

На наступному етапі відкидаються точки, розташовані уздовж ребер.Для цього використовуються властивості матриці Гессе

$$H = \begin{pmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{pmatrix}, \quad (1.7)$$

обчисленої в точках, відібраних на попередньому етапі. Оскільки головна кривизна вздовж ребра приймає набагато більше значення, ніж в перпендикулярному до нього напрямку, а власні числа матриці Гессе пропорціональні головній кривизні $D(x, y, \sigma)$, то для виключення зайвих точок досить порівняти власні числа.

Для інваріантного опису області навколо особливої точки може бути використаний метод PCA-SIFT [17]. Суть цього методу полягає в наступному. Після того як ключова точка була виявлена, навколо неї вибирається частина зображення заданого розміру. У кожній точці виділеного зображення обчислюється градієнт, після чого отриманий вектор градієнтів нормується.

1.6.2. Вилучення фону

Визначити об’єкти можна за допомогою використання представлення сцени, яка називається вилучення фону. Вилучення шукає зміни в зображенні для кожного наступного кадру ,значні зміни між кадрами означають ,що об’єкт рухається. Зокрема, багато комп'ютерних систем зору використовуються у зовнішніх сценах, інтенсивність світла зазвичай коливається, і тому надійний метод фонового моделювання повинен мати можливість адаптуватися до поступових змін світла в навколишньому середовищі.

Методи виявлення переднього плану можуть бути застосовані для ефективного розмежування об'єктів переднього плану, у тому числі рухомих або статичних об'єктів із фону, що дуже важливо при застосуванні відеоаналізу, особливо відеоспостереження. Фонові моделі можуть отримати хороші результати виявлення переднього плану.

Зміни освітлення не тільки прогресивні, але й раптові одноразові висвітлення також часто відбуваються на сцені. Наприклад, раптова зміна світла або сонячного світла, затемнення хмарами, сильно впливає на

зовнішній вигляд фонової моделі. Тому здатність швидко адаптуватися до раптових змін освітлення є ключовим моментом для оцінки методів моделювання фону.

1.6.3. Гауссові моделі змішування

Також використовують Гаусові моделі змішування для моделювання кольору пікселя. Метод полягав тому ,що кожен піксель зображення порівнюється з моделлю фону ,кожного Гаусіану моделі поки не буде знайдений потрібний. Після знаходження ,значення Гаусіану оновлюється ,або додається новий Гаусіан зі значенням пікселя поточного кольору.

$$P(x_t) = \sum_{i=1}^{K} w_{i,t} \times \eta(x_t, \mu_{i,t}\Sigma_{i,t}), \tag{1.8}$$

де K - це число гауссових розподілів, $\eta(x_t, \mu_{i,t}\Sigma_{i,t})$ - гауссова функція щільності імовірності, $w_{i,t}$ - його вага в момент t, $\mu_{i,t}$ - середнє значення і-го Гауссіана в суміші в момент часу t, а $\Sigma_{i,t}$ - коваріаційна матриця і-го Гауссіана в суміші в момент t. У практичних випадках матриця коваріації може бути прийнята як діагональна матриця, а K - від 3 до 5.



Рис 1.5 – Гауссова модель змішування для моделі заднього фону

Моделювання фону може бути здійснене іншим способом за допомогою спектрального розкладу матриць. Для вхідних кадрів I ,розміром $m \times n$ і матриці заднього фону B , розміром $k \times l$,де $l = m \times n$, тоді матриця розкладається по формулі:

$$C = B^T B, \tag{1.9}$$

Усі можливі освітлення з поля зору охоплюються отриманими власними векторами , як зображено на рис 1.5[18]:

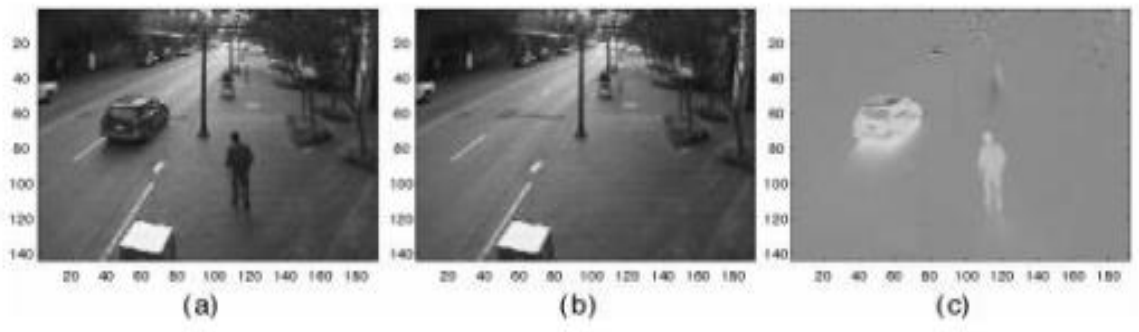


Рис 1.6 – Використання спектрального розкладу. Оригінальне зображення(a), отримане зображення заднього фону (b) , зображенні різниці (c)

Модель гауссової суміші вирівнює параметри за допомогою двох типів значень, маси компонентів суміші та компонентних засобів та коваріацій.

Якщо кількість компонентів відома, максимізація очікування - це техніка, яка найчастіше використовується для оцінки параметрів моделі суміші. У теорії ймовірностей, моделі, як правило, вивчаються за допомогою методів оцінки максимальної правдоподібності, які прагнуть максимально збільшити вірогідність або ймовірність спостережуваних даних з урахуванням параметрів моделі. На жаль, пошук рішення максимального правдоподібності для моделей сумішей шляхом диференціювання вірогідності журналу та вирішення для, як правило, аналітично неможливо.

Максимум очікування (EM) - це чисельна методика оцінки максимальної правдоподібності, і вона зазвичай використовується, коли вирази закритої форми для оновлення параметрів моделі можна обчислити . Максимум очікування - це ітеративний алгоритм, який має зручне властивість, що максимальна ймовірність даних суттєво збільшується з кожною наступною ітерацією, тобто гарантується підхід до локального максимуму.

Гаусова модель були використані недавно для вилучення функцій з мовних даних для використання в системах розпізнавання мовлення [7]. Вони також широко використовувались при відстеженні об'єктів декількох об'єктів, де кількість компонентів суміші та їх засоби передбачають розташування об'єктів на кожному кадрі в відеопослідовності [14]. Алгоритм використовується для оновлення компонентних засобів з часом, як оновлення відеокадрів, що дозволяє відстежувати об'єкти. [8]

1.6.4. Сегментація

Відеопотік може містити велику різноманітність рухомих об'єктів. Одні з них представляють інтерес для моніторингу - наприклад, зображення людей, транспортних засобів, тварин. Інші об'єкти, такі як гілки дерев, тіні, що обертаються двері, створюють перешкоди при відеоспостереження.

Для того щоб провести класифікацію об'єктів, що рухаються, потрібно спочатку відокремити кожен об'єкт від інших. Ця операція називається сегментацією об'єктів. Результатом проведення сегментації є або позначений набір пікселів переднього плану, або вказівку вершин мінімально можливого прямокутника , всередині якого знаходиться виділений об'єкт (див. Рис. 1.7).

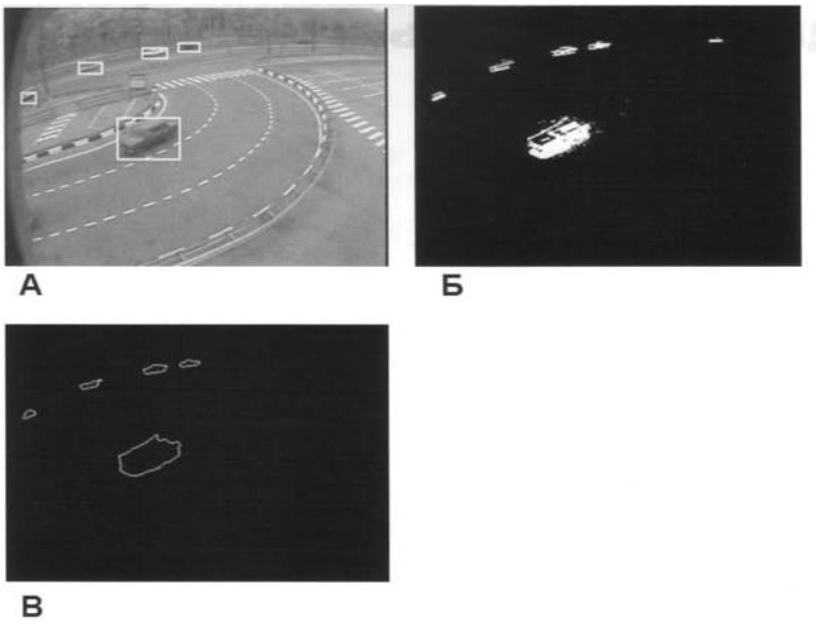


Рис. 1.7. Виділення рухомих об'єктів: А - за допомогою описаного прямокутника, Б - шляхом виділення блобів, В - описом кордону блобів

Практика показує, що після сегментації об'єктів переднього плану не слід відразу приступати до їх класифікації, а потрібно провести додаткову обробку, мета якої полягає в усуненні дефектів в зображеннях виділених об'єктів. Такі вади найпростіше усунути шляхом виконання відповідних морфологічних операцій, наприклад дилатації і подальшої ерозії . При правильному виборі для цих операцій структурного елемента цілісність про об’єктів буде відновлена, а контури об'єктів стануть більш плавними.

Сегментацією називається процес розбиття зображення на складові його об'єкти. Це завдання ретельно вивчається в більшості книг, присвячених цифровій обробці зображень. Що стосується завданням обробки відеоряду виникає ряд особливостей, як полегшують її рішення, так і ускладнюють.

Оскільки процесу сегментування піддаються тільки рухомі об'єкти, сумарна площа яких зазвичай істотно менше площі, займаної всім зображенням, то обсяг оброблюваної інформації менше, ніж при сегментації статичних зображень. До того ж рухомі об'єкти часто бувають ізольованими однин від одного . Ці фактори полегшують рішення поставленого завдання.

Існує також проблема збірки виділених об'єктів з частин, на які вони були поділені в результаті роботи детектора руху. Така проблема може виникнути, наприклад, в тому випадку, якщо частина рухомого об'єкту має колір, що співпадає з кольором фону, в результаті чого в цьому місці можуть з'явитися порожнечі химерної форми. А головною вимогою до вживаного алгоритму зазвичай є можливість його використання в реальному часі, що накладає дуже жорсткі обмеження [6,13].

Дуже ефективним з точки зору обчислювальних ресурсів є використання для сегментації методу кластерного аналізу. Суть кластеризації полягає в тому, що всі вихідні об'єкти розбиваються на кілька непересічних груп таким чином, щоб об'єкти, що потрапили в одну групу, мали подібні характеристики, в той час як у об'єктів з різних груп ці характеристики повинні значно відрізнятися .

У разі використання кольорових зображень вектори стають п’ятимірними. В результаті такого кодування в один кластер повинні потрапити групи пікселів, розташовані близько один до одного і при цьому котрі мають близьку інтенсивність або схожий колір. Сегментація розділяє зображення на схожі зони .Алгоритм вирішує проблеми гарного і ефективного розбиття зображення.

1.7 Адаптивні порогова обробка зображень

Найбільш досконалими системами розпізнавання об’єктів являються адаптивні , ці методи мають таку назву тому ,що вони настроюються відповідно до даних які обробляють. Такі системи виконують більш складні задачі ,такі як знаходження обличчя заданої людини.

Метод опорних векторів знаходить оптимальну площину ,перпендикулярну найкоротшому відрізку між випуклими оболонками різних класів. Опорні вектори це дані що лежать на границі гіперплощини і відповідають класам об’єктів та необ’єктним класам. Нелінійне відображення будується у випадку якщо дані не можуть бути розділені площиною. В деякому багатомірному просторі будується нелінійне відображення в якому дані можуть бути розділені. [19]

Різновиди адаптивних методів:

• штучні нейронні мережі,

• метод опорних векторів,

• методи кластеризації даних,

• методи бустінга.

1) Штучні нейронні мережі з'явилися в результаті спроб моделювання процесів розпізнавання мозком живих істот, тому навіть термінологія, що застосовується в цій області, частково запозичена з біології. У розділі розглядаються два найбільш часто вживаних типу нейронних мереж: прямого поширення і мереж з нейронами радіального базису. Велику увагу приділено

організації процесу настройки нейромережі, оскільки він, як правило, викликає великі труднощі навіть у кваліфікованих фахівців.

2) Метод опорних векторів був запропонований радянськими вченими В. Вапніка і А. Червоненкіса. В іноземній літературі він відомий як Support Vector Machine (SVM). Метод опорних векторів заснований на ідеї побудови оптимальної гіперплощини, що розділяє дві безлічі. Основна структурна відмінність SVM від нейромереж полягає в тому, що для нейромережі кількість параметрів коефіцієнтів має апріорно задаватися користувачем на підставі деяких евристичних міркувань. У разі застосування SVM кількість параметрів, що настроюються автоматично визначається під час налаштування і зазвичай істотно менше, ніж число векторів в навчальній послідовності. Відмінними від нуля залишаються тільки коефіцієнти при так званих опорних векторах, за допомогою яких і будується гіперплощину.

3) Методи кластерного аналізу використовуються для вирішення завдання, в якому потрібно розбити задану вибірку об'єктів на непересічні підмножини, названі кластерами, так щоб кожен кластер складався з схожих об'єктів, а будь-які два об'єкти, що належать різним кластерам, істотно відрізнялися. Процес побудови кластерів належить до розряду адаптивних алгоритмів, проте, на відміну від раніше розглянутих нейромереж і методу опорних векторів, він відноситься до класу методів навчання «без вчителя».

Це означає, що в прикладах, які використовуються при навчанні, присутні тільки вхідні дані і відсутня будь-яка інформація про їх належність до того чи іншого класу.

4) Методи бустінга призначені для підвищення якості системи, що розпізнає за рахунок використання набору експертів, яких також називають «елементарними розпізнавання», або класифікаторами. Як експерт може виступати будь-яка розпізнає система, наприклад заснована на застосуванні штучної нейронної мережі.

1.8 Методи відстеження об’єктів

Метою відстеження об’єкта є знаходження рухомого об’єкту на кожному кадрі ,а також створення та прогнозування траєкторії об’єкту. Задача встановлення зв’язку між станами об’єкта на кожному кадрі вирішується окремо або разом. Кожна обрана модель представлення об’єкту накладає відповідний ліміт на спосіб відстеження.

1.8.1 Моделі стеження за окремими точками

Відстеження об’єктів може відбуватися за окремими точками на кожному кадрі. Подібна задача являється складною через наявність на зображені перекриттів ,неправильного визначення ,входу і виходу об’єкта з кадру.

Методи поділяються на дві категорії :статичні та детерміністичні.

Детерміністичні методи визначаються за допомогою асоціації кожного об’єкту кожного попереднього кадру з попереднім . Супроводження точок обмежене наступними чинниками:

- Близькість
- Максимальна швидкість
- Незначні зміни швидкості (гладкий рух)
- Загальні рухи
- Жорсткість

Данні обмеження використовуються і статичному методі відстеження. Статичний метод заклечається у використанні Байесового фільтра ,котрий працює в два етапи : передбачення та корекція. Використання фільтра найбільш оптимальне через то ,що матриця камери завжди має шум. Більш того рух об’єктів ускладнюється випадковою траєкторією ,наприклад, автомобіль ,що виконує маневр.

Multiple Hypothesis Tracking**.** Алгоритм здатний проводити трекінг об’єктів ,що з’явилися у кадрі в процесі. Інтерактивний алгоритм створює гіпотези в яких намагається передбачити положення кожного об’єкту у

наступному кадрі. Прогнози порівнюються зі справжніми даними і на основі цього обновлюються .[18]

1.8.2 Відстеження ядра(Kernel Tracking)

Mean Shift. Порівнює гістограми регіонів зображення ,максимізуючи схожість вигляду інтерактивна. Проводить порівняння гістограми об'єктів Q, з вікном гіпотетичного знаходження об'єкта P.

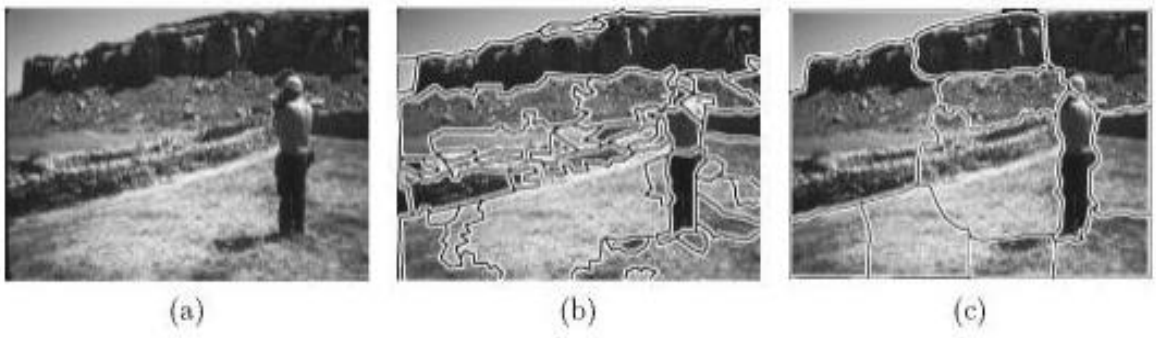


Рис 1.8. – Сегментація зображення(a) , MeanShift(b), розрізом графа(c)

Схожість гістограм визначається за формулою:

$$\sum_{b}^{u} = 1P(u)Q(u), \sum_{b}^{u} P(u)Q(u) \tag{1.10}$$

де b – кількість інтервалів, Q – гістограма об'єктів, P – вікно гіпотетичного знаходження об'єкта.

Метод Лукаса-Канаде використовується досить широко в задачах оцінки руху об'єкта і оснований на роботі з оптичним потоком . Являється локальним методом обчислення оптичного потоку, тому що за допомогою ключових точок інтерактивно вираховує переміщення кожного регіону.

Даний алгоритм припускає, що:

- зсув точок на поточному і попередньому кадрах є незначним;
- зсув точок в околі деякої точки однаковий;
- інтенсивність пікселей не змінюються в час

$$I(x, y, t) - I(x + \delta x, y + \delta y, t + \delta t) = 0 \tag{1.11}$$

де $I(x, y, t)$ - функція інтенсивності пікселя з координатами (x, y) на зображенні t і $(\delta x, \delta y)$ - зміщення пікселя між послідовними кадрами t і $t + \delta t$.

Припустимо, що $D = (q1, q2....qn)$ — набір точок навколо точки P.

При малому зміщенні при лінійному розкладанні в ряд Тейлора для кожної з точок, отримаємо систему рівнянь ,яку можемо розв'язати методом найменших квадратів[1]. За допомогою функція $W(x, y)$ визначимо вагу коефіцієнтів для пікселів на зображенні:

$$\varepsilon(\upsilon) = \sum_{x,y \in D} W(x, y) \times \left[I(x, y, t) - I(x + \delta x, y + \delta y, t + \delta t \right]^2 = \\ = \sum_{x,y \in D} W(x, y) \times \left(\frac{\partial I}{\partial x} \upsilon_x + \frac{\partial I}{\partial y} \upsilon_y + \frac{\partial I}{\partial t} \right)^2 \quad (1.12)$$

Метод вважається досить ефективним ,так як є швидким і простим у використанні. Основний недолік полягає в тому ,що його можна застосовувати тільки при невеликих зсувах об'єктів між кадрами. Усунути недолік можна за допомогою пірамідального алгоритму Лукаса-Канаде.

За основу взято алгоритм слідкування, розглянутий вище. На вході процедури спостереження маємо два послідовних зображення поточний і попередній кадри відеопослідовності, а також точку спостереження з її координатами на попередньому кадрі. На виході маємо отримати координати точки спостереження на поточному кадрі або не отримати, якщо процес слідкування буде зірвано.

Пірамідальний алгоритм Лукаса-Канаде працює так :

1)створимо цикл для послідовного отримання кадрів з камери;

2) отримаємо наступний кадр відеопослідовності;

3) побудуємо з отриманого кадру наступну піраміду зображення з заданою кількістю рівнів k за допомогою Гауссового фільтру і горизонтальне та вертикальне проріджения зображення для кожного рівня піраміди;

4) перевіримо наявність координат точки (x, y), за якою потрібно слідкувати (при її відсутності - переходимо до кроку вісім, при наявності - продовжуємо далі);

5) обчислимо координати точки для *k*-ого рівня піраміди ($x = x/2k$, $y = y/2k$);

6) порахуємо цикл за рівнями піраміди ($n = k$, поки $n \geq 0$, $n = n-1$);

7) процедура спостереження (вхід: зображення *n*-ого рівня поточної і попередньої піраміди, з координатами (x, y); вихід: координати (x, y); $x = x^2$, $y = y^2$);

8) попередня піраміда = поточна піраміда;

9) переходимо до кроку два.

З виявленням кожної нової ключової точки алгоритм перераховує за допомогою афінних перетворень якість супроводження . Якщо сума різниці квадратів старого та нового регіонів невелика ,точка визначається як ключова. За допомогою пірамідального методу можна визначати об’єкт спостереження кадр за кадром при його значно більшій швидкості .

2 РОЗРОБКА ПРОГРАМИ ТРЕКІНГУ ОБ'ЄКТІВ НА ВІДЕОПОСЛІДОВНОСТІ

2.1.Розробка алгоритму роботи програми

Було розглянуто та досліджено достатньо методів та алгоритмів визначення рухомих об'єктів. Найбільш продуктивним виявилися методи пошуку ключових точок та віднімання фону. Перший метод будує модель фону і визначає бінарну маску – де 1 це об'єкт, що рухається, 0 – фон. Значення маски перераховується за допомогою Гауссової моделі,а рекурсивний алгоритм вираховує інтенсивність кожного кадру. Для обробки кожного наступного кадру оновлюється математичне сподівання і середньоквадратичне відхилення. Коефіцієнт ваги характеризує зміну пікселів між кадрами.

Далі було виконано морфологічні операції для зниження шуму і зменшення хибних визначень об'єктів детектором на зображенні. Послідовне використання операцій розмикання та замикання. Розмикання усуває надто дрібні елементи для усунення неточних визначень і пришвидшення роботи програми. Замикання використовуємо щоб уникнути нарощення контуру.

Методи розглянуті в попередніх пунктах дозволяють визначити границі об'єктів. Для цього скористаємося процедурою виділення крайових точок по методу Канні. Отримані крайові точки для газоподібних об'єктів зазвичай бувають не пов'язані між собою. Для побудови замкнутої границі з таких країв необхідно застосувати морфологічні операції. Спочатку видаляємо дуже короткі краї, так як вони в основному являють собою вплив шуму.

Нехай X - це бінарне зображення виявлених крайових точок. Значення пікселя, рівне 1, означає, що цей піксель є крайової точкою. Визначимо наступні морфологічні операції:

Затирання

$$X \varnothing V_p \equiv \left\{ x : \left| (X \setminus V_c) \cap V_c \right| > 3 \right\} \tag{2.1}$$

Дилатація

$$X \oplus V_d \equiv \{x : X \cap V_d \neq 0\} \tag{2.2}$$

Згладжування

$$X \otimes V \equiv \{x : |X \cap V_d| \geq 3\} \tag{2.3}$$

Структурні елементи задаються як :

$$V_p = \begin{bmatrix} d & d & d \\ d & 0 & d \\ d & d & d \end{bmatrix} \tag{2.4}$$

$$V_c = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.5}$$

$$V_d = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{2.6}$$

$$V_b = \begin{bmatrix} 1 & 0 & 1 \\ 0 & d & 0 \\ 1 & 0 & 1 \end{bmatrix} \tag{2.7}$$

Тут d може приймати будь-яке значення 0 або 1

Тепер комбінація морфологічних операцій для побудови границі об'єкта з різних крайових точок виглядає наступним чином:

$$((X \varnothing V_c) \oplus V_d) \otimes V_b \tag{2.8}$$

Наступний крок полягає у відновленні внутрішньої області об'єкта за отриманою границею . Якщо області слабкої інтенсивності є дуже довгими, то навіть за допомогою морфологічних операцій неможливо відновити кордон.

Це означає, що виявлений поріг буде незамкненим і може не повністю

включати в себе внутрішню область об'єкта. З цього необхідно розрізняти області, що не охоплюються отриманої границею (див. Рис. 2.1), і самостійні окремі області.

Для цього будемо шукати перетин з границею в горизонтальному і вертикальному напрямках. Тільки якщо горизонтальна і вертикальна лінії, що проходять через даний піксель, перетинають одну і ту ж границю, будемо вважати що такий піксель належить внутрішній області об'єкта [16].

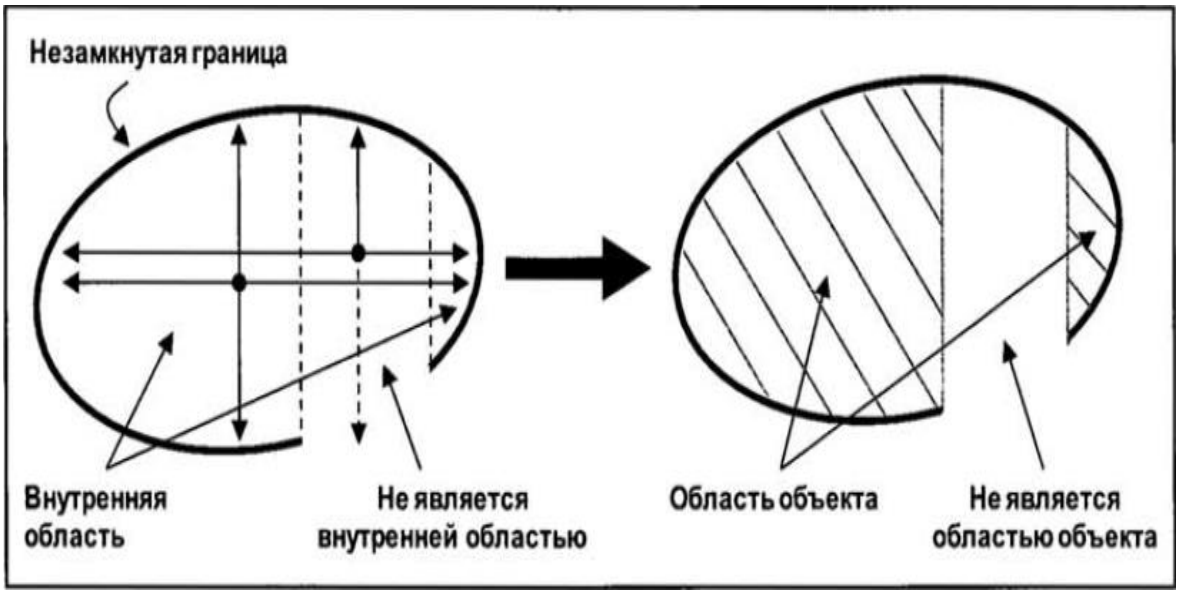


Рис.2.1 Реконструкція області об'єкта

Для цього застосовуємо детектор Канні ,робота якого складається з наступних кроків :

1) Згладжування зображення для зменшення кількості шуму.

2) Знаходження градієнтів. При максимальних значеннях градієнтів на зображені відмічаються границі.

3) Визначення зайвих точок. Тільки локальні максимуми позначаються як граничні точки

4) Повторна фільтрація. Двічі відфільтровуються границі і визначаються пороги.

5) Визначення найсильніших границь.

Застосування даного алгоритму дозволяє отримати набір контурів, які можна вважати готовими об’єктами для спостереження. Ще одним важливим

кроком для визначення об’єктів є їх порівняння між собою. Цей процес потрібен для встановлення нових об’єктів та визначення руху вже виділеного об’єкту між кадрами.

Використання алгоритму для призначення дозволяє швидко визначати нові об’єкти та перераховувати рух помічених об’єктів між кадрами. Основою алгоритму являється побудова матриці коштів, яка являє собою Евклідові відстані між об’єктами поточного і попереднього кадрів. Кроки алгоритму:

1) Побудова матриці.

2) Знаходження мінімального елементу кожного ряду і віднімання його від свого ряду.

3)Знаходження мінімального елементу кожного стовбцю і аналогічне його віднімання від стовбців.

4) Визначення найбільшого сполучення пар.

2.2 Фільтр Калмана

За допомогою алгоритму отримуємо координати визначених та нових об’єктів на відео. Заключним кроком роботи програмного продукту трекінгу рухомих об’єктів являється передбачення траєкторії руху об’єкту і коригування вимірів використовуючи фільтр Калмана.

Фільтр Калмана часто застосовується для відстеження об’єктів на зображеннях з шумом .[3] Щоб відстежити одиночний об’єкт використовують оцінку стану лінійної системи розділеної за Гаусом. Наступний етап екстраполяції використовує модель станів щоб передбачити траєкторію і нові стани рухомого об’єкту.

Модель фільтру Калмана записується за формулою:

$$x_k = F_k x_{k-1} + B_k u_k + w_k, \tag{2.9}$$

де x_k – стан системи у момент часу k, F_k – модель переходу стану, що застосовується до попереднього стану x_{k-1}, B_k – модель впливу керування,

що застосовується до вектору керування u_k , w_k – шум процесу з нульовим середнім значенням коваріації Q_k .

$$z_k = H_k x_k + v_k , \tag{2.10}$$

де H_k модель спостереження, яка відображає простір справжнього стану у просторі спостереження , і v_k шум спостереження, що вважається, ґаусовим білим шумом з нульовим середнім значенням і коваріацією R_k .

Формула передбачення:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k , \tag{2.11}$$

Де $\hat{x}_{k|k-1}$ – передбачення оцінки стану, B_k – модель впливу керування, що застосовується до вектору керування u_k

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \tag{2.12}$$

де $P_{k|k-1}$ – коваріація передбаченої оцінки, F_k – модель переходу стану, Q_k – матриця коваріації шуму.

Уточнення:

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \tag{2.13}$$

де – K_k оптимальний коефіцієнт передачі Калмана, H_k модель спостереження, R_k – коваріація шуму спостереження.

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - H_k \hat{x}_{k|k-1}) \tag{2.14}$$

де $\hat{x}_{k|k}$ – оновлена оцінка стану, $\hat{x}_{k|k-1}$ – передбачена оцінка стану.

$$P_{k|k} = P_{k|k-1} - P_{k|k-1} K_k H_k \tag{2.15}$$

Основним недоліком фільтру Калмана являється те ,що змінні повинні бути розподілені нормально. Інакше від роботи фільтру буде отриманий слабкий результат. Але за допомогою фільтру часток цього недоліку можна

уникнути. Густота ймовірностей за час t представлена набором зразків $\{s(n)t : n = 1,....,N\}$ часток з вагою $\pi(n)$. За допомогою ваги визначаються важливі зразки.

Алгоритм розроблено за умов стаціонарного положення відеокамери і частоті кадрів до 60 Гц. Розміри визначених об'єктів не менші за $10x10$ пікселів. Рух об'єктів прямолінійної чи криволінійної траєкторії,або періодичний. Швидкість об'єктів не більший за 100 пікселів за кадр.

2.3 Висновки

У розділі було наведено деталі алгоритму та архітектури програми. Розробка програмного продукту трекінгу рухомих об'єктів відбувається відповідно до спроектованої архітектури програми. Наведені аргументи підтверджуючі зручність та функціональність обраних алгоритмів і методів. Програмний продукт написано у пакеті Matlab. Для обробки зображень використано бібліотеку Computer Vision System Toolbox.

3.ТЕСТУВАННЯ ПРОГРАМИ

3.1 Оцінка роботи програми

Розроблений програмний продукт буде оцінюватися точністю розпізнавання програмою рухомих об’єктів. Так як використання програми передбачає нерухому положення камери, деякі об’єкти які являються шумом ,можуть визначатися програмою. Наприклад, коливання гілок дерев або снігопад ,можуть бути визначенні як рухомі об’єкти.[20] Також на роботу програми може впливати якість знятого відео, через розмитість зображення помилково можуть виділятися об’єкти.

Варіанти покращення роботи програми:

Головним недоліком являється сприйняття програмою шуму за об’єкти руху. Це зв’язане з розділенням зображення на передній і задній плани ,а також використання методу Гауса. Тому для покращення визначення можна використати наступні способи:

- Методи оптичного потоку визначають вплив освітлення на зображення.
- Використання шаблонів. Кожен визначений рухомий об’єкт порівнюється із шаблонами для його розпізнавання. Як шаблони можуть використовувати силуети людини ,машин і інших об’єктів.
- Використання камер більшої якості .
- Примінення штучних нейронних мереж. Спосіб є одним з найпопулярніших ,але разом з цим і найскладнішим . Для ефективної роботи потрібне багаторазове тренування зі значною кількістю даних.
- Одночасне використання декількох камер,для одночасної зйомки об’єкта з різних сторін. Допомагає уникнути оклюзії об’єктів іншими .

Кожен метод збільшує навантаження на оперативну систему котра використовує програму. Це значно сповільнює швидкість роботи програми ,проте збільшує якість трекінгу.

3.2 Система « Object Tracking »

3.2.1. Робота алгоритму

Зовнішній вид основної робочої форми програми « Object Tracking » можна побачити на Рис.3.1

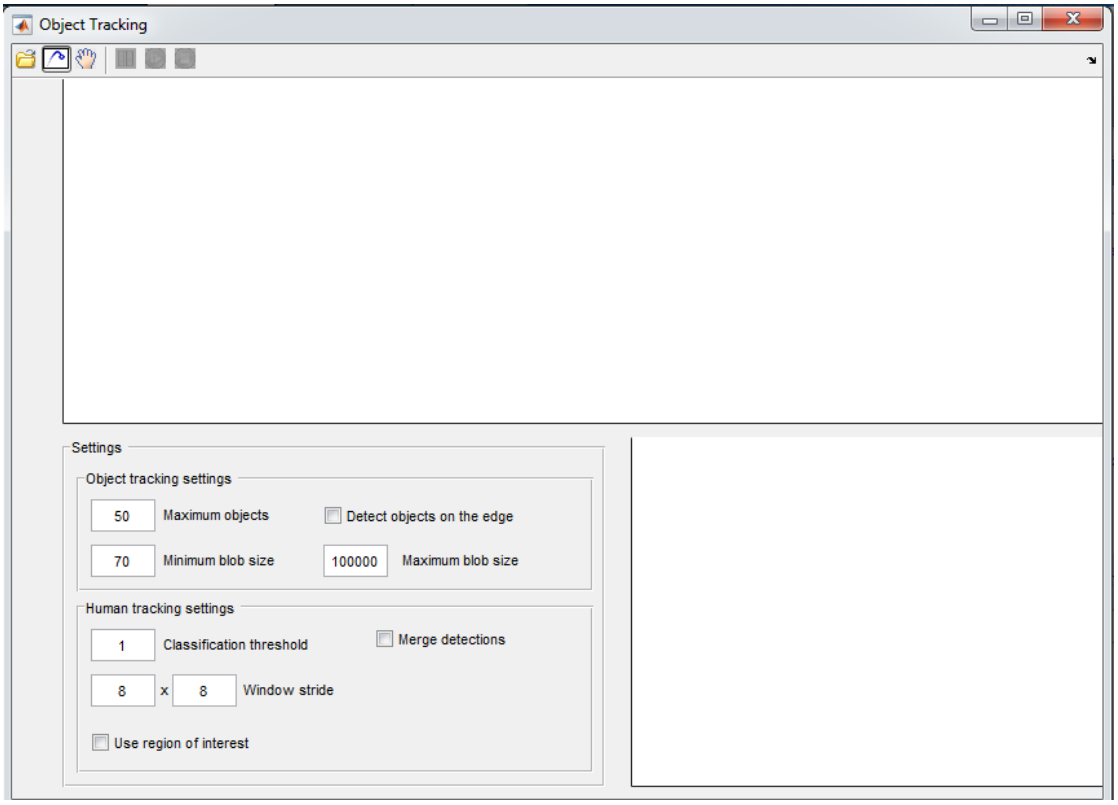


Рис.3.1 Зовнішній вид основної робочої форми програми

Першим кроком роботи з програмою є вибір і завантаження відеофайлу.

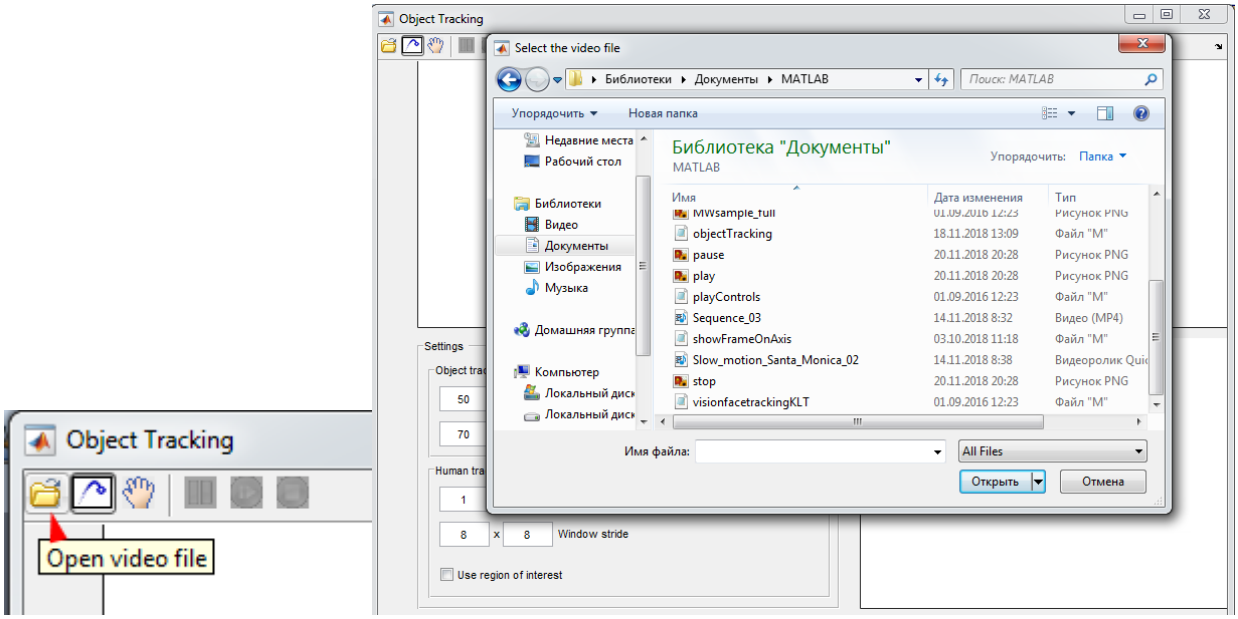


Рис.3.2 Вибір (а) і завантаження (б) відеофайлу

Вибираємо кнопку «Open video file» Рис.3.2 а, яка відкриває робочий стіл користувача Рис.3.2 б ,де він може обрати потрібний відео файл з комп'ютеру. Система дозволяє завантажувати відеофайли, збережені у будь-якому стандартному графічному форматі.

Після завантаження відео у програму ,користувачу стають доступні клавіші керування відео послідовністю : «play» , «pause» , «stop».(Рис 3.3)

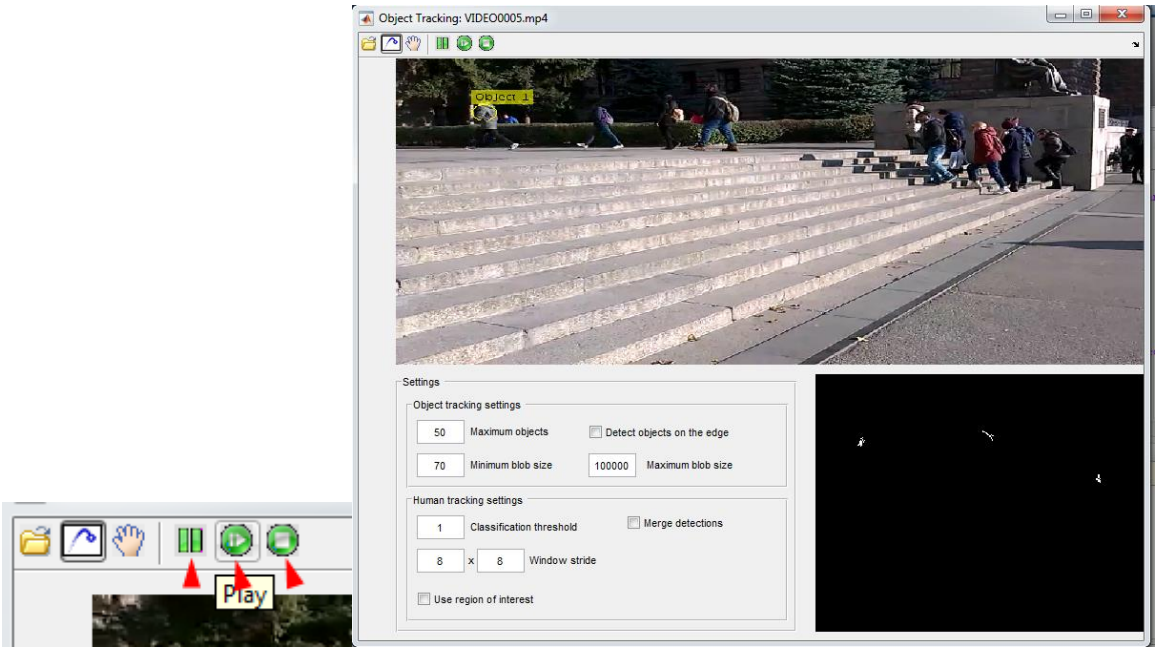


Рис.3.3 Клавіші керування відео

Також на робочому полі програми доступні клавіші «Detect Object» та «Detect Humans» – кнопки активуючі детектори визначення об'єктів та людей на відео послідовності відповідно(Рис.3.4).

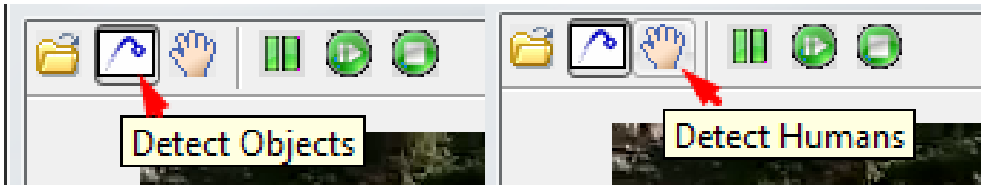


Рис.3.4 Функціональні детектори

У вікно програми виводиться оригінальне зображення, а також бінарна маска (справа нижнє). При використані «Detect Object» визначені рухомі об'єкти нумеруються і виділяються ярликом у вигляді кола. «Detect Humans» визначає точність визначення та виділяє об'єкти прямокутним ярликом.

Робота детектору «Detect Object» (Рис 3.5).

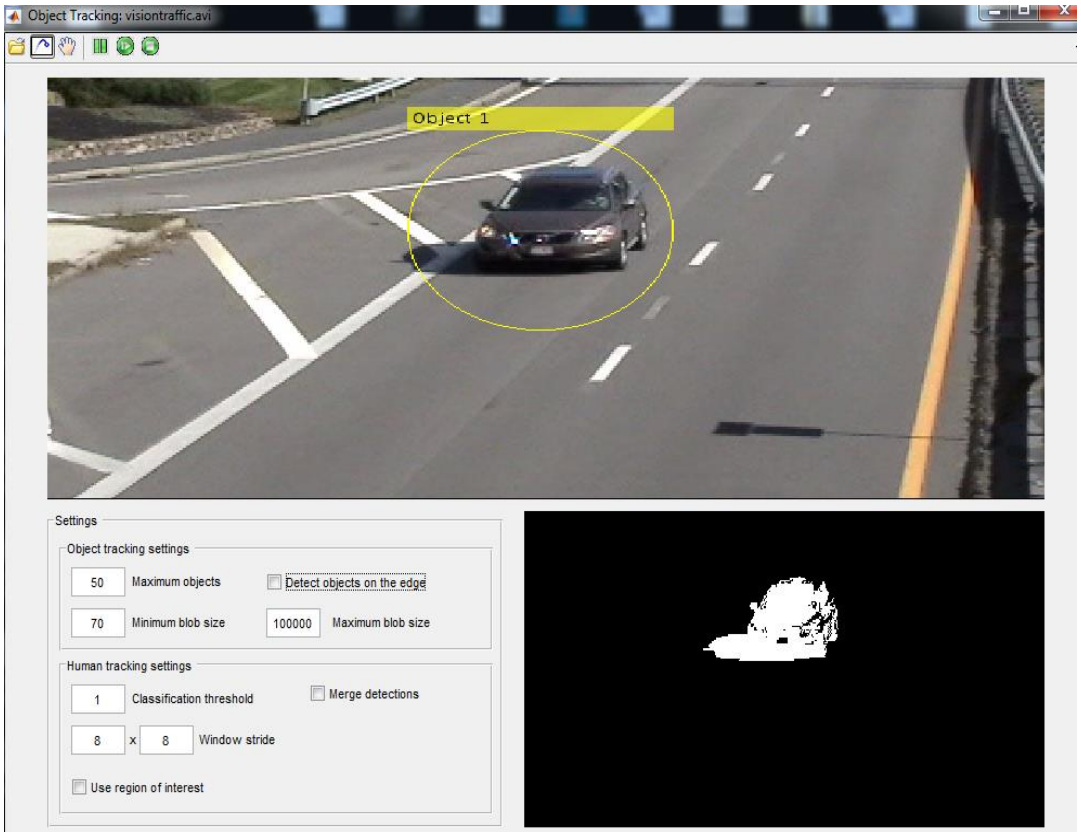


Рис.3.5 Приклад роботи «Detect Object».

Робота детектору «Detect Humans».(Рис 3.6)

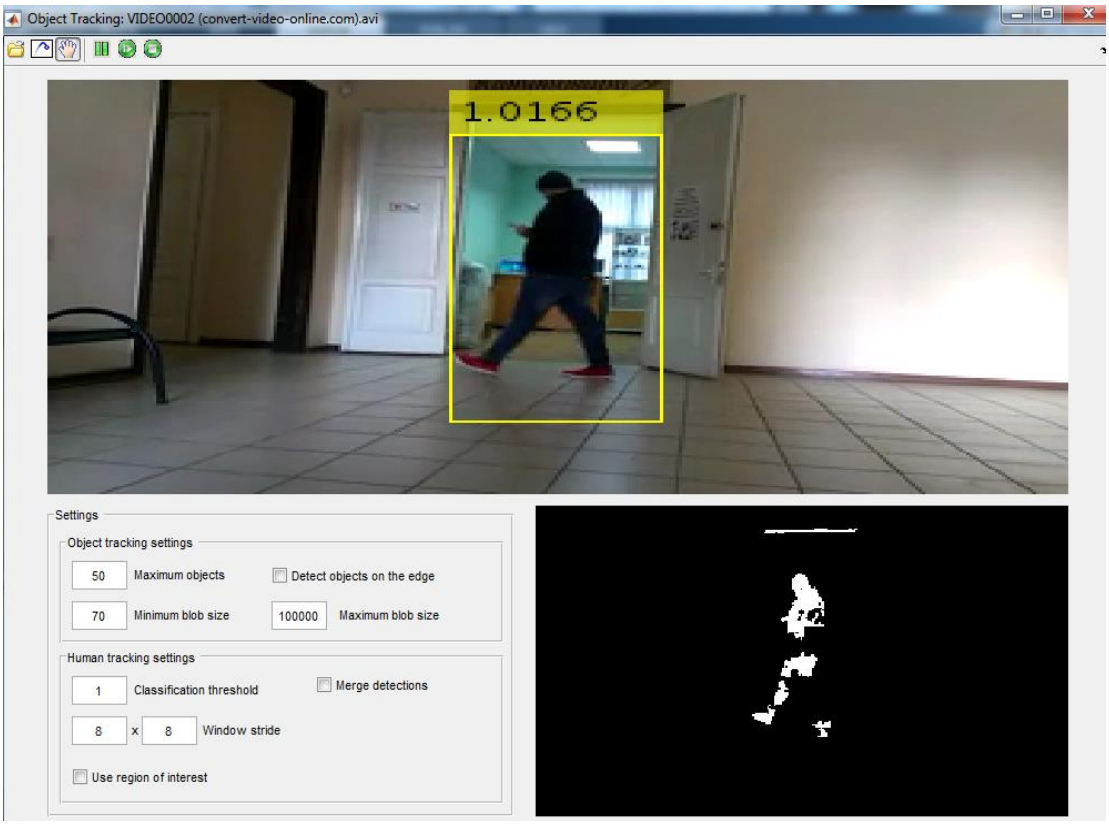


Рис.3.6 Приклад роботи «Detect Humans».(Рис.3.7)

Програма передбачає використання обох детекторів одночасно.

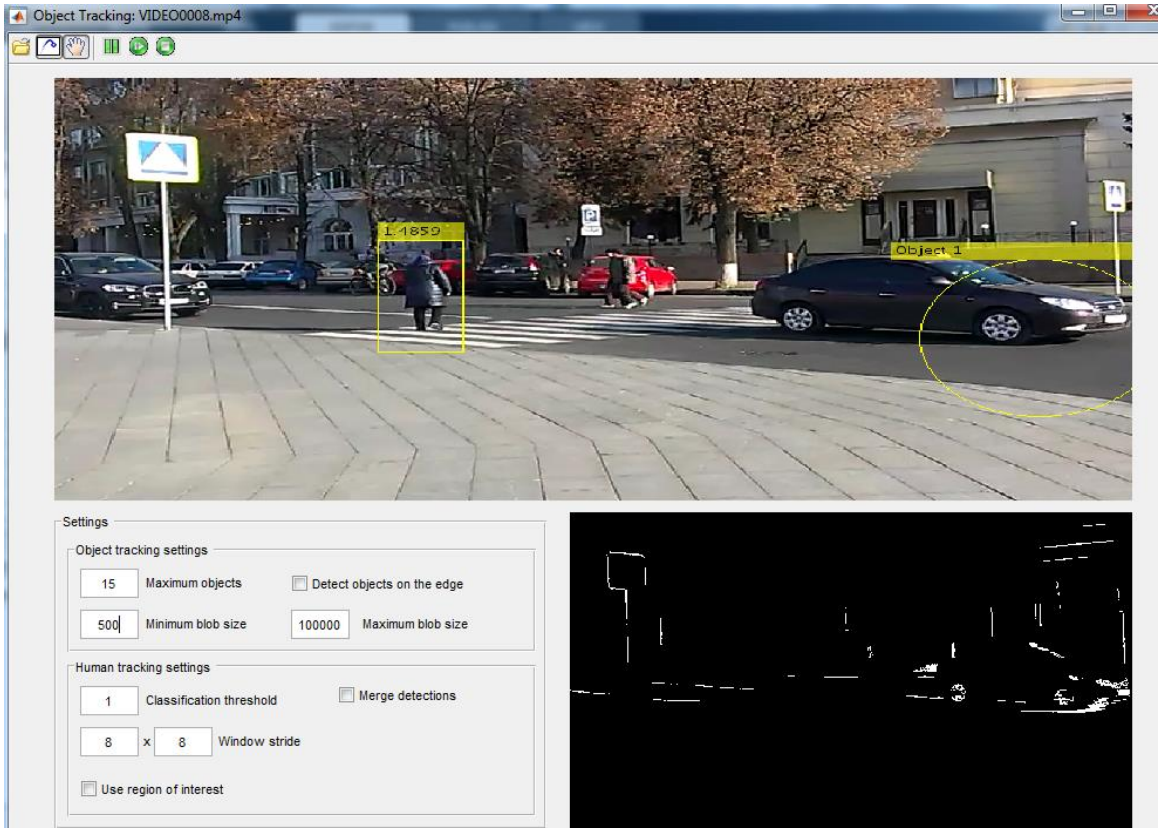


Рис.3.7 Використання двох детекторів одночасно.

Відео послідовність має об’єкти які рухаються за довільною траєкторією. Програма точно визначає та відслідковує рухомі об’єкти, поки вони знаходяться у полі зору камери. Незважаючи на те, що відео мають різний розмір та рівень шуму ,програмний продукт повністю задовольняє поставлені перед ним цілі.

3.2.2 Детектор «Detect Object» визначає об’єкти що знаходяться у відео послідовності

Для покращення роботи програми було створено поле керування програмою (Рис 3.8).

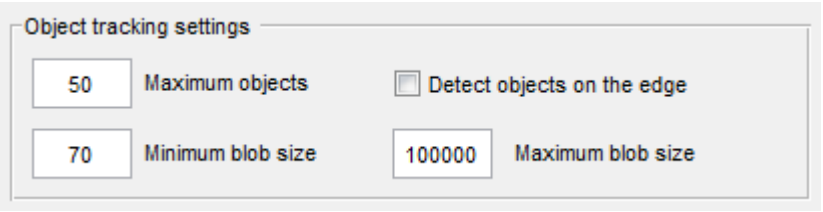


Рис 3.8 Панель керування «Detect Object».

«Maximum object»

Перша панель вибір максимальної кількості об’єктів на відео послідовності. Приклад роботи програми з визначенням 5 об’єктів (Рис 3.9)

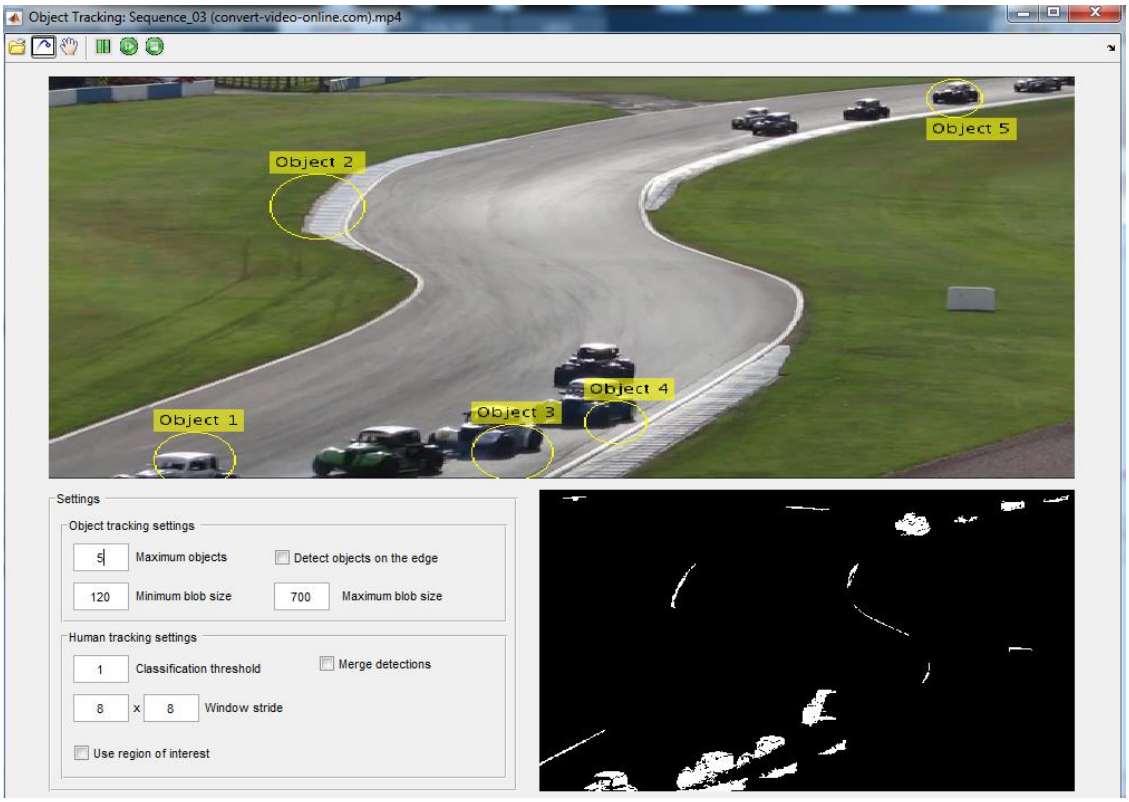


Рис 3.9 Визначення п’яти об’єктів за допомогою «Maximum object»

Приклад роботи програми з визначенням 12 обєктів (Рис 3.10).

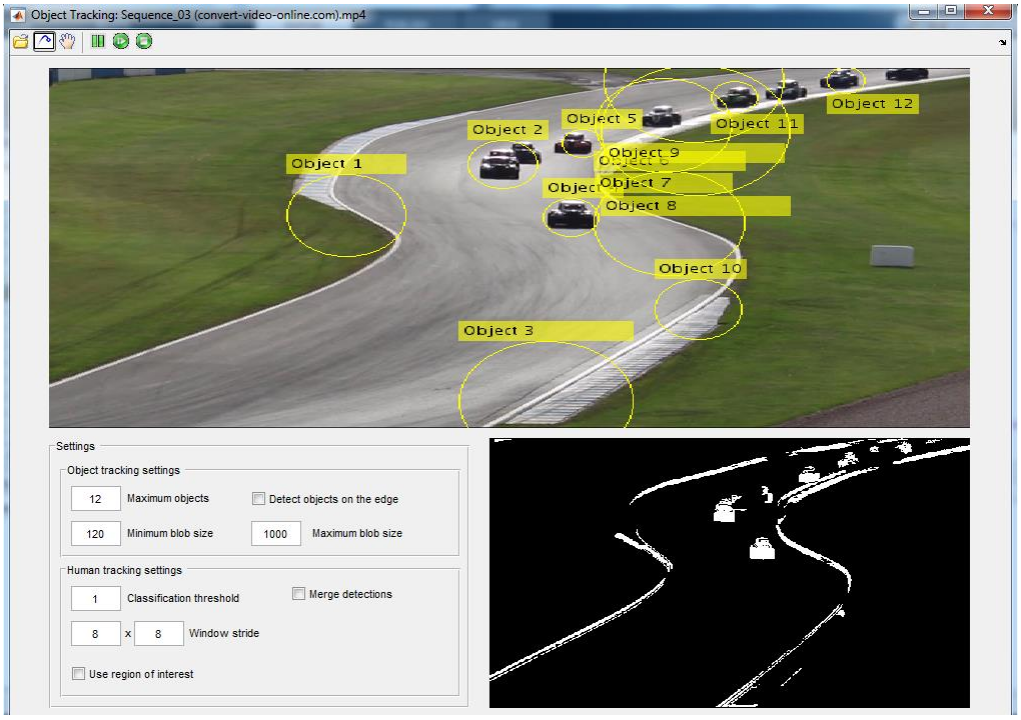


Рис 3.10 Визначення дванадцяти об’єктів за допомогою «Maximum object» «Detect object on the edge»

Друга визначення об’єкту коли він не повністю з’являється на зображенні



Рис 3.11 Використання «Detect object on the edge» «Minimum blob size» і «Maximum blob size»

Також можна задавати максимальний і мінімальний розмір об’єктів спостереження.

Програма визначає об’єкти розміром від 70 до 5000 пікселів.

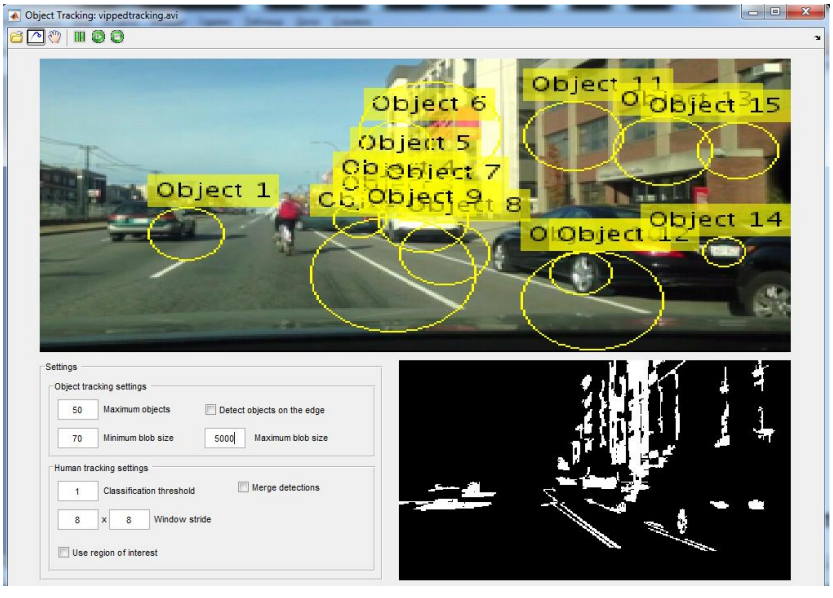


Рис 3.12 Використання «Minimum blob size» і «Maximum blob size»

На тому ж відео , визначення об’єктів розміром від 150 до 4000 пікселів.

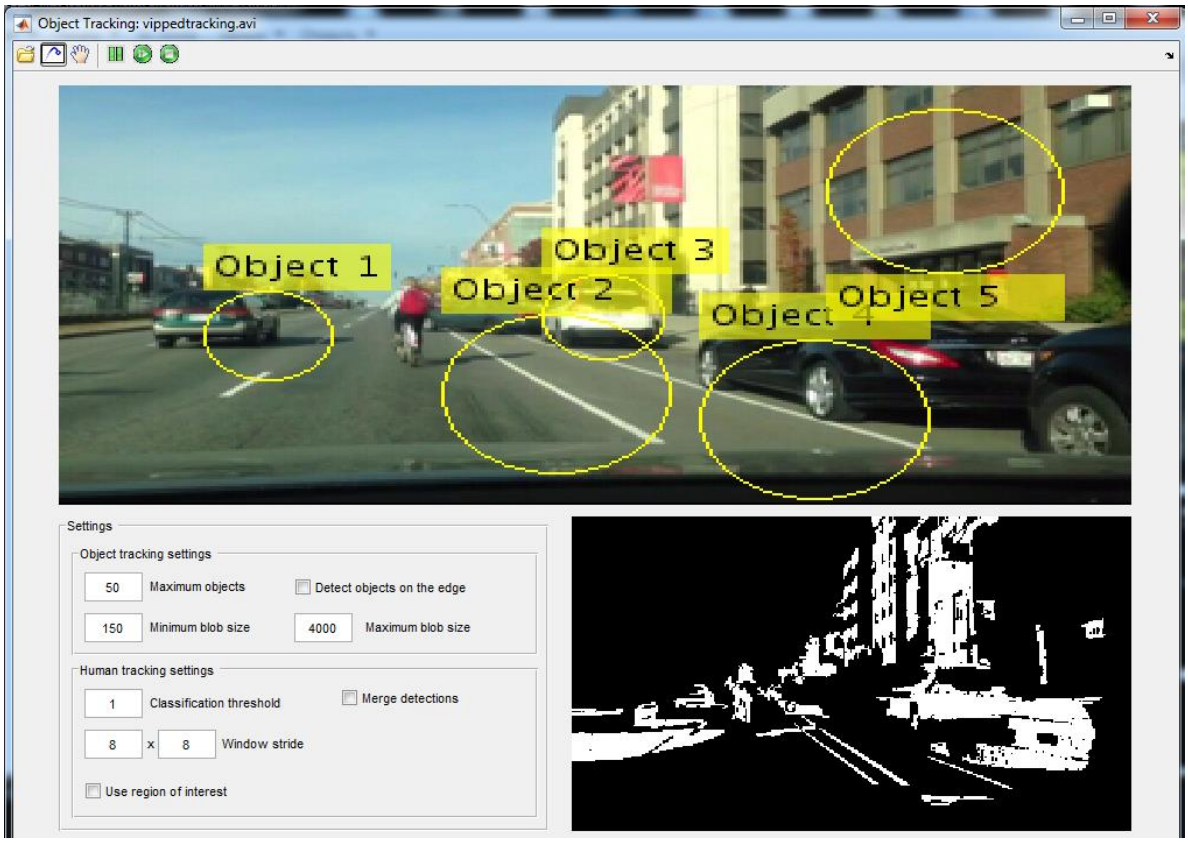


Рис 3.13 Використання «Minimum blob size» і «Maximum blob size»

3.2.3 Детектор «Detect Humans» визначає людей ,які рухаються у полі зору камери. Панель керування представлена на Рис 3.14.

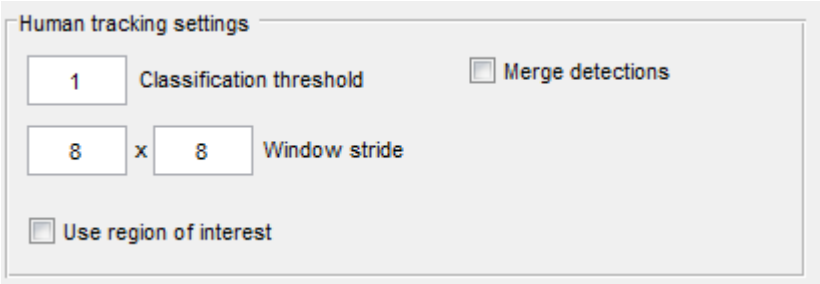


Рис 3.14 Панель керування «Detect Humans»

«Classification threshold»

Поріг класифікації людей ,використовується щоб контролювати розпізнавання окремих регіонів зображення під час багатозначного детектування. Поріг визначає виділяти об’єкт як особу чи ні. При помилковому визначені можна збільшити порогове значення ,тим самим

збільшивши жорсткість вимог для трекінгу. Цю властивість можна перебудувати. Діапазон значень порогу найчастіше коливається від 0 до 4.

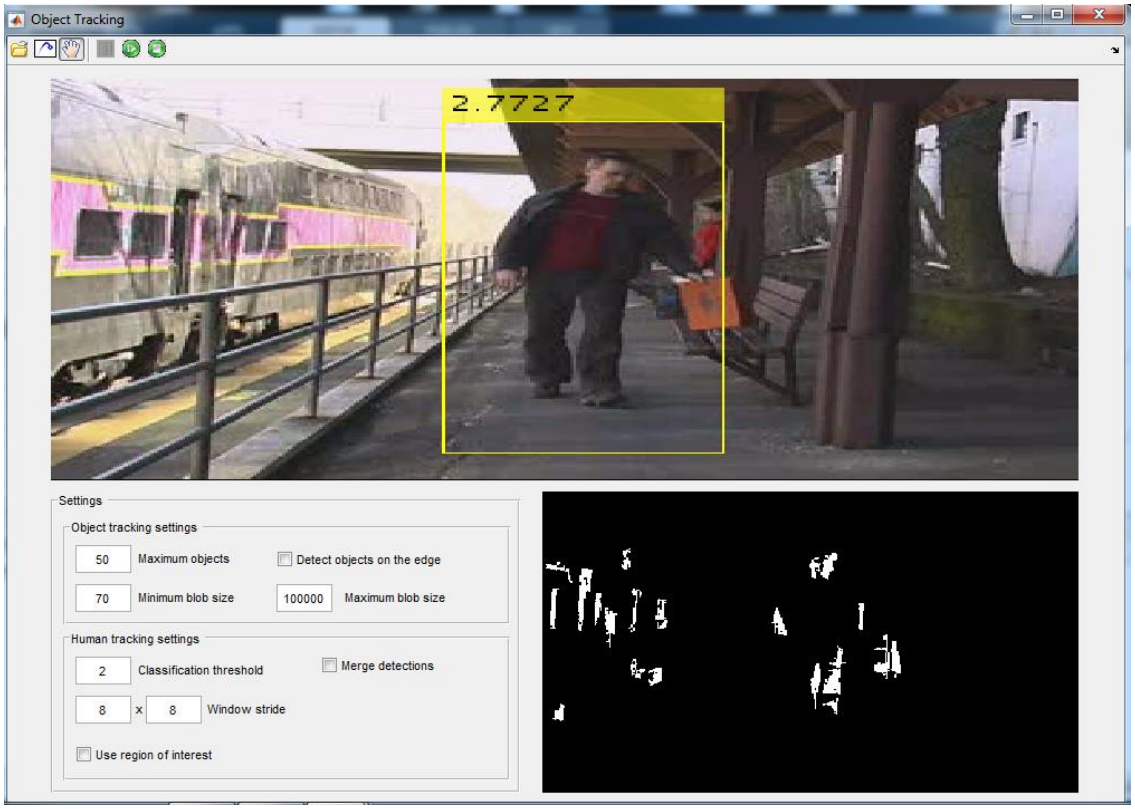


Рис 3.15 Використання «Classification threshold»

"Merge Detections"

Властивість яка контролює злиття подібних визначень. При використання цієї властивості об’єднуються усі визначення одного рухомого об’єкта і виводиться середній параметр. Якщо ж не використовувати злиття то на екрані будуть відображатися усі визначення рухів визначених об’єктів .

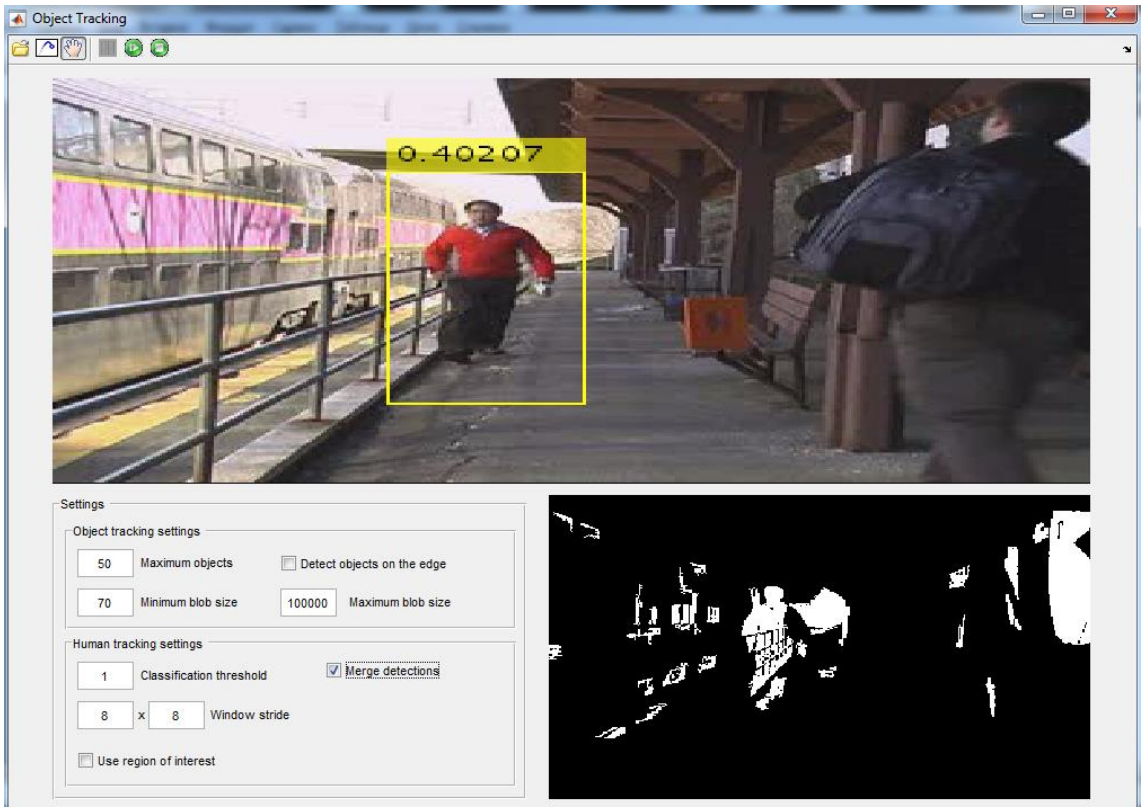


Рис 3.16 Використання "Merge Detections"

Якщо ж не використовувати злиття то на екрані будуть відображатися усі визначення рухів визначених об’єктів .

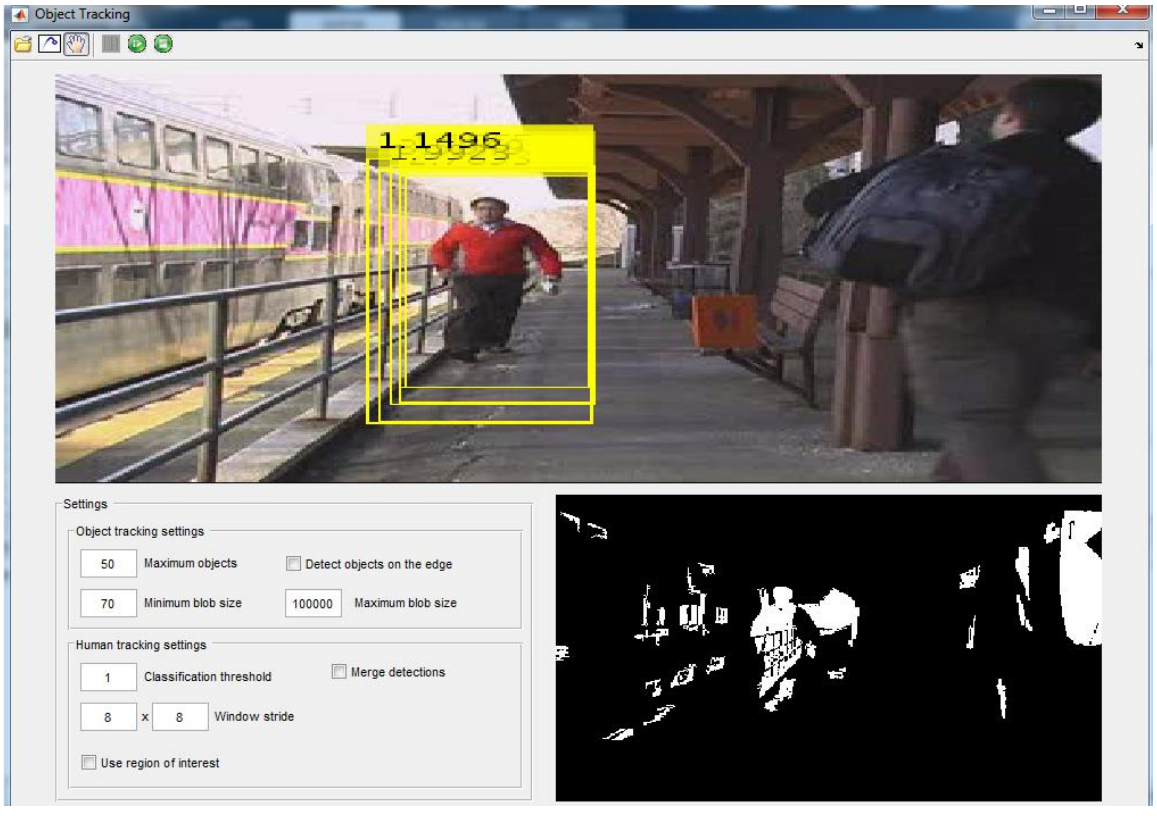


Рис 3.17 Без використання "Merge Detections"

"WindowStride"

Властивість використовує кроки ярликів областей визначення для того ,щоб скопіювати область виявлення всього зображення. Зменшення масштабу області підвищує точність визначення ,але знижає швидкість обробки відео.

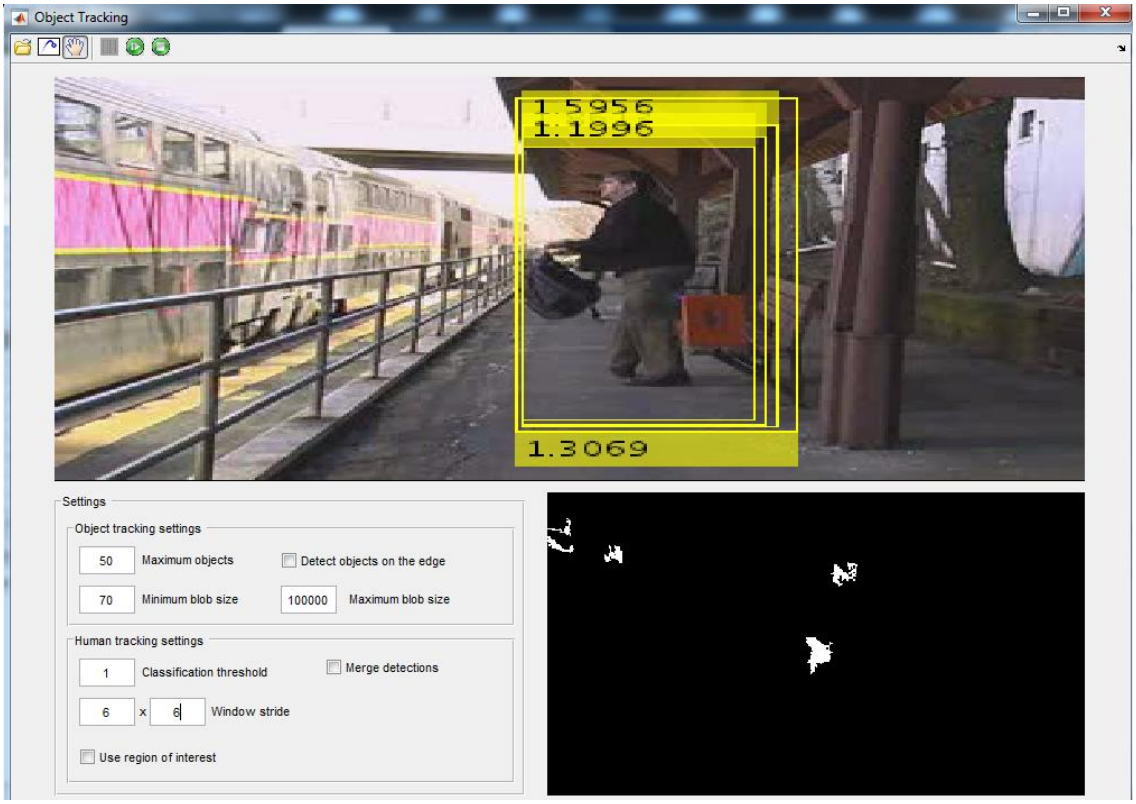


Рис 3.18 Використання "WindowStride"

«Use region of interest»

Визначає рухомі об’єкти у визначеній прямокутній області інтересу на зображенні.

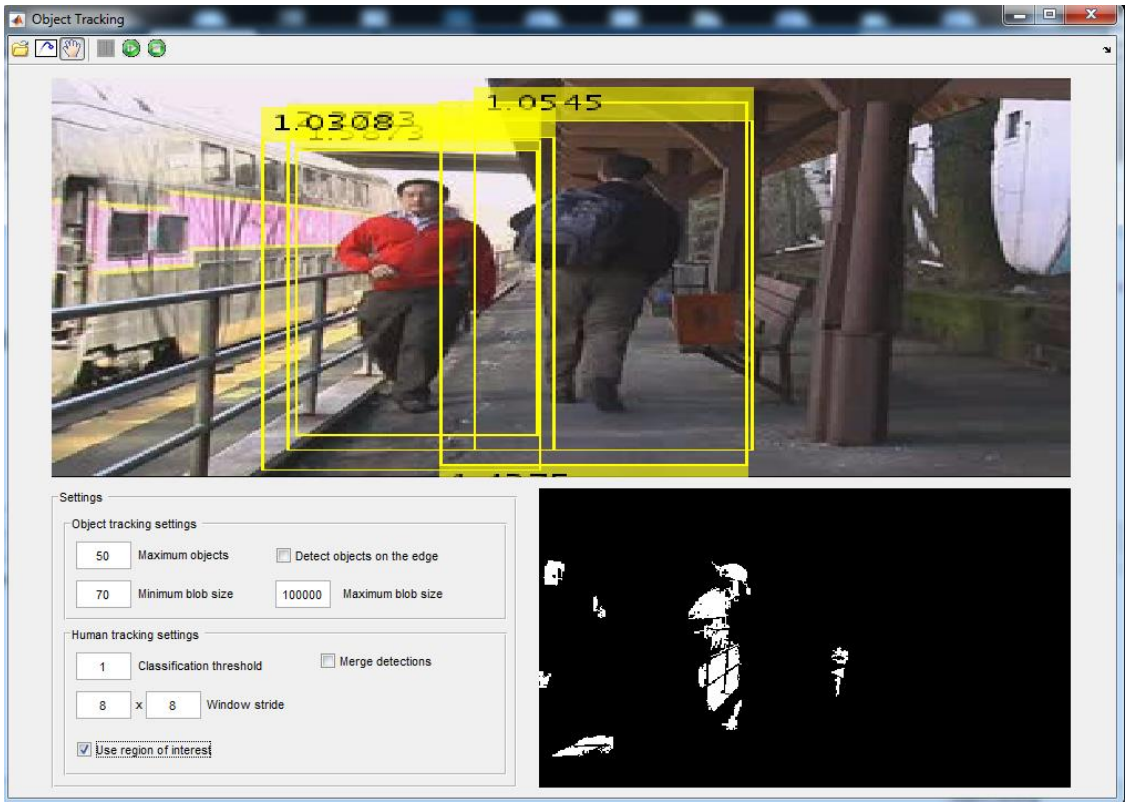


Рис 3.19 Використання «Use region of interest»

3.3 Висновки

Розроблений програмний продукт повністю відповідає поставленим задачам. Програма проводить трекінг рухомих об’єктів та осіб незалежно від складності траєкторії руху ,швидкості та розміру. Розміри відео впливають на час роботи програми ,але не на ефективність. Кількість шуму може привести до ускладнення роботи програми і хибних визначень. Проте було запропоновано деякі рішення подолання рівня шуму на зображенні.

Програма успішно долає проблему оклюзії об’єктів. Фільтр Калмана успішно прогнозує траєкторію руху об’єктів з моменту їх визначення у кадрі.

Програмний продукт справляється із задачею виділення і супроводження об’єктів у реальному часі.

ВИСНОВКИ

В результаті виконання дипломної роботи була розроблена програма трекінгу рухомих об’єктів у відео послідовності . Дана програма може використовуватися в навігації машин ,системах охорони та безпеки, стеження за підозрілою активністю.

За останні роки розробки в області трекінгу об’єктів досягли значного прогресу і привели до створення значної кількості технологій та методів визначення та відстеження рухомих об’єктів. Головною проблемою у даному напрямку являється розробка алгоритму ,який забезпечував би результат незалежно від якості знятого відео. Адже відео з дешевих камер частіше за все мають високий рівень шуму,стиснуті та неструктуровані.

Розглянуто особливості роботи алгоритмів трекінгу об’єктів ,проведений аналіз існуючих методів та технологій визначення та супроводження об’єктів.

Розроблено алгоритм і архітектуру програмного продукту у пакеті програм Matlab. Для створення алгоритму було використано наступні моделі та технології комп’ютерного зору : Гаусова модель змішування , детектор Кані, фільтр Калмана ,модель заднього фону.

Проведено тестування програмного продукту на різних відео зображеннях. Виявлено переваги та недоліки роботи програми. Найефективніше алгоритм працює при стаціонарному положенні камери і високій якості зйомки. Подано приклади покращення роботи програми та усунення похибок визначення та супроводження рухомих об’єктів.

Розвиток технологій в області комп’ютерного зору знаходить широке практичнее застосування в багатьох сферах. Визначення і відстеження рухомих об’єктів для подальшого аналізу і роботи з отриманими даними відповідно до поставлених цілей їх використання. .

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Baker S., Matthews I. Lucas—Kanade 20 Years On: A Unifying Framework // International Journal of Computer Vision (IJCV). — 2004. — V. 56, №3. P. 221—255.

2. Cavallaro A. Video tracking/ A.,E. Maggio, Cavallaro A. – Chichester, West Sussex, UK: Wiley, 2010.

3. Challa S. Fundamentals of object tracking/Chall S., Mark R. Morelande, Darko Mušicki // Cambridge UK: Cambridge University Press, 2011.

4.C. Harris, M. Stephens. A Combined Co rner and Edge Detector. In: Proc. Alve y Vision Conference, Manchester, p. 147-151,1988.

5. D. Lowe. Distinctive Image Featu res from Scale-invarian t Keypoints. In: International Journal o f Comp uter Vision, v. 60, p. 91-110,2004.

6. D. Zhang , G. Lu. Segmentation of Moving O bjects in Image Sequence: A Review. In: Circuits, Systems and Signal Processing, v. 20, p. 143-183,2001.

7.Deng, L. (2014). Automatic Speech Recognition- A Deep Learning Approach (pp. 6-8). Springer.

8.Gaussian Mixture Model. Brilliant.org. Retrieved , November 24, 2018, from https://brilliant.org/wiki/gaussian-mixture-model/

9. H. Moravec. Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover. Technical Report CM U-R I-TR-3 , Carnegie-Mello n University, Robotics In stitute, 170 p, 1980.

10. K. Mikolajczyk, C. Schmid. A Performance Evaluatio n of Local Descriptors. In: Proc. IE EE Computer Society Conference on Computer Vision and Pattern Recognition, Madison, p. 1305-1312,2003.

11. Martínez-Martín E. Robust motion detection in real-life scenarios./ A. Pobil, E. Martínez-Martín// – London: Springer, 2012.

12.M. H. Miroslav Trajkovii. Fast corner detection, 1998.

13. R Salembier, J. Serra. Fla t Zones Filtering, Connect ed Op era tor s, and Filters by Recon stru ctio n. In: IE EE Transactions on Image Processing, v.4, p. 1153-1160,1995.

14.Santosh, D. (2013). Tracking Multiple Moving Objects Using Gaussian Mixture Model. International Journal of Soft Computing and Engineering, 3-2, 114-119.

15.T. Shi. Good Features to Track, 1994

16. W.-H. Kim, C. Brislawn. Plume Detection and Tracking in Video Data. FY99 Progress Report LAUR-00-15, Los Alamos National Laboratory, 1999.

17. Y. Ke, R. Sukthankar. PCA-SIFT: A More Distinc tive Represe nta tio n for Local Image Descriptors. In: Proc. IE EE Com puter Society Conference on Computer Vision and Pattern Rec ognition, Washington, p. 506-513,2004.

18. Yilmaz A. Object tracking: A survey /A. Yilmaz, O. Javed and M. Shah// ACM Comput. Surv, 2006, vol. 38, no. 4,

19. Борисенко Д. И. Методы поиска угловых особенностей на изображениях // Молодой ученый, 2011. — №5. Т.1. — С. 120-123.

20. Цифровая обработка видеоизображений / А. А.Лукьяница, А.Г.Шишкин. – М.: «Ай-Эс-Эс Пресс», 2009.

Додаток А

```
function objectTracking
f=figure('units','pixels','position',[40,40,1000,660],'menubar','none','Name','Object
Tracking','NumberTitle','off');
tb=uitoolbar;
isPaused=false;
isStopped=false;
isDetecting=false;
blobAnalyzer=0;
peopleDetector=0;
img = double(imread(fullfile(matlabroot,...
        'toolbox','matlab','icons','file_open.png')));
img(img==0)=60000;
img=img/65536;
btnOpen=uipushtool(tb,'TooltipString','Open                                                    video
file','CData',img,'clickedcallback',@LoadVideo);
img = double(imread(fullfile(matlabroot,...
        'toolbox','matlab','icons','plotpicker-comet.png')));
% img(img==0)=60000;
 img=img/256;
btnObject=uitoggletool(tb,'TooltipString','Detect Objects','CData',img,'state','on');
img = double(imread(fullfile(matlabroot,...
        'toolbox','matlab','icons','tool_hand.png')));
img(img==0)=60000;
img=img/65536;
btnHuman=uitoggletool(tb,'TooltipString','Detect Humans','CData',img,'state','off');
img = double(imresize(imread('pause.png'),[16 16]));
img(img==0)=220;
img=img/256;
btnPause=uipushtool('TooltipString','Pause','clickedcallback',@pauseVideo,'Separator','on','CDat
a',img,'enable','off');
img = double(imresize(imread('play.png'),[16 16]));
img(img==0)=220;
img=img/256;
```

```
btnPlay=uipushtool('TooltipString','Play','clickedcallback',@playVideo,'CData',img,'enable','off');
img = double(imresize(imread('stop.png'),[16 16]));
img(img==0)=220;
img=img/256;
btnStop=uipushtool('TooltipString','Stop','clickedcallback',@StopVideo,'CData',img,'enable','off')
vPlayer=axes('units','pixels','position',[40,290,920,360],'xtick',[],'ytick',[]);
mPlayer=axes('units','pixels','position',[480 10 480,270],'xtick',[],'ytick',[]);
pan1=uipanel('units','pixels','position',[40,10,420,270],'Title','Settings');
pan2=uipanel('parent',pan1,'units','pixels','position',[10 150, 400,95],'Title','Object tracking
settings');
pan3=uipanel('parent',pan1,'units','pixels','position',[10 10, 400,135],'Title','Human tracking
settings');
MCount=uicontrol('style','edit','parent',pan2,'position',[10,45,50,25],'string','50','callback',@MCo
unt_change);
tx1=uicontrol('style','text','parent',pan2,'position',[65,40,130,25],'string','Maximum
objects','horizontalalignment','left');
DetectEdgeObjects=uicontrol('style','checkbox','parent',pan2,'position',[190,45,200,25],'string','D
etect objects on the edge','Value',0,'callback',@MCount_change);
MinBlob=uicontrol('style','edit','parent',pan2,'position',[10,10,50,25],'string','70');
tx2=uicontrol('style','text','parent',pan2,'position',[65,5,130,25],'string','Minimum blob
size','horizontalalignment','left','callback',@MCount_change);
MaxBlob=uicontrol('style','edit','parent',pan2,'position',[190,10,50,25],'string','100000');
tx3=uicontrol('style','text','parent',pan2,'position',[250,5,130,25],'string','Maximum blob
size','horizontalalignment','left','callback',@MCount_change);
ClassThresh=uicontrol('style','edit','parent',pan3,'position',[10,85,50,25],'string','1','callback',@PD
_CB);
tx4=uicontrol('style','text','parent',pan3,'position',[65,80,130,25],'string','Classification
threshold','horizontalalignment','left');
WSx=uicontrol('style','edit','parent',pan3,'position',[10,50,50,25],'string','8','callback',@PD_CB);
WSy=uicontrol('style','edit','parent',pan3,'position',[73,50,50,25],'string','8','callback',@PD_CB);
tx5=uicontrol('style','text','parent',pan3,'position',[63,45,10,25],'string','x','horizontalalignment','lef
t');
tx6=uicontrol('style','text','parent',pan3,'position',[127,45,100,25],'string','Window
stride','horizontalalignment','left');
```

```
MDetections=uicontrol('style','checkbox','parent',pan3,'position',[230,90,200,25],'string','Merge detections','Value',0,'callback',@PD_CB1);
USEROI=uicontrol('style','checkbox','parent',pan3,'position',[10,10,200,25],'string','Use region of interest','Value',0,'callback',@PD_CB1);
%    video = vision.VideoFileReader('viptrain.avi');
%  video = vision.VideoFileReader('visiontraffic.avi');
%   video = vision.VideoFileReader('singleball.avi');
% video = vision.VideoFileReader('shuttle.avi');
video=[];
   function PD_CB(~,~)
      if peopleDetector~=0
set(peopleDetector,'ClassificationThreshold',str2num(ClassThresh.String),'WindowStride',[str2num(WSx.String) str2num(WSy.String)])
      end
   end
   function PD_CB1(~,~)
      if peopleDetector~=0
         release(peopleDetector)
         peopleDetector =
vision.PeopleDetector('ClassificationThreshold',str2num(ClassThresh.String),'WindowStride',[str2num(WSx.String) str2num(WSy.String)], ...
   'MergeDetections',MDetections.Value,'UseROI',USEROI.Value);
      end
   end
   function MCount_change(~,~)
      if (blobAnalyzer~=0)
         release(blobAnalyzer)
         blobAnalyzer = vision.BlobAnalysis('AreaOutputPort', true, 'MinimumBlobArea', str2num(MinBlob.String),'MaximumBlobArea',str2num(MaxBlob.String), 'CentroidOutputPort', true,'MaximumCount',str2num(MCount.String), ...
   'ExcludeBorderBlobs',~(DetectEdgeObjects.Value));
      end
   end
function LoadVideo(~,~)
```

```
   [FileName,PathName] = uigetfile({'*.avi';'*.mpeg';'*.wmv';'*.asf';'*.asx'},'Select the video file');
   if FileName~=0
      video = vision.VideoFileReader(strcat(PathName,FileName));
      set(btnPause,'enable','on');
      set(btnPlay,'enable','on');
      set(btnStop,'enable','on');
      set(f,'Name',['Object Tracking: ',FileName])
   end
end
function playVideo(~,~)
isStopped=false;
reset(video)
fgDetector = vision.ForegroundDetector('NumTrainingFrames', 10, 'InitialVariance', 0.05);
blobAnalyzer = vision.BlobAnalysis('AreaOutputPort', true, 'MinimumBlobArea', str2num(MinBlob.String),'MaximumBlobArea',str2num(MaxBlob.String), 'CentroidOutputPort', true,'MaximumCount',str2num(MCount.String), ...
   'ExcludeBorderBlobs',~(DetectEdgeObjects.Value));
peopleDetector = vision.PeopleDetector('ClassificationThreshold',str2num(ClassThresh.String),'WindowStride',[str2num(WSx.String) str2num(WSy.String)], ...
   'MergeDetections',MDetections.Value,'UseROI',USEROI.Value);
% blobAn2 = vision.BlobAnalysis('BoundingBoxOutputPort', true,'MinimumBlobArea', 70);
while ~isStopped
   while  ~isDone(video) && ~isPaused && ~isStopped
      frame = step(video);
      I = rgb2gray(frame);
      fgMask = step(fgDetector,I);
      fgMask = bwareaopen(fgMask,45);
      if strcmp(get(btnHuman,'state'),'on')
         [bboxes, scores] = step(peopleDetector,frame);
         if strcmp(get(btnHuman,'state'),'on')
            frame = insertObjectAnnotation(frame,'rectangle',bboxes,scores);
         end
      end
```

```
        if strcmp(get(btnObject,'state'),'on')
        [~,centroid,bbox] = step(blobAnalyzer,fgMask);
        if ~isempty(centroid)
            for i=1:size(bbox,1)
                position = centroid(i,:);
                position(:,3) = ceil(max(bbox(i,3:4))/2)+2;
                if strcmp(get(btnObject,'state'),'on')
                    frame = insertObjectAnnotation(frame,'circle',position,['Object ',num2str(i)]);
                end
            end
        end
        end
        showFrameOnAxis(vPlayer, frame);
        showFrameOnAxis(mPlayer, fgMask);
        drawnow
    end
    if ~isPaused
        reset(video)
    end
    drawnow
end
reset(video)
end
function pauseVideo(hObject,~)
    isPaused=~isPaused;
    if strcmp(get(hObject,'String'),'Pause')
      set(hObject,'String' ,'Continue')
    else
        set(hObject,'String' ,'Pause')
        playVideo(0,0)
    end
end
function StopVideo(~,~)
    isStopped=true;
    isPaused=false;
```