

Наукова робота під шифром “ Дорога ”

Комп’ютерна система виявлення та аналізу дефектів поверхонь

2019 р.

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. ЗАГАЛЬНІ СПОСОБИ РЕАЛІЗАЦІЇ СИСТЕМ КОМП'ЮТЕРНОГО ЗОРУ	6
1.1. Загальне поняття зображення	6
1.2. Реалізація систем комп'ютерного зору	7
1.1.1. Реалізація способу отримання зображень	8
1.1.2. Реалізація попередньої обробки зображення	9
1.1.3 Пошук і виділення необхідних об'єктів	10
РОЗДІЛ 2. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАСОБУ ВИЯВЛЕННЯ НЕРІВНОСТЕЙ ПОВЕРХНІ ЗАСОБАМИ КОМП'ЮТЕРНОГО ЗОРУ	12
2.1. Опис програмного засобу для вирішення основного завдання роботи....	12
2.2. Технічне завдання на розробку програмного засобу для виявлення пошкоджень дорожнього полотна засобами комп'ютерного зору	12
2.3. Архітектура програмного засобу	16
2.4. Інструкція використання	19
2.5. Тестування розробленого програмного засобу	25
ВИСНОВКИ.....	29
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	30
ДОДАТКИ.....	32

ВСТУП

Актуальність. На даний момент технології досягли значного розвитку і продовжують розвиватися. Людство вже давно намагається створити собі подібні механізми і інші засоби що володіють такими ж можливостями як людина. У наш час більшість наукових розробок людства так чи інакше пов'язані зі штучним інтелектом та методами його використання у вирішенні задач будь-якого характеру, що і привернуло велику увагу до теми комп'ютерного зору так як це один із способів сприйняття навколишнього світу для машин. Комп'ютерне зір це також спроба копіювання людини де в якості очей виступають камери.

Нині кількість нових рішень і актуальних додатків для комп'ютерного зору продовжує зростати. Сфера комп'ютерного зору може бути охарактеризована як дуже різноманітна, молода та динамічна. Поняття зору можливо розділити на такі сфери як комп'ютерний зір, фотограмметрія, машинний зір, обробка зображень, розпізнавання та аналіз образів.

Проблеми створення і застосування методів комп'ютерного зору досліджували Сезеліскі Р., Даугман Дж., Стокман Дж., Шапиро Л., Чанг Шу, Конушин А., Раджеш Рао та інші.

Багато з методів і додатків все ще знаходяться в стадії фундаментальних досліджень, але все більше число методів знаходить застосування в комерційних продуктах, де вони часто складають частину більшої системи, яка може вирішувати складні завдання наприклад, в області медичних зображень або вимірювання і контролю якості в процесах виготовлення. Також мабуть однією з найбільших областей застосування комп'ютерного зору являється військове застосування. Одними з нових областей застосування є автономні транспортні засоби, включаючи підводні, наземні такі як роботи і машини, повітряні. Рівень автономності змінюється від повністю автономних (безпілотних) до транспортних засобів, де системи, засновані на комп'ютерному зорі, підтримують водія або пілота в різних ситуаціях.

У роботі розглядається тема «Комп'ютерна система виявлення та аналізу дефектів поверхонь» а саме виявлення пошкоджень дорожнього полотна за допомогою засобів комп'ютерного зору на підставі відео з відеореєстраторів звичайних автомобілів.

Об'єкт – комп'ютерний зір в сфері виявлення нерівностей поверхонь.

Предмет – програмний засіб для виявлення пошкоджень дорожнього полотна за допомогою засобів комп'ютерного зору на підставі відео з відеореєстраторів.

Мета – розробка програмного засобу для виявлення пошкоджень дорожнього полотна за допомогою засобів комп'ютерного зору.

Завдання роботи:

1. Проаналізувати наукову, спеціальну літературу та мережні джерела з метою вивчення досвіду створення та використання систем та алгоритмів комп'ютерного зору.

2. Дослідити існуючі систем комп'ютерного зору пов'язаних з темою.

3. Розробити технічне завдання до програмного засобу по темі роботи.

4. Розробка системи комп'ютерного зору для виявлення нерівностей поверхні.

У процесі реалізації поставлених завдань були використані такі **методи наукового дослідження**: *теоретичні*: теоретичний аналіз алгоритмів роботи комп'ютерного зору та програмування з метою вибору методів розробки та проектування програмного засобу; аналіз програм та ресурсів Інтернет; *емпіричні*: проектуванні та моделювання програмного засобу; аналіз даних тестування програмного коду та розробленого програмного засобу; інтерпретаційні методи, які дають можливість узагальнення і пояснення встановлених фактів і їх взаємозв'язку; *практичні*: розробка системи комп'ютерного зору для виявлення нерівностей поверхні.

Наукова новизна та практична значущість – полягає у розробці програмного засобу, метою якого буде полегшення виявлення пошкоджень дорожнього полотна на основі відео з відеореєстраторів автомобілів.

Апробація результатів дослідження здійснювалась шляхом публікації статті та подання документів на отримання авторського права: «Особливості розробки програмного засобу для виявлення пошкоджень дорожнього полотна за допомогою засобів комп'ютерного зору» (м. Житомир, 2019 р.).

Впровадження здійснювалось на базі кафедри інформатики та кібернетики Мелітопольського державного педагогічного університету імені Богдана Хмельницького.

Структура роботи: робота складається зі вступу, двох розділів, висновків, списку використаних джерел та додатків. Повний обсяг роботи складає 33 сторінок, з них 29 сторінок – основна частина.

РОЗДІЛ 1

ЗАГАЛЬНІ СПОСОБИ РЕАЛІЗАЦІЇ СИСТЕМ КОМП'ЮТЕРНОГО ЗОРУ

1.1. Загальне поняття зображення

При роботі систем комп'ютерного зору вони маніпулюють цифровими зображеннями.

Цифрове зображення це двовимірне зображення, представлене в цифровому вигляді. Залежно від способу опису, зображення може бути растровим або векторним. Растрове зображення представляється як прямокутний двовимірний масив чисел, при цьому кожне число відповідає одному елементу зображення або пікселя. Ці масиви часто передаються і зберігаються в стислому вигляді. Растрові зображення виходять за допомогою різних пристроїв, таких як цифрові фотоапарати, цифрові відеокамери, сканери і т. д., Або синтезуються штучно засобами машинної графіки. Векторні зображення виходять шляхом математичного опису елементарних геометричних об'єктів, зазвичай званих примітивами, таких як точки, лінії, сплайни, криві Безьє, кола та кола, багатокутники [1].

Найчастіше для зберігання цифрових зображень використовується колірний простір RGB. У ньому кожної з трьох осей (каналів) присвоюється свій колір: червоний, зелений і синій. На кожен канал виділяється по 8 біт інформації, відповідно, інтенсивність кольору на кожній осі може приймати значення в діапазоні від 0 до 255. Всі кольори в цифровому просторі RGB виходять шляхом змішування трьох основних кольорів.

На жаль, RGB не завжди добре підходить для аналізу інформації. Експерименти показують, що геометрична близькість квітів досить далека від того, як людина сприймає близькість тих чи інших квітів один до одного [16].

Але існують і інші колірні простору. Наприклад простір HSV (Hue, Saturation, Value). У ньому присутній вісь Value, що позначає кількість світла.

На нього виділено окремий канал, на відміну від RGB, де це значення потрібно обчислювати кожен раз. Фактично, це чорно-біла версія зображення, з якої вже можна працювати. Hue представляється у вигляді кута і відповідає за основний тон. Від значення Saturation залежить насиченість кольору. HSV набагато ближче до того, як ми уявляємо собі кольору [14].

1.2. Реалізація систем комп'ютерного зору

Для реалізації систем комп'ютерного зору найкраще застосовувати вже готові рішення і методи і оптимізувати їх за себе. На даний момент вже існує велика кількість методів і рішень, які збірнені в бібліотеки прикладом служать такі бібліотеки як OpenCV, CcV, scikit-learn, EmguCV і т.д.

Наприклад OpenCV це бібліотека з відкритим вихідним кодом. Вона включає в насіння безліч алгоритмів комп'ютерного зору, обробки зображень та чисельних алгоритмів загального призначення з відкритим кодом. Реалізована на C / C ++, також розробляється для Python, Java, Ruby, Matlab, Lua та інших мов. До цієї бібліотеки існує безліч відео уроків, лекцій, практик.

Бібліотека ccv написана на C і позиціонується як компактна, легка альтернатива OpenCV, тому з неї старанно вилучені всі несуттєві функції. Таким чином, ccv призначена не для експериментів з різними алгоритмами, а для практичного використання в конкретних додатках [7].

Emgu CV є крос-платформних .Net доповненням для бібліотеки OpenCV для обробки зображень. Розроблено для роботи з .NET сумісними мовами, такими як C #, VB, VC ++, IronPython і т.д., може бути використано в Visual Studio, Xamarin, працює з Windows, Linux, Mac OS X, IOS, Android і Windows Phone [5].

Для реалізації проекту на тему «виявлення нерівностей поверхні засобами комп'ютерного зору» буде застосовуватися бібліотека Emgu CV так як проект буде виконуватися в Visual Studio 2013 на мові програмування C # на підставі того що C # дозволяє стартувати розробку швидше, а це дозволяє швидше

отримати прототип рішення . Швидкість розробки на C # на початкових етапах проекту значно вище в порівнянні з C ++. Також C # зручніше налагоджувати та збирання C ++ проектів помітно складніше збірки проектів C #. Звичайно C # не без недоліків але в даному випадку використовувати краще його.

1.1.1. Реалізація способу отримання зображень

Цифрові зображення виходять від одного або декількох датчиків зображення, які крім різних типів світлочутливих камер включають датчики відстані, радари, ультразвукові камери і т. Д. В залежності від типу датчика, що виходять дані можуть бути звичайним 2D зображенням, 3D зображенням або послідовністю зображень. Значення пікселів зазвичай відповідають інтенсивності світла в одній або декількох спектральних смугах, але можуть бути пов'язані з різними фізичними вимірами, такими як глибина, поглинання або відбиття звукових або електромагнітних хвиль, або ядерним магнітним резонансом [2].

В процесі розробки після створення проекту підключення до нього бібліотеки Emgu CV та створення форми потрібно вирішити проблему отримання даних для системи комп'ютерного зору. У нашому випадку система служить для виявлення пошкоджень дорожнього щільна на підставі відео з відеореєстратора автомобіля [24]. Для початку вирішення даної проблеми потрібно отримати дані відео файлу для цього можна скористатися вбудованим класом OpenFileDialog. Створюємо об'єкт класу OpenFileDialog і встановлюємо тип фільтра для вибору тільки відео файлів [10, с. 120]. Далі потрібно зберегти відео файл в екземпляр класу від Emgu CV під назву Capture що відповідає за захоплення відео потоку в ньому буде зберігатися відео файл і дані про нього. З об'єкта Capture ми отримуємо такі дані як загальна кількість кадрів, FPS, і зображення кожного кадру для обробки під час програвання відео [3].

1.1.2. Реалізація попередньої обробки зображення

Перед тим як на кожному окремому кадрі відео виконувати пошук нерівностей потрібно провести з кожним зображенням обробку так як перед тим, як методи комп'ютерного зору можуть бути застосовані до відеоданих необхідно обробити відеодані, з тим щоб вони задовольняли деяким умовам, в залежності від використовуваного методу. У нашому випадку спробуємо прибрати шуми і збільшити контрастність зображення для того, щоб потрібна інформація могла бути виявлена [5].

Процедури фільтрації покликані вирішити конкретні завдання предоброботки: усунути адитивні високочастотні перешкоди на оригінальному документі, підкреслити контурні переходи на зображенні, виключити з початкового зображення різкі перепади основний фонової складової і т.д.

Типи фільтрації зображень [12]:

- Широкополосная низькочастотна лінійна фільтрація.
- Спрямована низькочастотна лінійна фільтрація
- Медіанні фільтри.
- Векторні медіанний фільтри.

Всі алгоритми процедури контрастування, так чи інакше, виконують обробку зображення, використовуючи гістограму розподілу яскравості елементів зображення, або в локальній області, або всього масиву даних.

Найбільш часто використовуваних процедур контрастування:

- глобальне розтягнення
- локальне розтягнення

Локальне рівномірний перетворення [4].

1.1.3 Пошук і виділення необхідних об'єктів

Для реалізації пошуку пошкоджень дорожнього полотна в проекті використовувалося два різних методи пошук за кольором і пошук по контуру.

Найпоширенішим способом виділити об'єкт є колір. Однією з найважливіших проблем пошуку за кольором цей вплив безлічі різних факторів таких як наприклад освітленість. Не можна також забувати, що видимий колір - це результат взаємодії спектра випромінюваного світла і поверхні. Для створення методу для виявлення місць пошкоджень дорожнього покриття по квіти ми розбили зображення на шари що були збережені в відтінках сірого кольору для подальшої обробки на одному з шарів. У цьому шарі пікселі колір яких вище певного порогу перефарбовуватися в сірий поріг може регулюватися в ручну. Даний метод полягає в тому що пошкожені ділянки дорожнього полотна будуть темніші ніж цілі що дозволяє виділити їх чорним кольором. Недоліком даного методу є те що на нього впливає безліч сторонніх чинників. Наприклад після дощу в калюжах буде відображатися небо що унеможливить визначити пошкодження цим методом так як пошкожені ділянки будуть світліше також на цей метод дуже сильно впливає освітлення і наявність тіней на ділянці дороги [6].

Другий реалізований метод це метод пошуку по контуру. Контурний аналіз - це один з важливих і дуже корисних методів опису, зберігання, розпізнавання, порівняння та пошуку графічних образів / об'єктів. Контур це зовнішні обриси об'єкта.

При проведенні контурного аналізу:

- потрібно було, що контур містить достатню інформацію про форму об'єкта;
- внутрішні точки об'єкта до уваги не беруться.

Таке положення, накладають суттєві обмеження на область застосування контурного аналізу, які, в основному, пов'язані з проблемами виділення контуру на зображеннях:

- через однакової яскравості з фоном об'єкт може не мати чіткої межі, або може бути зашумлений перешкодами, що призводить до неможливості виділення контуру;

- перекриття об'єктів або їх угруповання призводить до того, що контур виділяється неправильно і не відповідає кордоні об'єкта.

Контурний аналіз має досить слабку стійкість до перешкод, і будь-який перетин або лише часткова видимість об'єкта призводить або до неможливості детектування, або до помилкових спрацьовувань, але простота і швидкодія контурного аналізу, дозволяють цілком успішно застосовувати даний підхід.

Звичайна послідовність дій при розпізнаванні об'єктів методом контурного аналізу:

- попередня обробка зображення (згладжування, фільтрація перешкод, збільшення контрасту);

- бінаризація зображення;

- виділення контурів об'єктів;

- первинна фільтрація контурів (по периметру, площі і т.п.);

- еквалізація контурів (приведення до єдиної довжині, згладжування) - дозволяє домогтися інваріантності до масштабу;

- перебір всіх знайдених контурів і пошук шаблону, максимально схожого на даний контур або ж сортування контурів по будь-якою ознакою, наприклад, площі [8].

РОЗДІЛ 2

РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАСОБУ ВИЯВЛЕННЯ НЕРІВНОСТЕЙ ПОВЕРХНІ ЗАСОБАМИ КОМП'ЮТЕРНОГО ЗОРУ

2.1. Опис програмного засобу для вирішення основного завдання роботи

Метою проекту є розробка програмного модуля який спрощує виявлення пошкоджених ділянок дорожнього полотна на підставі відео з відеореєстраторів автомобілів.

Програмний засіб може бути застосоване Укравтодором або іншими подібними підприємствами для визначення рівня якості дороги або аналізу рівня її пошкодження на підставі відео з відеореєстраторів звичайних користувачів автомобілів.

Для розробки інтерфейсу застосовувалося Windows Form C #, розробка інтерфейсу проводилася в середовищі Microsoft Visual Studio. Для реалізації засобів комп'ютерного зору було використано Emgu CV яке є крос-платформним .Net доповненням для бібліотеки OpenCV для обробки зображень.

За для покращення процесу роботи із програмним засобом, він має відповідати таким вимогам: даний програмний продукт повинен бути легким, з точки зору навантажень на ПК користувача і не займати великої кількості місця на жорсткому диску. Використання програмного продукту не повинно вимагати установки, а так само не повинно вимагати установки додаткового ПЗ [7].

2.2. Технічне завдання на розробку програмного засобу для виявлення пошкоджень дорожнього полотна засобами комп'ютерного зору

Робота над створенням системи комп'ютерного зору з виявлення нерівностей поверхні проводилася в кілька етапів.

1. Розробка технічного завдання, яке включає в себе формування мети і завдань, опис очікуваних результатів, змісту і основних вимог до системи.
2. Вибір джерел теоретичних матеріалів.
3. Розробка структури системи.
4. Створення інтерфейсу.
5. Реалізація системи.

Це технічне завдання визначає вимоги і порядок виконання програмного засобу для виявлення пошкоджень дорожнього полотна засобами комп'ютерного зору.

Предметом розробки є система комп'ютерного зору для виявлення нерівностей поверхні. Підставою для розробки є завдання, видане кафедрою інформатики та кібернетики.

Мета - розробка програмного засобу для виявлення пошкоджень дорожнього полотна засобами комп'ютерного зору.

Програмний засіб може бути застосоване Укравтодором або іншими подібними підприємствами для визначення рівня якості дороги або аналізу рівня її пошкодження на підставі відео з відеореєстраторів звичайних користувачів автомобілів [11].

Для розробки інтерфейсу застосовувалося Windows Form C #, розробка інтерфейсу проводилася в середовищі Microsoft Visual Studio.

Результати що очікуються: системи комп'ютерного зору що дозволить виявляти нерівності поверхні за допомогою камери.

Системні вимоги:

– автономність програми, тобто її запуск не має передбачати наявності на комп'ютері кінцевого користувача спеціальних програмних засобів, окрім операційної системи;

– розміри програми не перевищують 1500 Мб й програма являє собою .exe-файл, що встановлюється на комп'ютер за допомогою копіювання .exe-файлу та не вимагає спеціальної інсталяції.

Початок роботи з програмою проходить після запуску .exe-файлу.

Вимоги до функціональності програмного засобу:

- вільне використання;
- надійність – реалізується за допомогою програмного забезпечення високого рівня;

– зручність використання системи;

– працездатність – визначається вимогами до обладнання;

Вимоги до дизайну:

– інтуїтивно зрозумілий інтерфейс – доцільно зрозуміле застосування об'єктів, що сприяє зручності користування;

– ергономічність – просторове розміщення інформації, виділення головного об'єкту простору;

– інтерактивність – досягається завдяки зручній навігації та швидкій обробці і виведенню даних;

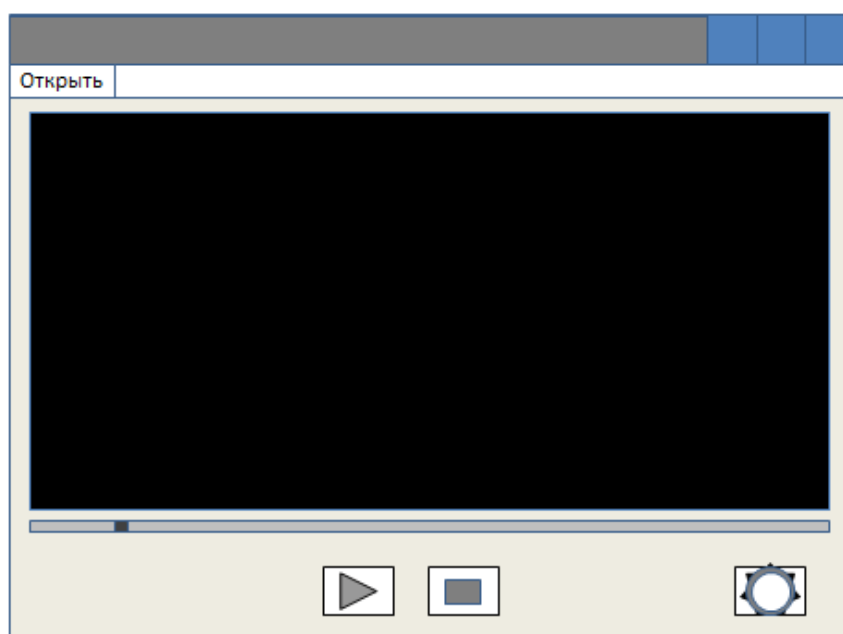


Рис. 2.1 Загальний інтерфейс

Інтерфейс розроблюваного програмного засобу є загальний інтерфейс (рис. 2.1). Загалом інтерфейсі присутні такі елементи як кнопка відкриття нового файлу для обробки, область для відтворення обраного файлу, смуга

прокрутки, кнопки взаємодії з відео файлом і кнопка виклику форми налаштувань [17].

Вікно налаштувань представляється формою, яка викликається поверх поточної (рис. 2.2). У ній реалізовані способи вибору методів і установка їх параметрів для виявлення місць пошкоджень на дорожньому полотні [21].

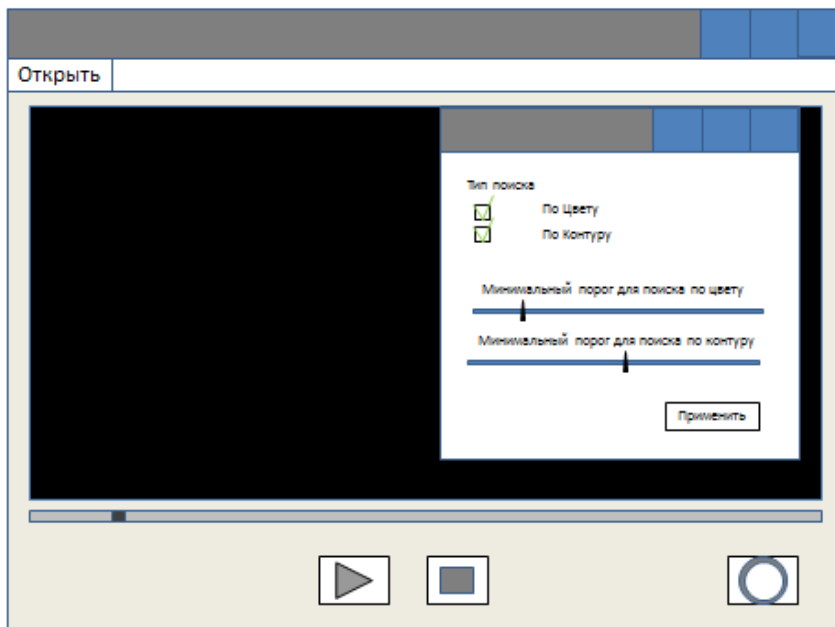


Рис. 2.2 Форма настроек

При відкритті нового відео файлу воно буде виглядати також як в оригіналі без будь-якої обробки (рис. 2.3).

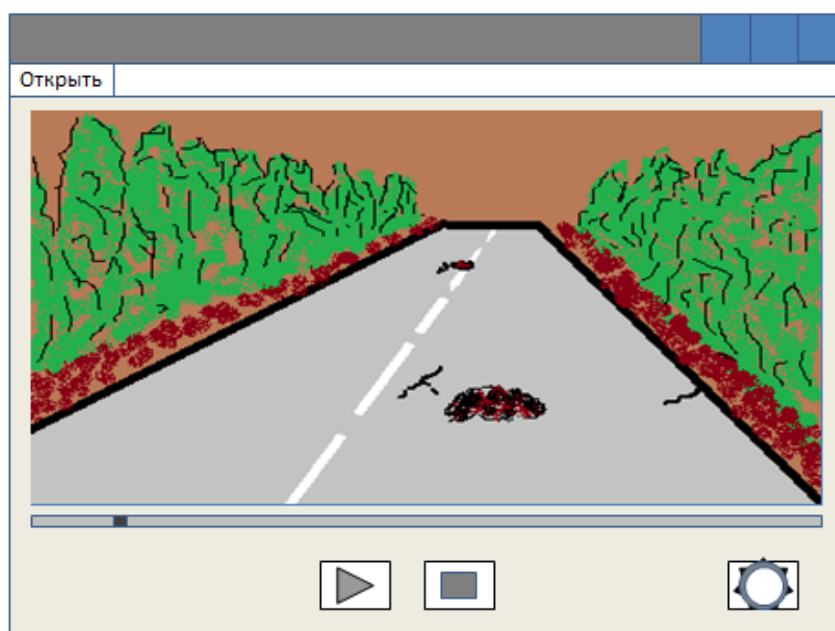


Рис. 2.3 Зображення без обраного типу пошуку

Після застосування методу пошуку за кольором на кожному кадрі зображення зміниться на версію де ділянки темнішого кольору виділяться в чорним кольором [14].

Після застосування другого методу по контуру на кожному кадрі зображення зміниться на версію де на зображенні чорного кольору залишаться тільки контури (рис. 2.4).

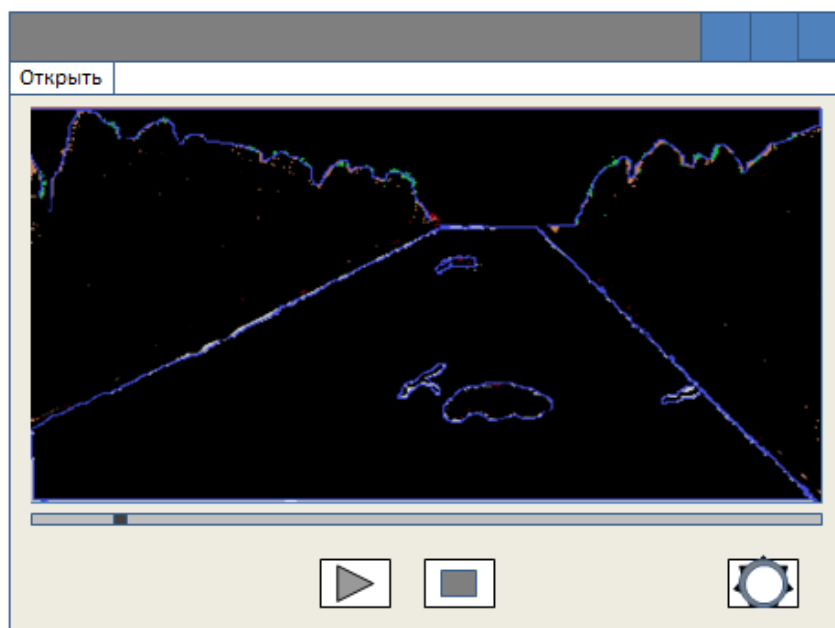


Рис. 2.4 Зображення при пошуку по контуру

Також передбачена можливість змінювати параметри чутливості обох методів і їх одночасне застосування [22].

2.3. Архітектура програмного засобу

Для розробки програмного засобу було створено три класи: клас Form1, клас VideoHendler та клас Setings [15].

Клас Form1 реалізує собою головну форму програмного засобу та реалізує методи відкриття нового відео та методи взаємодії с ним такі як програти, пауза, стоп, методи для взаємодії з класом Setings та VideoHendler (рис. 2.4).

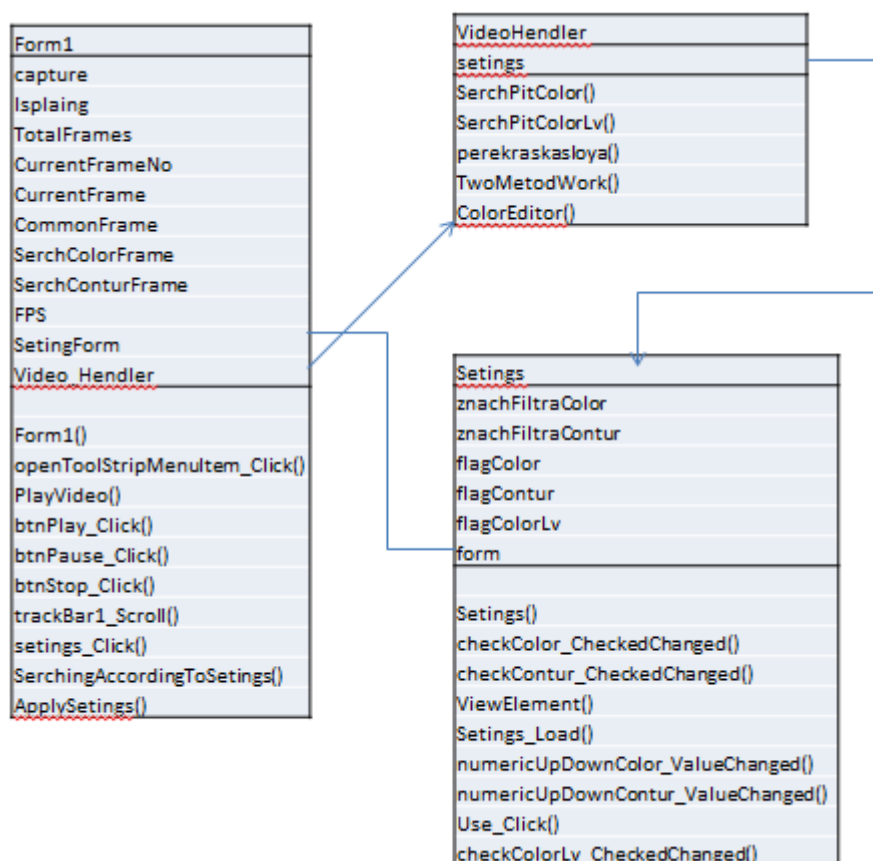


Рис. 2.4. Архітектура програмного засобу

Клас Setings реалізує собою форму налаштувань в програмному засобі та методи передачі цих значень цих налаштувань в інші класи. В цьому класі вибирається метод пошуку пошкоджень дорожнього полотна, та значення фільтрів для роботи цього методу.

Клас VideoHendler представляє собою набір методів які реалізують обробки кожного окремого кадру або різні методи пошуку нерівностей [21].

Для реалізацій комп'ютерної системи виявлення та аналізу дефектів поверхонь було обрано Emgu CV яка є крос-платформенною платформою .Net для бібліотеки обробки зображень OpenCV. Дозвіл OpenCV функцій які потрібно використовувати в .NET сумісних мов таких як C #, VB, VC ++, IronPython т.д. оболонка може бути складена Visual Studio, Xamarin студії і єдності, він може працювати на Windows, Linux, Mac OS X, прошивкою, Android і Windows Phone.

Так як для реалізації проекту найбільш придатною являється бібліотека OpenCV завдяки тому що в ній вже присутнє велика кількість рішень для різних завдань комп'ютерного зору і обробки зображень також існує велика кількість відео уроків, інструкцій застосування, лекцій і тому подібного в інтернеті. Все це сильно допомагає в розробці системи комп'ютерного зору [20, с. 65].

Основні модулі бібліотеки:

сxscore - ядро містить базові структури даних і алгоритми: базові операції над багатовимірними числовими масивами; матрична алгебра, математичні функції, генератори випадкових чисел; запис / відновлення структур даних в / з XML; базові функції 2D графіки; CV - модуль обробки зображень і комп'ютерного зору; базові операції над зображеннями (фільтрація, геометричні перетворення, перетворення колірних просторів і т. д.); аналіз зображень (вибір відмінних ознак, морфологія, пошук контурів, гістограми); аналіз руху, спостереження за об'єктами; виявлення об'єктів, зокрема осіб; калібрування камер, елементи відновлення просторової структури

Highgui - модуль для введення / виведення зображень і відео, створення призначеного для користувача інтерфейсу: захоплення відео з камер і з відео файлів, читання / запис статичних зображень; функції для організації простого UI (всі демо додатки використовують HighGUI).

Svaux - експериментальні і застарілі функції: просторів. зір: стерео калібрації, саме калібрації; пошук стерео-відповідності, кліки в графах; знаходження і опис рис обличчя;

CvCam - захоплення відео: дозволяє здійснювати захоплення відео з цифрових відео-камер (підтримка припинена і в останніх версіях цей модуль відсутній).

OpenCV написана на C++ і її основний інтерфейс також реалізовано на C++, але бібліотека і досі представляє старіший C інтерфейс. На даний момент реалізовано інтерфейс на мовах Python, Java і MATLAB/OCTAVE. Оболонки для інших мов, таких як C#, C#, Ruby були розроблені з метою охоплення

ширшої аудиторії. Так як для роботи використовується мова програмування C# було обрано крос-платформенну платформу .Net для бібліотеки обробки зображень OpenCV яка називається Emgu CV.

На даний момент основними обгортками бібліотеки OpenCV є наступні інструменти: EmguCV; OpenCV#; CV.NET; SharperCV; CodeProject.

На основі порівняння наведеного у таблиці А.1 (додаток А) за основу системи було обрано EmguCV.

Для можливості повного використання сучасних технологій програмного та апаратного забезпечення, під час проектування базової моделі системи було обрано обгортку над бібліотекою OpenCV – EmguCV, та відповідно мову програмування C#.

Emgu CV має два шари оболонки (додаток Б, рис. Б.1). Основний шар (рівень 1) містить функції, структуру та перерахування, які безпосередньо відображають ті, що вказані в OpenCV.

Другий шар (шар 2) містить класи, які змішуються в перевагах зі світу .NET.

2.4. Інструкція використання

Для початку роботи даного програмного засобу необхідно запустити exe файл з цією програмою.

Після запуску exe файлу відкриватися вікно головної форми (рис. 2.5).

Дане вікно з рядка меню в якому присутні кнопки відкриття файлу і закриття програми, виділено ділянку для програвання відео файлу а також його смуга прокрутки, кнопка програвання і паузи а також кнопки зупинки і налаштувань.

Для подальшої роботи з програмою необхідно вибрати файл з відео за яким буде проводитися виявлення. У програмному засобі за замовчуванням встановлений необхідний формат відео для роботи. Відкриття файлу відбувається при натисканні кнопки відкрити в рядок меню пункту файл. В

процесі вибору відеофайлу бажано уникати вибору відео файлів з великим розширенням, так як відео файли з великим розширенням будуть сильно уповільнювати роботу даної програми, так як на кожен кадр буде йти більше часу для обробки [9].

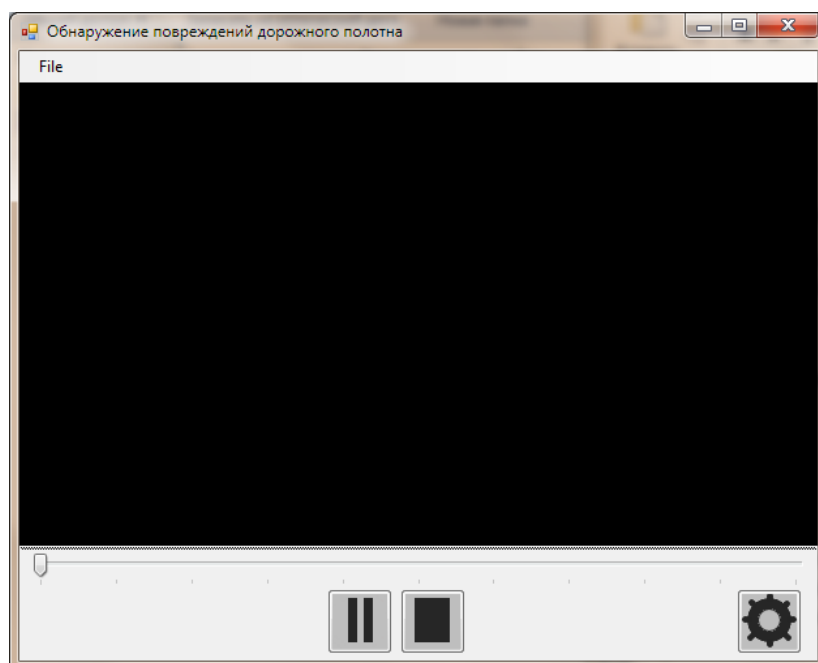


Рис. 2.5. Головна форма

Відразу після відкриття файлу він буде програватися, без будь-яких методів пошуку, так як це встановлено за замовчуванням. Для зміни способу виявлення місць пошкоджень дорожнього полотна необхідно перейти в меню налаштувань для чого і існує кнопка настройки.

Кнопка налаштувань служить для виклику форми настройки поверх поточної форми. Ця форма являє собою вікно, в якому можна вибрати спосіб пошуку різними методами. У програмному засобі реалізовані два різних методу пошуку які посредством установки галочки можна вибрати [13]. При відсутності галочок на обох методах, що встановлено за замовчуванням, відео буде програватися без використання будь-яких методів. У програмі передбачено чотири види роботи перший це той що без обраного методу він відтворює відео в його оригінальному вигляді (рис. 2.6).

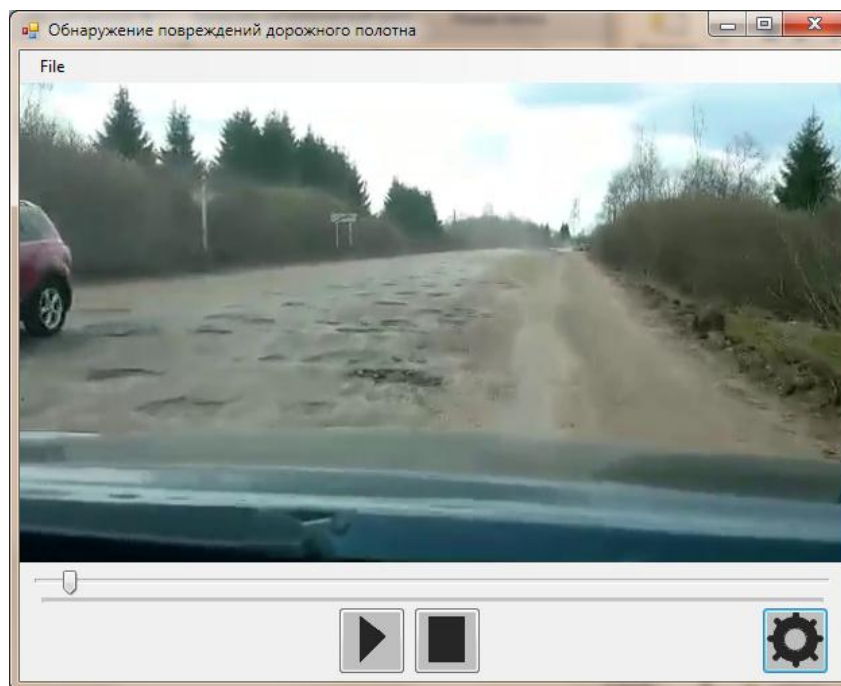


Рис. 2.6. Работа файла без выбора метода

Другим методом пошуку є метод пошуку за кольором. Цей метод досить сильно поширений і слугує для виділення об'єктів на підставі його кольору. Але даний метод схильний до впливу безлічі факторів, що впливає на точність його роботи. У програмі він працює на підставі зображення переведеного з кольорового в відтінки сірого і перефарбування біліше темних ділянок в чорний. Принцип роботи такий що він використовує значення фільтра і переводить в відтінок сірого кольору і перефарбовувати його і все що темніше в чорний. Цей метод дозволяє підбирати потрібний рівень фільтра в ручну що дозволяє зробити його більш універсальним так як настройка може бути змінена при зміна освітленості дороги [19].

Прикладом роботи даного методу служать зображення Пошук за кольором з встановленим значенням фільтра в 100 (рис 2.7) і Пошук за кольором з встановленим значенням фільтра в 120 (рис 2.8).

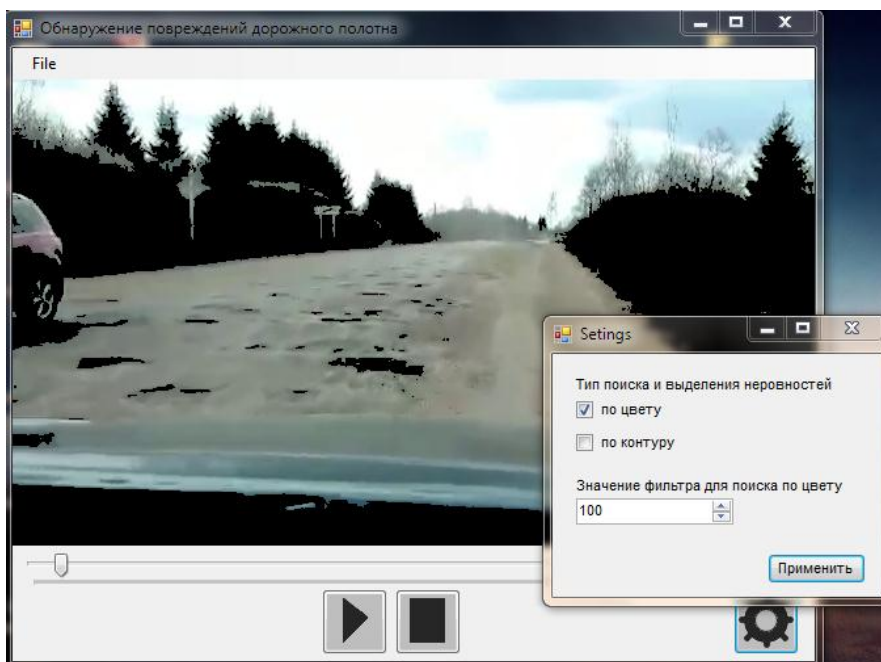


Рис. 2.7. Пошук за кольором з встановленим значенням фільтра в 100

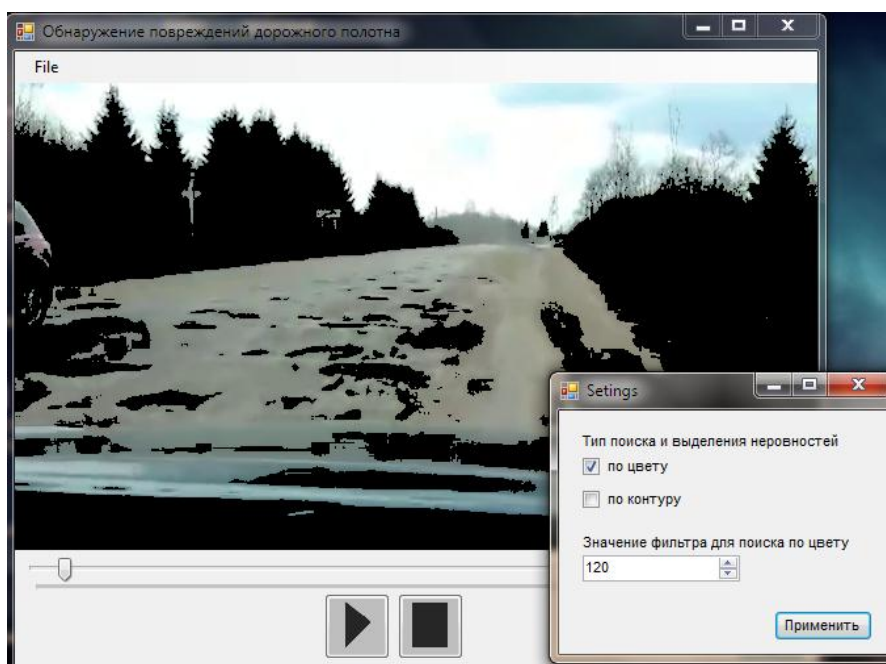


Рис. 2.8. Пошук за кольором з встановленим значенням фільтра в 120

Також можливо проводити пошук за кольором виводячи декілька кольорів в залежності від затемнення та значення фільтру для пікселів які темніші за значення фільтру будуть перемальовуватися в чорний колір, для другого кольору який позначається червоним кольором значення фільтру береться на 5 більше що повинно виводити на зображенні ділянки з менш затемненою частиною що сприймаються як менші пошкодження, третім

кольором є помаранчевий значення фільтру береться на 10 більше встановленого цим кольором позначаються ділянки з найменшими виявленими пошкодженнями (рис. 2.9).

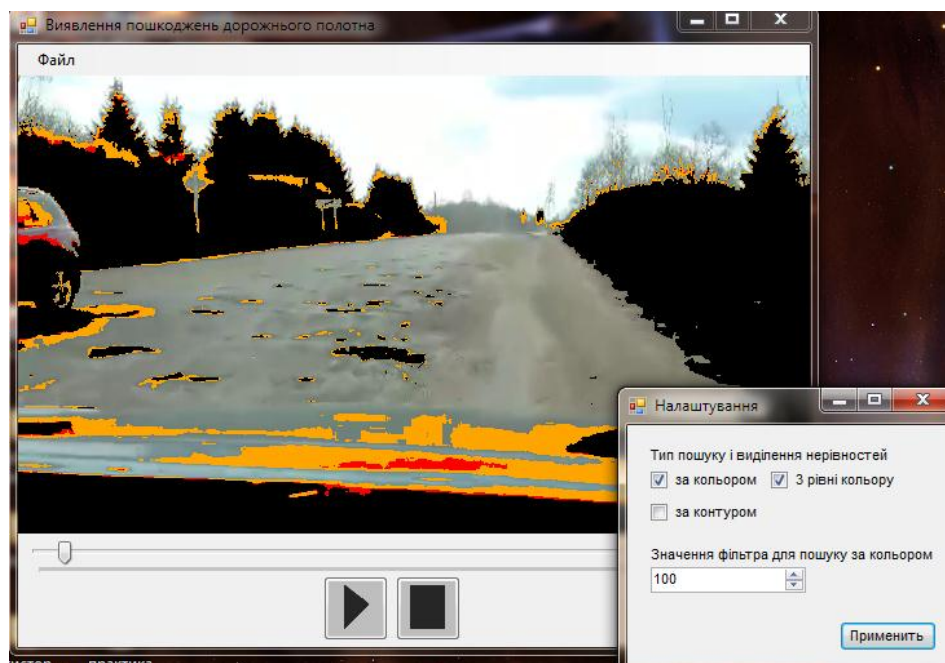


Рис. 2.9. Пошук за кольором с 3 фільтрами

Третім методом пошуку є метод пошуку по контуру. Це один з важливих і дуже корисних методів опису, зберігання, розпізнавання, порівняння та пошуку графічних об'єктів. Принцип роботи такий, що він використовує значення фільтра, перевіряє різкість зміни кольору і виділяє ділянки з нею контурами. Різкість визначається на підставі значення фільтра, яку можна змінити в ручну, що дозволяє зробити його більш універсальним так як настройка може бути змінена при зміна освітленості дороги. В даному методі великий вплив на працездатність мають шуми і схожість кольору необхідних ділянок з фоном.

Прикладом роботи даного методу служать зображення Пошук по контуру з встановленим значенням фільтра в 100 (рис 2.11) і Пошук по контуру з встановленим значенням фільтра в 120 (рис 2.10).

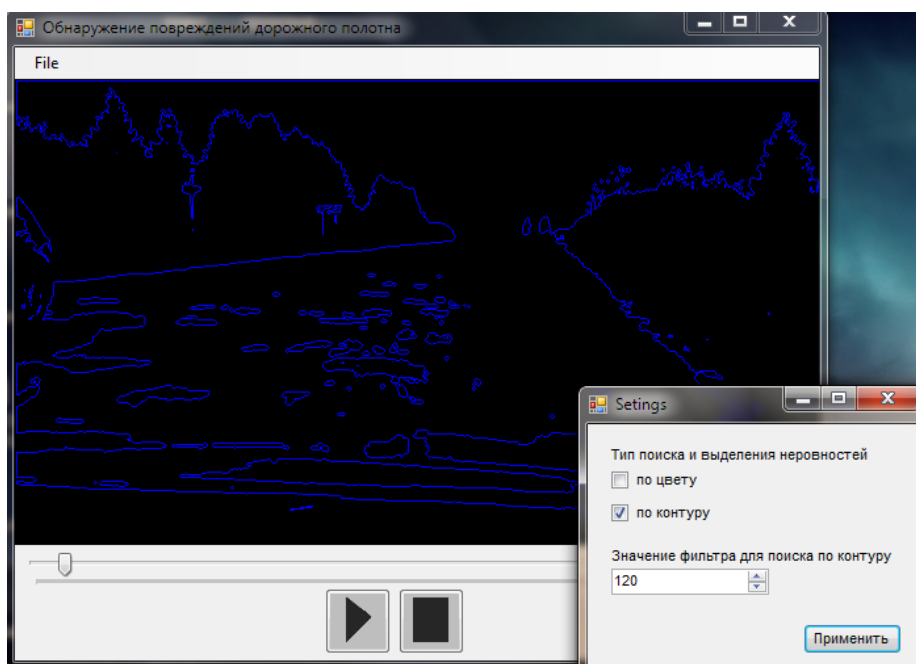


Рис. 2.10. Пошук по контуру з встановленим значенням фільтра в 120

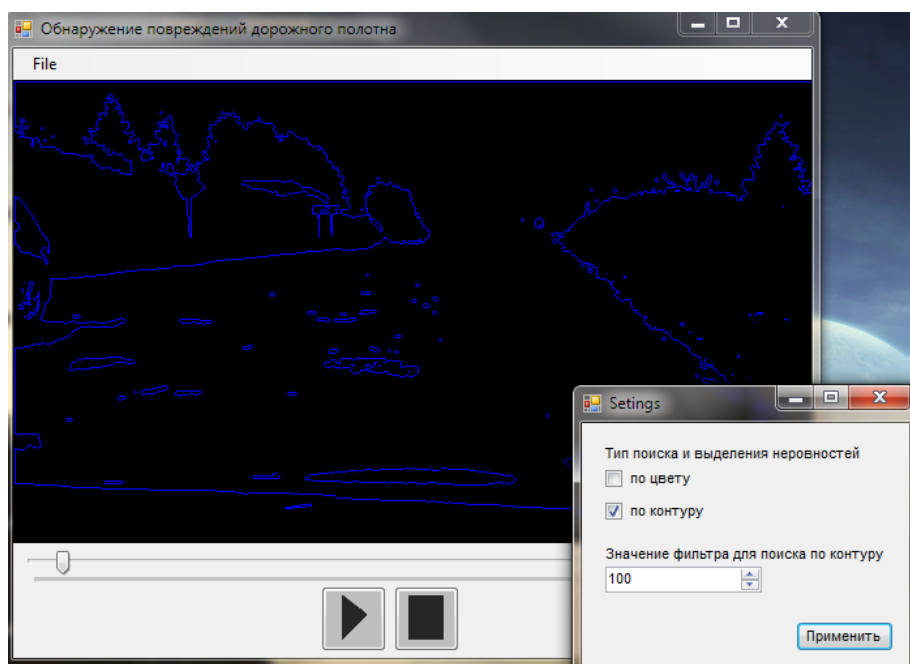


Рис. 2.11. Пошук по контуру з встановленим значенням фільтра в 100

Останнім способом пошуку є одночасне застосування двох методів пошуку. Метод полягає в тому щоб на зображення, яке з'явиться результатом виконання методу пошуку за кольором накласти результат виконання методу пошуку по контуру, таким чином щоб перенести тільки контури без чорного фону. Даний метод в деяких випадках дозволить цим двом методам

компенсувати недоліки один одного. Ще даний метод може допомогти визначити різкість зміни глибини ям при різних встановлених значеннях фільтрів для обох методів як на зображенні (рис. 2.12)

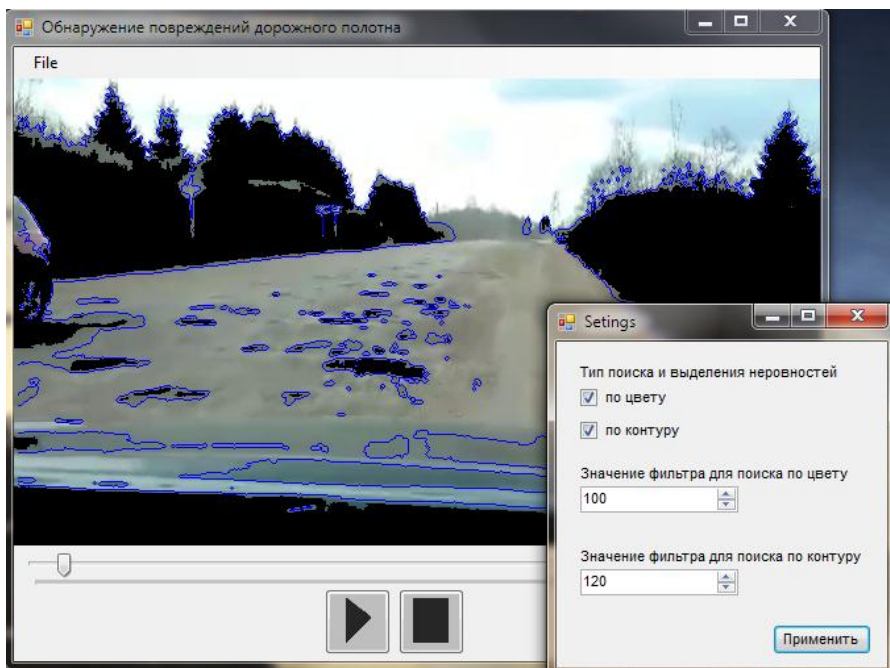


Рис. 2.12. Пошук засобами обох методів одночасно

2.5. Тестування розробленого програмного засобу

Для оцінки результатів роботи будемо розглядати наскільки точно програма розпізнає нерівності на відео з відеореєстратора автомобіля. Оскільки програма працює тільки з послідовностями зображень, які мають більш менш статичний фон. Деякі зміни яскравості в продовж відео впливають на якість роботи програмного засобу наприклад такі як відблиски зміна освітленості, тіні дерев на дорозі і тому подібні. Неякісне зображення може привести до помилкових визначень що також зменшує точність результатів.

Для тестування програмно засобу скористаємося відео з відео реєстраторів автомобілів при різних значеннях освітлення, у тінях та інших можливих ситуаціях з фіксованими налаштуваннями фільтрів.

В першому тесті перевіро працездатність програмного засобу з нормальною освітленістю (рис. 2.7). Також на прикладі бачимо що тіні також в

процесі пошуку також вважаються як пошкодження дорожнього полотна бо за кольором вони схожі (рис. 2.13). Для рішення цієї проблеми краще усього встановити значення фільтрів кольору вручну поки робота не буде задовільнювати вимогам.

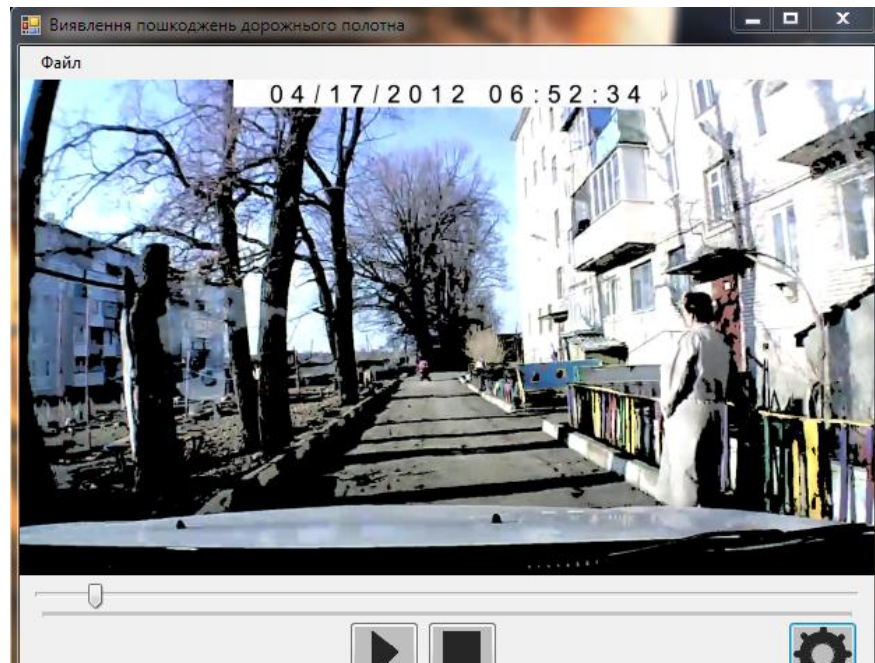


Рис. 2.13. Пошук при нормальному освітленні

В другому тесті перевірмо працездатність програмного засобу при праці у тіні або з недостатньою освітленістю. При недостатньому освітленні програмний засіб потребує зміни значення фільтру з стандартного значення в 100 на значення 50-70 для задовільної працездатності програмного засобу. При стандартному значенні програмний засіб працює не коректно, але при значенні фільтру 50-70 пошук пошкоджень дорожнього полотна проходить задовільно (рис. 2.14). Значення фільтру краще всього підирати вручну при випадках зміни фонового освітлення, кольору дороги наприклад після дощу бо вона стає темнішою, та інших впливаючих факторах . В процесі тестування є моменти значного погіршення якості праці які не можливо виправити змінами в значенні фільтру для покращення працездатності прикладом таких випадків є момент переходу з освітленого місця в не достатньо освітлене але в даному випадку

людський зір також погано бачить бо за освітленню частиною все виглядає чорним.

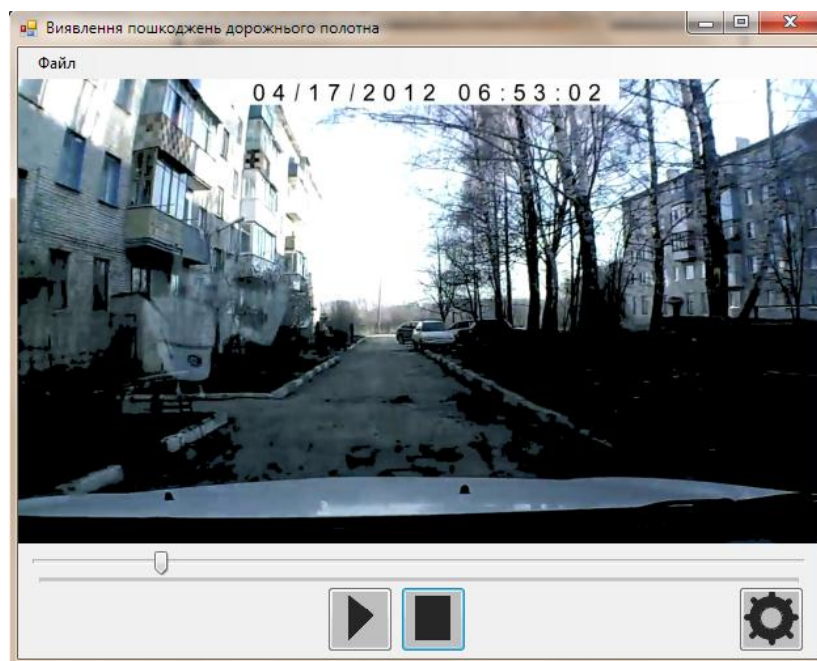


Рис. 2.14. Пошук при недостатньому освітленні

В третьому тесті перевірмо працездатність програмного засобу при праці з занадто яскравим освітленням спереду автомобіля. При такому освітленні якість роботи програмного засобу знижується незалежно від значення фільтру так як при стандартному значенні в 100 знаходження пошкоджень дорожнього полотна на відстані проходить нормально а біля автомобілю значення фільтру в 100 занадто високе тому виділяться занадто багато. При підборі такого значення при якому якість знаходження пошкоджень дорожнього полотна біля авто задовільна то пошкодження що знаходяться спереду автомобіля на деякій відстані не знаходяться (рис. 2.15).



Рис. 2.15. Пошук при занадто яскравому освітленні з переду

Як висновок тестування програмного засобу можна сказати що він працює але якість сильно залежить від різних факторів які в процесі праці можуть змінюватися і зменшувати якість виконаної роботи. При ідеальних умовах програмний засіб працює добре але підтримувати ідеальні умови завжди на дорозі не є можливим. Для роботи програмного засобу в різних умовах треба змінювати значення фільтрів що вимагає втручання людини для ручного налаштування значень фільтрів в процесі роботи. Також при деяких визначених умовах існують випадки коли через якісь фактори програмний засіб працює не коректно і в якості результату видає не вірне рішення. Ще однією проблемою є швидкість роботи програмного засобу яка не дозволяє застосовувати весь функціонал програмного засобу одночасно також швидкість праці занадто низька при роботі з відео з великим розширенням для оптимальної роботи розширення має бути не більшою ніж 720 пікселів бо це дуже сильно впливає на швидкість як і кількість робочих фільтрів.

ВИСНОВКИ

Комп'ютерне зір є досить молодий і добре розвивається областю. У наш час комп'ютерний зір застосовується у великій кількості різних сфер. Ми можемо і не помічати цього, але різні системи комп'ютерного зору застосовуватися в нашій звичайному житті. Одним з найбільш важливих застосувань комп'ютерний зір в сфері медицини для обробка зображень з рентгенографії, мікроскопії, томографії, ультразвукових досліджень і ангиографії. Також з нових областей застосування комп'ютерного зору останнім часом є автономні транспортні засоби такі як наприклад безпілотні автомобілі та інших видів транспортних засобів. Так що розробка програмних засобів у сфері комп'ютерного зору є досить перспективною і затребуваною.

Метою даної роботи було розробка програмного засобу для виявлення нерівностей поверхні засобами комп'ютерного зору, для досягнення цієї мети були поставлені та виконані 5 основних завдань роботи.

Під час роботи була вивчена література і особливості систем комп'ютерного зору. Після цього був проведений аналіз можливостей застосування комп'ютерного зору напрямків його розвитку на практиці. Результатом аналізу став висновок про те що розробка програмного засобу в сфері комп'ютерний зір службовця для виявлення нерівностей поверхні є перспективною.

Обравши методи розробки та проектування було зроблено висновок, що оптимальною мовою розробки навчального програмного засобу оптимальною середою програмування є Microsoft Visual C#, тому що вона задовольняє усім вимогам технічного завдання та об'єктно-орієнтованого методу програмування, є простою у розробці та подальшому використанні програмного засобу.

Результатом роботи став програмний засіб для виявлення пошкоджень дорожнього полотна за допомогою засобів комп'ютерного зору.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Audi and NVIDIA team up to bring fully automated driving to the roads starting in 2020 accelerated with artificial intelligence | Audi USA [Електронний ресурс]. – Режим доступу: <https://www.audiusa.com/newsroom/news/press-releases/2017/01/audi-and-nvidia-to-bring-fully-automated-driving-in-2020>
2. Autonomous Car Development Platform | NVIDIA DRIVE AGX [Електронний ресурс]. – Режим доступу: <https://www.nvidia.com/en-us/self-driving-cars/drive-platform/>
3. Berthold K. Robot Vision. / Berthold K. P. –1986. – 48с.
4. BMW to introduce HoloActive Touch at the 2017 CES [Електронний ресурс]. – Режим доступу: <https://www.bmwblog.com/2016/12/15/bmw-introduce-holoactive-touch-2017-ces/>
5. EMGU CV Tutorial / 2008. – 36с.
6. Home of Smarter Appliances - Connect to Your Kitchen Like Never Before. [Електронний ресурс]. – Режим доступу: <https://smarter.am>
7. OpenCV / RoboCraft. Роботы? Это просто! [Електронний ресурс]. – Режим доступу: <http://robocraft.ru/page/opencv/>
8. System Vision Design: High Speed Image Processing [Електронний ресурс]. – Режим доступу: <http://www.k2.t.u-tokyo.ac.jp/vision/DPM/>
9. Working with Images - Emgu CV: OpenCV in .NET (C#, VB, C++ and more) [Електронний ресурс] / Working with Images – Режим доступу: http://www.emgu.com/wiki/index.php/Working_with_Images
10. Астелс Д. Практичний посібник з екстремального програмування: Пер. з англ. / Д. Астелс, Г. Миллер, М. Новак – М .: Видавничий дім "Вільямс", 2002. - 320 с.
11. Брадски Г. Learning OpenCV . / Г.Р. Брадски, А.Каехлер - O'Reilly Media. – 2008, – 555с.

12. Васильев В. Комп'ютерна графіка: Навчальний посібник / В.Е. Васильев, А.В. Морозов – СПб.: СЗТУ, 2005. – 101с.
13. Волошин Н.В. Методи автоматичного розпізнавання автомобільних номерних знаків / Волошин Н.В., Федорів П.С. Тези доповіді на III Всеукраїнській науковотехнічній конференції «Теоретичні та прикладні аспекти радіотехніки і приладобудування». - Тернопіль, ТНТУ, 2017. – с. 137-138.
14. Гонсалес Р. Цифрова обробка зображень. / Р.С. Гонсалес, Р.Е. Вудс. - Москва: Техносфера, 2012. – 1104 с.
15. Давидова Н. А. Програмування [Електронний ресурс]: навчальний посібник / Н.А.Давидова, Є.В.Боровська. – М.: БИНОМ. Лабораторія знань, 2012. – 238 с.
16. Дьомін А. Основи комп'ютерної графіки: Навчальний посібник. / А.Ю. Дьомін – Томск: Изд-во Томского политехнического университета, 2011. – 191 с.
17. Зуева Е. Комп'ютерний зір в ім'я В.М.Келдиша ран – історія розвитку. /Е.Ю.Зуева - Математичні машини і системи, 2009, № 4, – С. 18-26.
18. Катаев.А Введення в комп'ютерне зір / А. Катаев, А. Алексеев - Singularis Lab, Ltd., Москва, 2016. – 40с.
19. Крилов В. С. Комп'ютерний зір: інноваційний віртуальний лабораторний практикум / В. С. Крилов - Науковий часопис НПУ імені М. П. Драгоманова. Серія 2 : Комп'ютерно-орієнтовані системи навчання. - 2012. - № 13. - С. 181-185.
20. Кручинін А. Розпізнавання образів з використанням OpenCV / А. Коручінін - 2011. – 171 с.
21. Кузін С. Алгоритми і методи попередньої обробки зображень. / С.Г. Кузін, А.Ф. Смірнов, Ю.Г. Васин Методичний посібник для виконання лабораторного практикуму – Новгород - ННГУ, 2001. – 10 с.
22. Липа В.В. Програмна інженерія: методичні рекомендації: підручник / В.В.Липа. – М. - Берлін: Директ-Медіа, 2015. – 608 с.

ДОДАТКИ

Додаток А

Таблиця А.1

Порівняння обгортки бібліотеки OpenCV

Технологія	EmguCV	OpenCV#	CV.NET	SharperCV	CodeProject
OpenCV 2.2	+	+	+	-	-
Навантаження GPU	+	+	+	-	-
Машинне навчання	+	+	+	-	+
Intel TBB	+	-	-	-	-
Intel IPP	+	-	-	-	-
Intel C++	+	+	-	-	-
Обробка винятків	+	+	-	-	-
Візуалізація відладки	+	+	+	-	-
Ліцензія	GPL	LGPL	Non-comm	Academic use	Non-comm
Підтримка Fedora	+	+	-	-	-
Підтримка Ubuntu	+	+	-	-	-
Активність проекту (оновлення не рідше ніж один раз на рік)	+	+	+	-	-

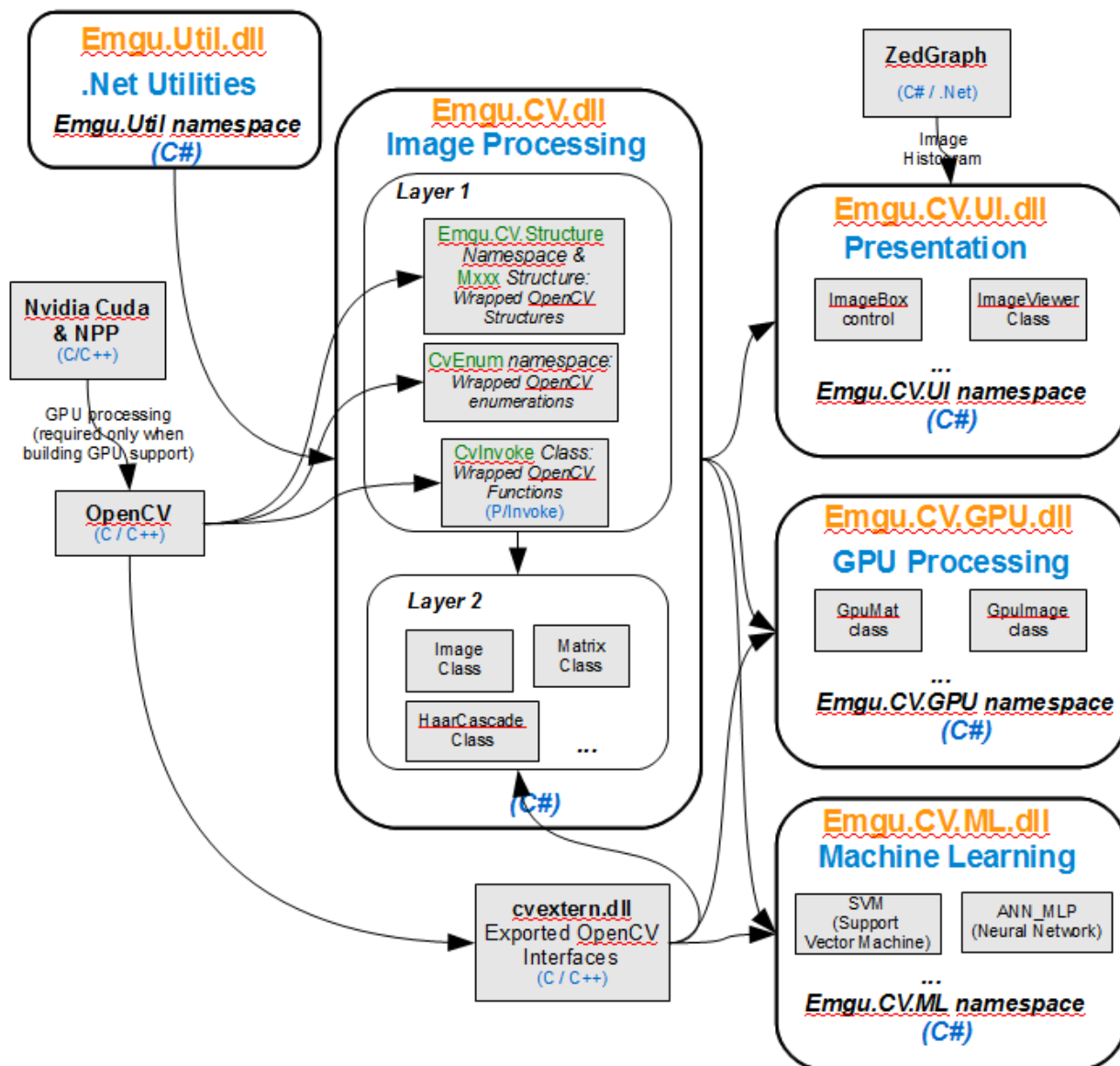


Рис. 2.1 Вигляд архітектури