



Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

Wydział Elektrotechniki, Automatyki,
Informatyki i Inżynierii Biomedycznej

Praca dyplomowa inżynierska

Wideo detekcja pojazdów samochodowych w ruchu
drogowym

Video-based vehicle detection for road traffic
applications

Autor:

Kierunek studiów:

Opiekun pracy:

Rafał Prusak

Automatyka i Robotyka

dr inż. Zbigniew Marszałek

Kraków, 15 listopada 2015

Oświadczam, świadomy(-a) odpowiedzialności karnej za poświadczenie nieprawdy, że niniejszą pracę dyplomową wykonałem(-am) osobiście i samodzielnie, i nie korzystałem(-am) ze źródeł innych niż wymienione w pracy.

.....

Spis treści

1. Wstęp	2
2. Metody pomiarów parametrów ruchu drogowego	3
3. Systemy wizyjne w zastosowaniach komercyjnych	5
3.1. NeuroSoft	5
3.2. Cat Traffic	5
4. Zastosowane narzędzia.	6
5. Opis programu	10
5.1. Instalacja	10
5.2. Struktura programu	10
5.3. Interfejs graficzny	11
5.4. Ustawienia	13
6. Algorytm detekcji i klasyfikacji pojazdów.	15
7. Analiza wyników	16
8. Rozwój oprogramowania	17
9. Podsumowanie	19
Bibliografia	19

1. Wstęp

Tutaj będzie wstęp do pracy inżynierskiej
Burzliwy rozwój nauki i techniki w XX wieku...

2. Metody pomiarów parametrów ruchu drogowego

W wyniku rozwoju motoryzacji oraz rozbudowy infrastruktury drogowej pojawiła się konieczność zbierania i analizy informacji o ruchu pojazdów.

Generalny Pomiar Ruchu Drogowego

W Polsce od roku 1965, co 5 lat, dokonywany jest Generalny Pomiar Ruchu Drogowego (GPRD). Ma on na celu pozyskanie charakterystyk ruchu drogowego na wszystkich odcinakach dróg krajowych i wojewódzkich. Uzyskiwane w wyniku GPRD pomiary są wykorzystywane przy planowaniu rozbudowy infrastruktury drogowej, organizacji ruchu, utrzymaniu dróg, określaniu przyczyn wypadkowości czy badaniu wpływu ruchu na komfort życia mieszkańców i środowisko naturalne.

W czasie pomiarów pojazdy są klasyfikowane do jednej spośród 8 kategorii:

1. motocykle,
2. samochody osobowe,
3. lekkie samochody ciężarowe,
4. samochody ciężarowe bez przyczepy,
5. samochody ciężarowe z przyczepami/ naczepami,
6. autobusy,
7. ciągniki rolnicze,
8. rowery.

Metodologię dokonywania pomiarów, terminy pomiarów, zadania obserwatorów, etc reguluje zarządzenie nr 38 Generalnego Dyrektora Dróg Krajowych i Autostrad z dnia 1 września 2014 r #dodać odnośnik do bibliografii

Parametry ruchu drogowego

Podczas dokonywania pomiarów dąży się do pozyskania zarówno indywidualnych informacji o pojazdach jak również ogólnej charakterystyki ruchu. Wyróżnia się w tym celu szereg parametrów:

- Prędkość - chwilowa, jazdy, podróży, miarodajna.
- Czas podróży - czas pokonania odcinka drogi.
- Natężenie ruchu - liczba pojazdów na jednostkę czasu.
- Gęstość ruchu - liczba pojazdów znajdujących się na odcinku drogi na jednostkę czasu.
- Zajętość drogi - procent drogi zajęty przez pojazdy.
- Bezpieczeństwo ruchu.
- Wpływ na środowisko.
- Zużycie paliwa.

Pomiarów ruchu drogowego dokonuje się w punkcie, na krótkim odcinku drogi, w zdefiniowanym obszarze, na całej trasie lub za pomocą pojazdu/pojazdów próbnych poruszających się po drodze. Rozróżnia się 4 typy ruchu pojazdów: jazda swobodna, jazda przy maksymalnym przepływie, jazda w warunkach dużej gęstości oraz zator.

3. Systemy wizyjne w zastosowaniach komercyjnych

Rozdział ten stanowi przegląd dostępnych na rynku systemów pomiaru ruchu drogowego, w których zastosowanie znalazły algorytmy wizyjne.

3.1. NeuroSoft

3.2. Cat Traffic

4. Zastosowane narzędzia.

Poniższy rozdział zawiera krótki opis języka programowania, w którym została zaimplementowana praca inżynierska, zastosowanych zewnętrznych bibliotek oraz innych narzędzi użytych przy wykonaniu programu.

Python

Algorytm wideo detekcji pojazdów został zaimplementowany w języku programowania Python, w wersji 3.4. Język ten posiada wiele korzystnych cech, które miały istotny wpływ na proces powstawania programu:

- Jest językiem obiektowym wysokiego poziomu. Obiektość kodu, w przeciwieństwie do takich popularnych języków jak Java czy C#, nie jest jednak odgórnie wymuszana przez składnię języka. Powoduje to, że w Pythonie można pisać proste liniowe skrypty, służące np. do testowania funkcjonalności programu.
- Gramatyka języka jest bardzo prosta, podobna do składni języka angielskiego, przez co kod jest jasny i zrozumiały. Dodatkowo język ten wymusza na programiście stosowanie standardu PEP8, co powoduje jednolity i przejrzysty styl kodu w powstającym programie.
- Posiada bogatą bibliotekę standardową zawierającą moduły do programowania sieciowego, pracy z plikami html, xml, json, tworzenia interfejsów gui czy przetwarzania wielowątkowego.
- Jest językiem interpretowanym, więc programy wykonane w tym języku są niezależne od platformy i mogą być łatwo przenoszone na inne systemy operacyjne bądź platformy sprzętowe (komputer PC, systemy embeded).
- Ogromna popularność języka Python i szeroka społeczność skupiona wokół niego, skutkuje w ogromnej bazie gotowych rozwiązań, porad i tutoriali dostępnych w sieci.

OpenCV

OpenCV jest biblioteką „open source” służącą do cyfrowego przetwarzania obrazu wizyjnego. Jest ona bezpośrednio zaimplementowana w języku C/C++. Dostępna jest na

wszystkich systemach operacyjnych stacjonarnych i mobilnych: Windows, Linux, MacOS, Android, iOS. Biblioteka zapewnia API dla wielu języków, takich jak: Python, Ruby, Java, Matlab. Została napisana z myślą o programach potrzebujących najwyższej wydajności i wykorzystaniu w aplikacjach czasu rzeczywistego. Najnowsze wydanie, w wersji 3.0, przynosi optymalizację biblioteki na wielu poziomach: algorytmów, wykorzystania wątków oraz rdzeni czy rozkazów dla CPU. Najwięksi producenci sprzętu komputerowego, tacy jak Intel czy Nvidia, także wzięli udział w tworzeniu biblioteki, dostarczając własne niskopoziomowe programy do wykonywania operacji na GPU.

Biblioteka jest podzielona na kilkanaście modułów, dostępnych w postaci bibliotek statycznych lub dynamicznych. Poniżej zostały omówione najważniejsze moduły, które znalazły zastosowanie w implementacji pracy inżynierskiej:

- core - podstawowe funkcje do działań na tablicach wielowymiarowych,
- highgui - obsługa graficznego interfejsu użytkownika, wyświetlanie zdjęć i wideo ,
- imgproc - funkcje do przetwarzania obrazu: filtracja, transformacje geometryczne, konwersje między przestrzeniami barw,
- imgcodecs - interfejs do odczytu i zapisu zdjęć,
- videoio - przechwytywanie obrazu wideo,
- video - analiza obrazu wideo(śledzenie ruchu, wyodrębnianie tła),
- features2D - znajdowanie krawędzi, dopasowywanie szablonów,
- objectdect - dopasowywanie i wykrywanie obiektów.

OpenCv znalazło wiele zastosowań w aplikacjach przemysłowych, takich jak: inspekcja gotowych produktów w fabrykach, medycyna, bezpieczeństwo, systemy wizyjne „wielowymiarowe”, robotyka.

NumPy

NumPy jest zewnętrzną biblioteką „open source”, służącą do przeprowadzania obliczeń na n-wymiarowych tablicach. Zapewnia ona funkcjonalność podobną do środowiska Matlab. Podstawowa typ tablicy, używany przez bibliotekę znacząco różni się od tablicy(listy) samego Pythona: lista jest kontenerem heterogenicznym, mogącym dynamicznie zmieniać swój rozmiar, natomiast „ndarray” jest tablicą homogeniczną o stałym rozmiarze. Rozwiązanie takie zapewnia wysoką wydajność obliczeń i możliwość bezpośredniej integracji z plikami binarnymi biblioteki napisanymi w języku C/C++.

SciPy

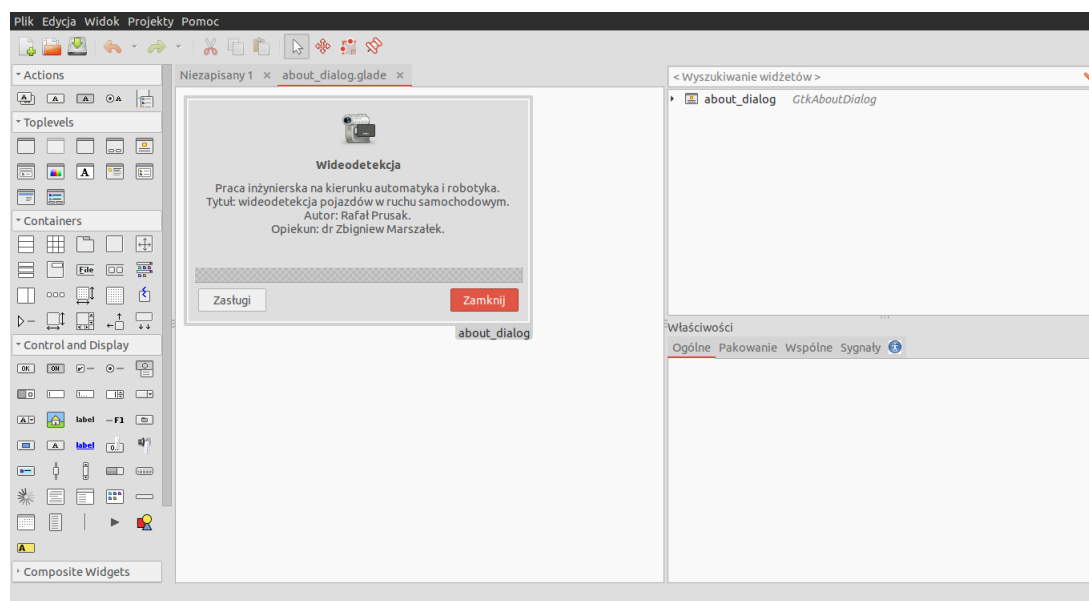
Biblioteka SciPy (skrót od „Scientific Python”) jest biblioteką „open-source” zawierającą funkcje do rozwiązywania zadań optymalizacji, algebry liniowej, przetwarzania sygnałów czy rozwiązywania równań różniczkowych. Biblioteka pozwala zrealizować większość funkcjonalności środowiska Matlab.

PyGtk + Glade

W celu zaprojektowania i stworzenia interfejsu graficznego użytkownika zostało wykorzystane PyGtk oraz program Glade.

PyGtk jest nakładką na bibliotekę Gtk+, pozwalającą na tworzenie graficznego interfejsu użytkownika przy użyciu języka Python. Znalazła ona zastosowanie w wielu popularnych aplikacjach: BitTorrent, Gedit, GIMP. Jest częścią linuksowego środowiska graficznego GNOME.

Glade jest programem do projektowania interfejsu graficznego użytkownika z wykorzystaniem „widgetów” z biblioteki Gtk. Stworzony projekt okna interfejsu jest zapisywany do pliku xml. Spreparowany w ten sposób plik, może być wykorzystany do zbudowania okna za pomocą klasy GtkBuilder.



Rysunek 4.1. Widok okna programu Glade

Sphinx

Sphinx jest to narzędzie służące do automatycznej generacji dokumentacji programu na podstawie komentarzy zawartych w kodzie. Program ten pozwala na tworzenie dokumentacji programów napisanych w Pythonie bądź C++. Aby dokumentacja mogła zostać

wygenerowana, komentarze muszą być zgodne ze standardem „reStructuredText”. Dokumentacja może być wygenerowana w postaci tekstu, htmla, w formacie pdf czy Latex.

Tutaj dać screen dokumentacji

Pycharm

Pycharm jest środowiskiem deweloperskim dla języka Python, tworzonym przez firmę JetBrains. Zapewnia ono szereg przydatnych funkcji, takich jak: podpowiadanie składni, detekcję błędów, proponowanie poprawek i rozbudowany debugger. Działa zgodnie, ze wszystkimi popularnymi implementacjami języka Python: CPython, IronPython, Jython. Środowisko oferuje wsparcie dla innych technologii, np. Html5, Css3, JavaScript, Angular.js. Za pomocą Pycharma programista może korzystać z frameworków webowych takich jak Django czy Flask. Środowisko wspiera również systemy kontroli wersji(Git, SVN), posiada wsparcie dla narzędzi do pracy z bazami SQL oraz umożliwia zdalne edytowanie kodu w oparciu o protokół SSH. Firma JetBrains udostępnia 3 wersje IDE: profesjonalną(płatną),

5. Opis programu

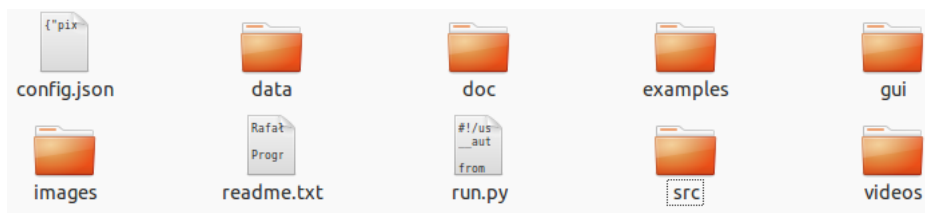
Poniższy rozdział opisuje interfejs i sposób działania zaimplementowanego programu.

5.1. Instalacja

Tutaj będzie opisany sposób instalacji programu, gdy ten zostanie doprowadzony do końca.

Instalacja z repozytorium githuba + instalacja bibliotek .

5.2. Struktura programu



Rysunek 5.1. Widok struktury programu

Program został podzielony na następujące pliki oraz foldery:

- config.json - plik konfiguracyjny programu, zawiera parametry algorytmów oraz opcje dotyczące prezentacji wyników programu,
- data - folder do którego program zapisuje dane o wykrytych pojazdach do baz danych SQLite,
- doc - dokumentacja programu w formacie .html wygenerowana przy użyciu narzędzia Sphinx,
- examples - spreparowane nagrania testowe,
- gui - pliki w formacie .glade używane do wyświetlania interfejsu graficznego użytkownika,
- images - zdjęcia wykrytych samochodów, opisane danymi samochodu

- readme.txt - plik tekstowy zawierający informacje o potrzebnych do uruchomienia programu bibliotekach i sposobie ich instalacji,
- run.py - skrypt uruchamiający program,
- src - pliki źródłowe programu,
- videos - zapisane przez program wyniki analizy w postaci plików wideo.

5.3. Interfejs graficzny

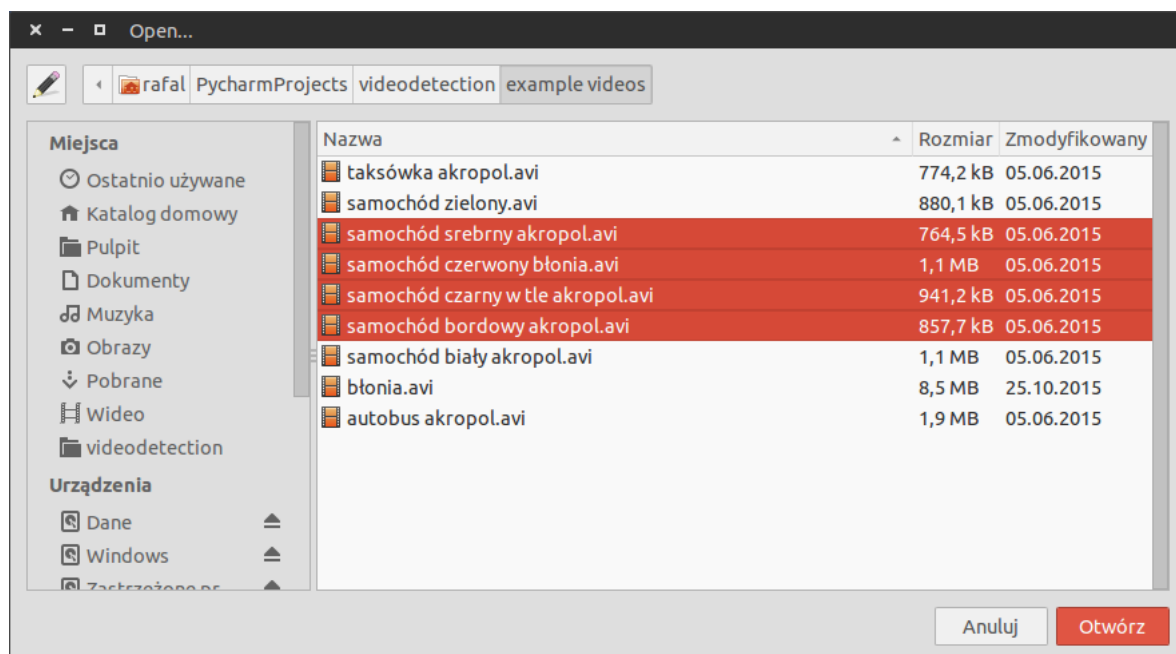
Po uruchomieniu programu skryptem z pliku run.py pojawia się główne okno programu. Ma ono początkowo postać zwykłego paska złożonego z przycisków.



Rysunek 5.2. Główne okno programu

Główne okno programu zapewnia następujące funkcje:

- „Otwórz plik wideo” - otwiera poniższe okno dialogowe, umożliwiające wybór jednego bądź większe ilości plików wideo do analizy.



Rysunek 5.3. Okno wyboru plików

- „Otwórz obraz z kamery” - uruchamia podprogram, który zajmuje się analizą obrazu z kamery w czasie rzeczywistym ### dodać screen podprogramu

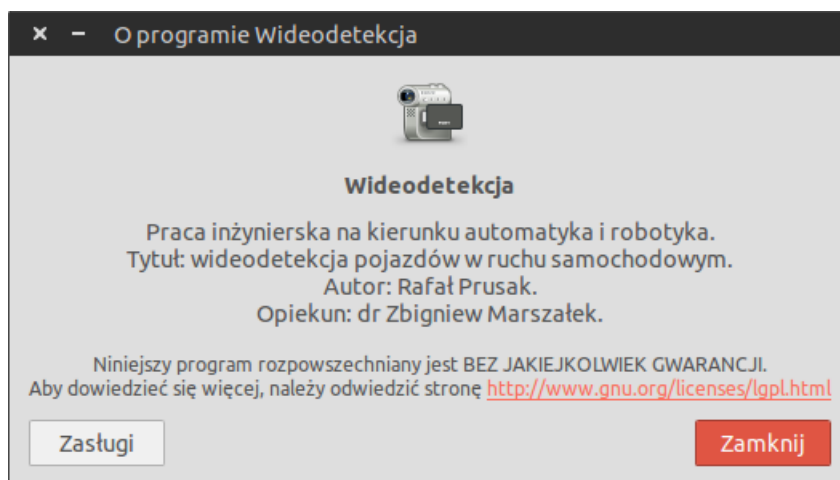
- „Otwórz plik bazy danych” - otwiera wskazaną bazę danych i prezentuje zawarte w niej dane o pojazdach. ### dodać screen
- „Przeglądaj obrazy” - włącza przeglądarkę zapisanych przez program zdjęć pojazdów.



Rysunek 5.4. Widok przeglądania obrazów

- „Usuń poprzednie dane” - usuwa zdjęcia, filmy i dane zapisane przez poprzednie sesje programu.
- „Włącz/wyłącz algorytm” - umożliwia wyłączenie algorytmu. Program dokonuje wtedy jedynie detekcji poruszających się obiektów bez klasyfikowania oraz logowania danych do bazy danych. Opcja wyłączenia algorytmu może być przydatna w sytuacji, w której użytkownik chciałby dokonać kalibracji ustawień.
- „Odtwarzaj wczytane pliki” - rozpoczyna wyświetlanie i analizę uprzednio wybranych plików wideo.
- „Zatrzymaj odtwarzanie” - zatrzymuje odtwarzanie obecnego pliku.
- „Zakończ odtwarzanie” - kończy odtwarzanie i analizę wszystkich plików.
- „Włącz/wyłącz nagrywanie” - umożliwia nagrywanie obrazu poddanego analizie do pliku wideo.
- „Ustawienia” - pokazuje okno ustawień programu. Okno ustawień zostało szczegółowo opisane w następnym podrozdziale.

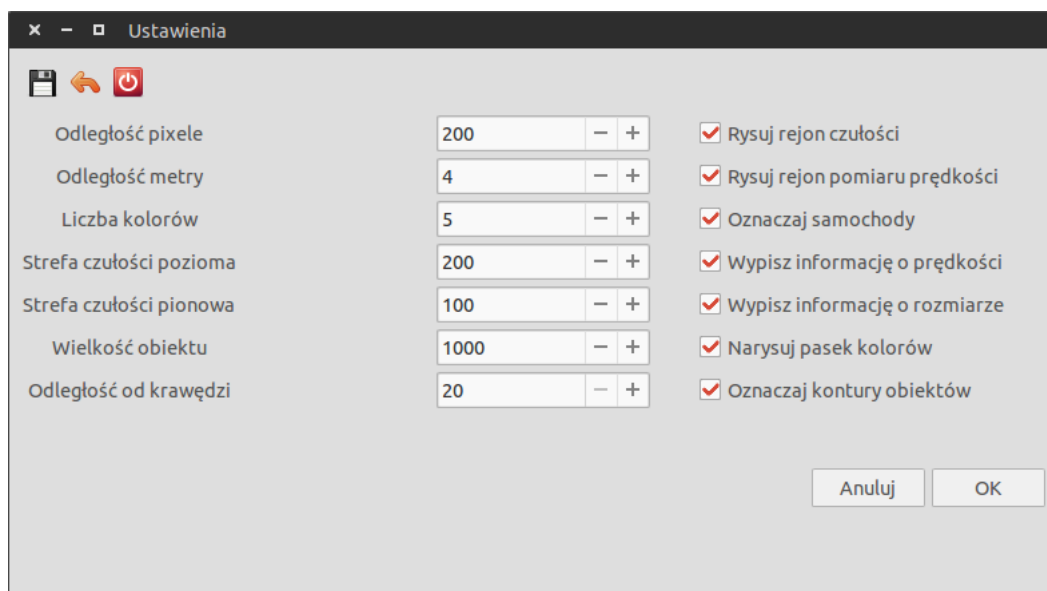
- „Dokumentacja” - uruchamia stronę html z dokumentacją.
- „O programie” - uruchamia okienko dialogowe pokazujące krótką informację o programie.



Rysunek 5.5. Informacja o programie

- „Zakończ” - wyłącza program.

5.4. Ustawienia



Rysunek 5.6. Okno ustawień programu

Okno ustawień umożliwia dostosowywanie parametrów algorytmu oraz

- Odległość pixele - wartość tego pola służy do wyliczenia przelicznika pixeli na obrazie na rzeczywistą odległość w metrach.

- Odległość metry - wartość tego pola ma również zastosowanie w przeliczaniu pixeli na metry. Procedura kalibracji pomiaru została opisana w poprzednim rozdziale
- Liczba kolorów - określa liczbę kolorów jaką program ma rozpoznać na obiekcie.
- Strefa czułości pozioma - określa ona obszar w poziomie, w którym obiekty są wykrywane i śledzone.
- Strefa czułości pionowa - określa ona obszar w pionie, w którym obiekty są wykrywane i śledzone.
- Wielkość obiektu - obiekt o mniejszej wartości są ignorowane po fazie binaryzacji
- Odległość od krawędzi

6. Algorytm detekcji i klasyfikacji pojazdów.

W tym rozdziale zostały opisane kolejne etapy zaimplementowanego algorytmu.

7. Analiza wyników

8. Rozwój oprogramowania

Nie zrealizowane funkcjonalności

W trakcie realizacji pracy inżynierskiej nie udało się autorowi zrealizować wszystkich pierwotnych założeń.

W trakcie pracy nad algorytmem okazało się, że detekcja pojazdów w ruchu na drodze dwupasmowej jest zadaniem bardzo trudnym, jeżeli program ma do dyspozycji jedynie obraz wideo z pobocza drogi. Zachodzi wtedy wiele „niekorzystnych” zjawisk:

- Mijanie się samochodów - w tej sytuacji dochodzi do zlewania się obiektów w jeden na obrazie binarnym a pojazdy mogą dotrzeć do przeciwnych końców drogi w odwrotnej kolejności. W celu zabezpieczenia programu przed wynikającymi z tej sytuacji przekłamaniami, należałoby dokonać poważnych zmian w algorytmie: zamiast pobierać informację o pojeździe na końcach obszaru „czułości” kamery, program musiałby śledzić obiekt w całym polu, wykryć kierunek ruchu i nadać mu szacowaną prędkość (przyrost pozycji w pixelach na kratkę obrazu).
- Wyprzedzanie się pojazdów - ta sytuacja może powodować, że dwa obiekty staną się dla programu jednym, jeżeli manewr zakończy się po za polem widzenia kamery. Skutkiem tej sytuacji będzie całkowita utrata wiadomości o obiekcie.
- Wzajemne rzucanie cienia i odbicia światła od samochodów - Powoduje to zniekształcenia na obrazie po wyodrębnianiu tła, takie jak połączenia się dwóch pojazdów w jeden w skutek odbicia światła, albo podział pojazdu na dwa niezwiązane ze sobą obiekty.

Rozwiązaniem powyższych problemów mogłoby być całkowite przemodelowanie algorytmu: odejście od prostej, ale wydajnej metody wyodrębniania tła na rzecz bardziej skomplikowanych algorytmów np. „cam shift”, „mean shift” czy „optical-flow”.

Kolejnym założeniem było zrealizowanie funkcjonalności do pracy z obrazem ruchu drogowego „on-line” streamowanym w sieci Internet. Niepowodzenie było spowodowane dwoma faktami :

- Znacząca większość dostępnych w sieci obrazów z kamer monitorujących filmuje ruch drogowy z pozycji „nad drogą” a nie z pobocza drogi.

- Udostępniany w sieci obraz jest obrazem w skali szarości, niskiej jakości i o bardzo małej ilości klatek, rzędu 2-3 na sekundę, co utrudnia detekcję poruszających się obiektów.

Przyszłe zmiany w programie

W ciągu dalszego rozwoju oprogramowania możliwe byłyby następujące zmiany:

- Podział kodu na klasy i metody operujący na tych samych „poziomach” abstrakcji - kod programu miejscami jest niezrozumiały, kilka metod jest zdecydowanie za długie, mieszają się operacje o różnej abstrakcji, co skutkuje niestety w trudnościach w modyfikacji i dodawaniu do programu nowych funkcjonalności.
- Dodanie obsługi baz MySQL - dodanie tej funkcjonalności umożliwiłoby automatyczne dokonywanie pomiarów „w terenie” po przez użycie mini-komputera klasy RaspberryPi i kamery, natomiast obserwacja wyników mogłaby odbywać się zdalnie za pomocą strony WWW.
- Dodatkowe opcje - ustawienia programu należało by poszerzyć o parametry wszystkich algorytmów.
- Zmiany w GUI - istnieje możliwość dodania do gui rysowania wykresów/opisywania statystyk przejechanych pojazdów.

9. Podsumowanie

A tutaj będzie podsumowanie pracy inżynierskiej, gdy już napiszę to gównio

Spis rysunków

4.1	Widok okna programu Glade	8
5.1	Widok struktury programu	10
5.2	Główne okno programu	11
5.3	Okno wyboru plików	11
5.4	Widok przeglądania obrazów	12
5.5	Informacja o programie	13
5.6	Okno ustawień programu	13

Bibliografia