



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE
Wydział Fizyki i Informatyki Stosowanej

Analiza i przetwarzanie obrazów

Zliczanie samochodów

Łukasz Ujma

Wydział Fizyki i Informatyki Stosowanej
Informatyka stosowana

Kraków, rok akademicki 2016/2017

Spis treści

1	Wstęp.....	3
2	Założenia projektu.....	3
3	Opis programu	3
4	Wnioski	5

1 Wstęp

Celem projektu było napisanie programu który wykonałby analizę przekazanego wideo po kątem wykrycia, śledzenia i zliczania przejeżdżających samochodów. Tak wykonany projekt może zostać wykorzystany do monitorowania i tworzenia statystyk ruchliwości danej ulicy.

2 Założenia projektu

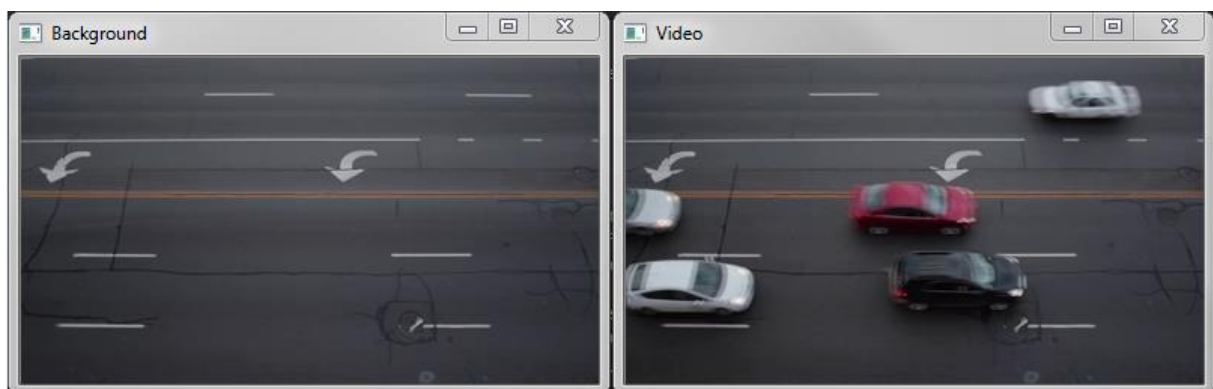
Głównym założeniem projektu było wyodrębnienie samochodów na przekazanym wideo. Na każdej klatce samochody wykrywane są niezależnie, dlatego wymogiem jest również korelowanie ich pozycji między kolejnymi klatkami. Dodatkowo inne ruchome elementy muszą być odróżniane od samochodów by otrzymać poprawny wynik.

3 Opis programu

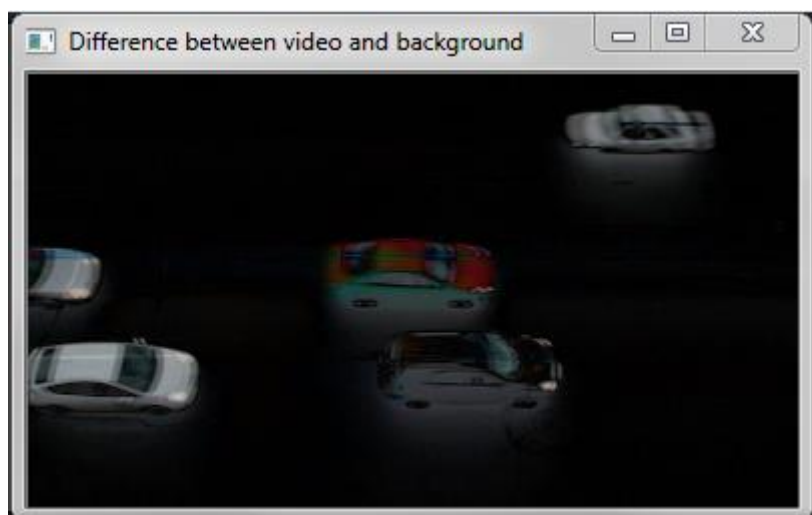
Projekt został zrealizowany przy użyciu Visual Studio (C++) i biblioteki OpenCV 3.2.

Cała funkcjonalność znajduje się w pliku main.cpp i w dwóch dodatkowych klasach BackgroundSubtraction.cpp i MyPoint.cpp.

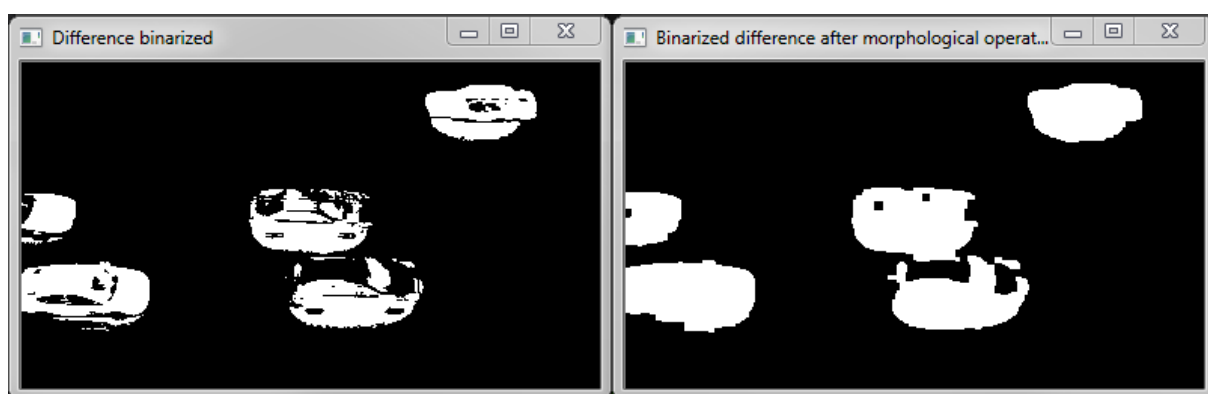
W pierwszym kroku wczytywane do pamięci jest wideo z podanej ścieżki z którego następnie ekstraktowane jest tło. Do wygenerowania tła wykorzystywana jest przygotowana klasa BackgroundSubtraction która korzystając z funkcji OpenCV accumulateWeighted uśrednia klatki wynikiem czego jest ulica bez samochodów.



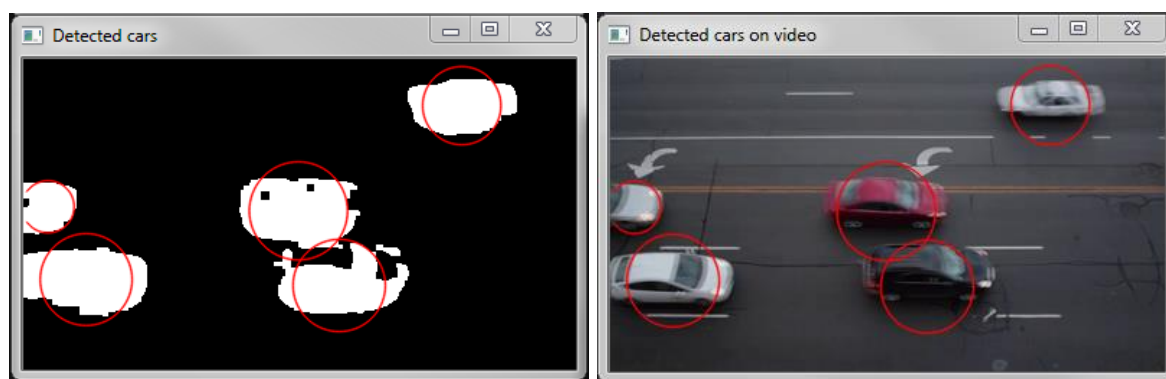
Kolejnym krokiem jest wyodrębnienie samochodów z tła. W tym celu wykorzystano prostą różnicę między aktualną klatką a tłem.



Tak otrzymany obraz poddawany jest prostemu progowaniu i operacjom morfologicznym by wyznaczyć tylko ruchome elementy obrazu. Wartości progowania i operacji morfologicznych mocno zależą od jakości wideo, oświetlenia itd., z tego powodu zaimplementowano możliwość ich modyfikacji za pomocą suwaków.



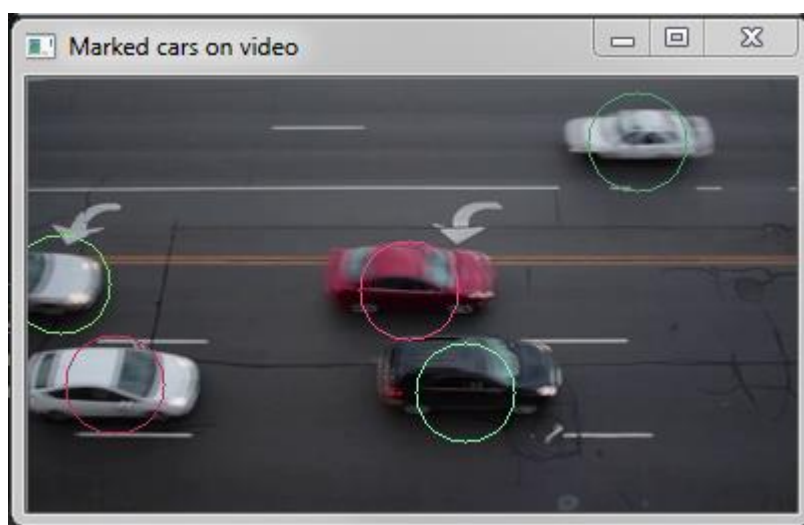
Na zbinaryzowanym obrazie można wykorzystać SimpleBlobDetector, dzięki któremu dostajemy informacje o białych plamach na obrazie w tzw. keyPoint. Przekazując odpowiednie parametry do SimpleBlobDetector eliminujemy wykrywanie elementów zbyt małych, zbyt dużych lub o złym kształcie. Funkcja drawKeypoints zaznacza na obrazie wykryte elementy.



Korzystając z otrzymanego wektora keyPoints w którym zapisane są współrzędne środków wykrytych obiektów tworzony jest wektor wektorów punktów path. W path przechowywane i odpowiednio kategoryzowane są wszystkie wykryte punkty. Pierwszym wymiarem tego wektora są

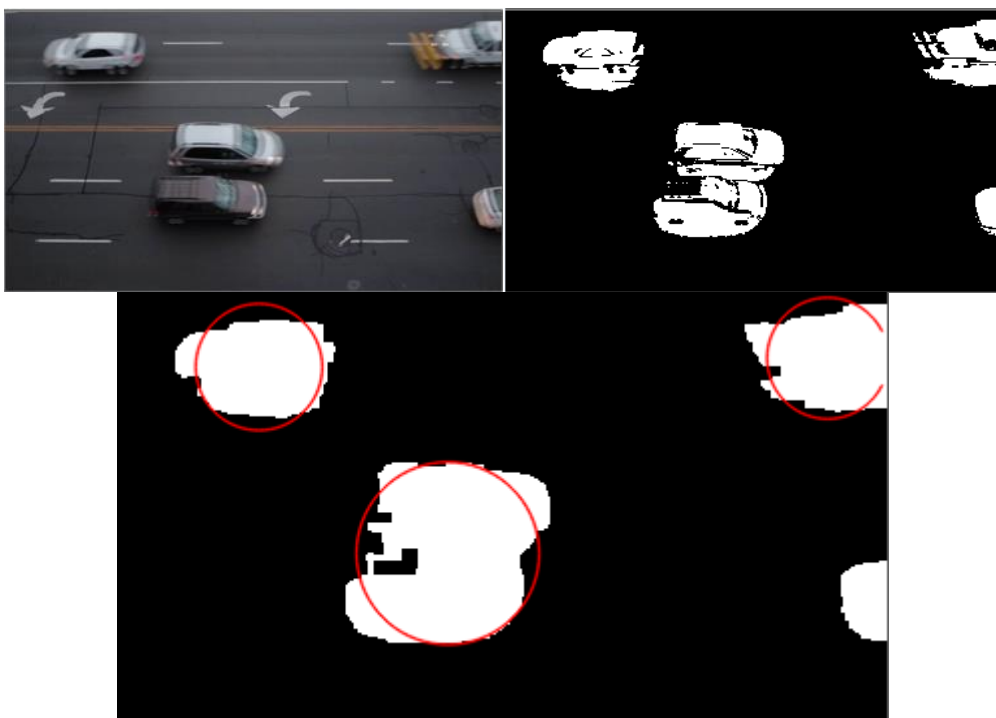
kolejne wykryte ścieżki/samochody, natomiast drugi wymiar odpowiada za i-ty punkt na trasie danego samochodu. To czy wykryty punkt zostanie dodany na końcu istniejącej ścieżki czy stworzona zostanie nowa ścieżka determinowane jest przez odległość od ostatnich punktów na ścieżkach. Jeśli odległość danego keyPointu jest większa od wybranej na suwaku oznacza to, że pojawił się nowy samochód na obrazie i tworzona jest nowa ścieżka. W przypadku gdy dana ścieżka nie została zaktualizowana przez wybraną ilość klatek sugeruje to, że pojazd wyjechał poza kadr, w takim przypadku ta ścieżka zostaje pomijana przy sprawdzaniu kolejnych punktów.

W ostatnim kroku na obrazie zaznaczane są wykryte samochody funkcją circle oraz wypisywana jest ilość wykrytych samochodów w konsoli. Każdemu nowemu samochodowi przypisywany jest losowy kolor symbolizujący ścieżkę do której przypisywane są jego współrzędne. (Na poniższym obrazie każdy zaznaczony samochód ma przypisany inny kolor. Niefortunna losowa generacja kolorów.)



4 Wnioski

Razem z projektem zamieszczono kilka nagrań na których testowano aplikację. W większości przypadków program działał prawidłowo i odpowiednio wykrywał i zliczał samochody. Pojawił się tylko jeden problem którego nie udało się rozwiązać. Z obrazu wyodrębniany jest również cień samochodu, ponieważ w tym miejscu obraz znacząco różni się od uśrednionego tła. W przypadku gdy dwa samochody poruszają się równolegle wyodrębniony cień powoduje, że dwa samochody po binaryzacji zlewają się w jeden przez co końcowy wynik nie jest poprawny.



Ten problem prawdopodobnie można rozwiązać zmieniając algorytm wyróżniania samochodów od tła (czego podjęto bezskuteczną próbę), bądź wykorzystując metodę typu Watershed. Niestety nie udało się naprawić tego błędu w projekcie.

Przedstawiony projekt pokazuje jak skomplikowanym zagadnieniem jest analiza obrazów. Przed praktycznym wykorzystaniem tej aplikacji należałoby poprawić parę elementów takich jak przedstawiony wcześniej problem z cieniami. Przykładowym rozwiązaniem mogłoby być ustawienie kamery bezpośrednio nad ulicą.

Projekt można w przyszłości rozwinąć o dodatkowe funkcjonalności. W pierwszej kolejności na myśl przychodzi obliczanie i wyświetlanie prędkości aut na podstawie ich współrzędnych.