POLITECHNIKA WARSZAWSKA WYDZIAŁ ELEKTRYCZNY INSTYTUT STEROWANIA I ELEKTRONIKI PRZEMYSŁOWEJ

PRACA DYPLOMOWA MAGISTERSKA na kierunku Automatyka i Inżyniera Komputerowa



Michał ZUBRZYCKI Nr albumu: 214964

> Rok akad.: 2011/2012 Warszawa 06.03.2012

ANALIZA RUCHU POJAZDÓW W SEKWENCJACH OBRAZÓW

Zakres pracy:

- 1. Wprowadzenie i sformułowanie celu pracy
- 2. Analiza wybranych metod wykrywania ruchu, śledzenia i rozpoznawania obiektów
- 3. Weryfikacja skuteczności badanych metod na rzeczywistych sekwencjach obrazów
- 4. Podsumowanie i wnioski

Kierujący pracą:

WOOLS

dr inż. Witold Czajewski

Termin wykonania: 2012.09.15 Praca wykonana i zaliczona pozostaje własnością Instytutu, Katedry i nie będzie zwrócona wykonawcy. (Podpis i pieczątka

Kierownika Zakładu Dydaktycznego)

Meling

Temat pracy: Analiza ruchu pojazdów w sekwencji obrazów

Streszczenie

Celem pracy jest porównanie ogólnie dostępnych metod wykrywania i śledzenia ruchu pojazdów w sekwencjach obrazów. Jako materiał źródłowy wykorzystano filmy przedstawiające ruch samochodowy na drogach szybkiego ruchu i w środowisku miejskim. Dzięki napisanemu przez autora oprogramowaniu oznaczono na wybranych filmach pojazdy, tworząc w ten sposób dane rzeczywiste, do których będą porównywane wyniki działania badanych metod.

W pierwszej części pracy opracowano porównanie metod wykrywania ruchu zawartych w bibliotece BGS. Zawiera ona 32 różne metody, różniące się stosowanymi modelami matematycznymi, oparte o powszechnie dostępne artykuły naukowe. Sprawdzono działanie algorytmów reprezentujące najpopularniejsze rozwiązania. Porównano ich wyniki zarówno ze względu na precyzję wykrywania ruchomych obiektów, fragmentaryzację wykrytych konturów, ilość generowanych szumów, jak i szybkość działania. Z powodu dużej liczby aspektów branych pod uwagę, wybrano trzy algorytmy: najszybszy, najskuteczniejszy, ale jednocześnie najwolniejszy oraz pośredni. Dodatkowo sprawdzono jaki wpływ ma zmiana rozdzielczości sekwencji wejściowych na jakość wykrywania w nich ruchu.

W drugiej części pracy zostały zbadane metody śledzenia wykrytych wcześniej ruchomych obiektów. Najpierw sprawdzono te zawarte w bibliotece OpenCV. Niestety nie były one w stanie stabilnie śledzić poruszających się obiektów np. w przypadku ich mijania się. W związku z tym zaproponowano, a następnie zaimplementowano i porównano dwie inne metody śledzenia ruchu. Pierwsza najprostsza wykorzystuje jedynie pozycje wykrytych konturów. By poprawić jej działanie stworzono drugą metodę, wykorzystującą filtr Kalmana jako predyktor. W porównaniu do metod wcześniej badanych, użycie predyktora spowodowało znaczną redukcję błędów i dokładniejsze śledzenie poruszających się obiektów. Zniwelowało także problem łączenia się konturów różnych wykrytych pojazdów ze sobą. Sprawdzono także wpływ rozdzielczości na jakość śledzenia wykrytych ruchomych obiektów.

W trakcie przeprowadzonych badań zrealizowano cel pracy poprzez wybór najskuteczniejszych algorytmów wykrywania ruchu i zaproponowanie dokładniejszych metod jego śledzenia.

Thesis subject: Vehicle motion analysis in image sequences

Abstract

The objective of this thesis is the comparison of publicly available methods of detecting and tracking movement of vehicles in image sequences. Clips of car traffic in highway and urban areas are used as the input material. A program written by the author was used to mark vehicles in selected clips creating the ground-truth data that will be compared with the results of evaluated methods.

The first part of the thesis describes the comparison of movement detection algorithms included in the BGS library. It contains 32 different methods, based on freely available scientific papers. These methods use different mathematical models. They were tested both for the precision of moving object detection, fragmentation of the detected contours, amount of generated noise, as well as the processing speed. Because of the large number of the aspects taken into consideration, three methods were selected: the fastest, the most accurate but the slowest one and one in between. Additionally, the methods sensitivity to image resolution was verified.

In the second part of the thesis algorithms for tracking of the previously detected moving objects were tested - initially by the methods included in the OpenCV library. Unfortunately, they were unable to provide the accurate tracking of moving objects, for example when these objects pass by each other. Due to the above, two methods were proposed and implemented. First, a simple one, using only position of detected contours. In order to improve its performance, the second method was created. It uses Kalman filter as a predictor. The second algorithm, in comparison to previously tested ones, resulted in significant error reduction and more accurate tracking of moving objects. It also reduced errors caused by connecting of multiply contours generated by different moving objects. Additionally, the methods sensitivity to image resolution was analyzed.

In the course the conducted research, the objective of this thesis has been achieved by selecting the most effective algorithms of motion detection and proposing more precise methods of tracking.

Warszawa, 26.11.2013

OŚWIADCZENIE

Świadomy odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa magisterska pt. "Analiza ruchu pojazdów w sekwencji obrazów"

- została napisana przeze mnie samodzielnie

- nie narusza niczyich praw autorskich

nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.
 Oświadczam, że przedłożona do obrony praca dyplomowa nie była wcześniej podstawą postępowania związanego z uzyskaniem dyplomu lub tytułu zawodowego w uczelni wyższej.

Jestem świadom, że praca zawiera również rezultaty stanowiące własności intelektualne Politechniki Warszawskiej, które nie mogą być udostępniane innym osobom i instytucjom bez zgody Władz Wydziału Elektrycznego.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Imię i Nazwisko dyplomanta: Michał Zubrzycki Podpis dyplomanta: <u>Michał Zubrzycki</u>

Spis treści

1.Wstęp	1
1.1.Cel pracy	2
1.2.Układ pracy	2
2.Obszary zastosowania i podział metod wykrywania ruchu	4
2.1.Obszary zastosowania	4
2.1.1.Liczenie samochodów	4
2.1.2.Kontrola i pobieranie opłat	4
2.1.3. Wykrywanie wypadków i łamania przepisów	4
2.2.Podział metod wykrywania ruchu.	5
3. Wprowadzenie do badań	7
3.1.Etapy badań	7
3.2.Sprzet i oprogramowanie	7
3.2.1. Sprzet	7
3.2.2.Biblioteka OpenCV	8
3.2.3.Biblioteka BGSLibrary	9
3.2.4.Materiał wideo.	11
3.2.5. Stworzone oprogramowanie.	14
4. Badania metod wykrywania ruchu	
4.1. Wstep	17
4.2. Metoda badania algorytmów wykrywania ruchu	18
4.3. Analiza danych	
4 4 Wyniki badania wybranych metod	22
4.4.1 Metoda różnicowa	
4.4.2. AdaptiveMedianBGS	24
4.4.3. Weighted Moving Mean BGS.	
4.4.4.WeightedMovingVarianceBGS	
4 4 5 DPMeanBGS	34
4.4.6.GrimssonGMM	
4.4.7 LBMixtureOfGaussians	
4.4.8 PixelBasedAdaptiveSegmenter	
4.4.9. MultiLaverBGS	
4.5. Zestawienie wyników	
4.6. Wykrywanie ruchu na skrzyżowaniu	60
4.7.Wpływ rozdzielczości	63
5. Badanie algorytmów śledzenia obiektów	69
5.1.Śledzenie obiektów w OpenCV	69
5.2. Wykrywanie konturów.	70
5.3. Metoda prosta	71
5.4. Problemy występujące przy śledzeniu	72
5.4.1. Ślędzenie pieszych	73
5.4.2. Łaczenie się różnych konturów	73
5.4.3.Cienie	74
5.4.4.Zatrzymywanie sie pojazdów	74
5.4.5.Duże pojazdy	75
5.5.Śledzenie wspomagane filtrem Kalmana	75
5.6.Porównanie działania obu metod.	79

5.6.1.Obiekt nr 8 – człowiek	80
5.6.2.Obiekt nr 12 – skręcający samochód	80
5.6.3.Obiekt nr 22- jadący rowerzysta	82
5.6.4.Obiekt nr 28 – samochód poruszający się z zmienną prędkością	83
5.6.5.Obiekt nr 50 – samochód ciężarowy poruszający się z zmienną prędkościa	ı. 84
5.7. Wpływ rozdzielczości na śledzenie obiektów	86
6.Podsumowanie i wnioski	89
Literatura:	91

1. Wstęp

W ostatnich latach zauważa się znaczny wzrost liczby kamer stosowanych do monitorowania zarówno dróg szybkiego ruchu, jak i obszarów miejskich. Służa one do obserwacji przemieszczających się osób i pojazdów. Coraz większa liczba kamer oznacza wzrost ilości danych analizowanych przez operatorów systemów monitoringu. Powoduje zatrudniania większej liczby to konieczność pracowników lub stworzenia oprogramowania wspierającego ich pracę. Dzięki rosnącej mocy obliczeniowej układów procesorowych oraz postępowi w technikach analizy obrazów, automatyczne metody wykrywania i śledzenia obiektów, wykazują się coraz większą skutecznością i są coraz częściej stosowane. Głównym celem tych programów jest wsparcie i/lub zastąpienie operatorów systemów monitoringu w obserwowaniu i analizowaniu obrazów z kamer.

Miejsce zastosowania systemu monitoringu w znacznym stopniu wpływa na złożoność problemu automatycznego wykrywania ruchu. Ze względu na obszar objęty monitoringiem, rozróżnia się dwa podstawowe rodzaje miejsc obserwowanych: drogi szybkiego ruchu (autostrady) i obszary miejskie. Analiza ruchu pojazdów wewnątrz miast jest trudniejsza i bardziej złożona. Obrazuje to porównanie skuteczności wykrywania i klasyfikacji pojazdów. Według [2], skuteczność rożnych algorytmów wykrywania ruchu w przestrzeni miejskiej wynosi od 61% do 85%, a na autostradzie od 89% do 95%. Na spadek skuteczności metod wykrywania i śledzenia ruchu w miastach wpływa fakt obserwowania bardziej skomplikowanego otoczenia, ruchu nie tylko w jednym kierunku (skrzyżowania, parkowanie, zakręty), a także udział w ruchu pieszych i cyklistów.

Głównym problemem wykorzystania danych z istniejącego monitoringu jest zazwyczaj niska jakość obrazu wyjściowego z kamer w nim zastosowanych, co skutkuje ograniczoną ilością szczegółów umożliwiających wykrycie i rozróżnienie poruszających się obiektów. Dodatkowo monitoring wymaga przetwarzania informacji w czasie rzeczywistym, co ogranicza złożoność algorytmów. Zwykle_stosowane są kamery bez możliwości obrotu, jednak coraz częściej montowane są kamery ruchome. Wprowadzają one kolejne czynniki utrudniające skuteczne wykrywanie i śledzenie pojazdów. Są one sterowane przez operatora lub poruszają się zgodnie z zadanym programem. Obejmują one większy obszar, zatem algorytmy wykrywania ruchu wymagają tworzenia bardziej skomplikowanych modeli obszaru, który monitorują. Oprócz tego program musi rozpoznawać, na który fragment obszaru jest w danej chwili skierowana kamera np. poprzez rozpoznanie punktów charakterystycznych lub informacje o kącie obrotu kamery. Innym czynnikiem utrudniającym skuteczne wykrywanie i śledzenie pojazdów, są zmienne warunki pogodowe i oświetleniowe. Algorytmy powinny być skuteczne zarówno w słoneczny dzień, jak i w nocy, przy sztucznym oświetleniu oraz w przypadku rożnych warunków atmosferycznych np. występowania mgły, opadów deszczu lub śniegu.

Systemy automatycznego monitorowania ruchu mogą być wspomagane przez inne czujniki niż tylko kamery. Najczęściej stosowane są pętle indukcyjne, stacje pogodowe, czujniki laserowe, radarowe i czujniki pracujące w zakresie podczerwieni lub mikrofalowym. Istnieje bardzo duży potencjał w jednoczesnym przetwarzaniu danych z różnych źródeł [2]. Jednak zbieranie dodatkowych danych wymaga zamontowania innego rodzaju sensorów, co podnosi koszty systemów i zwiększa złożoność oprogramowania. Dlatego dąży się do skutecznego wykorzystania tylko sekwencji obrazów z kamer.

1.1. Cel pracy

Podstawowym celem pracy jest analiza i porównanie różnych metod wykrywania ruchu w sekwencjach obrazów. Założeniem jest zastosowanie nieruchomej kamery i metod wykorzystujących jako danych wejściowych jedynie sekwencje obrazów z pojedyńczej kamery, bez dodatkowych danych z innych czujników. Wybrano do tego szereg algorytmów reprezentujących najpopularniejsze metody matematyczne zastosowane do tego celu. Wykorzystano funkcje zaimplementowane w bibliotece BGS, opartej na bibliotece OpenCV (ang. Open Source Computer Vision Library). Analizowane jest ich działanie i skuteczność zarówno w przypadku prostego układu drogi na przykładzie autostrady dwupasmowej, jak i bardziej skomplikowanej na przykładzie skrzyżowania w formie litery T.

Dodatkowym celem pracy jest zaproponowanie i porównanie metod śledzenia obiektów wykrytych w wyniku działania algorytmów wykrywania ruchu. Porównano działanie dwóch algorytmów stworzonych i zaimplementowanych przez autora niniejszej pracy.

1.2. Układ pracy

Praca zaczyna się od wprowadzenia do tematyki wykrywania ruchomych obiektów. Kolejna część poświęcona jest wybranym metodom wykrywania ruchu w sekwencji obrazów z kamery. Opis każdej metody zawiera opis teoretyczny algorytmu na którym się opiera, a także przedstawienie wyników z jej badań na wybranym filmie. Ostatnia część pracy obejmuje teoretyczny opis metod śledzenia oraz wyniki badania ich skuteczności. Praca podzielona jest na rozdziały zawierające:

Rozdział 2: wprowadzenie do tematyki wykrywania ruchu w sekwencjach obrazów, omówienie obszaru ich zastosowania, podział i klasyfikacja metod,

Rozdział 3: opis kolejnych etapów prowadzonych badań, wykorzystanego sprzętu i oprogramowania, zestawienie wykonanych i użytych filmów oraz stworzonego przez autora oprogramowania,

Rozdział 4: badania metod wykrywania ruchu na filmie przedstawiającym ruch na autostradzie. Zawiera: opis metody badania, opis algorytmów wykorzystanych przez poszczególne metody, wyniki badań i wpływ poszczególnych parametrów na jakość

badania. Następnie dla wybranych algorytmów przeprowadzono analogiczne badanie dla filmu przedstawiającego skrzyżowanie w formie litery T i wpływ rozdzielczości na efektywność wykrywania ruchu,

Rozdział 5: opis i porównanie skuteczności dwóch algorytmów śledzenia obiektów zaproponowanych przez autora, opis problemów spotykanych przy śledzeniu i wpływu rozdzielczości na jego dokładność,

Rozdział 6: podsumowanie i wnioski oraz przedstawienie możliwości dalszego rozwoju opisywanych zagadnień.

2. Obszary zastosowania i podział metod wykrywania ruchu

2.1. Obszary zastosowania

2.1.1. Liczenie samochodów

Obecnie jedną z często stosowanych metod liczenia przejeżdżających samochodów jest użycie pętli indukcyjnej [2]. Zapewnia ona wysoką dokładność zliczania pojazdów, niestety nie dostarcza wystarczająco wiele danych do dokładnej klasyfikacji pojazdów. W badaniach gdy potrzebne są dodatkowe parametry do dokładniejszej analizy ruchu takie jak rodzaj, marka, rozmiar pojazdów sama pętla jest niewystarczająca. Ponadto wymaga ingerencji w strukturę drogi co poważnie podnosi koszty jej stosowania. Z tych przyczyn stosuje się analizę obrazu z kamer wideo, głównie w celu liczenia i klasyfikowania przejeżdżających pojazdów, szczególnie na autostradach. Niektóre systemy zostały też zaadaptowane do środowiska miejskiego. Wówczas kamery montuje się na wysokich słupach lub bramownicach nad drogą. Są one specjalnie przeznaczone do tego celu, gdyż standardowe kamery monitoringu miejskiego montuje się niżej [2]. Zapewnia to lepszy kąt obserwowania obszaru, który ogranicza nakładanie się jadących blisko siebie pojazdów.

2.1.2. Kontrola i pobieranie opłat

System kontroli i pobierania opłat bazuje na rozpoznawaniu numerów rejestracyjnych samochodów. Służy m. in. pobieraniu opłat za przejazd, identyfikacji pojazdów, rozpoznawaniu pojazdów poszukiwanych. Kamery są tak ustawione by zapewnić duże powiększenie na tablicę rejestracyjną pojazdów. Takie umieszczenie kamery ogranicza znacznie jej pole widzenia. Często stosuje się dodatkowe oświetlenie promieniami podczerwonymi by ułatwić rozpoznanie numeru rejestracyjnego. System wykrywania ruchu umożliwia wykrycie pojazdu zanim rozpoznane zostaną tablice pojazdu, a następnie śledzenie go. Rozwiązania takie stosowane są m. in. przy automatycznych stacjach poboru opłat na autostradach, pobieraniu opłat za wjazd do centrum Londynu lub Singapuru [2] oraz przy kontroli wjeżdżających i wyjeżdżających samochodów z płatnego parkingu przy warszawskim Lotnisku Chopina.

2.1.3. Wykrywanie wypadków i łamania przepisów

Systemy służące do wykrywania wypadków i przypadków łamania kodeksu drogowego (przepisów), wymagają znajomości układu drogi, która jest monitorowana. Konieczne jest wprowadzenie do systemu geometrii drogi oraz modeli niebezpiecznych i niedozwolonych zachowań oraz sytuacji. Opisane wyżej systemy są testowane między innymi na autostradzie A1 niedaleko Genewy w Szwajcarii [2] i Szwecji [3]. Środowisko miejskie wymaga jeszcze szerszej bazy wykrywanych zależności między poszczególnymi pojazdami i układem drogi. Powoduje to konieczność większego poziomu rozumienia

geometrii obszaru monitorowanego przez algorytm. Algorytmy wykrywania i śledzenia ruchu stosuje się m. in. do wykrywania nielegalnego parkowania, zawracania pojazdów w miejscach zabronionych, monitorowania pasów przeznaczonych dla autobusów czy identyfikacji wypadków drogowych [2].

2.2. Podział metod wykrywania ruchu

Istnieje wiele różnych algorytmów wykrywania i śledzenia ruchu pojazdów. Opisane niżej podejścia można uogólnić na śledzenie dowolnych poruszających się obiektów, a nie tylko samochodów. By ułatwić klasyfikację algorytmów śledzenia można je podzielić ze względu na podejście do zagadnienia na dwie grupy: "top-down" (problem rozbijany jest na coraz mniejsze podproblemy, aż do uzyskania pożądanego efektu) i "bottom-up" (system łączy szereg kolejnych drobnych elementów) [2].



Rys. 2.1. Schemat blokowy algorytmów "top-down"



Rys. 2.2. Schemat blokowy algorytmów "bottom-up"

Metody "top-down", inaczej nazywane obiektowymi, polegają najpierw na ekstrakcji tła i pierwszego planu. Są one oparte o model tła, powstały w trakcie trwania fazy nauki algorytmu. Pierwszy plan definiuje się jako maskę na której zaznaczono

obszary klatki w których wykryto ruch. Następnie te obszary są grupowane, klasyfikowane i śledzone (Rys. 2.1). Tego rodzaje metody zostały zbadane w pracy.

Algorytmy "bottom-up" najpierw wykrywają i klasyfikują fragmenty obrazu jako konkretne obiekty np. koło samochodowe, tablica rejestracyjna, głowa przechodnia. Klasyfikacja opiera się o wcześniejszy "trening" algorytmów na podstawie danych rzeczywistych wprowadzonych przez człowieka. Następnie rozpoznane obiekty są grupowane w większe obiekty np. samochód, człowiek i śledzone w kolejnych klatkach (Rys. 2.2). Te metody nie opierają się na detekcji ruchu więc nie są rozważane w niniejszej pracy.

W pracy badano metody typu "top-down". Ze względu na rodzaje wykorzystywanych w nich modeli matematycznych algorytmy śledzenia ruchu można podzielić na kilka głównych grup:

- proste, np: statyczna różnica klatek, różnica klatek, metoda uśredniania, kroczącej średniej i kroczącej mediany [4], [19],

- statystyczne, wykorzystujące jeden rozkład Gaussa dla każdego piksela [5], [20],

- statystyczne, wykorzystujące wiele rozkładów Gaussa dla każdego piksela [6], [7], [8], [9],

- wykorzystujące logikę rozmytą [10],

- statystyczne oparte o kolory i parametry tekstur [11], [21], [22],

- używające sieci neuronowych [28], [29],

- inne, np. PBAS [12], GMG [13].

Metody najprostsze działają najszybciej i w badanej implementacji na komputerze typu PC, jako jedyne umożliwiają analizę obrazu z kamery w czasie rzeczywistym. Wynika to z mniejszej liczby obliczeń przypadających na jedną klatkę w tych metodach. Niestety wadą tych metod jest słabsze wykrywanie poruszających się obiektów i wprowadzanie większej liczby błędów, co pokazują wyniki przeprowadzonych w tej pracy badań. Najpopularniejszym podejściem jest wykorzystanie w algorytmach jednego lub kilku rozkładów statystycznych Gaussa. Te metody uzyskują dobre wyniki przy wykrywaniu poruszających się obiektów, jednak wymagają większej liczby obliczeń, co wydłuża czas ich wykonania dla każdej klatki. Inne metody różnią się modelami matematycznymi stosowanymi przy wykryciu ruchu, co prowadzi zarówno do różnej skuteczności, jak i czasu potrzebnego na wykonanie algorytmu. Poziom złożoności modelu tła stosowany w różnych metodach także w znaczącym stopniu wpływa na czas analizy kolejnych klatek, co także obrazują wyniki badań.

3. Wprowadzenie do badań

3.1. Etapy badań

W trakcie badań sprawdzono działanie 21 algorytmów zawartych w bibliotece BGS Library [17]. Uzyskanie danych wyjściowych testując algorytmy przy różnych wartościach ich parametrów zajęło około 70 godzin pracy komputera, w trakcie których uniemożliwione było jakiekolwiek inne jego wykorzystanie by nie wprowadzić zakłóceń do pomiaru czasu wykonania algorytmów. Wyniki działania zbadanych algorytmów odniesiono do danych rzeczywistych tzn. położenia pojazdów w kolejnych klatkach oznaczonego przez autora. Najpierw zbadano metody przy różnych ustawieniach parametrów na filmie obrazującym minutę ruchu na amerykańskiej autostradzie. W pracy zamieszczono wyniki badania metod (nazwy przetłumaczone z oryginalnych nazw algorytmów): różnicy klatek, mediany adaptacyjnej, ruchomej średniej ważonej, ruchomej wariancji ważonej, uśrednionego tła oraz metod opartych na artykułach [5], [6], [9], [11], [12]. Dodatkowo sprawdzono działanie pozostałych metod z biblioteki BGS. Uzyskiwały one wyniki zbliżone do innych np. metody na podstawie artykułów [7], [8] do innych opartych o modele statystyczne Gaussa (np. oparte o artykuły [6], [9]) lub wyniki wyraźnie gorsze m. in. metody używające o logiki rozmytej [10], wnioskowanie Bayesa [13]. Z tego powodu nie zamieszczono ich wyników w niniejszej pracy.

W kolejnym etapie badań sprawdzono działanie najlepszych spośród zbadanych algorytmów na drugim filmie przedstawiające skrzyżowanie w formie litery "T" znajdujące się w mieście. Porównano też wyniki ich działania przy różnej rozdzielczości filmu wejściowego.

Ostatnim elementem badań była weryfikacja metod śledzenia wykrytych obiektów. W tym celu porównano dwie metody zaproponowane przez autora, a jako dane referencyjne wykorzystano ręcznie oznaczone na filmach rzeczywiste pozycje poruszających się obiektów.

3.2. Sprzęt i oprogramowanie

3.2.1. Sprzęt

Do stworzenia niezbędnego oprogramowania oraz przeprowadzenia analizy algorytmów detekcji ruchu i śledzenia obiektów wykorzystano komputer klasy PC. Specyfikacja jego istotnych elementów znajduje się w tabeli poniżej.

Procesor	Intel Core I5 750 @ 2.67GHz		
Płyta główna	Gigabyte P55-UD3		
Pamięć RAM	8GB DDR3		
System operacyjny	Windows 7 Home Premium		

Tabela 3.1. Specyfikacja podstawowych parametrów komputera

Filmy z Trasy Łazienkowskiej i skrzyżowania ulic Topiel i Tamka w Warszawie zostały nagrane przy pomocy aparatu: Sony Nex-5 z matrycą 14,2 MP Exmor APS HD CMOS. W celu stabilizacji zamontowano aparat na statywie. Użyto dwóch rodzajów obiektywów: szerokokątnego o ogniskowej 16mm i normalnego o zmiennej ogniskowej 18-55mm. W drugim przypadku filmy nagrano z ogniskową 18mm.

Film obejmujący skrzyżowanie typu T ulic Robotniczej i Śrubowej we Wrocławiu został nagrany przy pomocy kamery Bosch NBN-921-IP z obiektywem Bosch VLG-3V3813-MP3. Film ten został udostępniony przez firmę Neurosoft z Wrocławia.

W celu stworzenia oprogramowania potrzebnego do badań skorzystano z środowiska Visual Studio 2010 oraz bibliotek OpenCV w wersji 2.4.5 i BGS Library w wersji 1.6. Dodatkowo analizowano i przetwarzano filmy w programach Blender i VirtualDub. Analizę danych przeprowadzono w programie OppenOffice Calc.

3.2.2. Biblioteka OpenCV

Biblioteka OpenCv jest wieloplatformową biblioteką funkcji programistycznych, przeznaczonych do przetwarzania obrazów głównie w czasie rzeczywistym. Prace nad biblioteką rozpoczęła firma Intel w roku 1999. Od tego czasu biblioteka jest ciągle rozwijana i rozbudowywana – od połowy 2008 roku przez niezależnych deweloperów. Biblioteka została oparta o licencję open source BSD, co umożliwia darmowe korzystanie z niej także w celach komercyjnych. Obecnie posiada aktywną bazę użytkowników liczącą ponad 47 000 osób i liczbę pobrań przekraczającą 6 milionów [16]. Wersja biblioteki wykorzystana w pracy to 2.4.5 z dnia 11 kwietnia 2013 roku. Jej głównymi zadaniami jest:

- rozwój i optymalizacja algorytmów widzenia komputerowego,
- przekazywanie wiedzy o analizie obrazów poprzez dostarczenie łatwo rozumianych i transferowalnych na inne platformy funkcji,
- umożliwienie tworzenia zaawansowanych, dobrze zoptymalizowanych komercyjnych aplikacji opartych o widzenie komputerowa.

Główne zastosowania biblioteki:

– analiza obrazów 2d i 3d,

- rozpoznawanie twarzy,
- rozpoznawanie gestów,
- interakcja człowiek-komputer,
- roboty mobilne,
- identyfikacja i śledzenie obiektów,
- segmentacja i rozpoznawanie elementów obrazów,
- wykrywanie i śledzenie ruchu na obrazach,
- stereowizja.

Biblioteka działa na następujących systemach operacyjnych: Windows, Android, Maemo, FreeBSD, OpenBSD, iOS, BlackBerry 10, Linux i OS X.

Oryginalnym językiem programistycznym biblioteki był C, ale obecnie większość funkcji została także napisana w C++, który stał się podstawowym językiem OpenCV. Oprócz tego stworzono pełne interfejsy dla takich języków/środowisk jak: Phyton, Java, Matlab/Octave. We wrześniu 2010 roku stworzono interfejs wsparcia kart graficznych z technologią CUDA, a od października 2012 roku rozwijany jest interfejs oparty o karty graficzne w technologi OpenCL.

3.2.3. Biblioteka BGSLibrary

BGSLibrary jest biblioteką programistyczną zbudowaną na podstawie funkcji z biblioteki OpenCV przez Andrewsa Sobrala. Dostarcza ona narzędzia do ekstrakcji tła i pierwszego planu w przypadku scen zawierających obiekty ruchome. Działa zarówno na systemie operacyjnym Windows jak i Linux. Obecnie posiada 32 algorytmy dostarczone przez wielu autorów. Jej kod jest udostępniany w zgodzie z licencją GNU GPL v3 open source i jest darmowy do wykorzystania [17].

Poniżej zamieszczono spis algorytmów dostępnych na dzień 26.07.2013 w bibliotece BGS. Nazwy algorytmów zachowano oryginalne, przetłumaczono jedynie nazwy kategorii. Numer przed nazwą odpowiada numerowi na wykresie na rys. 3.1, brak numeru oznacza brak wykresu danej metody na rys. 3.1. W nawiasie podano rok publikacji artykułu, na podstawie którego zostały zaimplementowany dany algorytm.

Spis algorytmów z biblioteki BGS:

Metody podstawowe

- 1. Static Frame Difference,
- 2. Frame Difference,
- 3. Weighted Moving Mean,
- 4. Weighted Moving Variance,

5. Adaptive Background Learning,

6. Temporal Mean,

7. Adaptive Median of McFarlane and Schofield (1995) [19],

8. Temporal Median of Cucchiara et al (2003) and Calderara et al (2006) [4].

Metody oparte o logikę rozmytą:

9. Fuzzy Sugeno Integral (with Adaptive-Selective Update) of Hongxun Zhang and De Xu (2006) [25],

10. Fuzzy Choquet Integral (with Adaptive-Selective Update) of Baf et al (2008) [26],

11. Fuzzy Gaussian of Sigari et al (2008).

Metody statystyczne wykorzystujące pojedynczy rozkład Gaussa:

12. Gaussian Average of Wren (1997) [5],

13. Simple Gaussian of Benezeth et al (2008) [20].

Metody statystyczne wykorzystujące wiele rozkładów Gaussa:

14. Gaussian Mixture Model of Stauffer and Grimson (1999) [6],

15. Gaussian Mixture Model of KadewTraKuPong and Bowden (2001) [7],

16. Gaussian Mixture Model of Zivkovic (2004) [8],

17. Gaussian Mixture Model 2 of Zivkovic (2004) [8],

18. Gaussian Mixture Model of Baf et al (2008) [9].

Metody oparte o logikę rozmytą typ 2:

19. Type-2 Fuzzy GMM-UM of Baf et al (2008) [10],

Type-2 Fuzzy GMM-UV of Baf et al (2008) [10],

Type-2 Fuzzy GMM-UM with MRF of Zhao et al (2012) [27],

Type-2 Fuzzy GMM-UV with MRF of Zhao et al (2012) [27].

Metody statystyczne używające parametrów kolorów i tekstur:

Texture BGS of Heikkila et al. (2006) [21],

Texture-Based Foreground Detection with MRF of Csaba Kertész (2011) [22],

20. Multi-Layer BGS of Jian Yao and Jean-Marc Odobez (2007) [11].

Metody nieparametryczne:

21. Pixel-Based Adaptive Segmenter (PBAS) of Hofmann et al (2012) [12],

22. GMG of Godbehere et al (2012) [13],

23. VuMeter of Goyat et al (2006) [23],

KDE of Elgammal et al (2000) [24].

Metoda oparta o wartości własne:

24. Eigenbackground / SL-PCA of Oliver et al (2000) [14]

Metody oparte o sieci neuronowe:

25. Adaptive SOM of Maddalena and Petrosino (2008) [28],

26. Fuzzy Adaptive SOM of Maddalena and Petrosino (2010) [29].



Rys. 3.1. Porównanie metod z bibliotek BGS [17]

Na rysunku 3.1 autor biblioteki przedstawił porównanie poziomów wykorzystania procesora, pamięci i średniego czasu przetworzenia klatki dla wszystkich metod zawartych w bibliotece. Widać na nim wyraźne różnice w wykorzystaniu pamięci oraz czasie przetwarzania dla poszczególnych metod. Oczywiście, dla innych danych wejściowych (w szczególności dla wyższych rozdzielczości), wyniki mogą być nieco inne, co widać w przeprowadzonych w ramach niniejszej pracy badaniach, gdzie najwolniejszą okazała się metoda nr 21, w przeciwieństwie do metody nr 20 jak na rysunku 3.1.

3.2.4. Materiał wideo

Przed przystąpieniem do badań zebrano materiał filmowy przedstawiający ruch pojazdów. Trzy filmy zostały pobrane z portalu Youtube [18]. Przedstawiają one

dwujezdniową autostradę o dwóch pasach ruchu w każdą stronę znajdującą się w USA. Nagrane zostały z kamery umieszczonej na wiadukcie centralnie nad środkiem drogi.

W celu powiększenia biblioteki materiałów nakręcono filmy z kładki dla pieszych nad trasą Łazienkowską w Warszawie, obok Parku Ujazdowskiego. Drugim obszarem filmowanym było skrzyżowanie ulic Tamka i Topiel. Aparat był umieszczony na balkonie na 8 piętrze budynku stojącego przy skrzyżowaniu. Nagranie wykonano przy pomocy aparatu Sony Nex-5 umieszczonego na stabilnym statywie. Użyto dwóch rodzajów obiektywów (wąsko i szerokokątnego) oraz trybu wyostrzania ręcznego i automatycznego. W przypadku wyostrzania ręcznego, ostrość ustawiano na środek kadru. Filmy nagrywano w styczniu i lutym przy różnych warunkach pogodowych. Dodatkowo otrzymano nagrania od firmy Neurosoft z Wrocławia. Przedstawiają one skrzyżowanie w formie litery T, ulic Robotniczej i Śrubowej we Wrocławiu, nagrane z kamery umiejscowionej na poziomie 2 piętra.

W tabeli 3.2 przedstawiono zestawienie zebranych materiałów. W sumie przeanalizowano ponad 130 min. materiałów filmowych i porównywano intensywność ruchu oraz poziom jego skomplikowania. Do badań wybrano filmy: "Movie 1" (autostrada amerykańska) i "fragmenty 1" (skrzyżowanie Robotniczej/Śrubowej). Pierwszy etap badań (porównanie metod wykrywania ruchu) przeprowadzono z użyciem filmu "Movie 1". Wybrano go ze względu na korzystne warunki nagrywania (powstawanie niewielkich cieni przy pojazdach) i fakt, że zawiera ruch w obu kierunkach zarówno samochodów ciężarowych, osobowych jak i motocykli. Dodatkowym atutem jest umiarkowana liczba poruszających się pojazdów, co ograniczyło czas konieczny na ręczne stworzenie danych rzeczywistych (wymagało to około 8 godzin pracy). Pojazdy widoczne na filmie poruszają się zarówno w grupach, gdy jedne pojazdy zakrywają lub zlewają się z innymi, jak i pojedynczo. Dzięki temu można sprawdzić działanie algorytmów wykrywania ruchu w różnych warunkach.

Do dalszych, dokładniejszych badań, wybrano film "fragmenty 1". Sprawdzono na nim działanie wybranych trzech algorytmów wykrywania ruchu oraz algorytmów śledzenia obiektów. Film "fragmenty 1" stanowi zbiór interesujących wycinków filmu przedstawiającego skrzyżowanie ulic Robotniczej i Śrubowej we Wrocławiu. Pozwala on na zbadanie działania badanych metod w środowisku miejskim, gdy występują zarówno piesi, jak i pojazdy poruszający się, w przeciwieństwie do autostrady, z zmienną prędkością i kierunkiem ruchu. Film zawiera zróżnicowane sytuacje do jakich dochodzi na skrzyżowaniu, w tym ruch jednostajny, zwalnianie i zatrzymywanie przed pasami, a także ruszanie, zmianę kierunku jazdy. Wybrano do badań skrzyżowanie ulic Robotniczej i Śrubowej, gdyż nagranie było wykonane z wysokości odpowiadającej wysokości montażu kamer monitoringu miejskiego, w przeciwieństwie do filmów ze skrzyżowania ulic Tamka i Topiel.



Rys. 3.2 Przykładowe klatki z filmów (od lewej): "Movie 1", "Fragmenty 1", "T_szar"



Rys. 3.3 Przykładowe klatki z filmów (od lewej): "MAH07460", "MAH07467", "MAH07455"

Miejsce	Nazwa pliku	Długość	Warunki pogodowe	Uwagi
	Movie 1	1 min.	Słońce	
Autostrada USA	Movie 2	31 sek.	Zachmurzenie	
	Movie 3	1 min.	Słońce	Drgająca kamera
	MAH07447	5 min 13sek.	Pochmurno	AF/szerokokątny
	MAH07446	5 min.	Pochmurno	MF/szerokokątny
	MAH07449	5 min 1sek.	Pochmurno	AF/wąskokątny
	MAH07448	5 min 14sek.	Pochmurno	MF/wąskokątny
	MAH07451	4 min 1sek.	Pochmurno	AF/wąskokątny
	MAH07450	4 min 4sek.	Pochmurno	MF/wąskokątny
Trace	MAH07459	4 min 1sek.	Opady śniegu	AF/szerokokątny
Irasa	MAH07458	4 min 31sek.	Opady śniegu	MF/szerokokątny
Lazielikowska	MAH07460	2 min 36sek.	Opady śniegu	MF/wąskokątny
	MAH07464	4 min 1sek.	Noc	AF/szerokokątny
	MAH07463	4 min 9sek.	Noc	MF/szerokokątny
	MAH07467	4 min 3sek.	Noc	AF/wąskokątny
	MAH07468	4 min 2 sek.	Noc	MF/wąskokątny
	MAH07469	2 min 6sek.	Noc	AF/wąskokątny
	MAH07470	2 min 9sek.	Noc	MF/wąskokątny
	MAH03365	4 min 8sek.	Słońce	AF/szerokokątny
	MAH03366	5 min 4sek.	Słońce	AF/wąskokątny
	MAH07443	8 min 47sek.	Pochmurno	AF/szerokokątny
	MAH07442	6 min 52sek.	Pochmurno	MF/szerokokątny
Skrzyzowanie Tomko Topiol	MAH07444	5 min 57sek.	Pochmurno	MF/wąskokątny
	MAH07455	5 min 1sek.	Opady śniegu	AF/szerokokątny
	MAH07454	5 min 34sek.	Opady śniegu	MF/szerokokątny
	MAH07457	4 min 53sek.	Opady śniegu	AF/wąskokątny
	MAH07456	5 min 26sek.	Opady śniegu	MF/wąskokątny
	T_szar	2 min 4sek.	Słońce	Film szarościowy
	T_kolor	1 min. 4sek.	Słońce	
Skrzyżowanie "T"	T_kolor 2	14 min. 58sek.	Słońce	
j ⁻ ⁻ ⁻	Fragmenty1	1 min. 52sek.	Słońce	Film z fragmentów "T_kolor 2"

Tabela 3.2. Zestawienie materiałów wideo

3.2.5. Stworzone oprogramowanie

Do przeprowadzenia badań stworzono trzy programy: "Dane rzeczywiste", "BGFG extraction" i "Tracker".

- Program "Dane rzeczywiste"

Pierwszy z programów służy do stworzenia danych rzeczywistych, do których będą porównywane wyniki działania algorytmów wykrywania i śledzenia ruchu. Przy jego pomocy użytkownik może w prosty sposób oznaczyć elipsą każdy wybrany obiekt w kolejnych klatkach filmu (rys. 3.4). Wybrano elipsę, gdyż umożliwia ona dokładniejsze określenie położenia obiektu i dobrze dopasowana obejmuje mniejszą powierzchnię poza interesującym nas obiektem niż prostokąt. Dodatkowo określanie położenia obiektów

elipsami wymaga wielokrotnie mniej czasu, niż ręczne wyznaczanie ich dokładnych konturów. Program umożliwia wybranie skoku co ile klatek ma następować oznaczenie, w celu ograniczenie ilości danych i czasu potrzebnego do ich stworzenia. Wybrano oznaczanie położenia obiektów co 5 klatek. Mimo to samodzielne oznaczenie wszystkich poruszających się obiektów na pierwszym filmie zajęło około 8 godzin, a na drugim 14 godzin. Każda elipsa jest definiowana poprzez położenie swojego środka, długość obu półosi i kąt obrotu względem osi poziomej klatki. Te dane są zapisane w każdej z klatek, w których nastąpiło oznaczenie. Dodatkową informacją o każdym obiekcie jest rodzaj obiektu podany przez użytkownika (pieszy, rower/motocykl, samochód osobowy, autobus, samochód ciężarowy). W celu usprawnienia procesu oznaczania, stworzono prosty system przewidywania pozycji elips w kolejnych klatkach. Bazuje on na pozycji elips w dwóch poprzednich klatkach, w których były oznaczone i na tej podstawie przewiduje gdzie będa w kolejnej. Zdecydowanie przyśpiesza to proces oznaczania, gdyż po kilku pierwszych oznaczeniach, program sam zamieszcza kolejne elipsy w przewidywanych miejscach a użytkownik tylko je poprawia by lepiej dopasować. Wszystkie informacje są zapisywane do pliku .txt, co umożliwia łatwe ich przenoszenie między programami i stanowiskami komputerowymi.



Rys. 3.4 Przykład oznaczenia pojazdu elipsą

- Program "BGFG extraction"

Program stworzony na podstawie programu "DEMO" autorstwa Andrewsa Sobrala z biblioteki BGS. Służy do analizy metod wykrywania ruchu zawartych w tej bibliotece. Wczytuje wybrany film i plik tekstowy z zapisem rzeczywistego położenia analizowanych obiektów, a następnie przetwarza film klatka po klatce przy pomocy wybranej metody z biblioteki BGS. Elementem wyjściowym z algorytmów są kolejne czarno-białe klatki, gdzie białe piksele zostały określone jako te w których nastąpił ruch. Kolejnym etapem jest porównanie obrazów wyjściowych z danymi rzeczywistymi. Porównywanie następuje co tyle klatek, co ile są oznaczone dane rzeczywiste tzn. co 5. Wyniki analizy służą porównaniu działania rożnych metod, przy różnym ustawieniu ich parametrów oraz pomiędzy sobą. Wyniki są zapisywane w formacie csv w pliku tekstowym. Proces został dokładnie opisany w rozdziale 4.1. - Program "Tracker"

Program służy do weryfikacji działania i określenia skuteczności metod śledzenia wykrytych obiektów. Zaimplementowano w nim dwie metody zaproponowane przez autora. Program wczytuje film wyjściowy z programu "BGFG extraction" oraz informacje z pliku tekstowego z danymi rzeczywistymi. Parametry wykrytych i śledzonych obiektów są porównywane z danymi rzeczywistymi. Wyniki tych analiz służą porównaniu skuteczności obu metod i wykryciu jak różne parametry pojazdów (wielkość, kolor) rodzaj ich ruchu i rozdzielczość nagrań wpływają na jakość śledzenia. Proces został dokładnie opisany w rozdziale 5.

4. Badania metod wykrywania ruchu

4.1. Wstęp

Badania metod wykrywania tła przeprowadzono na dwóch filmach. Pierwszy "Movie 1" przedstawia jednominutowe ujęcie ruchu na amerykańskiej autostradzie w słoneczny dzień, przy takim ustawieniu słońca, że nie powodowało powstawania dużych cieni i przy nieruchomej kamerze. Znajdują się na nim różne rodzaje pojazdów: motory, samochody osobowe, samochody ciężarowe. Film jest nagrany w formacie FLV, z odstępem między klatkami kluczowymi równym 60. Powoduje to powstawanie zwiększonej liczby błędnie wykrytych elementów w klatkach będących wielokrotnością liczby 60. W współczesnym kodowaniu filmów stosowanie klatek kluczowych jest powszechne w związku z tym nie odrzucono tych danych ani nie wprowadzono dodatkowych przekształceń, których celem jest eliminacja tych błędów.

Długość filmu	1min.
FPS	30
Liczba klatek	1800
Rozdzielczość	800x480 px.
Liczba oznaczonych obiektów	48
Pole najmniejszego oznaczonego obiektu	75 px.
Pole największego oznaczonego obiektu	30115 px.

Tabela 4.1. Parametry filmu "Movie 1"

Do badania pierwszego filmu zastosowano następujące metody z biblioteki BGS:

- Metoda różnicowa
- AdaptiveMedianBGS
- WeightedMovingMeanBGS
- WeightedMovingVarianceBGS
- DPMeanBGS
- GrimssonGMM
- LBMixtureOfGaussians
- PixelBasedAdaptiveSegmenter
- MultiLayerBGS

Drugi film zawiera nagranie skrzyżowania ulic Robotniczej i Śrubowej we Wrocławiu w formie litery T. Sytuacje, które na nim się znajdują, są trudniejsze do analizy dla algorytmów wykrywania ruchu, z powodu przemieszczania się obiektów w rożnych kierunkach z zmienną prędkością. Użyto go do porównania dla trzech metod:

- Metoda różnicowa
- LBMixtureOfGaussians
- PixelBasedAdaptiveSegmenter

Uzasadnienie wyboru znajduje się w rozdziale 4.6.

4.2. Metoda badania algorytmów wykrywania ruchu

W opisie badań występują dwa pojęcia: obiektów i konturów. Obiektami lub obiektami rzeczywistymi określa się oznaczone przez użytkownika przy pomocy elips poruszające się pojazdy i pieszych. Mianem konturów nazywa się fragmenty obrazu wyjściowego algorytmu detekcji obiektów ruchomych, które składają się z 4 lub więcej białych pikseli.

Pierwszym krokiem było oznaczenie na filmie pozycji pojazdów z wykorzystaniem programu "Dane rzeczywiste". W ten sposób utworzono zestaw danych rzeczywistych, do których będą porównywane wyniki badania poszczególnych metod. Oznaczenie zastosowano co 5 klatkę w celu uniknięcia nadmiaru danych. Każdy pojazd w każdej badanej klatce został opisany elipsą najlepiej dopasowaną do obszaru w którym się znajduje.

Film i dane rzeczywiste w formacie .txt są wprowadzane do programu "BGFG extraction" wykrywającego tło i pierwszy plan. Analizuje on materiał wejściowy klatka po klatce, stosując jedną z wybranych do badania metod, wraz z zadanymi parametrami. Piksele z wprowadzanej klatki są opisane wartościami kolorów zapisanymi w przestrzeni RGB o zakresie wartości 0-255 (8-bitów na kanał). Klatka wyjściowa jest zbiorem pikseli w przestrzeni binarnej, gdzie 1(biały) oznacza wykryty ruch, a 0 (czarny) brak ruchu - tło. Co 5 klatkę następuje porównanie klatki wyjściowej do danych rzeczywistych. W dalszej części pracy używane są pojęcia "biały piksel" -piksel w którym program wykrył ruch i odpowiednio "czarny piksel" - piksel w którym ruch nie został wykryty.

W procesie porównywania klatki wyjściowej z danymi rzeczywistymi, najpierw następuje analiza całej klatki. Zliczana jest liczba białych pikseli znajdujących się w elipsach rzeczywistych, liczba białych pikseli poza nimi, czas wykonania analizy danej klatki w milisekundach (tylko analizy wykrywającej pierwszy plan/tło, bez dalszej analizy danych). Następnie w klatce wykrywane są kontury przy pomocy funkcji findContours z biblioteki openCV, której parametry zostały ustawione tak by wykrywać tylko zewnętrzne kontury i nie traktować "dziur" w konturach jako osobnych konturów.

Zdecydowano się na taką metodę, ze względu na fakt, że przy wykrywaniu pojazdów, "dziury" w konturach oznaczają najczęściej niewykryty fragment pojazdu, a nie brak pojazdu. Każdy kontur jest dodatkowo opisany najlepiej dopasowaną do niego elipsą. Następnie jest zapisywana ilość wszystkich konturów oraz konturów dużych w danej klatce. Konturami dużymi określa się te kontury, których powierzchnia jest większa niż 65 pikseli w przypadku pierwszego filmu i 461 w przypadku drugiego. Stanowi to odpowiednio 85% i 75% pola powierzchni najmniejszego obiektu rzeczywistego w danym filmie. Wybrano wartość mniejszą od pola obiektów rzeczywistych, ze względu na fakt, że elipsa nie opisuje doskonale położenia obiektu w klatce i zawierają się w niej elementy tła. Jednocześnie liczone jest pole powierzchni wszystkich obiektów rzeczywistych (ich elips rzeczywistych, którymi zostały opisane ręcznie). Dane zostają zapisane w odpowiednim formacie do pliku txt.

Kolejnym etapem jest analiza każdego obiektu rzeczywistego w klatce. Liczone jest pole powierzchni jego elipsy rzeczywistej, liczba białych pikseli i pole wszystkich wykrytych konturów znajdujących się wewnątrz tej elipsy. Pole konturów może być większe od liczby białych pikseli, ponieważ znajdujące się wewnątrz konturu "dziury" (czarne piksele) wliczane są do powierzchni konturu ("dziury" nie są traktowane jako osobne kontury). Dodatkowo liczona jest liczba konturów, których środki elips opisanych na nich znajdują się wewnątrz elipsy opisującej obiekt rzeczywisty. Dane zostają zapisane w odpowiednim formacie do kolejnego pliku txt.

4.3. Analiza danych

Analiza danych odbywa się w programie OpenOffice Calc. Dla każdej klatki liczone są ilorazy: liczby konturów do liczby obiektów rzeczywistych, liczby wszystkich wykrytych konturów dużych do liczby obiektów rzeczywistych, liczba białych pikseli w elipsach rzeczywistych do pola powierzchni tych elips (stopień wypełnienia elips) oraz liczba białych pikseli poza elipsami do pola powierzchni klatki poza elipsami (szum). Następnie obliczane są wartości średnie i odchylenia standardowe z tych ilorazów oraz z czasu analizy klatek. Wartości średnich i odchyleń standardowych liczone są dla wszystkich klatek i osobno dla drugiej połowy filmu. Wartości policzone tylko na podstawie drugiej połowy filmu służa porównaniu działania algorytmów, gdy metody już mają optymalnie stworzony obraz tła (są nauczone). Analogiczne obliczenia dokonywane sa dla każdego obiektu w każdej klatce. Dla każdego obiektu obliczany jest stosunek białych pikseli wewnątrz elipsy rzeczywistej do pola tej elipsy, pola konturów wykrytych wewnątrz elipsy do pola elipsy. Dodatkowo obliczana jest średnia i odchylenie standardowe czasu analizy klatek i liczby środków elips wewnątrz elipsy rzeczywistej. Wyniki analiz przedstawiają skuteczność wykrywania ruchu przez dane metody i czy obiekt jest określony jednym konturem czy ich większą liczbą. Ich zestawienie zamieszczono w tabelach przy każdej metodzie.

Ostateczną ocenę metody oparto na następujących założeniach:

-Liczba wykrytych konturów, a zwłaszcza dużych konturów powinna być zbliżona do

liczby obiektów oznaczonych na danej klatce (średnia stosunków liczby konturów do liczby obiektów jak najbliższa 1),

- możliwie mała fragmentacja obiektów,

-Średni czas analizy powinien mieć jak najniższą wartość,

-Liczba pikseli dobrze wykrytych powinna być jak najbliższa pola powierzchni obiektów rzeczywistych(średnia stosunków liczby białych pikseli do powierzchni obiektów jak najbliższa 1) – możliwie jak najlepsze wypełnienie obiektów,

-Liczba pikseli źle wykrytych powinna być jak najmniejsza (średnia stosunków liczby czarnych pikseli do powierzchni klatki poza obiektami jak najbliższa 0%) - możliwie jak najmniejszy szum poza obiektami,

-Liczba wykrytych środków elips wewnątrz obiektu powinna być zbliżona do 1 (jeden obiekt opisany jednym konturem) – kolejny parametr określający fragmentację obiektów,

-Pole konturów wewnątrz elipsy rzeczywistej powinno być zbliżone do pola powierzchni tej elipsy (obliczana średnia ich stosunku jak najbliższa 100%) - kolejny parametr określający wypełnienie obiektów.

Na podstawie tych danych dokonano analizy metod wykrywania ruchu. Dla każdej metody przeprowadzono badania przy różnych jej parametrach, by znaleźć optymalny ich zestaw dla tego filmu. W pracy przedstawiono wybrane zestawy parametrów najlepiej obrazujące skuteczność działania metody. W tabelach podano wartości średnie wyników z analizy danych dla całego filmu i dla jego drugiej połowy. Ze względu na duże wartości ilorazów w kolumnach 1, 2 i 5 nie podano ich w procentach, by poprawić czytelność tabel. Kolejne kolumny w tabelach określają:

-Kolumna nr 1 "Liczba konturów": Średnia ilorazów liczby konturów wykrytych do liczby obiektów rzeczywistych,

-Kolumna nr 2 "Liczba dużych konturów": Średnia ilorazów liczby dużych konturów wykrytych do liczby obiektów rzeczywistych,

-Kolumna nr 3 "Czas [ms]": Średni czas analizy klatek w milisekundach,

-Kolumna nr 4 "Liczba pikseli dobrze wykrytych": Średnia ilorazów liczby białych pikseli wewnątrz konturów rzeczywistych do pola powierzchni tych obiektów,

-Kolumna nr 5 "Liczba pikseli źle wykrytych": Średnia ilorazów liczby białych pikseli poza konturami rzeczywistymi do pola powierzchni klatki poza tymi obiektami,

-Kolumna nr 6 "Liczba środków elips wewnątrz obiektu": Średnia liczba elips konturów, których środek znajduje się wewnątrz obiektów rzeczywistych,

-Kolumna nr 7 "Pole konturów wewnątrz obiektów": Średnie pole konturów wewnątrz obiektów rzeczywistych.

Dla obrazowego porównania metod i parametrów zamieszczono klatkę

wyjściową nr 1205 z analizy pierwszego filmu, a także wykresy przedstawiające stosunek liczby dużych konturów do liczby obiektów rzeczywistych w funkcji numeru klatki filmu. W przypadku gdy w danej metodzie czas analizy klatki zmieniał się w czasie dodatkowo zamieszczono jego wykres. Wykresy pozostałych wyników analizy znajdują się w plikach załączonych na płycie CD. Jako nazwy kolejnych metod, w celu łatwiejszej ich identyfikacji, zastosowano oryginalne nazwy użyte przez autora biblioteki BGS. Jako bazowe wartości parametrów przyjęto te zaproponowane przez autora biblioteki. Do nich porównywano wpływ na zmiany wyników poszczególnych parametrów.

4.4. Wyniki badania wybranych metod

4.4.1. Metoda różnicowa

Metoda różnicowa opiera się na bezwzględnej różnicy klatek. Wartości składowych koloru każdego piksela bieżącej klatki są odejmowane od odpowiadających im wartości z klatki poprzedniej. Wynik końcowy jest wartością bezwzględną tej różnicy. Dla obrazów kolorowych dokonywana jest konwersja do wersji w odcieniach szarości poprzez zastosowanie sumy ważonej poszczególnych składowych (czerwony: 0,299, zielony: 0,587, niebieski: 0,114). Obraz wyjściowy jest progowany progiem binarnym o zadanej wartości, która to wartość jest jedynym parametrem w tej metodzie.

Metoda różnicowa jest możliwie najprostszą i najszybszą spośród wszystkich metod wykrywania ruchu, wymaga minimalnych nakładów obliczeniowych, lecz niestety nie osiąga zadowalających rezultatów. Nie jest odporna na zmiany ostrości obrazu i okresowy ruch obiektów tła np. drzew, co przekłada się na znaczną liczbę błędnie wykrytych pikseli. Czas analizy poszczególnych klatek jest niezależny od wartości parametru metody oraz liczby klatek, które zostały już przetworzone (metoda róznicowa nie wykorzystuje danych historycznych). W ramach przeprowadzonych eksperymentów porównano działanie metody różnicowej przy użyciu trzech progów o wartościach: 10, 15 i 20. Wyniki porównania omówiono oraz pokazano w tabelach i na ilustracjach poniżej.

 Tabela 4.2. Wartości średnie w zakresie klatek 1-1800

 Analiza klatek
 Analiza obiek

 Liczba
 Liczba środków

			Analiza obiektów				
		Liczba				Liczba środków	
	Liczba	dużych		Liczba pikseli	Liczba piksli	elips wewnątrz	Pole konturów
	konturów	konturów	Czas [ms]	dobrze wykrytych	błędnie wykrytych	obiektu	wewnątrz obiektu
Próg: 10	105,66	1,23	8,0	55,84%	0,43%	2,21	67,88%
Próg: 15	32,80	1,12	8,2	47,11%	0,28%	2,65	57,85%
Próg: 20	22,60	1,10	8,3	40,72%	0,23%	3,32	48,87%

Tabela 4.3. Wartości średnie w zakresie klatek 900-1800

			Analiza obiektów				
		Liczba				Liczba środków	
	Liczba	dużych		Liczba pikseli	Liczba pikseli	elips wewnątrz	Pole konturów
	konturów	konturów	Czas [ms]	dobrze wykrytych	błędnie wykrytych	obiektu	wewnątrz obiektu
Próg: 10	96,27	1,40	8,0	60,80%	0,53%	2,01	75,54%
Próg: 15	30,30	1,31	8,3	52,01%	0,36%	2,39	66,38%
Próg: 20	22,07	1,31	8,5	45,38%	0,30%	3,26	57,25%

Czas analizy pojedynczej klatki jest w przybliżeniu stały i niemal niezależny od wartości jedynego parametru metody. Różnice przy poszczególnych progach wahają się w granicach 0,2-0,3 ms. w stosunku do średniego czasu przetwarzania klatki wynoszącego około 8 ms. Zastosowanie najniższego progu równego 10 przekłada się na większą czułość algorytmu. Niestety powoduje to znaczący wzrost błędnie wykrytych pikseli w stosunku do dwóch pozostałych progów (odpowiednio o 54% i 87%). Jednakże jego zaleta jest wzrost liczby poprawnie wykrytych pikseli, który jednak wynosi znacznie mniej (odpowiednio 19% i 37%) od wzrostu liczby błędnie wykrytych. Dodatkowo, liczba dużych konturów jest większa o odpowiednio 10% i 12%. Obrazuje to rysunek 4.1, gdzie wyraźnie widać, że wykres dla progu o wartości 10 jest wyżej niż dla innych jego wartości. Liczba wszystkich wykrytych konturów jest zdecydowanie największa także dla progu o wartości 10 (o 222% większa niż przy progu 15 i 368% niż przy progu 20). W efekcie, obraz ruchu dla progu o wartości 10 zawiera znacznie wiekszą liczbę fałszywych konturów niż dla wyższych progów, co widać na rys. 4.2 (liczne niewielkie "kropki" w miejscach, gdzie nie występowały obiekty ruchome). Główną (i jedyną) zaletą niskiego progu jest stosunkowo dobre wypełnienie obrazu poruszających się obiektów, a więc najmniejsza fragmentacja konturów, co obrazuje fakt, że liczba wyznaczonych środków elips dla progu o wartości 10 jest mniejsza o odpowiednio 17% i 33% dla pozostałych progów.



Rys. 4.1. Stosunek liczby dużych konturów wykrytych do liczby obiektów rzeczywistych



Rys. 4.2. Klatka nr 1205, próg (od lewej): 10, 15, 20

Do dalszych porównań został wybrany próg o wartości 15, ponieważ ogranicza on liczbę błędnie wykrytych pikseli, a jednocześnie nie powoduje znacznego spadku liczby dobrze wykrytych pikseli. W stosunku do niego próg o wartości 20 pozwala wprawdzie na znaczące zmniejszenie liczby błędnie wykrytych pikseli (o 28%), przy jednoczesnym niewielkim ograniczeniu liczby dobrze wykrytych (o 14%) ale powoduje też większą fragmentację obiektów dobrze wykrytych (3,32 środków elips w porównaniu do 2,65 w przypadku progu o wartości 15, co stanowi wzrost aż o 25%). Widać to dobrze na rys. 4.2, gdzie wraz ze wzrostem wartości progu, sylwetki poruszających się obiektów stają się coraz bardziej "dziurawe". Średnia liczba dużych konturów dla progu o wartości 15 jest praktycznie taka sama jak dla progu najwyższego (jest większa o zaledwie 2% w całym filmie i identyczna w drugiej jego połowie), jednak dokładniejsza analiza tego parametru (rys. 4.1) pokazuje, że jego chwilowe wahania są wyraźnie większe dla progu o wartości 20.

4.4.2. AdaptiveMedianBGS

Metoda AdaptiveMedianBGS opiera się na aktualizowanym modelu tła. Nazwa pochodzi od faktu, że wartości pikseli modelu tła dążą do wartości środkowych (mediany) danego piksela w trakcie trwania całego analizowanego filmu. Parametrami tej metody są próg i okres próbkowania modelu (liczba klatek między aktualizacjami modelu tła (nazwana przez autora sampling rate)).

W każdej klatce jest prowadzone porównanie tej klatki do modelu w celu wykrycia pikseli w których nastąpił ruch. Dla każdego piksela liczone są odległości dla każdej składowej koloru. Są one równe bezwzględnej różnicy danego koloru piksela i odpowiadającej mu wartości z modelu. Jeżeli wszystkie trzy dystanse są mniejsze lub równe od zadanej wartości progu, wówczas piksel jest przypisywany do tła, natomiast gdy co najmniej jeden dystans jest większy od progu, to piksel jest przypisywany do pierwszego planu.

Model tła jest obrazem o rozmiarach klatki zapisanym w przestrzeni RGB. Przy uruchomieniu algorytmu jako model przyjmuje się pierwszą klatkę filmu. Następnie wszystkie piksele jest aktualizowany co wybraną liczbę klatek, równą ustawionej wartości okresu próbkowania ("sampling rate"). Wartość każdego koloru każdego piksela bieżącej klatki jest porównywana z odpowiadającymi im wartościami w modelu. Jeżeli wartość koloru z bieżącej klatki jest wyższa niż odpowiadająca jej wartość z modelu, to dana wartość koloru z modelu jest zwiększana o jeden, w przeciwnym razie zmniejszana o jeden. W ten sposób wartości kolorów pikseli modelu dążą do odpowiadającej im wartości środkowej z historii wybranych klatek. Poniżej w tabelach i na rysunkach zamieszczono wyniki badań.

			Analiza c	biektów			
	Liczba konturów	Liczba dużych konturów	Czas [ms]	Liczba pikseli dobrze wykrytych	Liczba pikseli błędnie wykrytych	Liczba środków elips wewnątrz obiektu	Pole konturów wewnątrz obiektu
Próg: 40 Okres: 7	17,12	1,60	55,6	50,66%	0,21%	2,94	49,77%
Próg: 40 Okres: 4	15,84	1,48	59,5	50,31%	0,19%	2,93	49,56%
Próg: 40 Okres: 2	14,96	1,44	68,8	49,43%	0,16%	2,99	48,91%
Próg: 20 Okres: 7	24,07	1,77	53,9	69,60%	0,39%	1,16	69,30%
Próg: 10 Okres: 7	150,41	2,28	52,9	79,89%	0,75%	0,94	78,47%
Próg: 20 Okres: 4	16,66	1,61	59,6	69,24%	0,34%	1,17	69,02%

Tabela 4.4. Wartości średnie w zakresie klatek 1-1800

Tabela 4.5. Wartości średnie w zakresie klatek 900-1800

			Analiza d	Analiza obiektów			
	Liczba konturów	Liczba dużych konturów	Czas [ms]	Liczba pikseli dobrze wykrytych	Liczba pikseli błędnie wykrytych	Liczba środków elips wewnątrz obiektu	Pole konturów wewnątrz obiektu
Próg: 40 Okres: 7	13,86	1,49	55,3	55,28%	0,22%	2,69	54,43%
Próg: 40 Okres: 4	14,01	1,48	58,9	54,99%	0,21%	2,73	54,28%
Próg: 40 Okres: 2	14,09	1,50	68,0	54,23%	0,20%	2,85	53,68%
Próg: 20 Okres: 7	8,66	1,47	54,8	73,30%	0,37%	1,05	72,58%
Próg: 10 Okres: 7	79,72	1,88	52,9	82,35%	0,62%	0,99	79,66%
Próg: 20 Okres: 4	8,15	1,41	59,4	73,12%	0,35%	1,06	72,53%

Wartość progu, w analogiczny sposób do poprzedniej metody, wpływa na ilość poprawnie wykrytych pikseli. Przy progu o wartości 20 wzrost wynosi 33%, a przy progu wartości 10 aż o 49% w stosunku do progu o wartości 40. Niestety następuje także duży wzrost liczby błędnie wykrytych pikseli, analogicznie aż o 68% i 182%. Wartość progu wpływa też na fragmentaryzacje konturów, zmniejszenie jego wartości z 40 do 20 powoduje spadek liczby wykrytych środków elips w obiektach o 61%, a przy progu równym 10 o 65%. Wpływ progu na czas analizy jest znikomy (rys. 4.5). Zmniejszanie jego wartości powoduje wzrost liczby wykrytych dużych konturów (w stosunku do progu o wartości 40 o 11 % przy zmniejszeniu progu do wartości 20 i o 43% do wartości 10), co oznacza pojawianie się dodatkowych konturów w miejscach poza obiektami rzeczywistymi (rys. 4.7). Obrazują to przebiegi wykresów na rys. 4.4.



Rys. 4.3. Stosunek liczby dużych konturów wykrytych do liczby obiektów rzeczywistych



Rys. 4.4. Stosunek liczby dużych konturów wykrytych do liczby obiektów rzeczywistych

W klatkach w których następuje aktualizacja modelu tła widać wyraźny skok czasu wykonywania algorytmu (rys. 4.5). Jest to spowodowane koniecznością ponownego odwołania się do każdego piksela i przeprowadzenia obliczeń dla każdego kanału koloru. Czas wykonania obliczeń dla takiej klatki jest w przybliżeniu dwa razy dłuższy od pozostałych. Częstsze aktualizowanie modelu tła (niższa wartość "Sampling rate") prowadzi do 5% spadku liczby błędnie wykrytych pikseli przy każdej kolejnej badanej wartości "sampling rate". Powoduje także niewielki spadek liczby poprawnie wykrytych pikseli (w stosunku do wartości 7 o 0,5% dla wartości 4 i o 2% dla wartości 2) oraz wydłużenia średniego czasu analizy klatki (analogicznie o 8% i o 23%). Wpływ częstości aktualizowania modelu tła na fragmentaryzacje konturów jest niewielki (o około 4% przy kolejnych wartościach badanych, rys. 4.6).

Do dalszych porównań wybrano próg 20 i okres próbkowania 4, jako kompromis między czasem analizy a ilością poprawnie i błędnie wykrytych pikseli (rys. 4.8).



Rys. 4.5. Czas analizy



Rys. 4.6. Klatka nr 1205, próg:40, Sampling rate (od lewej): 7, 4, 2



Rys. 4.7. Klatka nr 1205, Sampling rate: 7, próg (od lewej): 20, 10



Rys. 4.8. Klatka nr 1205, próg: 20, Sampling rate: 4

4.4.3. WeightedMovingMeanBGS

Metoda WeightedMovingMeanBGS używa modelu tła wykorzystującego wybraną średnią. Po wczytaniu każdej klatki najpierw jest tworzony model tła z wartości poszczególnych kolorów odpowiadających sobie pikseli z bieżącej klatki i dwóch poprzednich. W celu stworzenia modelu używana jest średnia arytmetyczna lub ważona. W przypadku średniej ważonej używane są wagi: 0,5, 0,3, 0,2 (odpowiednio dla bieżącej klatki i dwóch poprzednich). Następnie obliczany jest obraz wyjściowy. Od bieżącej klatki filmu odejmowany jest model tła. Wynik jest progowany progiem binarnym o zadanej wartości. Progowanie rozdziela piksele na należące do tła albo do pierwszego planu. Wartość progu i rodzaj średniej są jedynymi parametrami w tej metodzie. Jest to prosta metoda niewymagająca dużych nakładów obliczeniowych. Metoda nie jest odporna na cykliczne ruchy obiektów tła np. drzew lub gwałtowne zmiany oświetlenia np. zachmurzenie.

			Analiza	obiektów			
	Liczba konturów	Liczba dużych konturów	Czas [ms]	Liczba pikseli dobrze wykrytych	Liczba pikseli błędnie wykrytych	Liczba środków elips wewnątrz obiektu	Pole konturów wewnątrz obiektu
Próg: 10 Śr. ważona	20,40	1,16	115,7	46,31%	0,35%	2,99	54,80%
Próg: 10 Śr. arytmetyczna	28,03	1,22	118,3	54,26%	0,42%	2,21	65,12%
Próg: 15 Śr. ważona	20,19	1,25	119,7	36,53%	0,27%	4,49	41,37%
Próg: 15 Śr. arytmetyczna	19,42	1,21	121,5	45,15%	0,34%	3,17	52,93%
Próg: 20 Śr. ważona	20,72	1,27	117,6	29,18%	0,21%	5,51	31,87%
Próg: 20 Śr. arytmetyczna	19,53	1,24	115,0	38,16%	0,29%	4,24	43,39%

Tabela 4.6. Wartości średnie w zakresie klatek 1-1800

Tabela 4.7. Wartości średnie w zakresie klatek 900-1800

				Analiza obiektów			
	Liczba konturów	Liczba dużych konturów	Czas [ms]	Liczba pikseli dobrze wykrytych	Liczba pikseli błędnie wykrytych	Liczba środków elips wewnątrz obiektu	Pole konturów wewnątrz obiektu
Próg: 10 Śr. ważona	19,81	1,37	114,6	50,97%	0,46%	2,85	62,86%
Próg: 10 Śr. arytmetyczna	25,36	1,40	117,6	59,17%	0,55%	1,93	73,05%
Próg: 15 Śr. ważona	22,51	1,52	119,6	40,53%	0,36%	5,06	49,04%
Próg: 15 Śr. arytmetyczna	19,28	1,45	122,0	49,74%	0,45%	3,11	60,87%
Próg: 20 Śr. ważona	25,18	1,61	116,8	32,69%	0,28%	6,58	38,99%
Próg: 20 Śr. arytmetyczna	21,51	1,51	114,8	42,25%	0,39%	4,70	51,05%

Wybór parametrów (metody liczenia średniej i progu) nie ma znaczącego wpływu na czas trwania analizy. We wszystkich przypadkach czas trwania analizy jednej klatki wynosił około 117 ms. Wyraźna jest różnica w skuteczności wykrywania ruchu między obliczaniem tła przy pomocy średniej arytmetycznej i ważonej. Pierwsza metoda lepiej wykrywa poprawne piksele o średnio 9 punktów procentowych, co stanowi wzrost o ok 22% w stosunku do średniej ważonej. Użycie średniej ważonej wpływa też niekorzystnie na fragmentaryzację konturów, powodując wzrost liczby wykrytych środków elips o średnio 50%. Niestety średnia arytmetyczna w stosunku do ważonej powoduje także wzrost liczby błędnie wykrytych pikseli o średnio 24%.



Rys. 4.9. Stosunek liczby dużych konturów wykrytych do liczby obiektów rzeczywistych




Rys. 4.10. Stosunek liczby dużych konturów wykrytych do liczby obiektów rzeczywistych



Rys. 4.11. Klatka nr 1205, średnia ważona, próg (od lewej): 10, 15, 20



Rys. 4.12. Klatka nr 1205, średnia arytmetyczna, próg (od lewej): 10, 15, 20

Zmiana wartości progu wpływa analogicznie na jakość wykrywania jak w poprzednich metodach. Zmniejszenie jej o 5 powoduje średni wzrost liczby dobrze wykrytych pikseli o 25% i spadek liczby błędnie wykrytych o 28% (rys. 4.7 i 4.8) Zwiększa także liczbę wykrytych dużych konturów o średnio 5% (rys. 4.9 i 4.10). W znaczący sposób wpływa na fragmentaryzacje konturów. Używając średniej ważonej zwiększenie progu z 10 do 15 powoduje wzrost liczby środków elips wewnątrz obiektu o 78%, a wzrost progu do wartości 20 o 131%. W przypadku średniej arytmetycznej analogicznie wzrost wynosi 61% i 144%.

Do dalszych porównań wybrano próg 10 i metodę wykorzystującą średnią arytmetyczną, jako przedstawiającą najlepsze wyniki.

4.4.4. WeightedMovingVarianceBGS

Metoda WeightedMovingVarianceBGS wykorzystuje wariancję oraz średnią arytmetyczną albo ważona z obecnej klatki i dwóch poprzednich. W przypadku średniej ważonej używane są wagi analogiczne jak w metodzie WeightedMovingMeanBGS: 0,5, 0,3, 0,2. Obecna i dwie poprzednie klatki są dzielone przez 255 by maksymalna wartość koloru każdego piksela była 1. Następnie obliczany jest obraz uśredniony poprzez policzenie dla każdego piksela średniej arytmetycznej lub ważonej klatki obecnej i dwóch poprzednich. Następnie są generowane trzy obrazy wariancji (po jednym dla każdej z tych trzech klatek). Każdy obraz jest tworzony poprzez różnicę danej klatki i obrazu uśrednianego, a następnie wartości wszystkich jego pikseli są podnoszone do kwadratu. Kolejnym krokiem jest, wykorzystując wybraną średnią (arytmetyczną lub ważoną), stworzenie z trzech obrazów wariancji jednego. Wartości pikseli, tak otrzymanego obrazu, są pierwiastkowane, w celu otrzymania obrazu odchylenia standardowego. Następnie każdy kolor jest mnożony przez 255 i przetwarzany do przestrzeni jednokanałowej poprzez zastosowanie sumy ważonej poszczególnych składowych (czerwony: 0.299, zielony: 0.587, niebieski: 0.114, analogicznie jak w różnicy klatek). Obraz wyjściowy jest progowany progiem binarnym o zadanej wartości. Wartość progu i wybór metody uśredniania sa jedynymi parametrami badanymi w tej metodzie. Wyniki badań zamieszczono w tabelach poniżej.

			Analiza klatek			Analiza obiektów	
		Liczba		Liczba pikseli	Liczba pikseli	Liczba środków	Pole konturów
	Liczba	dużych		dobrze	błędnie	elips wewnątrz	wewnątrz
	konturów	konturów	Czas [ms]	wykrytych	wykrytych	obiektu	obiektu
Próg: 10	47.50	4 4 4	254.2	FF 200/	0.440/	2.06	62.020/
śr. ważona	17,50	1,14	204,3	55,29%	0,41%	2,00	03,82%
Próg: 15	15.00	1.02	254.0	45 64%	0.24%	2 71	51 05%
śr. ważona	15,00	1,05	204,0	45,04 %	0,34%	2,71	51,95%
Próg: 20	15 50	1.02	254.0	20.020/	0.200/	2 47	42 010/
śr. ważona	15,56	1,02	204,9	30,23%	0,29%	3,47	42,01%
Próg: 20	45.01	1 22	256.2	40 4 4 9/	0.559/	2.01	F2 60%
śr. arytmetyczna	40,21	1,33	200,3	49,44%	0,05%	2,91	55,69%

Tabela 4.8. Wartości średnie w zakresie klatek 1-1800

-				Analiza objektów				
		Liczba		Liczba pikseli	Liczba pikseli	Liczba środków	Pole konturów	
	Liczba	dużych		dobrze	błędnie	elips wewnątrz	wewnątrz	
	konturów	konturów	Czas [ms]	wykrytych	wykrytych	obiektu	obiektu	
Próg: 10	16.01	1.00	254.4	61.000/	0 540/	1 70	70.000/	
śr. ważona	10,21	1,00	204, 1	01,09%	0,04%	1,79	12,02%	
Próg: 15	15.00	1.05	254.2	E1 000/	0.459/	0.74	61 010/	
śr. ważona	15,25	1,20	204,2	51,00%	0,45%	2,74	01,2170	
Próg: 20	17.00	1.07	255.2	42.010/	0.200/	4.01	E0 760/	
śr. ważona	17,99	1,27	200,2	43,01%	0,36%	4,01	50,76%	
Próg: 20 śr. arytmetyczna	41,21	1,46	255,7	52,29%	0,66%	3,07	59,85%	

Tabela 4.9. Wartości średnie w zakresie klatek 900-1800



Rys. 4.13. Stosunek liczby dużych konturów wykrytych do liczby obiektów rzeczywistych

Wybór metody liczenia średniej jak i wartości progu nie wpływa na czas analizy klatki. Średnia arytmetyczna, w stosunku do ważonej, powoduje wzrost dobrze wykrytych pikseli o 22%, jednak także wzrost błędnie wykrytych pikseli o 74%. Dobrze obrazuje to porównanie rysunków 4.14 i 4.15. Także wzrasta liczba wykrytych dużych konturów o 15% (rys. 4.13) oraz liczba konturów o 127%. Z powodu znacznego wzrostu liczby błędnie wykrytych pikseli oraz liczby konturów wybrano metodę liczenia średniej ważonej.

Wartość progu wpływa na wyniki w analogiczny sposób jak w poprzednich metodach. Jej wzrost o 5 powoduje spadek zarówno liczby dobrze i błędnie wykrytych pikseli o 20%. Obrazuje to rys. 4.14. Widać na nim, że z wzrostem progu pojawia się coraz mniej białych pikseli i kontury obiektów są coraz mniej wypełnione. Wpływ

wartości progu na liczbę dużych wykrytych konturów jest nieznaczny. Każde zwiększeni wartości o progu o 5 powoduje spadek ich liczby o około 4%. Natomiast wartość progu w znaczący sposób wpływa na fragmentaryzacje obiektów. Próg o wartości 15, w stosunku do wartości 10, powoduje wzrost liczby środków elips w obiekcie o 53%, a próg o wartości 20 aż o 124%.

Do dalszych porównań została wybrana metoda używająca średniej ważonej i progu o wartości 10. Daje ona najlepsze wyniki wykrywania ruchomych elementów obrazu, przy dosyć znaczącym wzroście liczby pikseli błędnie wykrytych. Jednak są to głównie pojedyncze piksele, niepołączone w większe kontury, co pozwala na proste ich odfiltrowanie od większych konturów.



Rys. 4.14. Klatka nr 1205, średnia ważona, próg (od lewej): 10, 15, 20



Rys. 4.15. Klatka nr 1205, średnia arytmetyczna, próg: 20

4.4.5. DPMeanBGS

Metoda DPMeanBGS używa średniej ważonej przy tworzeniu modelu tła. W modelu każdy piksel ma zapisaną średnią ważoną jego wartości z poprzednich klatek. Przy sprawdzaniu czy piksel należy do tła czy pierwszego planu obliczany jest dla niego dystans równy sumie kwadratów różnic między wartościami wszystkich trzech kolorów z bieżącej klatki, a odpowiadającymi im wartościami średnich z modelu. Jeżeli dystans jest mniejszy od progu, to piksel jest uznawany za tło. Następnie dla każdego koloru każdego piksela obliczana jest średnia ważona jego wartości z obecnej klatki i średniej z poprzednich. Średnia z poprzednich klatek jest brana z wagą α , a wartość z obecnej klatki z wagą 1- α . Obliczona średnia jest używana przy liczeniu dystansu dla kolejnej klatki. Parametrami tej metody są: wartość progu i waga α . W tabelach poniżej zamieszczono wyniki badań różnych wartości parametrów.

			Analiza klatek			Analiza d	obiektów
	Liczba	Liczba		Liczba pikseli	Liczba pikseli	Liczba środków	Pole konturów
	konturów	dużych konturów	Czas [ms]	dobrze wykrytych	błędnie wykrytych	elips wewnątrz obiektu	wewnątrz obiektu
Próg: 1800 α : 1e-7	18,96	1,14	110,0	29,93%	0,16%	4,57	33,33%
Próg: 900 α: 1e-7	21,21	1,08	110,0	38,07%	0,20%	3,61	44,26%
Próg: 400 α: 1e-7	37,97	1,10	116,5	47,97%	0,28%	2,55	57,97%
Próg: 200 α: 1e-7	116,22	1,21	119,6	56,23%	0,46%	2,14	67,20%
Próg: 400 α: 1e-3	37,96	1,10	118,8	47,97%	0,29%	2,55	57,96%
Próg: 400 α: 1e-1	37,18	1,14	119,2	49,19%	0,32%	2,48	59,17%
Próg: 400 α: 2.5e-1	35,55	1,16	111,2	52,41%	0,39%	2,33	62,32%

Tabela 4.10. Wartości średnie w zakresie klatek 1-1800

Tabela 4.11. Wartości średnie w zakresie klatek 900-1800

			Analiza klatek			Analiza obiektów	
	Liczba	Liczba		Liczba pikseli	Liczba pikseli	Liczba środków	Pole konturów
	konturów	dużych	Czas [ms]	dobrze	błędnie	elips wewnątrz	wewnątrz
	Kontarow	konturów		wykrytych	wykrytych	obiektu	obiektu
Próg: 1800	22.04	1 1 2	100.7	22 770/	0.21%	5 25	40.96%
α:1e-7	22,04	1,42	109,7	55,7776	0,2170	5,55	40,00 /0
Próg: 900	01.07	1 20	100.2	40 500/	0.070/	2 70	ED E10/
α: 1e-7	21,07	1,29	109,5	42,00%	0,27%	3,78	52,51%
Próg: 400	25 14	1 20	115 1	F2 06%	0.270/	1 21	66 720/
α: 1e-7	55, 14	1,20	115,1	52,90%	0,3776	2,32	00,73%
Próg: 200	105.26	1 20	101.0	61 240/	0 56%	1 00	75 120/
α: 1e-7	105,50	1,30	121,3	01,34%	0,50%	1,90	75,1576
Próg: 400	25 12	1 29	119.0	52 05%	0.37%	0.20	66 73%
α: 1e-3	55,15	1,20	110,9	52,9576	0,37 /0	2,32	00,7378
Próg: 400	24.20	1 21	110.7	EA 120/	0.410/	2.27	67 649/
α: 1e-1	34,39	1,31	119,7	54, 1570	0,4170	2,21	07,04%
Próg: 400	32 50	1 3/	111.0	57 40%	0.51%	2 17	70 54%
α: 2.5e-1	52,59	1,34	111,0	57,40%	0,0170	2,17	10,0470

Czas analizy jednej klatki jest zbliżony, niezależnie od wybranych wartości parametrów. Wynika to z faktu, że parametry nie mają wpływu na liczbę wykonywanych obliczeń. Rozbieżność w czasach wykonania klatki dla różnych parametrów wynika z zakłóceń wprowadzanych przez system operacyjny Windows.

Najpierw sprawdzono wpływ wartości progu na skuteczność wykrywania ruchu przez algorytm. Obniżanie jego wartości wpływa analogicznie jak w poprzednich metodach. W porównaniu do wartości 1800 zmniejszenie progu do wartości 900 powoduje wzrost liczby dobrze wykrytych pikseli o 26%, ale także wzrost liczby błędnie wykrytych pikseli o 29%. Kolejne obniżenie progu do wartości 400 skutkuje wzrostem

dobrze wykrytych pikseli o kolejne 25% ale błędnie o 37%. Dalsze zmniejszenie wartości progu do 200 zwiększa liczbę dobrze wykrytych pikseli o dalsze 16% i błędnie aż o 51%, w stosunku do progu o wartości 400. Zmniejszanie wartości progu wpływa też na liczbę dużych konturów wykrytych. Ich największa liczba jest w przypadku progu o wartości 1800. Wynika to z faktu, że największe kontury opisujące pojedynczy pojazd zostały podzielone na wiele mniejszych. Jak zmiany wartości progu wpływają na skuteczność wykrywania dużych konturów przedstawiono na rys. 4.16 oraz jej wpływ na obraz wyjściowy na rysunku 4.18 i 4.19. Do dalszych porównań wybrano próg o wartości 400.



Rys. 4.16. Stosunek liczby dużych konturów wykrytych do liczby obiektów rzeczywistych

Następnie zbadano wpływ parametru uczenia się (α). Odpowiedni dobór jego wartości jest kluczowy dla skutecznego działania algorytmu. Jak widać na zamieszczonych na płycie filmach, jego zły dobór powoduje powstawanie "duchów" na początku uczenia się sceny. Wynika to z niedostatecznie szybkiej aktualizacji modelu i może powodować błędy przy szybkiej zmianie natężenia oświetlenia. Podstawową wartością α było 10⁻⁷, zaproponowane przez autora. Zmiana jej do wartości 10⁻³ nie wprowadziło praktycznie żadnych zmian, co do jakości wykrywania. Liczba dobrze i źle wykrytych pikseli nie uległa zmianie w drugiej połowie filmu, minimalnie zmieniła się w pierwszej połowie. Następnie zwiększono α do wartości 0,1, co spowodowało zwiększenie liczby wykrywanych dobrze pikseli o 2% i błędnie o 11%. Zmiana wpłynęła korzystnie na fragmentaryzację konturów, zmniejszając liczbę wykrytych środków elips o 2,5%. Zwiększenie wartości α do wartości 0,25, powoduje poprawę wykrywania dobrych pikseli o dalsze 6%, niestety liczba błędnie wykrytych wzrosła aż o 25%. Zmiany jakości

wykrywania ruchu w zależności od parametru α przedstawiono na wykresach na rysunku 4.17, a ich wpływ na obraz wyjściowy na rysunku 4.20.





Rys. 4.17. Stosunek liczby dużych konturów do liczby obiektów



Rys. 4.18. Klatka nr 1205, $\alpha = 10^{-7}$, próg (od lewej): 1800, 900, 400



Rys. 4.19. Klatka nr 1205, próg: 200, α =10⁻⁷



Rys. 4.20. Klatka nr 1205, próg 400, α (od lewej): 0,001, 0,1, 0,25

4.4.6. GrimssonGMM

Metoda GrimssonGMM[6] jest oparta na modelu tła, gdzie każdy piksel jest zamodelowany mieszanką rozkładów Gaussa (od 3 do 5). Każdy kolor posiada swoją wartość oczekiwaną ale wariancja jest jednakowa dla wszystkich kolorów piksela. Na podstawie historii zmian i wariancji każdego z rozkładów Gaussa, algorytm decyduje czy dany piksel w badanej klatce należy do tła. Takie podejście umożliwia uodpornienie systemu na zmiany oświetlenia, cykliczne ruchy obiektów (np. drzew) i śledzenie wolno poruszających się obiektów.

Dla każdego piksela przeprowadzane są obliczenia na podstawie każdego z rozkładów. Najpierw liczony jest dystans danego piksela od wartości oczekiwanych danego rozkładu. Obliczana jest różnica między wartością oczekiwaną danego koloru, a jego wartością w badanym pikselu. Dystans stanowi sumę kwadratów trzech różnic (jednej dla każdego koloru). Następnie jest sprawdzane czy dystans mieści się w zakresie określonym przez wariancję rozkładu i próg. Jeżeli tak, to piksel należy do tła i nie przeprowadza się obliczeń z wykorzystaniem pozostałych rozkładów. Następuje wtedy uaktualnienie wartości oczekiwanych i wariancji tego rozkładu. Stopień zmiany reguluje parametr α . Jeżeli dystans nie mieści się w badanym zakresie, przeprowadzana jest analogiczna analiza dla kolejnego rozkładu. Gdy żaden rozkład nie spełnia powyższego kryterium, tworzony jest nowy rozkład, którego wartościami oczekiwanymi są wartości kolorów bieżącego piksela. Zastępuje on najdawniej aktualizowany rozkład.

Parametrami w tej metodzie są: α- stopień aktualizacji rozkładów, treshold- próg określający jak daleko od wartości oczekiwanej może być piksel by przypisać go do pierwszego planu, gaussians- liczba rozkładów Gaussa dla każdego piksla.

		_	Analiza klatek			Analiza obiektów	
	Liczba konturów	Liczba dużych konturów	Czas [ms]	Liczba pikseli dobrze wykrytych	Liczba pikseli błędnie wykrytych	Liczba środków elips wewnątrz obiektu	Pole konturów wewnątrz obiektu
α: 9 Alfa: 0.01 Gaussians: 3	127,58	1,65	304,37	79,31%	0,62%	0,97	77,51%
α: 9 Alfa: 0.01 Gaussians: 4	107,61	1,60	448,39	78,55%	0,59%	1,00	76,81%
α: 9 Alfa: 0.01 Gaussians: 5	103,30	1,55	617,74	77,88%	0,57%	1,02	76,12%
α: 15 Alfa: 0.01 Gaussians: 3	34,74	1,38	261,12	74,57%	0,37%	1,10	72,71%
α: 18 Alfa: 0.01 Gaussians: 3	26,11	1,36	244,42	72,69%	0,32%	1,19	70,90%
α: 9 Alfa: 0.005 Gaussians: 3	144,96	1,68	280,61	78,84%	0,61%	0,95	77,05%
α: 9 Alfa: 0.001 Gaussians: 3	334,92	2,44	295,59	78,00%	1,23%	1,01	77,55%
α: 15 Alfa: 0.005 Gaussians: 3	46,95	1,48	252,18	74,43%	0,38%	1,07	72,95%

Tabela 4.12. Wartości średnie w zakresie klatek 1-1800

			Analiza klatek			Analiza obiektów		
	Liczba konturów	Liczba dużych konturów	Czas [ms]	Liczba pikseli dobrze wykrytych	Liczba pikseli błędnie wykrytych	Liczba środków elips wewnątrz obiektu	Pole konturów wewnątrz obiektu	
α: 9 Alfa: 0.01 Gaussians: 3	64,81	1,79	348,78	83,49%	0,69%	0,94	80,78%	
α: 9 Alfa: 0.01 Gaussians: 4	59,66	1,76	519,96	83,09%	0,67%	0,94	80,65%	
α: 9 Alfa: 0.01 Gaussians: 5	59,65	1,69	720,02	82,55%	0,66%	0,96	80,43%	
α: 15 Alfa: 0.01 Gaussians: 3	17,06	1,49	283,48	79,79%	0,45%	0,97	77,68%	
α: 18 Alfa: 0.01 Gaussians: 3	14,41	1,48	268,61	78,21%	0,41%	1,01	76,42%	
α: 9 Alfa: 0.005 Gaussians: 3	28,29	1,69	314,40	82,34%	0,55%	0,92	79,83%	
α: 9 Alfa: 0.001 Gaussians: 3	42,21	1,72	318,83	79,21%	0,48%	0,98	77,56%	
α: 15 Alfa: 0.005 Gaussians: 3	10,42	1,59	276,50	78,95%	0,42%	0,97	77,04%	

The first find to be and the first find the first state of the first s	Tabela 4.13.	Wartości	średnie w	zakresie	klatek	900-	-1800
--	--------------	----------	-----------	----------	--------	------	-------

Metoda wymaga etapu nauki tła. Na wykresach na rysunkach 4.21 i 4.22 widać że po około 200 klatkach następuje zdecydowana poprawa skuteczności wykrywania. Następuje też znaczący wzrost czasu analizy (rys. 4.23 i 4.24). Jest to spowodowane rosnącą ilością rozkładów dla poszczególnych pikseli i koniecznością dokonywania większej liczby obliczeń. Po osiągnięciu maksymalnej liczby rozkładów dla wszystkich pikseli, czas analizy zmienia się już w nieznacznym stopniu. Zmiana maksymalnej ilości rozkładów Gaussa dla każdego piksela w bardzo znaczący sposób wpływa na czas analizy klatki (wzrost między trzema a czterema rozkładami o 49%, a pomiędzy trzema i pięcioma rozkładami aż o 106%). Liczba rozkładów ma niewielki wpływ na jakość wykrywania ruchu po okresie nauki. Liczba dobrze wykrytych pikseli maleje o 0,5%, a błędnie wykrytych rośnie o 2% przy zwiększaniu liczby rozkładów o jeden. Analogicznie liczba wykrytych dużych konturów zmniejsza się o 3%. Widać, że większa liczba rozkładów Gaussa w niewielkim stopniu wpływała na skuteczność wykrywania, natomiast w bardzo znacznym na czas analizy pojedynczej klatki.



Rys. 4.21. Stosunek liczby dużych konturów wykrytych do liczby obiektów rzeczywistych



Rys. 4.22. Stosunek liczby dużych konturów wykrytych do liczby obiektów rzeczywistych

Wzrost wartości progu w znacznym stopniu ogranicza liczbę niepoprawnie wykrytych pikseli (przy wzroście progu z 9 do 15 spadek o 35%, a przy zwiększeniu do 18 o 41%), kosztem skuteczności poprawnie wykrywanych pikseli (analogicznie spadek o 4,5% i 6%). Korzystnie także ogranicza liczbę wykrytych dużych konturów (rys 4.21 i 4.22). Zwiększenie wartości progu z 9 do 15 obniża ich liczbę o 17%. Kolejne zwiększenie wartości progu do 18 powoduje zmianę już tylko o mniej niż 1%. Liczba wszystkich konturów wykrytych przy progu o wartości 15 maleje o 75% w stosunku do wartości 9, co widać na rys. 4.26. Znajduje się tam, w stosunku do rys. 4.25, zdecydowanie mniej małych konturów, które nie należą do żadnego z obiektów.



Rys. 4.23. Czas analizy



Rys. 4.24. Czas analizy

Współczynnik α w znaczący sposób wpływa na efekt analizy. Zmniejszenie go do wartości 0,005 z 0,01 powoduje spadek błędnie wykrytych pikseli o 20% i spadek liczby wykrytych konturów o 56%, przy spadku liczby dobrze wykrytych pikseli o tylko 1.5%. Dzięki temu obraz wyjściowy jest dużo mniej zaszumiony (rys. 4.27). Dalsze obniżenie α do wartości 0,001 powoduje spadek liczby błędnie wykrytych pikseli o 30% i dobrze wykrytych o 5% w stosunku do bazowej wartości. Jednak liczba wykrytych konturów jest większa o 50% niż dla α =0,005. Jak widać na rys. 4.22 powoduje także duży skok liczby wykrytych dużych konturów, największy ze wszystkich badanych parametrów.

Do dalszych porównań wybrano parametry: próg: 15, α =0,005, gaussians: 3 (rys. 4.28). W stosunku do parametrów bazowych liczba dobrze wykrytych pikseli jest mniejsza o 5%, przy spadku liczby błędnie wykrytych aż o 40%. Stosunek liczby wykrytych konturów spadł o 84%, będąc najbliżej wartości 1 w porównaniu z wszystkimi badanymi zestawami parametrami.



Rys. 4.25. Klatka nr 1205, próg 9, α=0,01, Gaussians (od lewej): 3, 4, 5



Rys. 4.26. Klatka nr 1205, Gaussians: 3, α=0,01, próg (od lewej): 15, 18



Rys. 4.27. Klatka nr 1205, próg 9, Gaussians: 3, α (od lewej): 0.005, 0.001



Rys. 4.28. Klatka nr 1205, Gaussians: 3, α=0,005, próg: 15

4.4.7. LBMixtureOfGaussians

Metoda LBMixtureOfGaussians[9] jest modyfikacją metody Grimssona [6]. Również opiera się na zamodelowaniu tła przy pomocy rozkładów Gaussa dla każdego koloru każdego piksela. Maksymalna liczba rozkładów jest równa 3. Stosowane jest podwójne progowanie i każdy rozkład kanału ma własną wariancję (w przeciwieństwie do jednej wspólnej dla wszystkich trzech kolorów w przypadku metody Grimmson'a). W trakcie analizy klatki dla pierwszego rozkładu piksela najpierw liczone są różnice między kolorem z bieżącej klatki a wartością oczekiwaną (dR, dG, dB). Następnie liczony jest dystans d₂ na podstawie wzoru:

$$d_2 = \frac{dR^2}{Rvar} \cdot \frac{dG^2}{Gvar} \cdot \frac{dB^2}{Bvar}$$
(4.1)

gdzie Rvar, Gvar, Bvar są wariancjami poszczególnych kanałów koloru tego rozkładu. Jeżeli d₂ jest mniejsze od progu określonego przez "czułość", to program przechodzi do

aktualizacji rozkładów. Jeżeli nie, to operacja jest powtarzana dla kolejnego rozkładu dla tego piksela. W przypadku gdy jeden z rozkładów spełnił warunek, to jest on aktualizowany. Aktualizowane są zarówno wartości oczekiwane, jak i wariancje dla każdego koloru. Stopień aktualizacji określa parametr α i wartości kolorów z bieżącej klatki. W przypadku, gdy wszystkie rozkłady nie spełniły warunku, tworzony jest nowy, na podstawie kolorów z bieżącej klatki i zastępuje on jeden z rozkładów, na podstawie parametrów opisujących ten rozkład. Następnie przy pomocy drugiego progu i parametrów wszystkich trzech rozkładów określane jest czy piksel należy do tła czy pierwszego planu. Dodatkowym parametrem jest "szum", która określa początkową wartość wariancji przy tworzeniu rozkładu. Zmiennymi parametrami metody są: szum, czułość, α - współczynnik uczenia się i próg.

			Analiza klatek			Analiza obiektów	
	Liczba konturów	Liczba dużych konturów	Czas [ms]	Liczba pikseli dobrze wykrytych	Liczba pikseli błędnie wykrytych	Liczba środków elips wewnątrz obiektu	Pole konturów wewnątrz obiektu
Czułość: 81 Próg: 83 Wsp. uczenia: 59 Szum: 206	69,95	1,89	112,8	78,41%	0,46%	0,94	77,00%
Czułość: 120 Próg: 83 Wsp. uczenia: 59 Szum: 206	18,86	1,54	103,9	72,43%	0,31%	1,08	72,01%
Czułość: 140 Próg: 83 Wsp. uczenia: 59 Szum: 206	13,58	1,42	121,5	69, 19%	0,27%	1,17	69,27%
Czułość: 120 Próg: 20 Wsp. uczenia: 59 Szum: 206	18,87	1,54	106,0	72,43%	0,31%	1,08	72,01%
Czułość: 120 Próg: 220 Wsp. uczenia: 59 Szum: 206	19,11	1,27	107,8	65,58%	0,22%	1,78	66,63%
Czułość: 120 Próg: 83 Wsp. uczenia: 30 Szum: 206	69,50	1,80	110,6	73,25%	0,47%	1,04	72,98%
Czułość: 120 Próg: 83 Wsp. uczenia: 90 Szum: 206	19,24	1,27	111,3	68,33%	0,27%	1,28	69,36%
Czułość: 120 Próg: 83 Wsp. uczenia: 59 Szum: 106	39,89	1,64	107,9	76,73%	0,38%	0,95	75,53%
Czułość: 120 Próg: 83 Wsp. uczenia: 59 Szum: 306	13,81	1,48	107,9	69,15%	0,27%	1,18	69,28%

Tabela 4.14. Wartości średnie w zakresie klatek 1-1800

			Analiza klatek			Analiza obiektów	
	Liczba konturów	Liczba dużych konturów	Czas [ms]	Liczba pikseli dobrze wykrytych	Liczba pikseli błędnie wykrytych	Liczba środków elips wewnątrz obiektu	Pole konturów wewnątrz obiektu
Czułość: 81 Próg: 83 Wsp. uczenia: 59 Szum: 206	32,73	1,84	114,1	81,11%	0,50%	0,96	78,74%
Czułość: 120 Próg: 83 Wsp. uczenia: 59 Szum: 206	9,67	1,65	105,4	75,94%	0,38%	1,01	75,04%
Czułość: 140 Próg: 83 Wsp. uczenia: 59 Szum: 206	8,35	1,53	127,5	73,14%	0,34%	1,04	73,04%
Czułość: 120 Próg: 20 Wsp. uczenia: 59 Szum: 206	9,58	1,65	105,8	75,94%	0,38%	1,01	75,04%
Czułość: 120 Próg: 220 Wsp. uczenia: 59 Szum: 206	16,83	1,44	109,1	70,17%	0,29%	1,63	71,88%
Czułość: 120 Próg: 83 Wsp. uczenia: 30 Szum: 206	9,42	1,67	112,2	76,24%	0,39%	1,00	75,31%
Czułość: 120 Próg: 83 Wsp. uczenia: 90 Szum: 206	17,97	1,39	115,0	73,14%	0,35%	1,16	73,95%
Czułość: 120 Próg: 83 Wsp. uczenia: 59 Szum: 106	18,85	1,68	109,0	79,99%	0,44%	0,94	77,90%
Czułość: 120 Próg: 83 Wsp. uczenia: 59 Szum: 306	8,39	1,57	109,4	72,87%	0,35%	1,07	72,84%

Tabela 4.15. Wartości średnie w zakresie klatek 900-1800



Rys. 4.29. Czas analizy

Wybór wartości parametrów ma niewielki wpływ na czas dokonywanych analiz, wynoszący w przybliżeniu 110 ms. (tabele 4.14 i 4.15). Różnice w czasach wynikają głównie z zakłóceń wprowadzanych przez system operacyjny Windows, wykonujący także inne własne procesy w trakcie prowadzenia analiz. Na wykresach na rys. 4.29 widać moment osiągnięcia maksymalnej liczby rozkładów, po którym średni czas analizy pozostaje w przybliżeniu niezmienny.

Czułość wpływa na jakość wykrywania w podobny sposób jak wartość progu w poprzednich metodach. Zwiększenie jej do wartości 140 powoduje spadek liczby dobrze wykrytych pikseli o 6% i spadek liczby błędnie wykrytych o 25%. Skutkuje ono także wykrywaniem mniejszej liczby konturów o 70% i dużych konturów o 10% (rys. 4.30). Oznacza to, że w znaczny sposób eliminuje powstawanie niewielkich konturów, które uznajemy za szum. Czułość o wartości 140 generuje spadek liczby dobrze wykrytych pikseli o 10% i błędnie o 32%, w stosunku do czułości równej 83. Analogicznie liczba wykrytych konturów spada o 74%. Dalsze zwiększanie czułości powoduje spadek liczby dobrze wykrywanych. Dalsze porównania wpływu parametrów wykonano dla czułości równej 120. Wyniki zmian czułości na obraz wyjściowy przedstawiono na rysunku 4.33.



Rys. 4.30. Stosunek liczby dużych konturów wykrytych do liczby obiektów rzeczywistych



Rys. 4.31. Stosunek liczby dużych konturów wykrytych do liczby obiektów rzeczywistych



Rys. 4.32. Stosunek liczby dużych konturów wykrytych do liczby obiektów rzeczywistych

Wartość progu wpływa w podobny lecz niewielki sposób na jakość wykrywania. Widać to na rysunku 4.30, gdzie większość wykresów pokrywa się z sobą. Zmniejszenie wartości progu do 20 praktycznie nie wpływa na wyniki. Zmiana pojawia się przy jej zwiększaniu. Jednak nawet przy prawie trzykrotnym zwiększeniu (wybrano wartość progu 220), zmiany w wynikach wykrywania są niewielkie. Liczba dobrze wykrytych pikseli maleje o 8%, a błędnie o 25%. Rośnie jednak znacząco liczba środków elips w obiektach o 60% i liczba wykrytych konturów o 75% co wskazuje na zdecydowanie większą fragmentaryzacje wykrytych konturów. Obrazują to obrazy wyjściowe zamieszczone na rysunku 4.34.

Zmniejszenie współczynnika α do wartości 30 (z bazowego 59) powoduje zmniejszenie liczby dobrze wykrytych pikseli o 1% i wzrost błędnie wykrytych o 50%, przy rozpatrywaniu całego filmu. Liczba konturów wzrosła o 268%, a liczba dużych konturów o 16%, natomiast liczba środków elips wewnątrz obiektów spadła o 4%. Oznacza to, że pojawiło się więcej konturów poza obiektami, które możemy uznać za wykrycie ruchu w miejscach gdzie nie występuje. Wynika to z używania w kompresji klatek kluczowych. Przy zmianie klatki ze zwykłej na kluczową, jej fragmenty tak się różnią, że zostają uznane za obszary w których nastąpił ruch. Jeżeli porównać wartości tylko z drugiej połowy filmu, są one bardzo zbliżone do tych gdy α =59. Powoduje to fakt, że poprzez wolniejszą naukę tła, w pierwszej połowie filmu pojazdy ruchome są słabiej wykrywane, natomiast w drugiej algorytm ma już dobrze zamodelowane tło w pamięci i wykrywa w sposób uzależniony już od innych parametrów. Zwiększenie parametru α do wartości 90, powoduje pogorszenie jakości wykrywania w trakcie całego filmu (rys. 4.35). W drugiej połowie, gdy algorytm powinien być już nauczony, liczba dobrze wykrytych pikseli jest mniejsza o 4%, a błędnie wykrytych o 8%. Jednak następuje większa fragmentaryzacja co objawia się w wzroście liczby konturów o 86%.



Rys. 4.33. Klatka nr 1205, próg 83, α=59, szum: 206, czułość (od lewej): 81, 120, 140



Rys. 4.34. Klatka nr 1205, czułość : 120, α=59, szum: 206, próg (od lewej): 20, 220



Rys. 4.35. Klatka nr 1205, czułość : 120, próg 83, szum: 206, α (od lewej): 30, 90



Rys. 4.36. Klatka nr 1205, czułość : 120, próg 83, α=59, szum: (od lewej): 106, 306

Zmniejszenie wartości szumu początkowego do 106 powoduje wzrost liczby dobrze wykrytych pikseli o 5%, a błędnie wykrytych o 16%. Prowadzi także do spadku liczby środków elips wewnątrz obiektów rzeczywistych poniżej 1, co oznacza, że nie wszystkie obiekty zostały wykryte. Zwiększenie szumu do wartości 306 powoduje zmniejszenie liczby dobrze wykrytych pikseli o 4%, a błędnie wykrytych o 8%.

Do dalszych porównań wybrano parametry: czułość: 120, próg: 83, α=59, szum: 206. Obraz wyjściowy dla tych parametrów znajduje się na rysunku 4.33.

4.4.8. PixelBasedAdaptiveSegmenter

W metodzie PixelBasedAdaptiveSegmenter[12] każdy piksel jest opisany wektorem N wartości z poprzednich klatek, w których był uznany za tło i dwoma parametrami: R- próg decyzyjny i T-parametr uczenia się. Każdy piksel ma swoje wartości R i T. Najważniejszą cechą metody jest dynamiczna zmiana progów R i T. Sprawdzenie, czy piksel należy do tła, odbywa się poprzez policzenie dystansów między obecną wartością piksla, a poszczególnymi wartościami z wektora N. Jeżeli odpowiednia liczba dystansów jest większa od progu R to piksel jest uznawany za tło. Jeżeli piksel należy do tła, to jedna z wartości wektora N jest zastępowana przez obecną wartość piksela, ale tylko z prawdopodobieństwem 1/T. Dodatkowo, z prawdopodobieństwem 1/T, aktualizowane są piksele z otoczenia badanego piksela, za pomocą ich aktualnych wartości. Następnie są aktualizowane wartości progów R i T analizowanego piksela, przy pomocy średniej arytmetycznej z minimalnych dystansów. Średnia ta określaj dynamikę zmian wartości tego piksla.

Z powodu dużej liczby parametrów w tej metodzie, zaprezentowano wyniki tylko dla zmian niektórych najbardziej istotnych. W tej implementacji stosowane jest rozmycie obrazu przed analizą. Parametr N określa liczbę elementów wektora wartości poprzednich (wektor N), Rincdec określa wartość o jaką zmieniany jest parametr R podczas jego aktualizacji.

			Analiza klatek			Analiza obiektów	
	Liczba konturów	Liczba dużych konturów	Czas [ms]	Liczba pikseli dobrze wykrytych	Liczba pikseli błędnie wykrytych	Liczba środków elips wewnątrz obiektu	Pole konturów wewnątrz obiektu
N: 20, Rincdec: 0,05	2,08	1,23	1535,4	79,16%	0,34%	0,88	74,93%
N: 20, Rincdec: 0,5	2,15	1,20	1461,4	78,95%	0,33%	0,90	74,91%
N: 20, Rincdec: 0,005	2,17	1,34	1518,8	82,31%	0,36%	0,87	78,22%
N: 20, Rincdec: 0,05 Bez rozmycia	4,18	1,39	1355,3	80,22%	0,34%	0,90	75,14%

Tabela 4.16. Wartości średnie w zakresie klatek 1-1800

			Analiza klatek			Analiza obiektów	
	Liczba konturów	Liczba dużych konturów	Czas [ms]	Liczba pikseli dobrze wykrytych	Liczba pikseli błędnie wykrytych	Liczba środków elips wewnątrz obiektu	Pole konturów wewnątrz obiektu
N: 20, Rincdec: 0,05	2,01	1,39	1490,5	86,19%	0,42%	0,80	83,18%
N: 20, Rincdec: 0,5	2,02	1,37	1537,0	86,09%	0,42%	0,80	83,19%
N: 20, Rincdec: 0,005	2,04	1,45	1495,9	86,88%	0,45%	0,80	84,00%
N: 20, Rincdec: 0,05 Bez rozmycia	3,52	1,43	1421,1	85,17%	0,41%	0,82	80,90%

Tabela 4.17. Wartości średnie w zakresie klatek 900-1800

Na czas analizy w niewielkim stopniu wpływają parametry metody, wyraźny jego spadek widać przy wyłączeniu rozmycia przed analizą. Jest to spowodowane faktem, że program nie wykonuje obliczeń odnośnie rozmycia, co obniża łączną liczbę obliczeń algorytmu. Jednak rozmycie w znaczącym stopniu wpływa na jakość wykrywania ruchu. Jego brak zwiększa liczbę wykrytych konturów aż o 75%, ale dużych konturów tylko o 3%, przy praktycznie niezmienionej liczbie błędnie wykrytych pikseli. Oznacza to, że w przypadku braku stosowania rozmycia obrazu na początku algorytmu, błędnie wykryte piksele zgrupowały się w niewielkie kontury. Stosowanie rozmycia ułatwia odfiltrowanie błędnie wykrytych pikseli.

Rincdec w badanym zakresie w niewielkim stopniu wpływa na jakość wykrywania ruchu (rys. 4.39). Zmiany liczby wykrytych dużych konturów są w przybliżeniu równe 3% (rys. 4.37 i 4.38) pomiędzy kolejnymi wartościami Rincdec. Analogicznie zmiany liczby dobrze wykrytych pikseli są poniżej 1%. Liczba wyznaczonych środków elips wewnątrz obiektów rzeczywistych jest równa 0,8, zatem mniejsza od oczekiwanej liczby 1. Wynika to z faktu dobrego wypełnienia obiektów i niskiej fragmentaryzacji dużych konturów. Z powodu łączenia się konturów pojazdów jadących obok siebie lub mijających się w niewielkiej odległości, średnia liczba środków elips jest mniejsza od 1.



Rys. 4.37. Stosunek liczby dużych konturów wykrytych do liczby obiektów rzeczywistych



Rys. 4.38. Stosunek liczby dużych konturów wykrytych do liczby obiektów rzeczywistych



Rys. 4.39. Klatka nr 1205, N=20, z rozmyciem, R (od lewej): 0.05, 0.5, 0.005



Rys. 4.40. Klatka nr 1205, N=20, z rozmyciem, R=0.05

Długi czas analizy klatki (średnio około 1,5 sekundy) uniemożliwia stosowanie tej metody, w tej implementacji, w systemach wymagających analizowania danych w czasie rzeczywistym. Rozmycie w znaczącym stopniu poprawia skuteczność działania, wpływając w znacznym stopniu na liczbę wykrywanych konturów (jego brak zwiększa ich liczbę o 75%). Metoda PBAS daje najlepsze wyniki wykrywania z wszystkich badanych metod.

4.4.9. MultiLayerBGS

Metoda MultiLayerBGS[11] prezentuje zupełnie inne niż dotychczas podejście do tematu wykrywania ruchu. Używa ona jednocześnie, dla każdego piksela, parametru opisującego wzór (teksturę) w okolicy danego piksla i wartości jego kolorów. Podstawowym parametrem na podstawie którego rozróżniamy tło i pierwszy plan jest Local Binary Pattern (LBP), a pomocniczym Photometric Invariant Color (PIC). Dodatkowo stosuje warstwy, których celem jest wykrycie ruchomych obiektów na tle, które były długo zakryte.

LBP opisuje wzór tekstury wokół każdego piksla jedną wartością liczbową. W celu obliczenia jej wartości algorytm wykorzystuje wartości pikseli w odcieniach szarości, dzięki czemu niweluje globalne i lokalne zmiany natężenia oświetlenia. Jej obliczanie jest stosunkowo szybkie i wymaga niewielkiej liczby parametrów. Wartości LBP są mało zróżnicowane w obszarach bez wyraźnych wzorów.

PIC jest wartością obliczaną dla każdego piksla na podstawie jego obecnej i historycznych wartości na podstawie kątów i odległości między poszczególnymi wartościami w przestrzeni RGB.

Wykrycie ruchomego tła polega na stworzeniu odpowiedniej mapy dystansów zarówno dla wartości LBP i PIC, która jest odpowiednikiem rozkładu

prawdopodobieństw w metodzie rozkładów Gaussa. Za pierwszy plan zostają uznane piksele których dystans globalny, obliczony na podstawie mapy dystansów, jest większy od ustalonego progu.

		Analiza obiektów				
Liczba konturów	Liczba dużych konturów	Czas [ms]	Liczba pikseli dobrze wykrytych	Liczba pikseli błędnie wykrytych	Liczba środków elips wewnątrz obiektu	Pole konturów wewnątrz obiektu
7,54	1,90	776,1	70,48%	0,33%	1,01	67,63%

Tabela 4.18. Wartości średnie w zakresie klatek 1-1800

Tabela 4.19. Wartości średnie w zakresie klatek 900-180	00
---	----

		Analiza obiektów				
Liczba konturów	Liczba dużych konturów	Czas [ms]	Liczba pikseli dobrze wykrytych	Liczba pikseli błędnie wykrytych	Liczba środków elips wewnątrz obiektu	Pole konturów wewnątrz obiektu
6,62	1,89	809,7	76,12%	0,39%	0,93	75,02%



Rys. 4.41. Stosunek liczby dużych konturów wykrytych do liczby obiektów rzeczywistych



Numer klatki Rys. 4.42. Czas analizy



Rys. 4.43. Klatka nr 300.



Rys. 4.44. Klatka nr 1205

Wyraźnie widać na obrazach (rys. 4.43), i wykresie (rys. 4.41), że ta metoda w klatkach kluczowych generuje dużą liczbę błędnie wykrytych pikseli zgrupowanych w duże obiekty. Powstały głównie na linii horyzontu, gdzie niebo łączy się z linią drzew. Są tam największe różnice pomiędzy klatkami zwykłymi i kluczowymi. Ich wpływ można ograniczyć poprawiając jakoś danych wprowadzanych do programu np. jako surowe

klatki bez kompresji lub stosując odpowiednie filtry. Czas wykonania algorytmu rośnie do czasu stworzenia historii danych o maksymalnej dopuszczanej długości (rys. 4.42). Po jej osiągnięciu jest bardzo stabilny. Jednak średni czas wynosi wtedy około 0,8 sekundy, zatem algorytm w tej implementacji, jest zbyt wolny do działania w czasie rzeczywistym. Dodatkowo algorytm ma problemy przy dużych jednolitych powieszaniach poruszających się pojazdów np. naczepach samochodów ciężarowych. Jak widać na rysunku 4.44, środkowa część naczepy pozostaje nie wykryta jako ruchoma. Jest to spowodowane jej jednolitym kolorem i wzorem, przez co algorytm nie zauważa dostatecznych zmian w danym obszarze, gdy przejeżdża przez niego obiekt tego rodzaju. Problem został rozwinięty w rozdziale 5.4.5.

4.5. Zestawienie wyników

W tabelach 4.20 i 4.21 zestawiono najlepsze wyniki wszystkich powyżej opisanych metod. Widać dużą rozbieżność zarówno w skuteczności działania metod jak i czasach potrzebnych na przeliczenie jednej klatki. Najskuteczniejszy okazał się algorytm PBAS, niestety potrzebuje on około 1,5 sekundy na przeliczenie jednej badanej klatki filmu co uniemożliwia, w tej implementacji, stosowanie go w czasie rzeczywistym. Algorytm LBMixtureOfGaussians (LBMG) cechuje się wysoką skutecznością. Potrzebuje około 0,1 s na przetworzenie jednej klatki, co pozwala na zastosowanie go w czasie zbliżonym do rzeczywistego. Z porównania działania algorytmów w drugiej połowie filmu, gdy powinny mieć wyuczone optymalne modele tła, wynika, że PBAS wykrywa dobrych pikseli o 9% więcej od LBMG. Niestety błędnie wykrywa też o 9% pikseli wiecej. Liczba środków elips wewnatrz obiektów jest mniejsza od 1. Wynika to z faktu, że część pojazdów jedzie tak blisko siebie, że dla algorytmu zlewają się w jeden ruchomy kontur. Metoda PBAS generuje pełne i jednolite kontury, przez co łatwiej następuje złączenie konturów pojazdów jadących obok siebie. To zjawisko zostało dokładniej opisane w rozdziale 5.4.2.. W pozostałych metodach liczba środków elips wewnątrz obiektów jest wyższa od 1, co oznacza, że średnio więcej niż jeden kontur tworzył każdy obiekt. Jest to skutkiem fragmentaryzacji wykrytych obiektów. Obrazuje to też stosunek liczby wykrytych konturów do liczby obiektów rzeczywistych. Dla metody PBAS jest on najbliższy oczekiwanej wartości 1. Oznacza to minimalna ilość błędnie wykrytych pikseli łączących się w kontury.

			Analiza klatek			Analiza	obiektów
	Liczba konturów	Liczba dużych konturów	Czas [ms]	Liczba pikseli dobrze wykrytych	Liczba pikseli błędnie wykrytych	Liczba środków elips wewnątrz obiektu	Pole konturów wewnątrz obiektu
Różnica klatek	32,80	1,12	8,2	47,11%	0,28%	2,65	57,85%
AdaptiveMedianBGS	16,66	1,61	59,6	69,24%	0,34%	1,17	69,02%
Weighted Moving MeanBGS	28,03	1,22	118,3	54,26%	0,42%	2,21	65,12%
Weighted Moving VarianceBGS	17,56	1,14	254,3	55,29%	0,41%	2,06	63,82%
DPMean	37,18	1,14	119,2	49,19%	0,32%	2,48	59,17%
DPWren	22,70	1,65	92,5	67,15%	0,33%	1,28	67,17%
GrimssonGMM	46,95	1,48	252,2	74,43%	0,38%	1,07	72,95%
LBMixtureOfGaussians	18,86	1,54	103,9	72,43%	0,31%	1,08	72,01%
PBAS	2,08	1,23	1535,4	79,16%	0,34%	0,88	74,93%
MultiLaverBGS	7.54	1.90	776.1	70.48%	0.33%	1.01	67.63%

Tabela 4.20. Zestawienie wartości średnie w zakresie klatek 1-1800

			Analiza klatek			Analiza	obiektów
	Liczba konturów	Liczba dużych konturów	Czas [ms]	Liczba pikseli dobrze wykrytych	Liczba pikseli błędnie wykrytych	Liczba środków elips wewnątrz obiektu	Pole konturów wewnątrz obiektu
Różnica klatek	30,30	1,31	8,3	52,01%	0,36%	2,39	66,38%
AdaptiveMedianBGS	8,15	1,41	59,4	73,12%	0,35%	1,06	72,53%
Weighted Moving MeanBGS	25,36	1,40	117,6	59,17%	0,55%	1,93	73,05%
Weighted Moving VarianceBGS	16,21	1,33	254,1	61,09%	0,54%	1,79	72,82%
DPMean	34,39	1,31	119,7	54,13%	0,41%	2,27	67,64%
DPWren	12,07	1,62	92,8	70,49%	0,35%	1,15	70,38%
GrimssonGMM	10,42	1,59	276,5	78,95%	0,42%	0,97	77,04%
LBMixtureOfGaussians	9,67	1,65	105,4	75,94%	0,38%	1,01	75,04%
PBAS	2,01	1,39	1490,5	86,19%	0,42%	0,80	83,18%
MultiLayerBGS	6,62	1,89	809,7	76,12%	0,39%	0,93	75,02%

Tabela 4.21. Zestawienie wartości średnie w zakresie klatek 900-1800

4.6. Wykrywanie ruchu na skrzyżowaniu

Do dalszych badań wybrano trzy metody:

- Różnice klatek ze względu na szybkość działania pozwala na przetwarzania obrazów w czasie rzeczywistym 25 klatek na sekundę,
- LB Mixture of Gaussians przykład metody wykorzystującej mieszankę modeli Gaussa, kompromis między szybkością a skutecznością,
- PBAS metoda najskuteczniejsza, ale niestety najwolniejsza.

Kolejne porównanie działania powyższych metod przeprowadzono na filmie "fragmenty 1" przedstawiającym skrzyżowanie w formie litery "T" ulic Robotniczej i Śrubowej we Wrocławiu. Film został nagrany kamerą umieszczoną na budynku przy drodze. W czasie badań sprawdzono około 30 minut nagrania i wybrano 4 fragmenty. W celu uniknięcia konieczności uczenia się tła przy każdym fragmencie połączono je w jeden film. Powoduje to powstanie chwilowej większej liczby błędów w pierwszych klatkach każdego kolejnego fragmentu. Numery klatek w których następuje łączenie fragmentów to: 944, 1359, 1835. Na filmie znajdują się przemieszczające samochody osobowe, piesi, rowerzyści, autobus i samochód ciężarowy, poruszające się zarówno w linii prostej jak i skręcające na skrzyżowaniu. Proces oznaczania ruchomych pojazdów na filmie zajął około 10 godzin. Podstawowe dane filmu:

Długość filmu	1min. 52 s.
FPS	25
Liczba klatek	2801
Rozdzielczość	1280x720 px. (720p)
Liczba oznaczonych obiektów	52
Pole najmniejszego oznaczonego obiektu	615 px.
Pole największego oznaczonego obiektu	396657 px.

Tabela 4.22. Parametry filmu "Fragmenty 1"

Przykładowe klatki z filmu:



Rys. 4.45. Przykładowe klatki z filmu "Fragmenty 1": 830, 1170, 2570

Jako próg powierzchni dużych obiektów wybrano wartość 461, co stanowi 75% wielkości najmniejszego oznaczonego obiektu. Jest on obniżony w stosunku do pierwszego filmu ze względu na występowanie pieszych. Nie stanowią oni zwartej bryły jak samochód, przez co są trudniejsi do opisania elipsą i stanowią mniejszy obszar wewnątrz tej elipsy. Badania przeprowadzono przy pomocy metod: Różnica klatek, LB Mixture of Gaussians i PBAS, stosując wcześniej wybrane parametry. Opis kolumn tabel znajduje się w rozdziale: 4.1.

				Analiza obiektów			
	Liczba konturów	Liczba dużych konturów	Czas [ms]	Liczba pikseli dobrze wykrytych	Liczba pikseli błędnie wykrytych	Liczba środków elips wewnątrz obiektu	Pole konturów wewnątrz obiektu
Różnica klatek	322,10	1,13	20,5	36,08%	0,39%	10,98	43,13%
LBMG	98,73	1,05	315,3	57,78%	0,87%	10,64	55,70%
PBAS	5,81	1,06	3396,2	75,09%	1,10%	2,38	65,87%

Tabela 4.23. Wartości średnie w zakresie klatek 1-1800

				Analiza obiektów			
	Liczba konturów	Liczba dużych konturów	Czas [ms]	Liczba pikseli dobrze wykrytych	Liczba pikseli błędnie wykrytych	Liczba środków elips wewnątrz obiektu	Pole konturów wewnątrz obiektu
Różnica klatek	344,27	1,14	20,3	30,38%	0,41%	13,86	36,90%
LBMG	119,89	1,10	341,2	49,29%	1,10%	14,45	50,75%
PBAS	6,88	1,10	3601,4	66,83%	1,32%	3,13	60,14%

Tabela 4.24. Wartości średnie w zakresie klatek 900-1800



Rys. 4.46. Stosunek liczby dużych konturów do liczby obiektów

W porównaniu do pierwszego analizowanego w pracy filmu, widać wyraźny wzrost stosunku liczby wykrytych konturów do liczby obiektów rzeczywistych, czasu analizy, liczby pikseli błędnie wykrytych i liczby środków elips wewnątrz obiektów. Natomiast liczba pikseli dobrze wykrytych i pole konturów wewnątrz obiektów zmniejszyły się. Przyczyną wyżej wymienionych zjawisk jest, przede wszystkim, zwiększona rozdzielczość. Powoduje ona konieczność wykonania większej liczby obliczeń dla każdej klatki, co zwiększa czas analizy. Większa liczba danych wejściowych skutkuje wykrywaniem większej ilości szumów, które potrafią się łączyć w dostatecznie duże kontury by zostały uznane za poruszający się obiekt. Badanie wpływu rozdzielczości na skuteczność działania metod wykrywania ruchu przeprowadzono w następnym rozdziale. Kolejnym czynnikiem utrudniającym skuteczne wykrycie ruchu na filmie ze skrzyżowania jest fakt, że pojazdy zajmują procentowo większy obszar niż na pierwszym filmie. Największy oznaczony obiekt na filmie z autostrady zajmuje w przybliżeniu 7,8% powierzchni całej klatki, podczas gdy największy obiekt na filmie ze

skrzyżowaniem 43%. Duże obszary pojazdów, jak np. dachy, maski, plandeki na samochodach ciężarowych, są stosunkowo jednolite pod względem koloru w przestrzeni RGB. Doskonałym przykładem jest tu ciężarówka przejeżdżająca pod koniec filmu drugiego (rys. 4.46). Jej duża jednolita plandeka jest w przestrzeni RGB zbliżona do koloru tła (drogi i chodnika). Dodatkowo jej rozmiar i jednolitość sprawia, że mimo przemieszczania się pojazdu, duże fragmenty klatki pozostają stosunkowo niezmienne w przestrzeni kolorów RGB. Utrudnia to wykrycie tego obszaru jako fragmentów obiektu ruchomego. Przyczyny te skutkują, tym że film drugi jest bardziej skomplikowany i jego analiza nie daje tak stabilnych wyników, jak w przypadku filmu pierwszego.

4.7. Wpływ rozdzielczości

Wysoka rozdzielczość obrazów wymaga większej liczby obliczeń w celu wykrycia ruchu, co utrudnia przetwarzanie obrazów czasie rzeczywistym. Sprawdzono na filmie ze skrzyżowaniem, jak rozdzielczość rejestrowanego filmu wpływa na jakość wykrywania ruchu. Dla wszystkich trzech wybranych metod przeprowadzono badania zmniejszając rozdzielczość filmu wejściowego 2 i 4-krotnie. Zmiana rozdzielczości filmu dokonywana jest przy pomocy biliniowej interpolacji.

Jak wynika z przeprowadzonych badań zmiana rozdzielczości analizowanego filmu w znaczący sposób wpływa na czas analizy jednej klatki i jest ważnym czynnikiem przy implementacji algorytmów do pracy w czasie rzeczywistym. Dwukrotna zmiana rozmiarów obrazu powoduje czterokrotną zmianę liczby pikseli. Czas analizy klatki jest proporcjonalny do liczby jej pikseli, zatem spadek czasu analizy powinien być proporcjonalny do liczby pikseli w klatce.

Podczas przeprowadzania badań próg wielkości dużych konturów też jest skalowany proporcjonalnie do zmiany rozdzielczości.

				Analiza obiektów			
	Liczba konturów	Liczba dużych konturów	Czas [ms]	Liczba pikseli dobrze wykrytych	Liczba pikseli błędnie wykrytych	Liczba środków elips wewnątrz obiektu	Pole konturów wewnątrz obiektu
Różnica klatek 1280x720	322,10	1,13	20,5	36,08%	0,39%	10,98	43,13%
Różnica klatek 640x360	20,59	0,81	5,8	34,43%	0,31%	4,75	39,01%
Różnica klatek 320x180	10,14	0,49	2,7	34,77%	0,35%	2,64	37,49%

Tabela 4.25. Różnica klatek - wartości średnie w zakresie klatek 1-1800

			Analiza klatek			Analiza obiektów		
	Liczba konturów	Liczba dużych konturów	Czas [ms]	Liczba pikseli dobrze wykrytych	Liczba pikseli błędnie wykrytych	Liczba środków elips wewnątrz obiektu	Pole konturów wewnątrz obiektu	
Różnica klatek 1280x720	344,27	1,14	20,3	30,38%	0,41%	13,86	36,90%	
Różnica klatek 640x360	25,29	0,77	5,8	28,94%	0,33%	5,79	32,90%	
Różnica klatek 320x180	12,47	0,51	2,7	29,15%	0,37%	3,08	31,70%	

Tabela 4.26. Różnica klatek - wartości średnie w zakresie klatek 900-1800

W metodzie różnicy klatek (tabele 4.25 i 4.26) nie widać tak znacznej redukcji czasu analizy, gdyż dla każdego piksela wykonywane jest tylko jedno odejmowanie i progowanie. Zmniejszenie rozdzielczości z 1280x720 do 640x360 powoduje spadek czasu o 72% natomiast jej zmniejszenie do 320x180 już tylko o kolejne 52%. Wynika to z faktu, że czasy analizy są już niewielkie i czasy wykonania pozostałych operacji systemowych takich jak: dostęp do dysku, wczytanie czy wyświetlenie obrazu zajmują znaczną część całkowitego czasu analizy jednej klatki.



Rys. 4.47. Różnica klatek - stosunek liczby dużych konturów do liczby obiektów

Zmiana rozdzielczości w niewielkim stopniu wpływa na liczbę dobrze wykrytych pikseli (różnice rzędu 4-5%). Zmniejszenie rozdzielczości o połowę korzystnie wpływa

na liczbę błędnie wykrytych pikseli obniżając ją o 20%. Spada też liczba konturów aż o 93%. Jednak niestety liczba dużych konturów zmniejsza się o 32% i wynosi 0,77 (rys. 4.47), zatem jest mniejsza od oczekiwanej wartości 1. Wynika to z faktu łączenia się konturów jadących blisko pojazdów i niewykrywania w dostatecznym stopniu mniejszych obiektów. Kolejne zmniejszenie rozdzielczości powoduje wzrost liczby błędnie wykrytych pikseli o 20% w stosunku do pośredniej rozdzielczości, jednak w dalszym ciągu ta liczba jest niższa niż w maksymalnej badanej rozdzielczości. Stosunek liczby wykrytych dużych konturów do liczby obiektów rzeczywistych spada do 0,51 w drugiej połowie filmu, zatem średnio co drugi pojazd jest wykrywany poprawnie.

			Analiza obiektów				
	Liczba konturów	Liczba dużych konturów	Czas [ms]	Liczba pikseli dobrze wykrytych	Liczba pikseli błędnie wykrytych	Liczba środków elips wewnątrz obiektu	Pole konturów wewnątrz obiektu
LBMG 1280x720	98,73	1,05	315,3	57,78%	0,87%	10,64	55,70%
LBMG 640x360	21,38	0,84	84,7	57,38%	0,86%	4,02	54,18%
LBMG 320x180	8,46	0,59	21,8	58,05%	0,91%	1,96	53,81%

Tabela 4.27. LBMG - wartości średnie w zakresie klatek 1-1800

Tabela 4.28. LBMG- wartości średnie w zakresie klatek 900-1800

			Analiza klatek			Analiza obiektów	
	Liczba konturów	Liczba dużych konturów	Czas [ms]	Liczba pikseli dobrze wykrytych	Liczba pikseli błędnie wykrytych	Liczba środków elips wewnątrz obiektu	Pole konturów wewnątrz obiektu
LBMG 1280x720	119,89	1,10	341,2	49,29%	1,10%	14,45	50,75%
LBMG 640x360	26,94	0,87	89,4	48,62%	1,08%	5,22	48,75%
LBMG 320x180	10,61	0,63	23,1	49,03%	1,14%	2,40	48,25%

W metodzie LBMG (tabele 4.27 i 4.28) zmiana czasu analizy jednej klatki przy zmianie rozdzielczości jest bardzo zbliżona do zmiany liczby pikseli w klatce. Dwukrotne zmniejszenie rozdzielczości powoduje czterokrotną zmianę liczby pikseli i w przybliżeniu taką samą zmianę czasu analizy jednej klatki.



Rys. 4.48. LBMG - stosunek liczby dużych konturów do liczby obiektów

Podobnie jak w przypadku różnicy klatek, zmiana rozdzielczości w bardzo małym stopniu wpływa na liczbę poprawnie wykrytych pikseli poprzez algorytm LBMG, różnice wynoszą około 1-2%. Także zmiana liczby źle wykrytych pikseli jest niewielka, wynosi od 2 do 5%. Jak widać na rysunku 4.48, zmiana rozdzielczości wpływa na liczbę wykrytych dużych konturów. Zmniejszenie jej do 640x360 powoduje spadek ich liczby o 20%, do średniej wartości 0,87. Kolejne obniżenie rozdzielczości powoduje spadek liczby dużych konturów o dalsze 28%, do wartości 0,63. Zmniejszanie rozdzielczości wpływa pozytywnie na fragmentaryzację konturów, zmniejszając średnią liczbę środków elips wykrytych wewnątrz obiektów o 64%, przy pierwszym zmniejszeniu rozdzielczości i o kolejne 54% przy zmniejszeniu rozdzielczości do 320x180. Korzystny przy tym jest fakt, że zmiana liczby dobrze wykrytych pikseli jest niewielka i pole wykrytych konturów wewnątrz obiektów rzeczywistych zmniejsza się o analogicznie o 4% i potem kolejny 1%.

				Analiza obiektów			
	Liczba konturów	Liczba dużych konturów	Czas [ms]	Liczba pikseli dobrze wykrytych	Liczba pikseli błędnie wykrytych	Liczba środków elips wewnątrz obiektu	Pole konturów wewnątrz obiektu
PBAS 1280x720	5,81	1,06	3396,2	75,09%	1,10%	2,38	65,87%
PBAS 640x360	2,21	0,89	954,8	74,29%	1,16%	1,35	63,57%
PBAS 320x180	1,23	0,65	261,4	70,26%	1,20%	0,94	56,18%

Tabela 4.29. PBAS- wartości średnie w zakresie klatek 1-1800
			Analiza obiektów				
	Liczba konturów	Liczba dużych konturów	Czas [ms]	Liczba pikseli dobrze wykrytych	Liczba pikseli błędnie wykrytych	Liczba środków elips wewnątrz obiektu	Pole konturów wewnątrz obiektu
PBAS 1280x720	6,88	1,10	3601,4	66,83%	1,32%	3,13	60,14%
PBAS 640x360	2,44	0,92	999,2	64,76%	1,38%	1,57	56,50%
PBAS 320x180	1,33	0,67	271,1	60,25%	1,42%	0,99	48,78%

Tabela 4.30. PBAS- wartości średnie w zakresie klatek 900-1800

W metodzie PBAS (tabele 4.29 i 4.30), tak jak w przypadku metody LBMG, zmiana czasu analizy jednej klatki przy zmianie rozdzielczości jest bardzo zbliżona do zmiany liczby pikseli w klatce. Pierwsze czterokrotne zmniejszenie liczby pikseli powoduje spadek czasu analizy o 72%, a następne czterokrotne zmniejszenie o kolejne 73%.



Rys. 4.49. PBAS - stosunek liczby dużych konturów do liczby obiektów

Liczba poprawnie wykrytych pikseli poprzez algorytm PBAS, przy redukowaniu rozdzielczości, zmienia się w większym stopniu, niż w przypadku różnicy klatek i LBMG. Mimo to zmiany są w dalszym ciągu niewielkie. Zmniejszenie rozdzielczości do 640x360 powoduje wzrost liczby poprawnie wykrytych pikseli o 3%, a w przypadku rozdzielczości 320x180 o kolejne 7%. Także liczba źle wykrytych pikseli ulega niewielkiej zmianie. Przy pierwszym obniżeniu rozdzielczości ich liczba wzrosła o 4,5%, a przy drugim o kolejne 3%. Jak widać na rysunku 4.49, zmiana rozdzielczości wpływa na liczbę wykrytych dużych konturów, jednak w wyraźnie mniejszym stopniu niż w

poprzednich dwóch metodach. Zmniejszenie rozdzielczości do 640x360 powoduje ich spadek o 14%, do średniej wartości 0,92, co jest wartością najbliższą 1, ze wszystkich badanych przypadków. Kolejne obniżenie rozdzielczości powoduje ich spadek o dalsze 27%, do wartości 0,67. Zmniejszanie rozmiarów klatki wpływa, analogicznie jak w metodzie LBMG, pozytywnie na fragmentaryzację konturów, zmniejszając średnią liczbę środków elips wykrytych wewnątrz obiektów. Przy pierwszym zmniejszeniu rozdzielczości, liczba maleje o 50%, do średniej wartości 1,57 i o kolejne 37%, do średniej wartości 0,99, przy zmniejszeniu rozdzielczości do 320x180. Jest to wartość najbliższa oczekiwanej wartości jeden. Korzystny przy tym jest fakt, że zmiana liczby dobrze wykrytych pikseli jest niewielka. Poniżej na rysunkach 4.50, 4.51 i 4.52 pokazano przykładowe wpływy zmiany rozdzielczości na skuteczność wykrywania metody PBAS. Wybrano ta metodę gdyż różnice są najwyraźniejsze przy jej badaniu. Rysunki z wszystkich badań znajdują się na płycie dołaczonej do pracy. Wyraźnie widać na nich, że zmniejszanie rozdzielczości powoduje sklejanie się mniejszych konturów, także różnych pojazdów.



Rys. 4.50. PBAS- rozdzielczość: 1280x720, klatki: 830, 1170, 2570



Rys. 4.51. PBAS- rozdzielczość: 640x360, klatki: 830, 1170, 2570



Rys. 4.52. PBAS- rozdzielczość: 320x180, klatki: 830, 1170, 2570

Jak wynika z badań, film w rozdzielczości 1280x720 nie wnosi dużo dodatkowych informacji przydatnych dla metod wykrywania ruchu, za to wprowadza więcej błędów. Rozdzielczość 640x360 zapewnia dużo lepsze wyniki wykrywania, przy niewielkim wzroście liczby błędów. Analizowanie sekwencji obrazów w mniejszej rozdzielczości wprowadza już więcej błędów i słabiej sobie radzi w wykrywaniu odpowiednich konturów co obrazuje znaczącym spadkiem wykrytych dużych konturów.

5. Badanie algorytmów śledzenia obiektów

Po wykryciu poruszających się obiektów, kolejnym krokiem jest śledzenie tych obiektów w czasie ich pojawiania się w filmie. Najpierw należy wykryć obiekty na podstawie ich konturów na czarno-białych obrazach wyjściowych z algorytmów wykrywania ruchu. Kolejnym etapem jest rozpoznanie, który kontur z poprzedniej klatki odpowiada konturowi z obecnej klatki. Najpierw sprawdzono metody zawarte w bibliotece OpenCV, w przykładzie blobtrack_sample. Niestety po wstępnych badaniach okazały się one zupełnie nieskuteczne. W pracy zaproponowano dwie metody: prostą, polegającą na porównywaniu położenia środków elips w klatce obecnej i poprzedniej oraz drugą używającą filtru Kalmana jako predyktora położenia obiektów w kolejnych klatkach. By ułatwić porównywanie kolejnych konturów, każdy z nich jest opisywany najlepiej dopasowaną elipsą i one są porównywane. Badania przeprowadzono na danych wyjściowych z algorytmu PBAS, gdyż daje on dane wyjściowe z najmniejszą liczbą błędów z badanych metod.

5.1. Śledzenie obiektów w OpenCV

W bibliotece OpenCV są zaimplementowane metody wykrywania i śledzenia ruchu. Znajdują się one w przykładzie blobtrack_sample. Składa się on z następujących modułów (rys. 5.1):



Rys. 5.1 Schemat modułów programu blobtrack_sample

-"Moduł detekcji ruchu (FG/BG)"-wykrywa ruch analogicznie jak badane wcześniej algorytmy i tak samo generuje czaro-białe obrazy z podziałem na pierwszy plan i tło. Ponieważ udostępnia tylko 3 metody (w porównaniu używana wcześniej biblioteka BGS 32) moduł został wyłączony. W jego miejsce wprowadzono klatki wyjściowe z algorytmu PBAS,

-"Moduł wykrywania nowych obiektów"- używa czarno-białych obrazów z poprzedniego modułu w celu wykrycia nowych obiektów pojawiających się w scenie w każdej klatce,

-"Moduł śledzenia obiektów"- śledzi każdy wykryty obiekt

-"Moduł generacji trajektorii"- zapisuje dane o pozycji wszystkich obiektów i zapisuje je na dysk twardy,

-"Moduł analizy trajektorii" - wygładza trajektorię obiektu, może być wyłączony.

Danymi wyjściowymi z programu są numery, pozycje i rozmiar kolejnych obiektów w każdej klatce. Dodatkowo generowany jest film z zaznaczonymi elipsami pozycjami obiektów. Po analizie obrazów wyjściowych stwierdzono, że program działa bardzo niedokładnie. Część dużych i wyraźnych obiektów jest niewykrywana lub śledzona tylko przez fragment czasu ich istnienia na filmie. Oznaczanie ich z reguły obejmuje obszar znacznie większy niż same obiekty. Wykryte elipsy są niestabilne, często przeskakują między pojazdami lub zanikają co oznacza, że obiekt przestaje być śledzony. Przykładowe klatki ilustrujące ten problem przedstawiono na rys. 5.2. Z powodu tak niepewnego i niestabilnego działania programu autor zdecydował się na jego niezastosowanie i napisał dwa własne algorytmy.



Rys. 5.2 Przykładowe klatki z wynikami działania programu blobtrack_sample

5.2. Wykrywanie konturów

Pierwszym krokiem w śledzeniu obiektów jest wykrycie konturów na obrazie z algorytmów wykrywania ruchu. Umożliwia to oznaczenie konkretnych obiektów. Funkcja wykrywania konturów została zaimplementowana przy pomocy funkcji findContours() z biblioteki OpenCV. Zwraca ona wektor wszystkich konturów wykrytych na danym obrazie i dodatkowo może stworzyć hierarchie wykrytych konturów (które się zawierają w innych konturach, relacje rodzic-dziecko). Posiada 4 rodzaje analizy zwracania listy konturów:

-CV_RETR_EXTERNAL – zwraca tylko kontury zewnętrzne (bez konturów w konturach),

-CV_RETR_LIST - zwraca wszystkie kontury, ale bez hierarchii,

-CV_RETR_CCOMP – zwraca wektor konturów z 2 stopniową hierarchią, pierwszy poziom to kontury zewnętrzne, a drugi to kontury "dziur" w konturach zewnętrznych,

-CV_RETR_TREE -zwraca wszystkie kontury i buduje ich hierarchię w formie drzewa.

Dodatkowo w bibliotece istnieją 3 sposoby zapisywania danych o konturach:

-CV_CHAIN_APPROX_NONE - przechowuje wszystkie punkty konturu

-CV_CHAIN_APPROX_SIMPLE – kompresuje wszystkie poziome, pionowe i ukośne odcinki konturu i zapisuje tylko punkty końcowe odcinków

-CV_CHAIN_APPROX_TC89_L1,CV_CHAIN_APPROX_TC89_KCOS-stosuje algorytm łańcucha aproksymacji Teh-Chin

W badaniach wybrano metodę CV_RETR_EXTERNAL, ponieważ analiza dotyczy śledzenia tylko zewnętrznych konturów określających obiekt. Wewnętrzne "dziury" traktuje się jako błędy algorytmu wykrywania ruchu. Jako metodę zapisu wybrano: CV_CHAIN_APPROX_NONE, żeby posiadać dostęp do wszystkich elementów każdego konturu. Konieczne jest to do zastosowania następnej funkcji fitEllipse().

W celu opisania każdego konturu elipsą zastosowano funkcję fitEllipse() z biblioteki OpenCV. By porównać działania metod, problemów występujących przy śledzeniu i wpływu parametrów filtru Kalmana wybrano 5 przykładowych poruszających się obiektów i przeanalizowano skuteczność ich śledzenia.

5.3. Metoda prosta

Najprostsza metoda opiera się na założeniu, że w trakcie analizy w kolejnych klatkach nie nastąpi znaczne przesuniecie się ruchomych obiektów. Dzięki temu można przyjąć, że środki obiektów z klatki n będą się zawierać wewnątrz środków obiektów na klatce n-1. Zaimplementowany algorytm działa w następujący sposób:

Stworzono wirtualna bazę obiektów z zapisanymi pozycjami poszczególnych wykrytych obiektów, w poszczególnych klatkach. Danymi wejściowymi są kontury z programu do detekcji ruchu. Na początku odrzucane są kontury o powierzchni mniejszej niż próg 'p'. Następnie każdy kontur opisywany jest najlepiej pasującą do niego elipsą. Kolejnym krokiem jest sprawdzenie czy środek elipsy zawiera się w jakiejkolwiek z elips opisujących obiekty w klatce poprzedniej. Schemat tego procesu został przedstawiony na rysunku 5.3.

Jeżeli środek elipsy z obecnej klatki znajduje się wewnątrz jednego z obiektów z poprzedniej klatki, to obiekt ten jest aktualizowany danymi tej elipsy. Natomiast gdy żaden z obiektów nie spełnia tego warunku, jest tworzony nowy obiekt na podstawie wykrytej elipsy.



Rys. 5.3. Algorytm sprawdzania wykrytych elips

5.4. Problemy występujące przy śledzeniu

Podczas badania skuteczności śledzenia poszczególnych obiektów zaobserwowano występowanie licznych błędów. Powodowały one niestabilne śledzenie, co objawia się przypisaniem wielu numerów różnych obiektów wirtualnych do jednego obiektu rzeczywistego. Uniemożliwia to automatyczne śledzenie ruchu i dokonywanie analiz na jego podstawie. Główną przyczyną tego problemu są dane wejściowe dostarczane przez algorytmy wykrywania ruchu. Zgodnie z założeniami pracy dostarczają one tylko informację o miejscach na obrazie w których wykryto ruch. Nie dostarcza żadnych dodatkowych danych jak np. kolor obiektów, które ten ruch wykonały. Dlatego algorytmy do śledzenia ruchu mogą pracować tylko na wykrytych konturach na kolejnych klatkach, co utrudnia rozróżnianie poszczególnych obiektów. Dodatkowym

problemem jest fragmentaryzacja konturów, wynikająca z niedoskonałości metod wykrywania ruchu. Utrudnia to spajanie konturów w jeden obiekt. Główne problemy wynikające z tych przyczyn zostały opisane poniżej.

5.4.1. Śledzenie pieszych



Rys. 5.4. Śledzenie pieszych

Badane metody śledzenia ruchu nie używającej modelu człowieka ani innego obiektu, polegają jedynie na analizie obrazu z wykrytym ruchem. Śledzenie pieszych stanowi duży problem, ponieważ ręce, nie będące przy tułowiu i nogi, w trakcie marszu, mają niewielką powierzchnię na analizowanym obrazie. Utrudnia to ich wykrycie i połączenie w cały kontur wszystkich elementów pieszego. Zostało to pokazane na rysunku nr 5.4. Na przemian dzielące się i łączące się kontury, składające się na wykrytą postać, mogą być za małe i potraktowane jako szum do usunięcia. Powoduje to zanikanie śledzonych obiektów w niektórych klatkach.

5.4.2. Łączenie się różnych konturów

Przy poruszających się w różnych kierunkach wielu obiektach nieunikniona jest sytuacja, kiedy są one na tyle blisko siebie, że pomiędzy nimi nie ma tła. Powoduje to zlanie się konturów tych obiektów w obrazie wyjściowym z algorytmu wykrywania ruchu (rys. 5.5). Jeżeli się potem te obiekty rozdzielą to idealnie by było, żeby zostały rozpoznane jako te same pojazdy, co przed złączeniem. W celu uniknięcia złączenia konieczne są dodatkowe informacje z analizowanego filmu (np. rozróżnienie poszczególnych ruchomych obiektów po kolorze). Mniej skuteczna jest metoda analizowania już czarno-białego filmu z wykrytym ruchem i próba rozpoznania połączenia się obiektów po zmianie ich rozmiarów, jak w założeniach pracy. W celu

rozpoznania tego samego obiektu po rozłączeniu zastosowano filtr Kalmana który, mimo braku wykrycia obiektów, cały czas przewiduje, gdzie obiekt powinien się znajdować w kolejnych klatkach.



Rys. 5.5. Maska wykrycia ruchu -łączenie się konturów

5.4.3. Cienie

Kolejnym problemem przy analizowaniu ruchu pojazdów, jest powstawanie cieni od przemieszczających się obiektów (rys. 5.5). Powoduje on zmianę wartości w przestrzeni RGB danego fragmentu obrazu co może także zostać wykryte jako ruch. Dodatkowo, jak na rysunku 5.5, może powodować zlewanie się pojazdów. Jego eliminacja wymaga dodatkowych analiz i algorytmów na poziomie etapu wykrywania ruchu, które eliminują przemieszczanie się cieni jako obszary w których nastąpił ruch.

5.4.4. Zatrzymywanie się pojazdów

Gdy pojazd się zatrzyma, algorytm wykrywania ruchu zaczyna go traktować jak element tła, gdyż nie ma żadnych zmian w danym obszarze klatki. Wiele metod stara sobie radzić z takim przypadkiem np. PBAS [12] powoduje stopniowe zanikanie konturów obiektu, który się zatrzymał. Innym podejściem jest zachowywanie kilku modeli tła na rożnych warstwach jak w [11]. Dodatkowo pojazd ruszający (np. ruszenie samochodu ze świateł) w obrębie klatki, powoduje powstanie dodatkowych błędów, poprzez stworzenie tak zwanego "ducha" (rys. 5.6). Jest to wykrywanie ruchu w kształcie pojazdu, który ruszył, w miejscu z którego ruszył, po jego przemieszczeniu się. Jest to spowodowane faktem, że obiekt był uznany za tło i w momencie ruszenia się nastąpiła gwałtowna zmiana wartości pikseli w przestrzeni RGB, w danym miejscu. Czas pozostania "ducha" w danym miejscu zależy od szybkości aktualizacji modelu tła. Szybka aktualizacja powoduje, krótki czas istnienia "ducha", ale też szybsze integrowanie nieruchomych obiektów do tła po ich zatrzymaniu się. Może być to utrudnione ciągłym ruchem obiektów przez dany obszar.



Rys. 5.6. "Duch" spowodowany ruszeniem samochodu

5.4.5. Duże pojazdy

Duże pojazdy stanowią kolejne źródło błędów podczas wykrywania i śledzenia ruchu, zwłaszcza jak posiadają jednolite obszary np. dach, plandeki na ciężarówkach (rys. 5.7). Jeżeli nie są kolorystycznie wyraźnie różne od tła, mogą pozostać niewykryte podczas ruchu. Widać to na przykładzie ciężarówki z filmu drugiego. Jej plandeka jest stosunkowo jednolita kolorystycznie na dużym obszarze i o kolorze zbliżonym w wartościach przestrzeni RGB do tła. Część pikseli klatki, odpowiadających plandece, ulega nieznacznej zmianie w trakcie ruchu ciężarówki. Jeżeli trwa to dłuższy czas (obszar ma znaczną powierzchnię), to skutkuje utrudnionym wykrywaniem danego obszaru jako ruchomego. Metodami niwelowania tego rodzaju błędów jest stosowanie dodatkowych informacji o otoczeniu piksela jak np. tekstury otoczenia każdego piksela [11].



Rys. 5.7. Przykład nieskutecznego wykrywania dużych jednolitych powierzchni w ruchu – plandeka samochodu ciężarowego

5.5. Śledzenie wspomagane filtrem Kalmana

W celu usprawnienia funkcji śledzenia zaimplementowano funkcję, która estymuje pozycję obiektów w następnej klatce, na podstawie historii ich ruchu. Wykorzystano równania opisujące zależność między położeniem, a prędkością danego obiektu w przestrzeni dwuwymiarowej:

$$X_k = X_{(k-1)} + t \cdot v_x \tag{5.1}$$

$$Y_k = Y_{(k-1)} + t \cdot v_y \tag{5.2}$$

Zapisano je w postaci dyskretnych równań stanu. Odstęp czasu między odczytem pozycji obiektów jest równy jednej klatce, dlatego przyjęto t=1. Zmiennymi stanu są: położenie obiektu we współrzędnych X i Y oraz prędkość obiektu Vx i Vy. W tym przypadku nie występują sygnały sterujące.

$$x_k = F x_{(k-1)}$$
 (5.3)

$$z_k = H x_k \tag{5.4}$$

$$x_{k} = \begin{bmatrix} X_{k} \\ Y_{k} \\ V_{x_{k}} \\ V_{y_{k}} \end{bmatrix} \qquad F = \begin{bmatrix} 1 \ 0 \ 1 \ 0 \\ 0 \ 1 \ 0 \ 1 \\ 0 \ 0 \ 1 \ 0 \\ 0 \ 0 \ 0 \ 1 \end{bmatrix} \qquad H = \begin{bmatrix} 1 \ 0 \ 0 \ 0 \\ 0 \ 1 \ 0 \\ 0 \ 0 \ 0 \ 1 \end{bmatrix}$$

Na podstawie tych równań stworzono predyktor w formie filtru Kalmana. Estymuje on nieznaną wielkość na podstawie jej pomiaru obarczonego nieznanym błędem (mającym charakter normalny). W tym przypadku służy do przewidywania (estymowania) położenia obiektu w kolejnej klatce. Dzięki temu łatwiej dopasować wykryte elipsy do odpowiadających im elipsom z klatek wcześniejszych. W programie została wykorzystana implementacja filtru z biblioteki OpenCV. Podstawowe równania filtru Kalmana:

$$x_k = F x_{k-1} + G u_{k-1} + w_k \tag{5.5}$$

$$z_k = H x_k + v_k \tag{5.6}$$

Z pierwszego równania (5.5) wynika, że każdy stan x_k jest liniową kombinacją swojej wcześniejszej wartości, sygnału sterującego u_k i szumu procesu w_k . W badanym przypadku sygnał sterujący wynosi 0. Drugie równanie (5.6) oznacza, że wartość mierzona z_k jest kombinacją liniową wartości zmiennych stanu i szumu pomiaru v_k . Szumy pomiaru i procesu są od siebie niezależne. Działanie filtru Kalmana można podzielić na dwie fazy: predykcji i korekcji.

W fazie predykcji filtr wyznacza przewidywaną estymacje zmiennych stanu \hat{x}_k (równanie 5.7) i błędu kowariancji \hat{P}_k (równanie 5.8). Równania fazy predykcji:

$$\hat{x}_{k} = F \, \hat{x}_{(k-1)} \tag{5.7}$$

$$\dot{\boldsymbol{P}}_{k} = \boldsymbol{F}\boldsymbol{P}_{k-1}\boldsymbol{F}^{T} + \boldsymbol{Q} \tag{5.8}$$

Następnie następuje faza korekcji:

$$K_{k} = \dot{P}_{k} H^{T} (H \dot{P}_{k} H^{T} + R)^{-1}$$
(5.9)

$$\hat{x}_{k} = \hat{x}_{k} + K_{k} (z_{k} - H \hat{x}_{k})$$
(5.10)

$$P_k = (1 - K_k H) \dot{P}_k \tag{5.11}$$

76

Najpierw jest obliczane wzmocnienie Kalmana K_k (równanie 5.9). Przy jego pomocy oraz wartości położenia i prędkości zmierzonych w bieżącej klatce z_k jest aktualizowana estymacja zmiennych stanu \hat{x}_k (równanie 5.10). Ostatnim krokiem jest zaktualizowanie wartości błędu kowariancji P_k (równanie 5.11). Macierze kowariancji szumu procesu (Q), jak i kowariancji szumu pomiarowego (R) filtru Kalmana przedstawiono poniżej:

$$Q = \begin{bmatrix} q & 0 & 0 & 0 \\ 0 & q & 0 & 0 \\ 0 & 0 & q & 0 \\ 0 & 0 & 0 & q \end{bmatrix} \qquad R = \begin{bmatrix} r & 0 & 0 & 0 \\ 0 & r & 0 & 0 \\ 0 & 0 & r & 0 \\ 0 & 0 & 0 & r \end{bmatrix}$$

W pracy badany jest wpływ wartości 'q' i 'r' na jakość przewidywania położenia obiektu.

Predyktor wykorzystujący filtr Kalmana jest jednym ze składników procesu śledzenia obiektów, którego główną część zaprezentowano na rys. 5.8. Jest to schemat blokowy analizy pojedynczej wykrytej elipsy. Analogicznie jak w poprzedniej metodzie najpierw następuje wykrycie konturów, które są większe od określonego progu i ich opisanie elipsami, które są kolejno analizowane. Każda z elips jest porównywana z kolejnymi istniejącymi obiektami (kolejne wartości "n" na rys. 5.8). Obiekt jest brany pod uwagę jeśli nie została już mu przypisana inna elipsa w bieżacej klatce lub nie zostało wykryte wcześniej, że opuścił obszar klatki (usunięty). Jeżeli obiekt był użyty wcześniej mniej niż 9 razy (faza uczenia się filtru Kalmana), to jest liczony dystans pomiędzy środkiem tego obiektu w poprzedniej klatce i środkiem elipsy. Natomiast gdy obiekt już jest po fazie uczenia się, dystans jest mierzony pomiędzy przewidywanym środkiem obiektu w danej klatce, a środkiem elipsy. Dodatkowo dystans musi być mniejszy od określonego progu, zależnego od rozmiarów klatki. W ten sposób badane są wszystkie obiekty dla każdej elipsy. Elipsa jest przypisywana do obiektu do którego ma najmniejszy obliczany wcześniej dystans (nr min na rys. 5.8). Jeżeli żaden z obiektów nie spełnia powyższych kryteriów (decyzja=0 na schemacie), tworzony jest nowy na podstawie badanej elipsy. Gdy wszystkie elipsy zostaną przetworzone, dla każdego obiektu przeprowadzana jest faza korekcji filtru Kalmana używając danych przypisanych im w bieżącej klatce i faza predykcji by uzyskać estymowaną pozycję w następnej klatce.



Rys. 5.8. Schemat śledzenia obiektu wykorzystującego predyktor Kalmana

5.6. Porównanie działania obu metod

W celu porównania działania obu metod śledzenia oraz wpływu parametrów filtru Kalmana wybrano kilka przykładowych obiektów poruszających się na filmie. Wybrano obiekty o numerach rzeczywistych:

8- przemieszczający się człowiek,

12- skręcający samochód na chwilę łączący się z samochodem jadącym w przeciwną stronę,

22- jadący rowerzysta,

28- samochód zatrzymujący się przed przejściem dla pieszych, a następnie ruszający,

50- ciężarówka zatrzymująca się przed przejściem dla pieszych, a następnie ruszająca.

Algorytmy śledzenia każdemu wykrytemu obiektowi przypisują swój numer. Porównując obie metody policzono ile razy zmienia się ten numer przypisany danemu obiektowi. Większa liczba zmian numeru przypisanego do każdego obiektu rzeczywistego wynika z nieidealnego śledzenia i przewidywania położenia obiektu w kolejnej klatce oraz z jego podziału na mniejsze kontury. Ma to znaczący wpływ zwłaszcza przy zatrzymujących się pojazdach, gdy po zatrzymaniu algorytm wykrywania ruchu powoli je zmniejsza od zewnątrz. Przez to "rozpadają" się na mniejsze fragmenty. Dodatkowe kontury mogą też się pojawić w przypadku złego śledzenia i nierozpoznania obiektu np. po ruszeniu i uznania go za nowy obiekt.

By porównać wpływ parametrów filtru Kalmana na jakość śledzenia obliczono średnią odległość między przewidywanym środkiem obiektu z predyktora a jego rzeczywistym położeniem w danej klatce. Zmieniano wartości parametrów w macierzach zarówno kowariancji szumu procesu (Q), jak i kowariancji szumu pomiarowego (R).

5.6.1. Obiekt nr 8 – człowiek



Rys. 5.9. Wykrycie ruchu obiektu nr 8

Obserwowany człowiek przemieszcza się w jednolitym tempie wzdłuż krawędzi drogi. Wychodzi z górnego fragmentu skrzyżowania i porusza się po chodniku do lewego końca klatki. W przypadku tego obiektu liczba zmian numeru przypisanego wynika z faktu, że osoba nie zawsze stanowi jeden kontur po fazie wykrywania ruchu (rys 5.9). Utrudnia to wykrycie i śledzenie całego konturu człowieka. Spośród badanych parametrów predyktora wartości: $q=10^{-3}$ $r=10^{-2}$ i $q=10^{-2}$ $r=10^{-2}$ powodują poprawę jakości śledzenia ruchu pieszego, pozostałe ją pogarszają.

		Średnie odległości	Liczba zmian
	q=10 ⁻² r=10 ⁻²	12,05	2
Droduktor	q=10 ⁻³ r=10 ⁻²	10,46	2
Predyktor	q=10 ⁻¹ r=10 ⁻²	42,56	5
	q=10 ⁻² r=10 ⁻³	42,56	5
Metoda prosta			4

Tabela 5.1. Porównanie skuteczności śledzenia obiektu nr 8

5.6.2. Obiekt nr 12 – skręcający samochód



Rys. 5.10. Wykrycie ruchu obiektu nr 12

Obserwowanym obiektem jest samochód skręcający na skrzyżowaniu w lewo (rys 5.10). Jednocześnie drugi samochód wyjeżdża z przeciwnej strony i skręca w prawo, mijając się z obiektem badanym. Oba obiekty po analizie algorytmem PBAS są dobrze

określone pojedynczymi konturami. Mimo to następuje sklejenie się konturów obu pojazdów na 6 klatek. To wystarcza by, w przypadku prostej metody, powstały już kolejne obiekty i nastąpiła zmiana ich numeracji. Zastosowanie predyktora wyeliminowało ten problem, powodując, że nie następuje żadna zmiana numeracji. Duży skok wartości błędu w klatce 830 (rys. 5.11) jest spowodowany rozłączeniem się obiektów, ale mimo to obiekty dalej mają te same numery i są dobrze śledzone aż do wyjechania poza kadr kamery. Wyniki śledzenia przy pomocy predyktora są zbliżone mimo zmian wartości parametrów.

		Średnie odległości	Liczba zmian
	q=10 ⁻² r=10 ⁻²	9,46	
Draduktor	q=10 ⁻³ r=10 ⁻²	9,53	
Predyktor	q=10 ⁻¹ r=10 ⁻²	9,41	
	q=10 ⁻² r=10 ⁻³	9,10	
Metoda prosta			5

Tabela 5.2. Porównanie skuteczności śledzenia obiektu nr 12



Rys. 5.11. Wykres błędów predykcji obiektu nr 12

5.6.3. Obiekt nr 22- jadący rowerzysta



Rys. 5.12. Wykrycie ruchu obiektu nr 22

Rowerzysta (rys. 5.12) porusza się jednostajnym ruchem wzdłuż jezdni. Podczas ruchu następuje nałożenie się jego sylwetki z sylwetką rowerzysty czekającego, a następnie ruszającego z drogi prostopadłej. To w przypadku wszystkich badanych parametrów, oprócz q=10⁻² i r=10⁻³, powoduje przypisanie konturowi rowerzysty nowego obiektu. Skutkuje to wzrostem liczby numerów przypisanych temu obiektowi. Wzrost wartości błędu w klatce 1160 wynika z tego faktu połączenia się konturu śledzonego obiektu z konturem drugiego rowerzysty, przez co środek wykrytej elipsy gwałtownie zmienia swoje położenie w kierunku innym niż ruch obiektu. W tabeli nr 5.3 i na rysunku 5.13 wyraźnie widać, że zastosowanie predyktora i parametrów q=10⁻² i r=10⁻³ zdecydowanie poprawia jakość śledzenia.

		Średnie odległości	Liczba zmian
	q=10 ⁻² r=10 ⁻²	9,16	4
Draduktor	q=10 ⁻³ r=10 ⁻²	10,95	3
Predyktor	q=10 ⁻¹ r=10 ⁻²	9,36	4
	q=10 ⁻² r=10 ⁻³	9,36	4
Metoda prosta			5

Tabela 5.3. Porównanie skuteczności śledzenia obiektu nr 22



Rys. 5.13. Wykres błędów predykcji obiektu nr 22

5.6.4. Obiekt nr 28 – samochód poruszający się z zmienną prędkością



Rys. 5.14. Wykrycie ruchu obiektu nr 28

Samochód pojawia się na klatce z prawej strony, dojeżdża do przejścia dla pieszych, gdzie zatrzymuje się, by przepuścić pieszych po czym rusza i wyjeżdża z ujęcia po lewej stronie. Obrazuje problem zatrzymania się pojazdu i znikania jego z obrazu wykrytego ruchu i pojawienia się "ducha" po jego ruszeniu (rys. 5.14). Widać wyraźny wpływ predyktora na jakość śledzenia. Obniża on ponad siedmiokrotnie liczbę zmian numerów przypisanych do tego obiektu, co w znaczący sposób wpływa na stabilność śledzenia. Wzrost wartości błędu od klatki 1525 (rys. 5.15) jest spowodowany połączeniem konturu z konturem drugiego samochodu, czekającego na wykonanie skrętu z ulicy podporządkowanej. Po ich rozłączeniu następuje stopniowy spadek wartości błędu. Mimo to oba pojazdy zachowują ten sam numer przez cały czas swojego ruchu w

kadrze filmu.

		Średnie odległości	Liczba zmian
	q=10 ⁻² r=10 ⁻²	13,98	3
Draduktor	q=10 ⁻³ r=10 ⁻²	14,43	3
Predyktor	q=10 ⁻¹ r=10 ⁻²	14,95	3
	q=10 ⁻² r=10 ⁻³	14,95	3
Metoda prosta			22

Tabela 5.4. Porównanie skuteczności śledzenia obiektu nr 28



Rys. 5.15. Wykres błędów predykcji obiektu nr 28

5.6.5. Obiekt nr 50 – samochód ciężarowy poruszający się z zmienną prędkością



Rys. 5.16. Wykrycie ruchu obiektu nr 50

Samochód ciężarowy porusza się i zatrzymuje się przed przejściem dla pieszych analogicznie jak obiekt nr 28. Jego duża biała plandeka o kolorze zbliżonym do koloru drogi jest trudno wykrywalna przez algorytmy wykrywania ruchu, co powoduje tworzenie dużej liczby małych konturów i utrudnia określenie całego konturu (rys. 5.16). Dodatkowo zatrzymanie skutkuje powolnym zanikaniem obiektu z obrazu wykrycia ruchu i powstawaniem mniejszych konturów, co powoduje mniej stabilne śledzenie obiektu. Zastosowanie predyktora Kalmana zwiększa liczbę zmian numerów przypisanych do tego obiektu. Pojazd zatrzymuje się w klatce 2430 a rusza w 2500. W tym czasie wykryty kontur jego ruchu zmniejsza się i dzieli na większą liczbę konturów. Tak duże skoki wartości błędu są spowodowane właśnie złym wykryciem całego konturu i fragmentaryzacją obiektu na mniejsze kontury.

Tabela 5.5. Porównanie skuteczności śledzenia obiektu nr 50

		Średnie odległości	Liczba zmian	
	q=10 ⁻² r=10 ⁻²	80,39		48
Draduktor	q=10 ⁻³ r=10 ⁻²	80,56		46
Pledyktol	q=10 ⁻¹ r=10 ⁻²	82,40		56
	q=10 ⁻² r=10 ⁻³	82,49		56
Metoda prosta				23



Rys. 5.17. Wykres błędów predykcji obiektu nr 50

Wszystkie badane wartości parametrów q i r dają zbliżone wyniki jakości

śledzenia obiektów. Najmniejsza liczba zmian numeru obiektów jest przy parametrach: $q=10^{-3}$ r= 10^{-2} . Nie powoduje ona najmniejszych różnic między przewidywanym położeniem obiektu a rzeczywistym, jednak są to wartości zbliżone do odpowiadających im w przypadku stosowania pozostałych parametrów. Z tych powodów wybrano wartości $q=10^{-3}$ r= 10^{-2} do dalszych badań.

5.7. Wpływ rozdzielczości na śledzenie obiektów

Z wyników badań wykrywania ruchu okazało się, że zmniejszanie rozdzielczości nie obniża w znacznym stopniu jakości wykrywania ruchu, a znacznie obniża czas analizy. W związku z tym sprawdzono jak zmniejszona rozdzielczość wpłynie na śledzenie obiektów. Zastosowano algorytm z filtrem Kalmana i parametrami q= 10^{-3} r= 10^{-2} , dobranymi na podstawie wcześniejszych badań. Rozdzielczość obniżono najpierw dwukrotnie do 640x360 a następnie jeszcze raz dwukrotnie do 320x180, analogicznie jak w przypadku badania algorytmów wykrywania ruchu. Porównanie przeprowadzono dla tych samych obiektów, co badanie parametrów algorytmu wykorzystującego filtr Kalmana.

Tabela 5.6. Porównanie skuteczności śledzenia obiektów w zależności od rozdzielczości

	Obiekt									
	8		12		22		28		50	
	Średnia		Średnia		Średnia		Średnia		Średnia	
Rozdzielczość	odległość	Liczba konturów								
1280x720	10,46	2,00	9,53	1,00	10,95	3,00	14,43	3,00	80,56	46,00
640x360	36,75	1,00	7,98	1,00	13,97	1,00	8,76	1,00	55,79	66,00
320x180	18,55	1,00	4,64	1,00	7,65	1,00	11,46	2,00	24,55	36,00



Rys. 5.18. Wykres błędów predykcji obiektu nr 12 w różnych rozdzielczościach



Rys. 5.19. Wykres błędów predykcji obiektu nr 28 w różnych rozdzielczościach



Rys. 5.20. Wykres błędów predykcji obiektu nr 50 w różnych rozdzielczościach

Obniżenie rozdzielczości nie powoduje jednoznacznej poprawy jakości śledzenia ruchu. Średnie błędy predykcji powinny każdorazowo zmniejszyć się około 4-krotnie (proporcjonalnie do zmniejszenia liczby pikseli). Zmniejszenie rozdzielczości do 640x360 pikseli powoduje spadek średnich błędów predykcji dla obiektów 12, 28 i 50 (samochód skręcający, samochód poruszający się z zmienną prędkością i samochód ciężarowy) o średnio około 30%. Wynika to z faktu, że predykacja już początkowo była dokładna i zmniejszenie rozdzielczości niewiele ją poprawiło. Natomiast w przypadku obiektów 8 i 22 (pieszy i rowerzysta) średnie błędy są większe, niż w przypadku pierwotnej rozdzielczości. Liczba zmian numerów przypisanych do jednego obiektu maleje dla wszystkich badanych oprócz samochodu ciężarowego. Poszczególne kontury lepiej się łacza w pojedyncze obiekty, co zaobserwowano podczas badania metod wykrywania ruchu. Zapewnia to łatwiejsze i stabilniejsze śledzenie wykrytych obiektów. Niestety samochód ciężarowy ulega większej fragmentaryzacji (wzrost liczby zmian o 30%) i kontury niewiele różnią się rozmiarem i odległością od siebie, co utrudnia stabilne śledzenie wybranego obiektu. Rozwiązaniem tego problemu byłoby lepsze wykrywanie całego konturu samochodu ciężarowego.

Kolejne obniżenie rozdzielczości spowodowało już nieznaczny spadek dystansów średnio dwukrotnie zamiast czterokrotnie. Przy tak niskiej rozdzielczości (320x180) odległości pomiędzy poszczególnymi obiektami są już niewielkie, co powoduje łatwiejszą wymianę numerów między obiektami znajdującymi się na filmie.

Zmniejszenie rozdzielczości do 640x360 niejednoznacznie wpływa na błędy w predykcji. W części przypadków są one obniżane, w pozostałych zwiększane. Wyraźnie to widać na rysunkach nr 5.18, 5.19, 5.20, zwłaszcza w przypadku najwyższych wartości błędu. Liczba zmian numeracji wskazuje na stabilniejsze śledzenie obiektów przy niższej rozdzielczości. Jedynym wyjątkiem jest samochód ciężarowy, jednak jest to spowodowane jego słabym wykryciem i fragmentaryzacją jego konturu. Dalsze zmniejszenie rozdzielczości powoduje już mniej stabilne śledzenie.

6. Podsumowanie i wnioski

Założeniem pracy było zbadanie i porównanie metod wykrywania i śledzenia ruchu w sekwencji obrazów pochodzących z jednej kamery. W tym celu nagrano i obejrzano około 130 minut filmów przedstawiających ruch pojazdów, zarówno na drogach szybkiego ruchu, jak i skrzyżowaniach. By porównać wyniki działania różnych algorytmów, konieczne było określenie danych rzeczywistych. By tego dokonać napisano odpowiedni program, umożliwiający oznaczanie przy pomocy elips pojazdów w kolejnych klatkach filmu i zapisanie informacji o nich w pliku csv. Zastosowano środowisko Microsoft Visual Studio 2010, język C++ i bibliotekę OpenCV.

W pracy przeprowadzono wstępne badania 21 algorytmów i wybrano metody osiągające najlepsze wyniki i reprezentujące różne rodzaje podejścia do wykrywania ruchu. Wykorzystano do tego bibliotekę BGS i na jej podstawie stworzono program wykrywający ruch na poszczególnych klatkach filmu. Wykonanie obliczeń dla poszczególnych metod, przy różnych wartościach ich parametrów zajęło około 70 godzin pracy komputera. Uzyskane wyniki umożliwiły porównanie skuteczności działania rożnych metod. Najlepszą skuteczność zaobserwowano dla metody PBAS, niestety jest ona też najwolniejsza.

Następnie trzy wybrane metody sprawdzono na filmie przedstawiającym ruch na skrzyżowaniu w formie litery T. Scena ta stanowi wyzwanie dla algorytmów wykrywania ruchu, gdyż samochody nie jadą tylko w jednym kierunku, skręcają, zmieniają prędkości i zatrzymują się. Dodatkowo na filmie pojawiają się piesi. W tej części badań zaobserwowano kilka zjawisk które utrudniają wykrywanie ruchu. Pojazdy znajdujące się na klatce, które ruszają w trakcie trwania filmu mogą na części metod zostawiać "duchy", zanim program nauczy się prawidłowego tła w obszarach, z których ruszyły pojazdy. Kolejnym problemem są jednolite powierzchnie poruszających się obiektów. Powoduje to wzrost fragmentaryzacji wykrytych konturów ruchu, co utrudnia ich złączenie w jeden obiekt opisujący cały pojazd. Szyby pojazdów często odbijają obraz tła dokoła niego, co powoduje, że się za mało różnią w przestrzeni kolorów RGB od tła, co wprowadza błędy do wykrywania ruchomych pojazdów. To samo dotyczy poruszających się obiektów o kolorze zbliżonym do koloru tła po którym się przemieszczają. Kluczowy przy niwelowaniu tych wszystkich błędów wykrywania jest odpowiedni dobór parametrów algorytmów.

W dalszej części pracy sprawdzono wpływ rozdzielczości wprowadzanego filmu na jakość wykrywania ruchu. Z wyników badań widać, że najlepsze wyniki wybrane algorytmy osiągnęły dla rozdzielczości 640x360, co stanowi połowę rozdzielczości oryginalnej. Dzięki zmniejszeniu rozdzielczości wykryte kontury łatwiej łączą się w większe, opisujące cały poruszający się pojazd, dzięki czemu zmniejszona jest fragmentaryzacja konturów. Dodatkowo obrazy w mniejszej rozdzielczości potrzebują mniej czasu by zostać poddane analizie. Kolejną częścią pracy było sprawdzenie działania metod śledzenia ruchu obiektów na filmach wyjściowych z pierwszej części badań. Najpierw sprawdzono działanie algorytmów zawartych w bibliotece OpenCV, przykładzie blobtrack_sample. Wynik ich działania był całkowicie niezadowalający, w związku z tym autor zaproponował i zaimplementował dwa swoje algorytmy. Pierwszy opierający się na rozpoznawaniu które środki elips wykrytych obiektów z aktualnej klatki znajdują się w środku elips obiektów w poprzedniej klatce. Na tej podstawie dopasowywano poszczególne kontury do numerów obiektów w kolejnych klatkach. Metoda wykazała wiele problemów na które napotykają programy śledzące ruch. Podstawowym jest łączenie się dużych konturów różnych poruszających się pojazdów w jeden kontur. Powoduje to zanik część obiektów na czas połączenia się konturów, co uniemożliwia ich śledzenie. Dodatkowo problemem jest zatrzymywanie się pojazdów i pojawianie się "duchów" po ich ruszeniu. Kolejne błędy wprowadzają pojazdy poruszające się z różną prędkością.

W celu poprawy działania algorytmu zastosowano predyktor w postaci filtru Kalmana. Umożliwia on przewidywanie pozycji pojazdu w kolejnych klatkach na podstawie historii jego ruchu. Poprawił on znacznie stabilność śledzenia pojazdów i wyeliminował w dużej mierze błędy, zwłaszcza te wprowadzane przez łączenie się konturów.

Dalsze ulepszanie algorytmu wykrywania i śledzenia konturów wymagałoby znacznego jego skomplikowania. Możliwa jest ekstrakcja dodatkowych danych w momencie wykrywania ruchu, takich jak kolory, wzorce geometryczne, co znacząco usprawniłoby rozpoznawanie i stabilne śledzenie kolejnych poruszających się obiektów. Niestety wiązałoby to się z spowolnieniem działania algorytmów, co byłoby przeszkodą w stosowaniu go w czasie rzeczywistym. Rozwiązaniem jest podjęcie kroków w celu skrócenia czasu jego wykonywania. Można to osiągnąć poprzez lepszą implementację np. poprzez podział obliczeń na wiele rdzeni procesora, użycie dedykowanego sprzętu specjalnie zoptymalizowanego do wybranych obliczeń lub zastosowanie technologii CUDA firmy Nvidia. W przypadku renderowania grafiki 3d potrafi ona przyśpieszyć obliczenia około dwudziestokrotnie, zatem dobrze zoptymalizowany algorytm używający karty graficznej wyposażonej w CUDA mógłby skutecznie wykrywać ruch w sekwencji obrazów w czasie rzeczywistym. Dzięki temu możliwe byłoby zaimplementowanie bardziej skomplikowanych i skuteczniejszych algorytmów do działania w czasie rzeczywistym.

Literatura:

- [1] J. Grębosz, "Symfonia C++ standard", Kraków, 2008, ISBN 978-83-7366-134-4
- [2] N. Buch, S. Valastin, J. Orwell, "A Review of Computer Vision Techniques for the Analysis of Urban Traffic" w *IEEE Transactions on intelligent transportation* systems, tom 12, wrzesień 2011
- [3] A. Laureshyn, A. Svensson, Ch. Hydén, "Evaluation of traffic safety, based on micro-level behavioural data: Theoretical framework and first implementation", Lund, 2010
- [4] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, "Detecting Objects, Shadows and Ghosts in Video Streams by Exploiting Color and Motion Information" w *IEEE Transactions on Pattern Analysis and Machine Intelligence*, wrzesień 2003
- [5] Ch. Wren, A. Azarbayejani, T. Darrell, A. Pentland, "Pfinder: Real-Time Tracking of the Human Body" w IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, tom 19, lipiec 1997
- [6] Ch. Stauffer, W.E.L. Grimson, "Adaptive background mixture models for realtime tracking", Cambridge, 1999
- [7] P. KadewTraKuPong, R. Bowden, "An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection" w Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems, wrzesień 2001
- [8] Z. Zivkovic, "Improved Adaptive Gaussian Mixture Model for Background Subtraction" w *Proc. ICPR*, 2004
- [9] T. Bouwmans, F. El Baf, B. Vachon, "Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey" w *Recent Patents on Computer Science*, 2008
- [10] F. El. Baf, T. Bouwmans, B. Vachon, "Type-2 Fuzzy Mixture of Gaussians Model: Application to Background Modeling" w ISVC 2008, Las Vegas : États-Unis d'Amérique, 2008
- [11] J. Yao, J-M. Odobez, "Multi-Layer Background Subtraction Based on Color and Texture" w IEEE THE CVPR VISUAL SURVEILLANCE WORKSHOP (CVPR-VS), MINNEAPOLIS, czerwiec, 2007
- [12] M. Hoffman, P. Tiefenbacher, G. Rigoll, "Background Segmentation with Feedback: The Pixel-Based Adaptive Segmenter", Monachium, 2012
- [13] A. B. Godbehere, A. Matsukawa, K. Goldberg, "Visual Tracking of Human Visitors under Variable-Lighting Conditions for a Responsive Audio Art Installation", 2012
- [14] M. Oliver, B. Rosario, A. Pentland, "A Bayesian Computer Vision System for Modeling Human Interactions" w IEEE TRANSACTIONS ON PATTERN

ANALYSIS AND MACHINE INTELLIGENCE, tom 22, sierpień, 2000

- [15] G. Bradski, A. Kaehler, "Learning OpenCV Computer Vision with the OpenCV Library", O'Reilly Media, wrzesień 2008
- [16] OpenCV 2.4.6.0 documentation [online], [dostęp: 25.06.2013], Dostępny w Internecie: http://opencv.org
- [17] BGS Library [online], Andrews Sobral [dostęp: 25.06.2013]Dostępny w Internecie: http://code.google.com/p/bgslibrary/
- [18] Homeland Security Corporation YouTube channel [online], [dostęp: 25.01.2013], Dostępny w Internecie: http://www.youtube.com/user/HSCC1000/videos
- [19] N. J. B. McFarlane C. P. Schofield, "Segmentation and tracking of piglets in images" w Machine Vision and Applications, tom 8, 1995
- [20] Y. Benezeth, "Review and evaluation of commonly-implemented background subtraction algorithms" z 19th International Conference on Pattern Recognition, 2008
- [21] M. Heikkila, "A texture-based method for modeling the background and detecting moving objects" w IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006
- [22] C. Kertész, "Texture Based Foreground Detection" w *Image Processing and Pattern Recognition*, tom 4, grudzień, 2011
- [23] Y. Goyat, "Vehicle trajectories evaluation by static video sensors" z *Intelligent Transportation Systems Conference*, 2006
- [24] A. Elgammal, D. Harwood, L. Davis, "Non-parametric Model for Background Subtraction", University of Maryland, 2000
- [25] Z. Hongxun, De Xu, "Fusing Color and Texture Features for Background Model" z Lecture Notes in Computer Science, tom 4223, 2006
- [26] F. El Baf, T. Bouwmans, B. Vachon, "Fuzzy Integral for Moving Object Detectionę, Hongkong, 2008
- [27] Z. Zhao, "A Fuzzy Background Modeling Approach for Motion Detection in Dynamic Backgrounds" z Communications in Computer and Information Science, tom 346, 2012
- [28] L. Maddalena, A. Petrosino, "A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications" z *IEEE Transactions on Image Processing*, tom 17, 2008
- [29] L. Maddalena, A. Petrosino, "A fuzzy spatial coherence-based approach to background/foreground separation for moving object detection" z Neural Computing and Application, tom 19, marzec, 2010