

**AGH**

**AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W  
KRAKOWIE**

**WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI, INFORMATYKI I INŻYNIERII  
BIOMEDYCZNEJ**

**KATEDRA AUTOMATYKI I INŻYNIERII BIOMEDYCZNEJ**

Praca dyplomowa magisterska

*Wizyjny system umożliwiający pomiar prędkości  
poruszających się pojazdów na podstawie analizy  
sekwencji wideo*

*Vision system that enables speed measurement of moving  
cars after analysis of video sequences*

Autor: *Marcin Siedlarz*  
Kierunek studiów: *Informatyka*  
Opiekun pracy: *dr hab. Adrian Horzyk*

Kraków, 2015

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

*Składam serdeczne podziękowania ...*



# Spis treści

<b>1. Wprowadzenie</b> .....	3
1.1. Cele pracy .....	3
1.2. Motywacja .....	3
1.3. Struktura pracy .....	3
<b>2. Zagadnienia teoretyczne</b> .....	5
2.1. System wizyjny .....	5
<b>3. Metody pomiaru prędkości</b> .....	9
3.1. Metody tradycyjne .....	9
3.1.1. Czujnik indukcyjny .....	9
3.1.2. Radar .....	9
3.2. Metody wykorzystujące analizę obrazu .....	10
3.2.1. Wideorejestратор .....	10
3.2.2. Odcinkowy pomiar prędkości .....	10
3.2.3. Osiągnięcia naukowe .....	11
3.2.4. Wykorzystanie długości pojazdów .....	12
3.2.5. Wykorzystanie statycznych informacji zawartych w scenie .....	12
3.2.6. Doskonalenie śledzenia pojazdów .....	13
<b>4. Koncepcja proponowanego rozwiązania</b> .....	15
4.1. Położenie kamery .....	15
4.1.1. Kamera umieszczona w samochodzie .....	15
4.1.2. Dwie kamery .....	16
4.1.3. Kamera zwrócona prostopadłe do linii drogi .....	17
4.1.4. Zastosowane rozwiązanie .....	17
4.2. Kalibracja kamery .....	18
4.3. Transformacja współrzędnych pola widzenia na współrzędne rzeczywiste .....	19

---

4.4.	Estymacja odległości .....	20
4.4.1.	Zewnętrzne dostarczenie danych.....	20
4.4.2.	Użycie Google Maps .....	21
4.4.3.	Analiza widocznej linii przerywanej.....	22
4.5.	Detekcja ruchu.....	23
4.6.	Identyfikacja i śledzenie obiektów .....	24
4.6.1.	Camshift .....	27
4.6.2.	Wykorzystanie centroidu .....	27
4.6.3.	Cechy charakterystyczne .....	28
4.7.	Kalkulacja prędkości .....	30
4.8.	Realizacja .....	30
<b>5.</b>	<b>Realizacja aplikacji .....</b>	<b>33</b>
5.1.	Użyte narzędzia i biblioteki.....	33
5.2.	Architektura aplikacji .....	33
5.3.	Ograniczenia aplikacji .....	35
5.4.	Konfiguracja testowa .....	36
<b>6.</b>	<b>Testy.....</b>	<b>39</b>
6.1.	Integracja .....	39
6.2.	Porównanie metod śledzenia obiektów .....	39
6.3.	Porównanie metod ewaluacji współczynnika dystansu.....	41
6.4.	Wyniki testów .....	42
<b>7.</b>	<b>Podsumowanie .....</b>	<b>45</b>
7.1.	Ocena realizacji celu.....	45
7.2.	Wnioski.....	45

# 1. Wprowadzenie

## 1.1. Cele pracy

Celem pracy jest stworzenie systemu wizyjnego pozwalającego wykonywać pomiary prędkości przejeżdżających pojazdów wykorzystując do tego informacje pozyskane dzięki analizie sekwencji wideo. Praca nie zakłada specjalnego przygotowania sceny ani zastosowania dodatkowych czujników poza odpowiednio wykadrowaną kamerą. Projekt przewiduje częściową interakcję z użytkownikiem podczas kalibracji środowiska oraz dalsze, samodzielne działanie systemu. Za główny cel postawiono stworzenie aplikacji, która będzie miała zastosowanie w realnym ruchu ulicznym oraz jej przetestowanie.

## 1.2. Motywacja

Motywacją do wyboru tematu pracy były wcześniejsze doświadczenia z cyfrowym przetwarzaniem obrazu zdobyte podczas tworzenia pracy inżynierskiej<sup>1</sup> i wynikająca z nich chęć pogłębienia wiedzy w wyżej wymienionej dziedzinie. Dodatkowym atutem była możliwość stworzenia systemu znajdującego rzeczywiste zastosowanie i korespondującego z motoryzacją, którą również się interesuję.

## 1.3. Struktura pracy

Krótki opis rozdziałów i ich zawartości:

- **Rozdział 1** - Cele pracy oraz motywacja
- **Rozdział 2** - Wstęp teoretyczny zawierający informacje o systemach wizyjnych
- **Rozdział 3** - Przegląd systemów pomiaru prędkości i stanu badań naukowych

---

<sup>1</sup>F. Misztal & M. Siedlarz. Nowe sposoby interakcji z komputerem i ich zastosowania w rozrywce, 2014

- **Rozdział 4** - Koncepcja tworzonego systemu z uwzględnieniem algorytmów oraz założeń projektowych
- **Rozdział 5** - Realizacja aplikacji oraz jej ograniczenia
- **Rozdział 6** - Testy stworzonego systemu
- **Rozdział 7** - Podsumowanie i wnioski

## 2. Zagadnienia teoretyczne

Analiza ruchu drogowego jest domeną, która jest naturalnym i idealnym kandydatem do zastosowania systemów wizyjnych. Obrazy z kamer monitoringu ruchu już od dawna służą do rozwiązywania takich problemów jak: detekcja korków ulicznych, zliczanie przejeżdżających pojazdów czy estymacja ich prędkości. Duża ilość dostępnych kamer i ich niewielka cena w porównaniu do innych rozwiązań sprawiają, że dziedzina ta rozwija się szybko i prężnie. Poniższy rozdział opisuje teoretyczny aspekt systemu, który zostaje tworzony.

### 2.1. System wizyjny

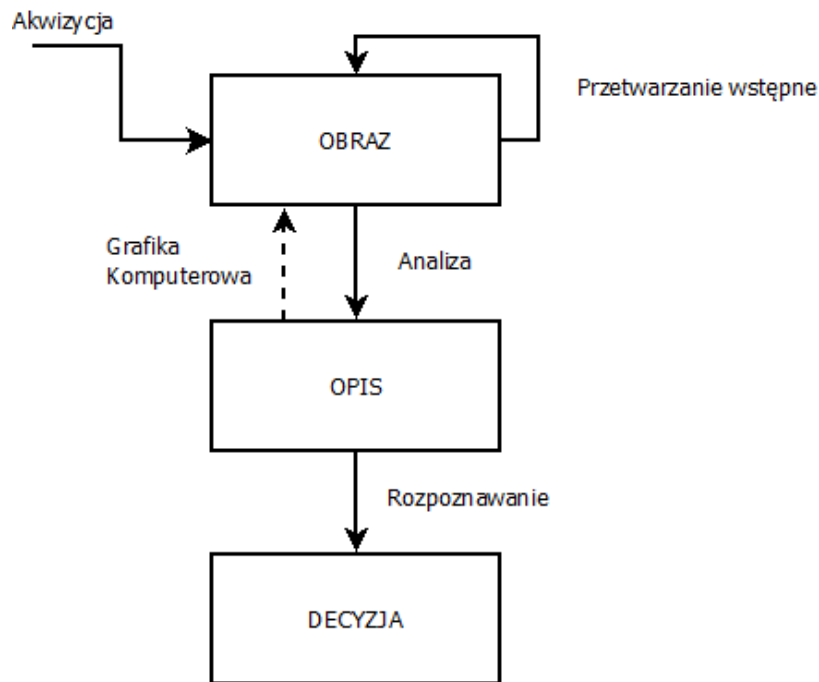
Stworzenie systemu wizyjnego jest złożonym i trudnym zadaniem. Dane dostarczone w formie zbioru pikseli muszą być przeanalizowane i zinterpretowane, aby dostarczać informacji potrzebnych do rozwiązania specyficznego problemu. To, co ludzki mózg robi naturalnie i szybko - jest najtrudniejszą częścią cyfrowego przetwarzania obrazów.

Zgodnie z tym, co opisano w publikacjach profesora Tadeusiewicza [1] i profesora Gorgonia [2], każdy taki system składa się z następujących kroków (rys. 2.1):

- akwizycji obrazu
- przetwarzania wstępnego obrazu
- analizy obrazu
- semantycznej interpretacji obrazu

Pierwsza faza - akwizycja, polega na zapisie cyfrowego sygnału z kamery w odpowiedniej strukturze danych cyfrowych. Dane te przechowują jasności poszczególnych kanałów (RGB/gray/HSV) w punktach obrazu i razem reprezentują widzialny efekt końcowy.

Druga faza algorytmu odpowiada za poprawę jakości i przygotowanie do analizy. Jest to zbiór operacji mających na celu wyostrenie obrazu, usunięcie zakłóceń i wyeksponowanie interesujących cech. Przetwarzanie wstępne realizuje się pod kątem osiągnięcia konkretnego



Rysunek 2.1: Schemat automatycznego systemu wizyjnego (opracowanie własne na podstawie [1]).

celu, dlatego dobór aplikowanych operacji może się różnić, ale w ich skład wchodzi między innymi: równoważenie histogramu, przekształcenia punktowe, geometryczne, filtracja czy edycja jasności.

Najtrudniejszym w realizacji etapem jest etap trzeci - analiza obrazu. W wyniku tej operacji diametralnie zredukowana zostaje ilość informacji. Dane zostają ograniczone do tych, które są potrzebne do dalszego etapu przetwarzania. Kosztem nieistotnych szczegółów znajdujących się w tle, zostają wyekstrahowane cechy, które na pierwotnym kadrze były trudne do zidentyfikowania. Wartością zwróconą nie musi być obraz. Mogą to być również same parametry obiektów, które nas interesują. Przykładami stosowanych metod są: segmentacja, indeksacja oraz pomiary momentów bezwładności.

Ostatnim etapem jest semantyczna interpretacja obrazu, czyli rozpoznanie. Dysponując odpowiednio przeanalizowanym obrazem, faza ta ogranicza się do interpretacji dostępnych parametrów obiektów względem konkretnej hipotezy. Ten krok algorytmu jest łatwiejszy i bardziej deterministyczny niż poprzedni.

Aspektem systemów wizyjnych czasu rzeczywistego, o którym nie sposób nie wspomnieć, jest ograniczenie czasowe wykonywanych operacji. W celu zachowania płynności i możliwości pracy, suma czasu trwania operacji przeprowadzanych na każdej klatce obrazu nie może przekraczać okresu pomiędzy klatkami. Rozważane standardy video przyjmują 25 lub 30 klatek na

sekundę. Wymusza to odpowiedni dobór i optymalizacje algorytmów utrudniając implementację systemu.



## **3. Metody pomiaru prędkości**

W poniższym rozdziale zostaną omówione zdalne metody pomiaru prędkości pojazdów. Rozwiązania podzielono na dwie kategorie: metody nie wykorzystujące analizy wideo do wyliczenia wyniku oraz metody, które w całości lub częściowo wynik opierają na cyfrowym przetwarzaniu obrazu.

### **3.1. Metody tradycyjne**

#### **3.1.1. Czujnik indukcyjny**

Indukcyjny czujnik pętlowy jest jedną z najczęściej stosowanych metod pomiaru parametrów ruchu drogowego. Wynika to ze stosunkowo niskiej ceny oraz dobrej dokładności uzyskiwanych wyników. Szczegółowa zasada działania popularnie zwanej 'pętlą indukcyjnej' opisana jest w pracy doktora Stencela [3], sam natomiast pomiar prędkości odbywa się z wykorzystaniem dwóch czujników umieszczonych w odpowiedniej od siebie odległości. Wynik zostaje uzyskany przez porównanie czasów aktywacji kolejnych czujników podczas najechania pojazdu. Dzięki specyfice swojej budowy oraz instalacji pod powierzchnią jezdni, czujnik ten jest odporny na warunki meteorologiczne, porę dnia i nocy oraz szereg zakłóceń mających wpływ na inne metody pomiaru. Niestety, ten sam aspekt ma swoje minusy - jeżeli podczas projektowania jezdni nie przewidziano instalacji czujnika, montaż urządzenia wymusi dodatkowo zerwanie nawierzchni, a tym samym zwiększenie kosztów jego montażu.

#### **3.1.2. Radar**

Inną popularną metodą pomiaru prędkości jest użycie radaru lub fotoradaru (ten drugi jest wzbogacony o wideorejestrację, ale zasada działania nie ulega zmianie). Urządzenie to korzysta ze zjawiska fizycznego zwanego efektem Dopplera:

Jeśli źródło fali znajduje się w ruchu, to czoła fal będą zagęszczone po stronie obserwatora, do którego źródło się zbliża, a rozrzedzone po stronie obserwatora,

od którego źródło się oddala.

Radar wysyła fale w kierunku pojazdu, aby te zostały od niego odbite i poddaje analizie częstotliwość fali powrotnej. W kontekście definicji - pojazd jest źródłem fali, a radar obserwatorem. Do wyliczenia prędkości korzysta się z faktu, że gdy samochód zbliża się do urządzenia, częstotliwość fal wzrasta, a gdy oddala, częstotliwość maleje. Urządzenia te cechują się dużą dokładnością i są głównym wyposażeniem policyjnej drogówki. Wadą są problemy z równoległymi pomiarami wielu pojazdów oraz wysoka cena.

## 3.2. Metody wykorzystujące analizę obrazu

Metody wykorzystujące przetwarzanie obrazu są szczególnie ważne w kontekście niniejszej pracy. Na podstawie wyciągniętych wniosków z analizy poniższych rozwiązań udało się udoskonalić koncepcję proponowanego systemu.

### 3.2.1. Wideorejestrator

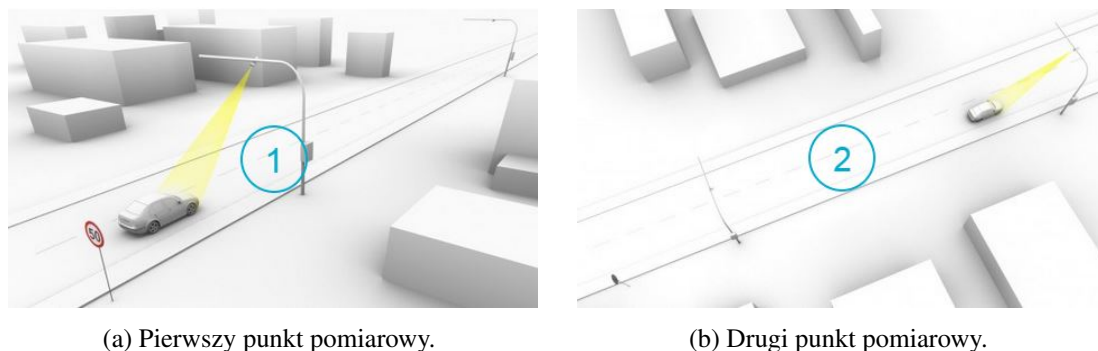
Jest to urządzenie montowane w pojeździe (zwanym dalej samochodem mierzącym) do wykonywania pomiarów w trakcie udziału w ruchu drogowym. Korzysta z zamontowanej, z przodu lub z tyłu samochodu, kamery, która umożliwia śledzenie wybranego pojazdu (kontrolowanego). System wykonuje pomiar pośredni opierając się na danych pochodzących z samochodu mierzącego oraz założeniu, że odległość od kontrolowanego obiektu jest stała. Dane potrzebne do wyliczeń pobierane są odpowiednio: czas - na podstawie taktowania zegara wideorejestratora, przejechana odległość - z przetwornika prędkości zainstalowanego w pojeździe. Analiza obrazu ogranicza się jedynie do akwizycji potencjalnych wykroczeń oraz walidacji utrzymania stałej odległości pomiędzy kontrolowanym, a mierzącym samochodem.

W opisanej metodzie nie zidentyfikowano interesujących aspektów przetwarzania zarejestrowanej sekwencji wideo, które warto włączyć do koncepcji budowanego systemu.

### 3.2.2. Odcinkowy pomiar prędkości

Idea odcinkowej kontroli prędkości zyskuje coraz większą popularność. W porównaniu do punktowych pomiarów, lepiej sprawdza się ona w zapewnieniu bezpieczeństwa ruchu drogowego. Dzięki jej użyciu kierowcy są zobowiązani do przestrzegania przepisów na całym odcinku objętym systemem.

Prędkość wyliczana jest na podstawie czasu pokonania dystansu pomiędzy dwoma bramkami urządzenia. Koncepcja punktów pomiarowych przedstawiona jest na rys. 3.1. Każdy z



Rysunek 3.1: Konceptcja systemu odcinkowego pomiaru prędkości. Źródło: Neurosoft.pl

tych punktów składa się z kamery i interfejsu przekazującego dane do algorytmu wyliczającego prędkość. Dystans pomiędzy nimi może wynosić od kilkudziesięciu metrów do kilku kilometrów. Najważniejszym aspektem dla skuteczności działania jest wiarygodna i odporna na zakłócenia identyfikacja pojazdów w punktach pomiarowych. Jej najczęstsza implementacja wykorzystuje automatyczne rozpoznawanie tablic rejestracyjnych [4]. Zakładając poprawną identyfikację pojazdu, zakłócenia na etapie przetwarzania obrazu nie wpływają na odchylenie poprawności wyniku w przeciwieństwie do metod przedstawionych w kolejnych podrozdziałach.

Funkcjonalność zaprezentowana w powyższym rozwiązaniu jest cenną częścią systemu kontroli prędkości. Algorytm automatycznego rozpoznawania tablic zostanie rozważony jako możliwe rozszerzenie proponowanego systemu.

### 3.2.3. Osiągnięcia naukowe

Powstało wiele opracowań naukowych dotyczących pomiaru prędkości na podstawie sekwencji wideo. Największą popularnością cieszy się idea korzystająca wyłącznie z jednej kamery. Motywacją do tego jest gotowa infrastruktura monitoringu ruchu, którą można by łatwo zaadaptować do nowych celów. W idealnym świecie, odpowiedni algorytm pozwoliłby na pomiar prędkości bez konieczności fizycznego modyfikowania miejsca kontroli czy wykonywania dodatkowych pomiarów. Aby wprowadzić to założenie w życie trzeba rozwiązać 2 najpoważniejsze problemy. Znaleźć przebyty dystans na podstawie zależności pomiędzy układem współrzędnym sceny, a układem współrzędnym obrazu oraz zidentyfikować i śledzić pojawiające się pojazdy. Ze względu na brak znajomości dokładnej kalibracji dostępnych kamer pierwszy problem wydaje się trudniejszy do rozwiązania.

### 3.2.4. Wykorzystanie długości pojazdów

Innowacyjnym podejściem do tematu wykazali się Daniel Dailey, F. W. Cathey and Suree Pumrin, [5]. Zgodnie z koncepcją przedstawioną w poprzednim akapicie, w swojej pracy korzystają ze standardowej kamery ruchu drogowego. Do identyfikacji poruszających się pojazdów wykorzystują algorytm detekcji krawędzi (tworząc różnicę 3 kolejnych klatek, krawędzie, które nie zmieniły swojego położenia zostają zniwelowane, a te które pozostają widoczne, reprezentują obrys poruszających się samochodów) oraz śledzenie centroidu. Ciekawszym natomiast aspektem jest rozwiązanie drugiego problemu.

Autorzy estymują przebyty dystans wykorzystując znany, średni rozkład długości pojazdów. Na potrzeby pracy stworzono model matematyczny, który korzystając z kilkunastu próbek - pojazdów - generuje wynik odzwierciedlający stosunek stopy (30,480 cm) do piksela w poszczególnych rzędach obrazu. Dzięki ograniczeniu algorytmu do przestrzeni klatki, w której samochody spełniają założone kryteria geometryczne, autorom udało się zniwelować błąd przybliżenia. Błąd ten diametralnie narasta w skrajnych obszarach obrazu. Otrzymana w ten sposób funkcja zostaje aproksymowana liniowo i jest punktem wyjścia do wyliczenia przebytej przez pojazd drogi.

Otrzymane tą metodą wyniki zapewniają satysfakcjonującą dokładność. Wykorzystanie tej metody może się doskonale sprawdzić do badania ogólnych trendów i aproksymacji parametrów ruchu drogowego.

### 3.2.5. Wykorzystanie statycznych informacji zawartych w scenie

Inne podejście do tego problemu zaprezentowano w pracy [6]. Do uzyskania referencyjnych informacji o odległości postanowiono wykorzystać statyczne cechy kadru i jezdni. Pierwszym etapem opisanego algorytmu jest obliczenie punktu zbieżności prostych wytyczonych względem zewnętrznych krawędzi jezdni. Na podstawie otrzymanego punktu wyliczona zostaje macierz transformacji potrzebna do przekształcenia płaszczyzny obrazu na płaszczyznę rzeczywistej sceny w rzucie pionowym. Po jej zaaplikowaniu zostaje uzyskany obraz, który cechuje się stałą  $k$  zależnością wymiarów pomiędzy układami współrzędnych.

Druga część algorytmu automatycznej kalibracji skupia się na wyliczeniu stałej  $k$ . Do jej wyliczenia skorzystano z obecności linii przerywanych znajdujących się na pasie ruchu. Dzięki temu, że ich długość i okres są niezmiennie, mogą posłużyć za model referencyjny. Po odpowiedniej identyfikacji pasów na przechwyconym kadrze, algorytm zlicza ich występowanie i określa stosunek metrów na piksel. Zarówno obliczenia prędkości jak i śledzenie obiektów odbywa się na podstawie klatki po przeprowadzonej transformacji, a uzyskanym wymiarem jest

prędkość w jednostkach: piksel na sekundę. Ostatnim etapem do uzyskania rzeczywistego wyniku jest pomnożenie wartości przez stałą  $k$  wyrażoną w metrach piksel.

Przedstawiona koncepcja wsparcia automatycznej kalibracji zostanie rozpatrzona w kontekście proponowanego rozwiązania.

### 3.2.6. Doskonalenie śledzenia pojazdów

Stworzony algorytm powinien sprawdzać się nie tylko w warunkach laboratoryjnych, ale przede wszystkim, w prawdziwym, ruchu drogowym. Przy dużym natężeniu ruchu, szczególnie przy małych prędkościach, gdy odległość pomiędzy pojazdami jest coraz mniejsza, ich odpowiednia ekstrakcja i śledzenie staje się prawdziwym wyzwaniem. W zależności od wykadrowania kamery i stosowanej metody, różne cechy uchwyconego kadru mogą negatywnie wpływać na identyfikację pojazdów. Rzucony cień, częściowe nakładanie pojazdów i coraz bardziej zakrzywiony kąt kadru na dalszym planie sprawiają, że pojazdy się 'zlewają' i ich odseparowanie staje się niemożliwe. Ma to szczególnie negatywny wpływ na metody śledzenia obiektów korzystające z centroidów [7].

Jedno z rozwiązań tego problemu opisano w pracy [8]. Dostrzegając zaistniały problem, autorzy skupili się na śledzeniu konkretnych cech pojazdów, zamiast jego geometrycznego środka. Na kadrze zaaplikowano algorytm identyfikujący cechy łatwe do śledzenia, głównie narożniki, a następnie monitorowano ich położenie. Każdy z przejeżdżających samochodów jest identyfikowany w kilku punktach, co pozwala kontynuować nawet, gdy część samochodu lub cech zostanie przesłoniętych. Dodatkowym problemem koniecznym do rozwiązania jest odpowiednie pogrupowanie cech należących do tego samego pojazdu, co zrealizowano za pomocą podobieństwa wektorów przesunięć.

Trasa uzyskana w ten sposób cechuje się mniejszymi odchyleniami wynikającymi z interakcji z otoczeniem samochodu. Sprawniejszy algorytm śledzenia pojazdu pozwala udoskonalić wyliczanie przebytej drogi, a co za tym idzie uzyskać dokładniejszy wynik prędkości. Rozwiązanie to było inspiracją do zaimplementowania podobnej metody opisanej w podrozdziale 4.6.3.



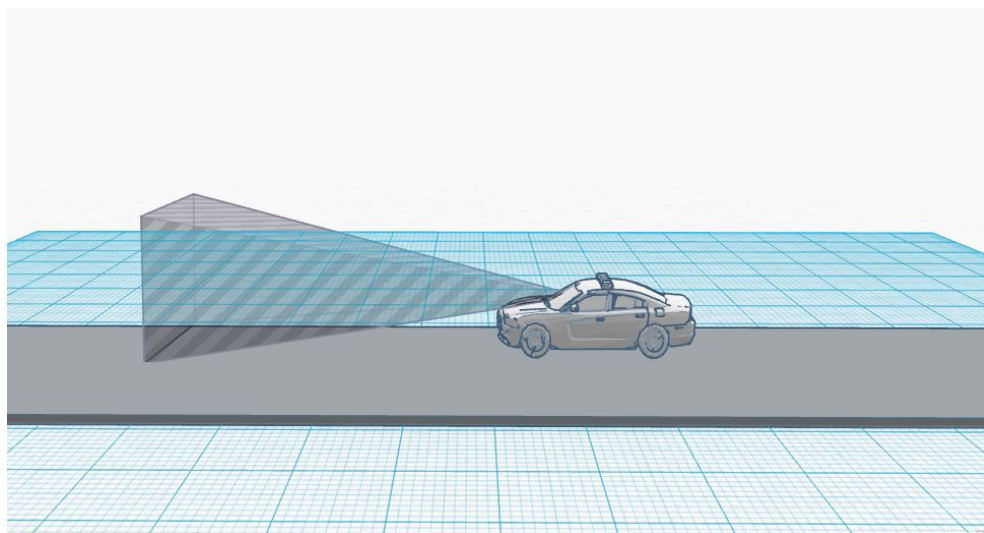
## 4. Koncepcja proponowanego rozwiązania

Poniższy rozdział przedstawia ewolucję tworzonego systemu, opis zastosowanych algorytmów oraz napotkane problemy i sposoby ich rozwiązania. Finalny koncept umieszczono w podrozdziale 4.8.

### 4.1. Położenie kamery

Temat pomiaru prędkości, pomimo pozornie jasno określonego celu, może być rozważany w wielu różnych płaszczyznach i zastosowaniach. Podczas wstępnych badań nad charakterystyką i scenariuszem użycia budowanego systemu opracowano szereg możliwych modeli, które zostały przedstawione i opisane poniżej.

#### 4.1.1. Kamera umieszczona w samochodzie



Rysunek 4.1: Model położenia kamery wewnątrz samochodu.

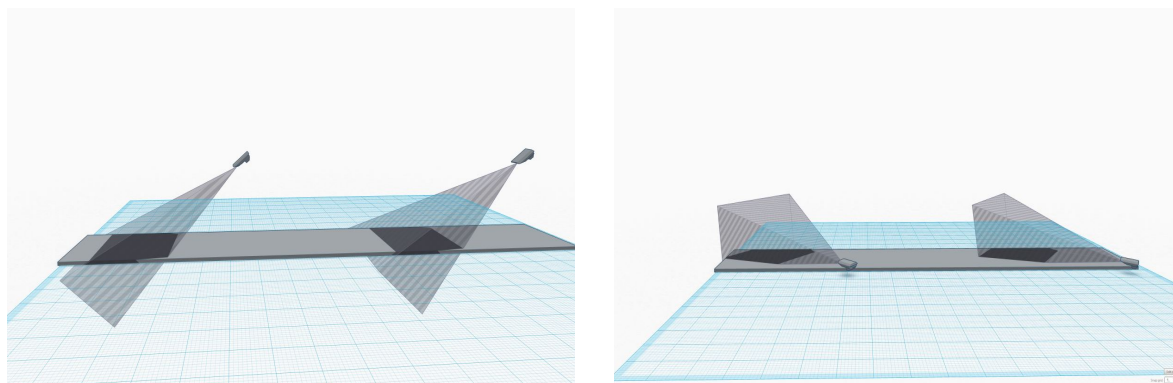
Wyróżniającą się wśród pozostałych była koncepcja korzystania z kamery zainstalowanej wewnątrz samochodu. Rozważany model nawiązywał by do wideorejestratora (podrozdział

3.2.1), ale służył do pomiaru prędkości własnej. Po zaimplementowaniu na platformie mobilnej, niewątpliwym plusem byłaby duża skalowalność rozwiązania. Zintegrowane kamery o dobrych parametrach i wysoka wydajność smartfonów wraz z dostępnością bibliotek do przetwarzania obrazów pozwoliły by na taką realizację. Algorytmiczny aspekt wyznaczania prędkości miał się opierać na śledzeniu odcinków linii przerywanej.

Koncepcja ta została porzucona z dwóch powodów. Rozwiązanie to, nawet wsparte innymi pomocniczymi punktami referencyjnymi było by niemożliwe do zastosowania na dużej części uczęszczanych dróg. Drugi i decydujący argument, w dobie dokładnych pomiarów z nowoczesnych systemów pokładowych oraz powszechności urządzeń mobilnych wyposażonych w GPS takie zastosowanie nie ma uzasadnienia biznesowego.

#### 4.1.2. Dwie kamery

Alternatywą było nawiązanie do drugiego opisanego przykładu - odcinkowego pomiaru prędkości. Rozważano wykorzystanie obrazu z dwóch kamer w różnych konfiguracjach (rys. 4.2). Każdy model skupiał się na stworzeniu dwóch przesuniętych względem siebie punktów pomiarowych, tak jak w podrozdziale 3.2.2. Podejście to jest bardzo dokładnie opracowane i zrealizowane w wielu komercyjnych projektach [UnicamVELOCITY<sup>1</sup>, Jenoptik Point-to-Point<sup>2</sup>, Neurocar Section Speed Control<sup>3</sup>]. Zmniejsza to możliwość wkładu badawczego w realizowany system i wartość naukową samej pracy. Dodatkowo od strony technicznej wymaga większej ilości przygotowań na miejscu pomiaru i większych zasobów sprzętowych.



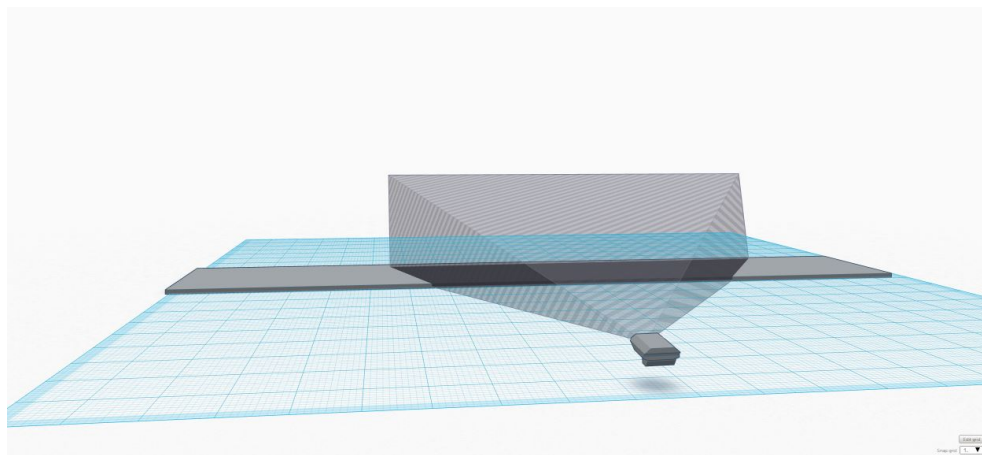
Rysunek 4.2: Model położenia kamery z użyciem dwóch kamer.

Pomimo możliwości otrzymania bardzo dokładnych wyników, podejście to niesie ze sobą wiele problemów, które przesądziły o fakcie, że koncepcja nie przeszła do fazy realizacji.

<sup>1</sup><http://www.camea.cz/en/traffic-applications/enforcement-systems/section-speed-control-average-speed-check/>

<sup>2</sup><http://www.jenoptik.com/en-section-speed-control-point-to-point-measurement>

<sup>3</sup><https://neurosoft.pl/produkty/neurocar/neurocar-section-speed-control/>



Rysunek 4.3: Model położenia kamery z widokiem bocznym.

### 4.1.3. Kamera zwrócona prostopadle do linii drogi

Stosunkowo mało zostało poczynione w celu pomiaru prędkości za pomocą kamery umieszczonej prostopadle do kierunku jezdni. Perspektywa ta (rys. 4.3) cechuje się brakiem zniekształceń obiektów rejestrowanych w obszarze pomiaru i zachowaniem ich właściwości geometrycznych. Z drugiej strony, nie dysponuje informacją o pasach ruchu oraz wprowadza nowy problem - nakładanie się mijających pojazdów. Dodatkowym minusem jest wymóg zapewnienia dużej odległości kamery od jezdni dla osiągnięcia odpowiednio szerokiego pola widzenia.

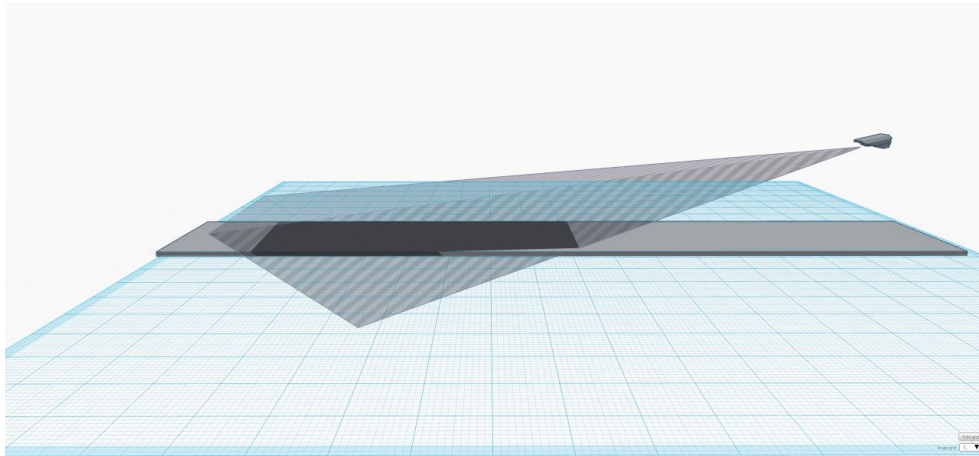
Warto wspomnieć, że budowa takiego czujnika ma duże perspektywy na rozwój w przyszłości. Powstaje coraz więcej systemów wspomagających kierowcę oraz projektów self-driving-car<sup>4</sup>. Wymaga to rozwiązania niespotykanych wcześniej problemów i automatycznej analizy otoczenia pojazdu [9], między innymi podczas wjazdu na skrzyżowania. Jeżeli jednym z założeń systemu, będzie możliwość jego instalacji w samochodzie, znajdzie on szerokie zastosowanie w nowatorskich systemach z wykorzystaniem sztucznej inteligencji. Niestety koncepcja stacjonarna cechuje się zbyt małym zastosowaniem oraz ograniczoną możliwością wyboru sceny, aby została zrealizowana.

### 4.1.4. Zastosowane rozwiązanie

Ostatecznie wrócono do najbardziej klasycznego, powszechnego i dającego najszersze możliwości modelu umieszczenia kamery - powyżej, centralnie i pod kątem od 20-45 stopni do jezdni (rys. 4.4).

Obraz wygenerowany z takiego kadru zawiera komplet informacji o parametrach ruchu drogowego. Widoczne są odseparowane pasy i nie występuje problem całkowitego nakładania

<sup>4</sup><http://www.google.com/selfdrivingcar/>



Rysunek 4.4: Model finalnego położenia kamery względem jezdni.

nia obiektów (wspomnianego w podrozdziale 4.1.3). Daje możliwość rozbudowy o algorytm automatycznego rozpoznawania tablic stosowany w systemach odcinkowych pomiarów prędkości (podrozdział 3.2.2). Wymaga najmniejszych przygotowań sceny wśród zaproponowanych koncepcji - ogranicza się do zainstalowania pojedynczej kamery i nie wymaga wykonywania lokalnych pomiarów.

Ostatnim ważnym argumentem jest fakt dostępnej infrastruktury. W Polsce rozmieszczonych jest około XXXX<sup>5</sup> kamer zamontowanych przez Generalną Dyрекcję Dróg Krajowych i Autostrad, z których około 50% spełnia założenia projektowe. Obecnie, służą one do monitorowania natężenia ruchu, naliczenia opłat oraz zliczania przejeżdżających samochodów. Nic nie stoi na przeszkodzie, aby umożliwić wzbogacenie ich w nową funkcjonalność.

## 4.2. Kalibracja kamery

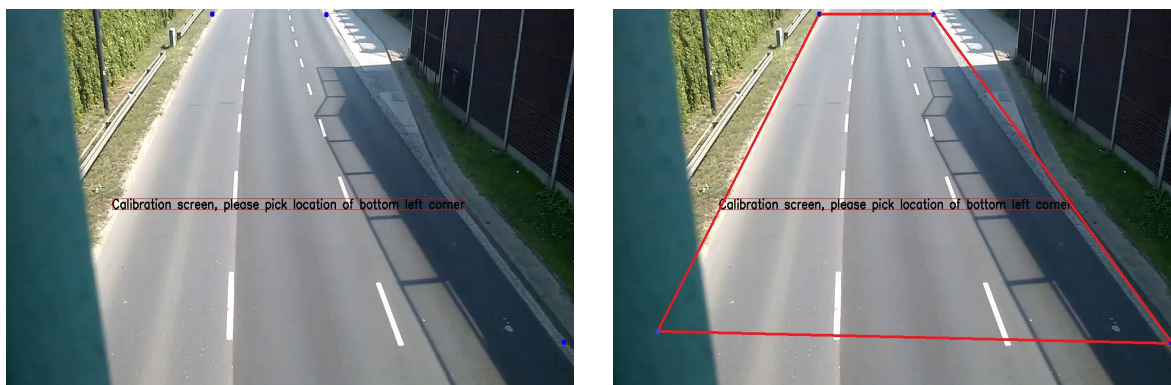
Poniższy podrozdział opisuje wyznaczenie z pola widzenia (Field of View - FOV) obszaru zastosowania (Region of Interest - ROI) algorytmu. Punktem wyjścia dla dalszego etapu jest trapezoid wytyczony na powierzchni jezdni odzwierciedlający obszar, w którym będzie dokonywany pomiar (rys. 4.5b).

Krok ten, wykonywany jest za każdym razem przed rozpoczęciem trybu pomiarowego i jego wynik jest aktualny do momentu zmiany położenia kamery. Algorytm kalibracji odbywa się na pasywnym kadrze niezawierającym pojazdów. Jeżeli taki kadr nie jest możliwy do uzyskania, można wykorzystać automatycznie wygenerowane tło (podrozdział 4.5).

Sceneria powinna spełniać poniższe warunki:

- maksymalne nachylenia drogi do osi Y w układzie współrzędnych obrazu jest równe 30°

<sup>5</sup>dane zostaną uzupełnione



Rysunek 4.5: Przykład manualnej kalibracji sceny.

- droga pozbawiona jest zakrętów na odcinku przynajmniej 50 metrów od pozycji kamery
- powierzchnia jezdni jest płaska i nachylona pod stałym kątem

Sposobem wytyczenia ROI jest manualne wyznaczenie narożników pożądanego obrazu (rys. 4.5). Interakcja z użytkownikiem pozwala na analizę bardzo skomplikowanej scenerii i rozwiązania problemów wynikających z nietypowego układu jezdni (skrajnego kąta nachylenia, dodatkowych pasów do skrętu w lewo lub w prawo).

### 4.3. Transformacja współrzędnych pola widzenia na współrzędne rzeczywiste

Dzięki założeniom projektu możemy przyjąć, że pojazdy, których prędkość będzie mierzona, mogą się poruszać tylko w kierunku kamery lub przeciwnym. Trasę przebytą przez pojazd można traktować jako prosty odcinek drogi w przestrzeni jednowymiarowej. W kontekście FOV (rys. 4.6a) odległość w przestrzeni obrazu ma liniową zależność do odległości w rzeczywistym układzie współrzędnych.

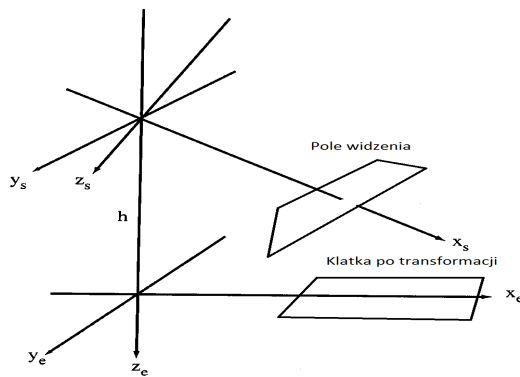
Na podstawie narożników ROI i prostych przez nie wytyczonych, zostaje wyliczona macierz transformacji. Po jej zaaplikowaniu otrzymujemy obraz 4.6b, na którym zbiegające linie z rysunku 4.6a są do siebie równoległe. Zmianie ulega stosunek odległości, który po transformacji jest stały, oraz kierunek ruchu pojazdów - na całkowicie wertykalny. Obraz wynikowy jest obrazem rzutu pionowego na powierzchnie jezdni co zaprezentowano na rysunku 4.6c. Identyfikacja przebytej odległości ogranicza się do analizy położenia punktu reprezentującego pojazd na osi Y.



(a) Obraz oryginalny.



(b) Obraz po transformacji perspektywy.



(c) Geometria kamery (obraz tymczasowy).

Rysunek 4.6: Transformacja współrzędnych obrazu na współrzędne rzeczywiste.

## 4.4. Estymacja odległości

Poniższy podrozdział opisuje operacje służące nadaniu relacjom między pikselami fizyczne wymiary. Efekt końcowy przedstawia współczynnik metrów/piksel w pionowym przesunięciu na obrazie wynikowym operacji opisanej w podrozdziale 4.3. Parametr ten w dalszej części pracy nazywany jest współczynnikiem dystansu.

### 4.4.1. Zewnętrzne dostarczenie danych

System zgodnie z założeniami pozwala na wprowadzenie długości referencyjnej manualnie przez użytkownika. Jest to konieczne do przeprowadzania pomiaru w miejscach, gdzie nie widnieje linia przerywana i nie można wykorzystać metod z podrozdziałów 4.4.3 i 4.4.2. Dodatkowo, funkcjonalność ta ma służyć do prototypowania algorytmów oraz pomiarów kontrolnych i porównania wyników końcowych dwóch pozostałych rozwiązań uzyskiwania odległości.

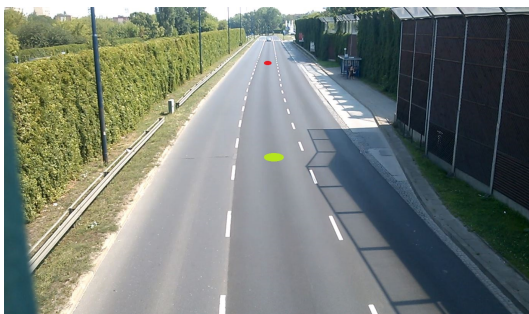
Do ustalenia współczynnika dystansu należy przeprowadzić fizyczny pomiar na miejscu sceny i wprowadzić wartość numeryczną do systemu wraz z zaznaczeniem obu końcowych punktów pomiarowych. Ważne, aby reprezentowane punkty znajdowały się w zasięgu ROI i dało się je odszukać w FOV.

Wprowadzona odległość nie musi reprezentować całego obszaru działania algorytmu, jest to jednak zalecane. Im dłuższa długość odcinka  $d_s$  jest większa, tym błąd całkowity  $\Delta R$  jest mniejszy zgodnie ze wzorem:

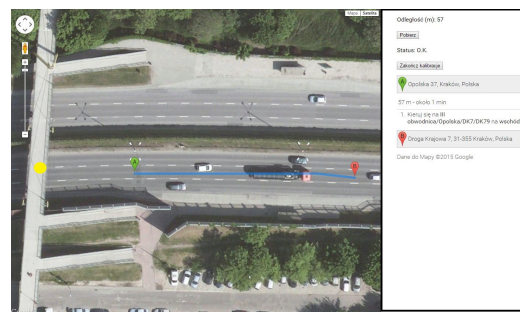
$$\Delta R = \frac{d_R}{d_s} * \Delta s \quad (4.1)$$

$d_R$  - długość obszaru pomiaru,  $\Delta s$  - błąd pomiaru odcinka referencyjnego

#### 4.4.2. Użycie Google Maps



(a) Field of Vision



(b) Panel do pomiaru odległości za pomocą Google Maps (żółty marker oznacza położenie kamery)

Rysunek 4.7: Przykład użycia map Google do określenia odległości.

Alternatywą ręcznego wykonywania pomiaru jest wykorzystanie silnika Google Maps. Rozwiązanie to podziela większość założeń przedstawionych w poprzednim podrozdziale, wymaga natomiast mniejszego wkładu pracy ze strony użytkownika.

Mapy Google udostępniają możliwość pomiaru odległości pomiędzy dwoma dowolnymi, wybranymi na mapie, punktami. Można założyć, że dokładna lokalizacja kamery jest zawsze znana, a jedynym warunkiem powodzenia tej metody jest możliwość zidentyfikowania przynajmniej 2 charakterystycznych punktów, aby użytkownik był w stanie dokonać markowania (rys. 4.7b). Koncepcja ta zakłada wykorzystanie map ze zdjęciami satelitarnymi do wyboru punktów A i B, które następnie muszą zostać oznaczone na FOV kadru kalibracyjnego (rys. 4.7a). Wyliczona odległość zostaje pobrana do systemu, a dalsze obliczenia przebiegają jak w koncepcji z sekcji 4.4.1.

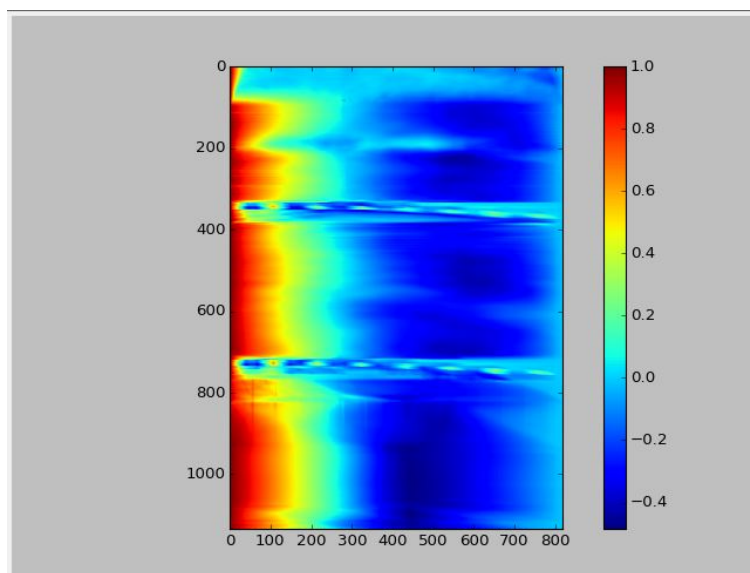
Plusem tej metody jest możliwość jej zastosowania pomimo braku warunków wystarczających dla metody opisanej w podrozdziale 4.4.3 oraz brak konieczności odwiedzania sceny.

### 4.4.3. Analiza widocznej linii przerywanej

Najlepsze rozwiązanie jakie można dostarczyć to w pełni automatyczny system kalibrujący się z minimalnym udziałem użytkownika. Wymaga to spełnienia większej ilości kryteriów i jest trudniejsze do osiągnięcia, ale w sytuacji, gdy operator dysponuje kilkoma tysiącami kamer, jest niebywałym plusem.

Pierwszy algorytm detekcji z użyciem pasów działał według następującej koncepcji. Na etapie kalibracji, na podstawie kadru po transformacji perspektywy (rys. 4.6b), zostają wyliczone proste odzwierciedlające linie przerywane (może ich być więcej niż jedna). System zlicza liczbę białych odcinków na każdej z linii i używa ich do wyliczenia odległości. Brane jest pod uwagę  $n+1$  odcinków i  $n$  odstępów. Skrajne fragmenty pasów dobierane są tak, aby nie miały punktów stycznych z obwodem ROI. Pozwala to uniknąć sytuacji, gdy ich część nie zmieściła by się w kadrze skracając długość odcinka.

Rozwiązanie to się nie sprawdziło, ponieważ długość każdego fragmentu linii jest bardzo małym ułamkiem całej wysokości obrazu. Dobór parametrów algorytmu detekcji powodował, że albo nie znajdował żadnych linii odpowiadających pasom, albo, gdy je wykrywał, identyfikował dużą liczbę obiektów postronnych.



Rysunek 4.8: Wykres autokorelacji obrazu.

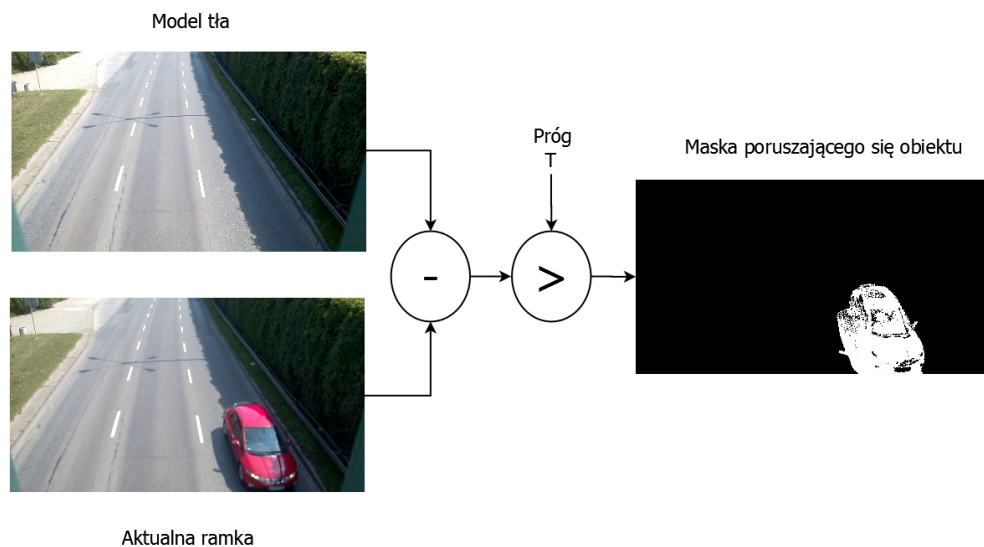
W finalnym rozwiązaniu wzorem pracy [6] zastosowano funkcję autokorelacji (AC). Na obrazie po transformacji perspektywy, dla każdej kolumny wyliczono wartość AC i zestawiono

dla całego obrazu (rysunek 4.8). Miejsca gdzie wartości AC przyjmują postać sinusoidalną reprezentują lokalizacje pasów. Identyfikacja największych pików odpowiada wyliczeniu okresu częstotliwości występowania pasów na obrazie po transformacji. Znajomość tego okresu wyrażonego w pikselach jest punktem wyjścia do wyliczenia współczynnika dystansu.

Zadaniem użytkownika jest wybór rodzaju widocznej linii, który przekłada się na charakterystykę odległości pasów. Według przepisów ruchu drogowego dotyczących znaków poziomych istnieje co najmniej 6 rodzajów takich pasów (odmiany znaków P-1, P-6 oraz P-7), z których każdy ma inne zastosowanie oraz długość i okresowość odcinków.

## 4.5. Detekcja ruchu

Identyfikacja obecności pojazdów w kadrze opiera się na algorytmie detekcji ruchu. Ideę jego działania przedstawiono na rysunku 4.9. Decydującymi czynnikami do wyboru tej metody były cechy obrazu wynikające z założeń projektowych: pojazdy które należy śledzić są w ciągłym ruchu oraz poruszają się po ściśle wytyczonej trasie, odseparowanej od innych ruchomych obiektów.



Rysunek 4.9: Schemat działania algorytmu ekstrakcji tła.

Na podstawie kilkunastu klatek sekwencji wideo stworzony zostaje model tła wykorzystywany do subtrakcji obrazu. Od niego bezpośrednio zależy skuteczność wyekstrahowania poruszających się elementów. Najprostszym podejściem jest generacja tła za pomocą ważonego uśredniania obrazu. Obszar wynikowy uzyskiwany jest zgodnie z poniższym wzorem ( $\alpha$  określa współczynnik adaptacji):

$$dst(x, y) \leftarrow (1 - \alpha) * dst(x, y) + \alpha * src(x, y) \quad (4.2)$$

Wynik takiego podejścia daje niewystarczające efekty, dlatego posłużono się adaptacyjnym algorytmem wykorzystującym mieszanki gaussowskie [10]. Rozwiązują to część problemów związanych ze zmianą oświetlenia sceny i drganiami obiektów, które nas nie interesują (gałęzi drzew lub cieni przez nie rzuconych).

W dalszym ciągu jednak błędy ekstrakcji są nieuniknione i mogą przyjmować różne formy:

rysunek 4.10a jeżeli adaptacja odbywa się zbyt wolno, pojazd zostaje uznany za część stacjonarną tła, co skutkuje detekcją ruchu w miejscu, w którym samochód znajdował się chwilę wcześniej (tzw: trailing). Rozwiązanie: dobór parametrów algorytmu.

rysunek 4.10b niekompletne wykrycie samochodu - cechy z bardziej kontrastujące z własnym otoczeniem (jak krawędzie elementów karoserii) zostają wykryte prawidłowo, ale całkowity obrys samochodu zawiera ubytki. Rozwiązanie: zastosowanie morfologicznej operacji zamknięcia, która stosunkowo nie zwiększa rozmiarów obiektu, zamyka natomiast małe luki i łączy pobliskie obiekty.

rysunek 4.10c szum - duża ilość małych obiektów rozsianych po całym obrazie wynikowym. Często jest wynikiem drgania kamery lub nagłej zmiany oświetlenia. Rozwiązanie: Filtracja obrazu względem powierzchni geometrycznej wykrytych obiektów. Usunięcie wszystkich, których ilość pikseli jest mniejsza niż 5% pikseli całego kadru.

rysunek 4.10d obiekty postronne - poruszane wiatrem gałęzie, piesi idący chodnikiem i inne ruchome elementy obrazu, które nie są interesujące w kontekście problemu. Rozwiązanie: ograniczenie działania algorytmu detekcji ruchu do obszaru ROI.

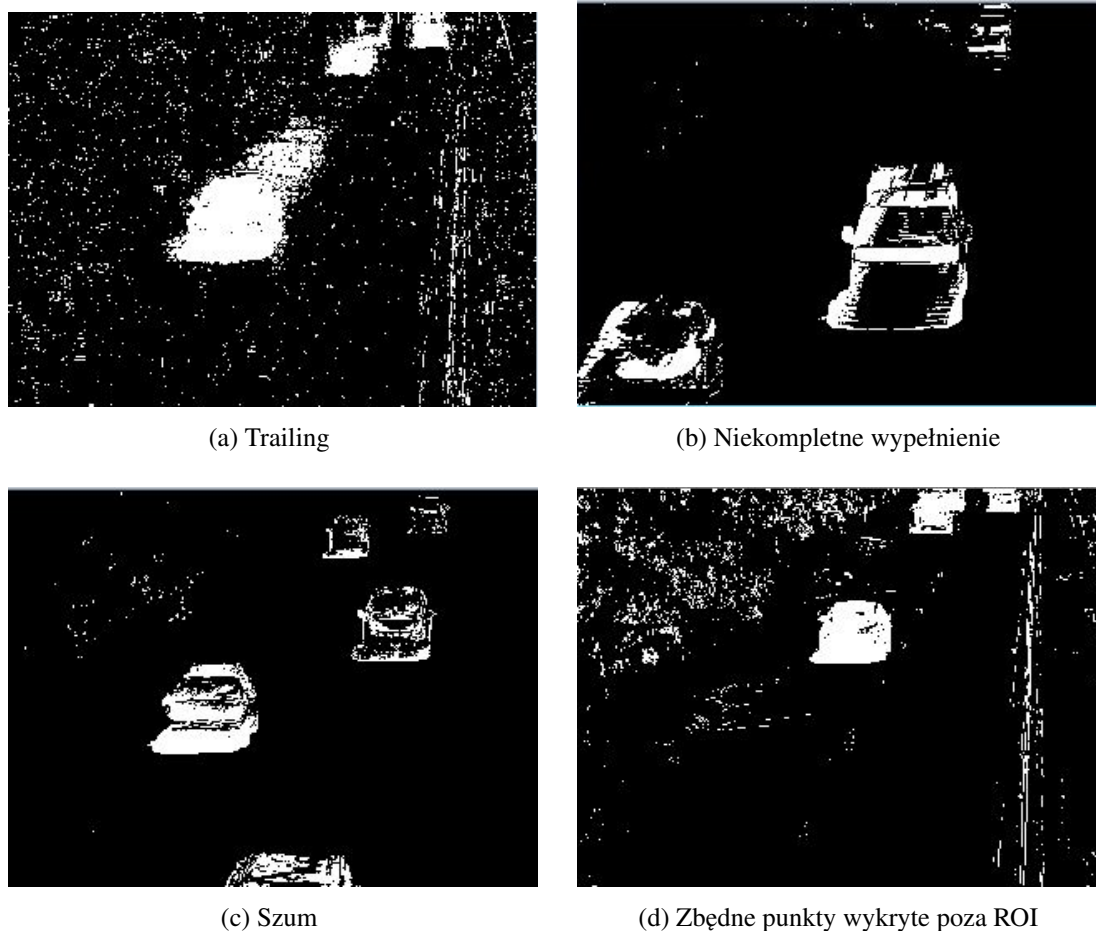
Wszystkie powyższe operacje zostają zaaplikowane w celu poprawienia wyniku algorytmu. Efektem tego kroku jest maska wynikowa, z której usunięto wszystkie zbędne elementy. Te które pozostały widoczne są reprezentacją pojazdów, które podlegają kontroli prędkości.

## 4.6. Identyfikacja i śledzenie obiektów

Kolejnym krokiem jest nadanie widocznym obiektom znaczenia i śledzeniu ich na kolejnych klatkach. Poniższy rozdział zawiera opis logiki odpowiadającej za interpretowanie sceny oraz opis trzech metod odszukiwania elementów na kolejnym obrazie.

Algorytm pracuje na wytyczonym w podrozdziale 4.2 ROI. Projekt zakłada możliwość użycia różnych reprezentacji obszaru (rysunek 4.11) i przetestowanie wpływu formy na pomiar. Wyniki zawarto w rozdziale 6.

Schemat algorytmu identyfikacji przedstawiono na diagramie 4.16. Do jego zastosowania system przechowuje dwa zestawy informacji o pojazdach: listę potwierdzonych samochodów,



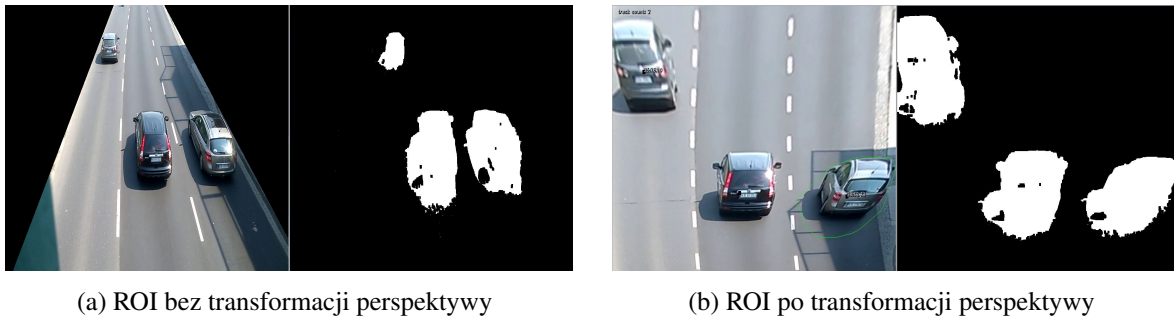
Rysunek 4.10: Problemy ekstrakcji poruszających się obiektów

które aktualnie znajdują się w kadrze (zwaną bazą pojazdów) oraz listę samochodów, które potencjalnie znajdują się w kadrze (zwaną listą oczekującą). Jest to konieczne, aby zminimalizować problem dublowania się wpisów o tych samych pojazdach pod różnymi ID. Zdarza się tak dlatego, że pomimo obecności na obrazie, samochód nie zawsze zostanie poprawnie wyszukany. Jeżeli ponownie uda się go znaleźć po dwóch lub trzech klatkach, system dąży do kontynuowania śledzenia, zamiast przypisywania pojazdowi nowego identyfikatora.

Dla każdej klatki z sekwencji wideo zostają wykonane trzy operacje.

Pierwsza jest odpowiedzialna za analizę obiektów z maski wyekstrahowanych pojazdów (rys. 4.16a). Każdy element jest wyszukiwany w bazie pojazdów. Jeżeli nie zostanie znaleziony, wyszukiwanie jest kontynuowane w liście oczekującej. Jeżeli tutaj również zakończy się niepowodzeniem, zostanie mu przypisany identyfikator i system doda go do listy pojazdów potencjalnie znajdujących się w kadrze.

Druga odpowiada za analizę listy oczekującej (rys. 4.16b). Jej zadaniem jest zwalidowanie



(a) ROI bez transformacji perspektywy

(b) ROI po transformacji perspektywy

Rysunek 4.11: Możliwe reprezentacje danych wejściowych do algorytmu identyfikacji i śledzenia.

poprawności obiektów przechowywanych w systemie. Algorytm podejmuje decyzje na podstawie aktualizacji trasy pojazdu:

- Jeżeli trasa samochodu była wprowadzana przez 10 kolejnych klatek, oznacza to, że samochód nie ma swojej reprezentacji w bazie pojazdów i należy go dodać. Wszystkie informacje od początku akwizycji zostają przekazane wraz z obiektem, dzięki czemu trasa z ostatnim 10 klatek nie zostaje utracona.
- Jeżeli samochód nie był uaktualniany przez 5 kolejnych klatek, oznacza to jeden z dwóch scenariuszy. Samochód, który znajduje się w bazie pojazdów, a z jakichś powodów nie został wykryty i stworzył nową reprezentację, został znowu odnaleziony w liście o wyższym priorytecie. Informacje o trasie aktualizowane są ponownie w bazie pojazdów. Drugą opcją jest możliwość, że obiekt nie był pojazdem tylko błędem ekstrakcji, który był widoczny przez kilka klatek, po czym zniknął. W obu wymienionych sytuacjach algorytm ma za zadanie usunąć reprezentacje obiektów z listy oczekującej.

Trzecia operacja jest odpowiedzialna za stan obiektów w bazie pojazdów (rys. 4.16c). Jej zadaniem jest analiza czasu aktualizacji pojazdów potwierdzonych. Jeżeli trasa samochodu nie była uaktualniana przez 5 kolejnych klatek, algorytm zakłada, że obiekt opuścił kadr i usuwa go z bazy.

Takie podejście okazało się skuteczne i zapewnia zmniejszenie zduplikowanych referencji dotyczących tego samego samochodu, równocześnie pozbywając się wpisów, które identyfikują błędy algorytmu jako nowe pojazdy. Jego końcowa wiarygodność jest ściśle zależna od skuteczności metod użytych do wyszukania obiektów. Ich opis znajduje się poniżej.

### 4.6.1. Camshift

Continuously Adaptive Meanshift [11]. Jest on przedstawionym w 1998 roku rozwinięciem algorytmu mean-shift. Służy do znajdowania nowej lokalizacji wybranego okna z uwzględnieniem jego rotacji i zmiany wielkości. Ideą działania jest odnajdywanie nowego centrum, w której dystrybucja cech takich jak kolor i tekstura odpowiada poszukiwanej.

Algorytm mean-shift ma zastosowanie nie tylko w przetwarzaniu obrazów. Jego celem jest znalezienie lokalnego ekstremum gęstości punktów. Jego koncepcja opiera się na danych ograniczonych wybranym oknem i działa następująco: wylicza wynik w obrębie określonego okna. Znajduje środek masy, po czym przesuwa okno, w miejsce uzyskanej wartości. Powtarza obliczenia aż do momentu, gdy okno przestaje się przesuwać - czyli, gdy centrum okna pokrywa się ze środkiem masy.

Zastosowanie tej metody (z uwzględnieniem jej usprawnienia przez Gary'ego Bradski'ego) do śledzenia elementów na obrazie składa się z następujących kroków:

1. Wybranie okna z interesującą cechą na poprzedzającej klatce.
2. Wyliczenie cechy danego obszaru korzystając z histogramu.
3. Obliczenie dystrybucji danej cechy na bieżącej klatce (rys. 4.12d).
- 4 Zastosowanie algorytmu mean shift
5. Uaktualnienie rozmiaru i rotacji okna do obszaru z najlepszą dystrybucją poszukiwanej cechy

Podczas testów aplikacji, metoda ta poprawnie śledziła pojazdy (rys. 4.12), ale wykazała się zbyt dużą złożonością obliczeń. Problem nie występował przy analizie pojedynczych samochodów, diametralnie natomiast wzrastał czas obliczeń przy dużej liczbie obiektów. Wynika to z konieczności wyliczenia cechy i dystrybucji cechy dla każdego z elementów osobno. Algorytm ten nie został włączony do koncepcji finalnej tworzonego rozwiązania.

### 4.6.2. Wykorzystanie centroidu

Analizując założenia projektu, zauważono, że algorytm śledzenia może ulec znacznemu uproszczeniu. Najważniejszym spostrzeżeniem jest fakt, że obiekty (pojazdy) nie mogą się przesłaniać i wchodzić w kolizje (dotyczy to tylko ROI - częściowe przesłanianie obiektów jest możliwe na dalszym planie i wynika z zakrzywienia perspektywy). Biorą to pod uwagę, do znalezienia samochodu pomiędzy klatkami posłuży informacja dostępna w oknie o wielkości prostokąta ograniczającego śledzony obiekt.



(a) Poprawne śledzenie obiektu(1)



(b) Poprawne śledzenie obiektu(2)



(c) Poprawne śledzenie obiektu(3)



(d) Dystrybucja poszukiwanej cechy

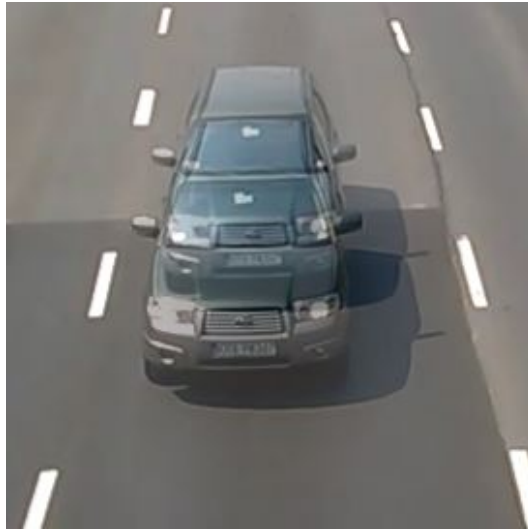
Rysunek 4.12: Przykład działania algorytmu camshift.

Do pewnej prędkości przejeżdżających pojazdów oraz przy częstotliwości klatek na sekundę równej 30fps różnica w lokalizacji (rys. 4.13a) pojazdów na klatce  $n$  i  $n+1$  jest tak niewielka, że dla znalezienia odwzorowania tego samego samochodu na 2 kolejnych obrazach, wystarczy znaleźć obiekt najbliższy poprzedniej lokalizacji. Dla osiągnięcia punktowej reprezentacji elementów wykrytych na etapie ekstrakcji pojazdów zastosowano dwie operacje. Znalezienie minimalnej otoczki wypukłej (rys. 4.13b - zielony wielokąt) oraz wyliczenie jej środka ciężkości (rys. 4.13b - czerwony kwadrat). Dodatkowym zyskiem z zastosowania powyższych operacji jest pozbycie się ubytków wewnątrz wykrytego samochodu, których nie zniwelowały operacje morfologiczne (rys. 4.13c).

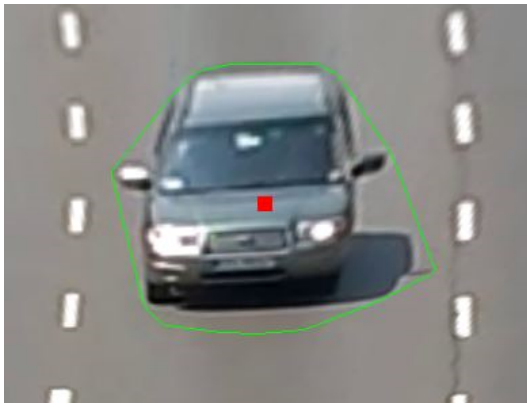
Algorytm poprawnie identyfikuje samochód, jeżeli środek ciężkości z klatki  $N+1$  zawiera się w otoczce zewnętrznej pojazdu z klatki  $N$ . Rozwiązanie to cechuje się niską złożonością obliczeniową i poprawnie realizuje założenie na testowym odcinku drogi.

### 4.6.3. Cechy charakterystyczne

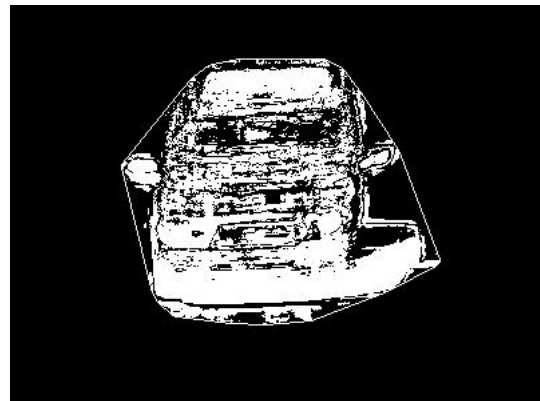
Analiza istniejących rozwiązań przeprowadzona w rozdziale 3 ukazała problem dokładności odzwierciedlenia obiektu na obrazie i jej wpływu na wynik pomiaru. Pomimo, że potwierdzono używanie centroidu jako wiarygodnego odwzorowania [7], nie jest to najlepsze z możliwych rozwiązań. Korzystając z wniosków wyciągniętych w podrozdziale 3.2.6 zbudowano koncepcje wykorzystującą śledzenie indywidualnych cech pojazdu, ale nie dzielącą problemu odpowiedniego grupowania uzyskanych wyników.



(a) Różnica położenia samochodu pomiędzy dwoma klatkami przy prędkości 70km/h



(b) Poprawne śledzenie obiektu(1)

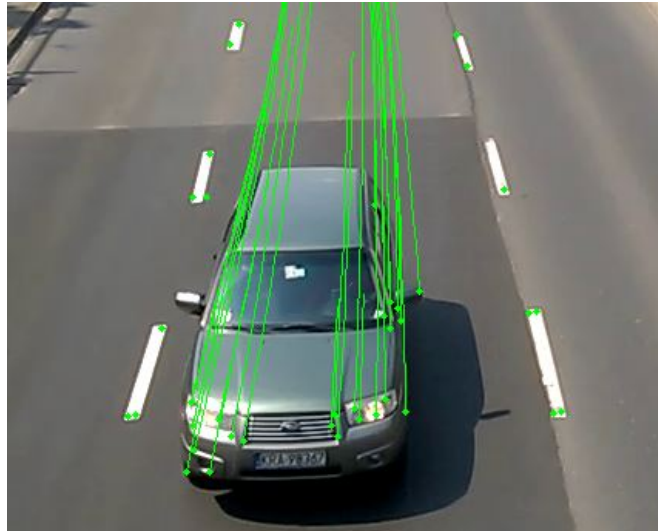


(c) Poprawne śledzenie obiektu(2)

Rysunek 4.13: Przykład działania algorytmu camshift.

Dla uniknięcia wspomnianego przyporządkowywania cech do poszczególnych samochodów, odwrócono sposób działania. System najpierw znajduje samochód (element z maski ekstrakcji), a potem w jego obrębie identyfikuje cechy nadające się do śledzenia. Są nimi punkty, których gradient jasności ma charakter dwuwymiarowy - narożniki. Krok ten jest wykonywany dla każdego z obiektów podczas dodawania do listy pojazdów oczekujących (podrozdział 4.6).

Identyfikacja pojazdów pomiędzy klatkami odbywa się za pomocą koncepcji przepływu optycznego (optical flow) realizowanej piramidalnym algorytmem Lucas-Kanade [12]. Jest to metoda pozwalająca wyliczyć wektory przesunięć dla zbioru pojedynczych punktów pomiędzy dwoma obrazami. Z zastosowania powyższego algorytmu na punktach przechowywanych w reprezentacji pojazdów, otrzymujemy kilka śladów dla każdego z obiektów (rys. 4.14). Do dalszych obliczeń przejechanej odległości wykorzystujemy najdłuższy, lub średnią z wszystkich



Rysunek 4.14: Przykład użycia przyptywu obrazu do śledzenia pojazdu.

przechowywanych.

## 4.7. Kalkulacja prędkości

Końcowa kalkulacja prędkości korzysta z czasu wynikającego z częstotliwości klatek na sekundę, współczynnika odległości wyliczonego na etapie kalibracji i trasy samochodu przebytej w pikselach.

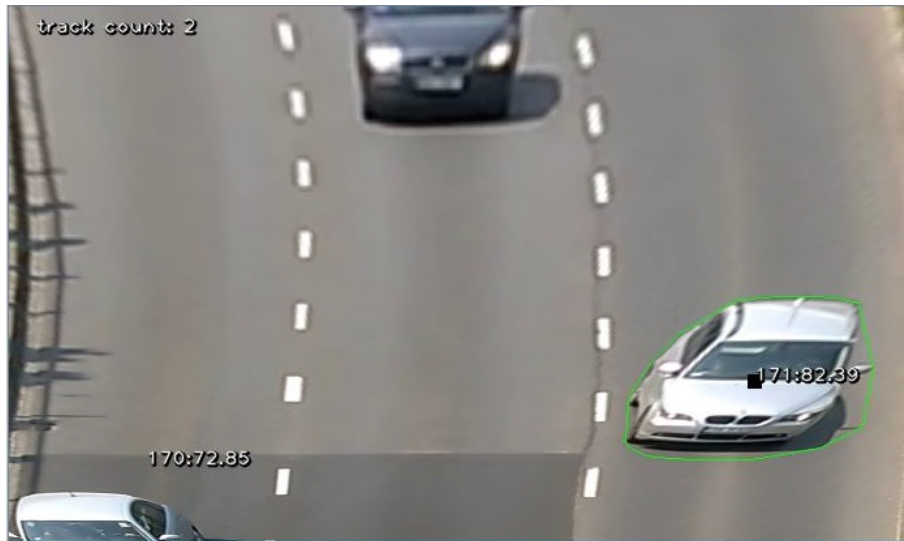
Algorytm rozwiązuje trzy poniższe równania ( $N$  - ilość klatek,  $s_{pxl}$  - droga w pikselach przebyta w  $N$  klatkach,  $fps$  - częstotliwość klatek,  $ratio$  - współczynnik dystansu):

$$t = N * \frac{1}{fps}, s = s_{pxl} * ratio, V = \frac{s}{t} \quad (4.3)$$

Koncepcja zakłada wyliczenie prędkości w dwóch trybach, ciągłym i końcowym. Podczas akwizycji samochodu w ruchu system dokonuje pomiaru prędkości chwilowej na podstawie danych z 5 ostatnich klatek. Przykład takiego wyniku widać na rysunku 4.15. Po opuszczeniu kadru przez samochód zostaje wyliczona prędkość średnia z całej przebytej trasy.

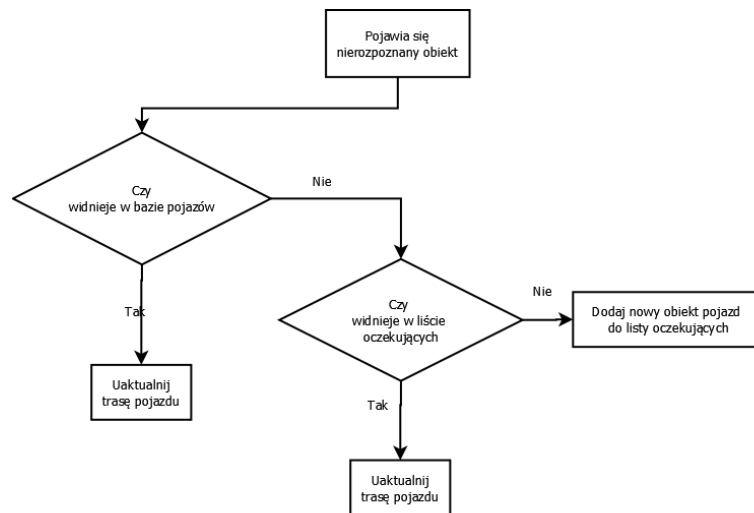
## 4.8. Realizacja

Podsumowując wcześniejsze podrozdziały system opiera się na śledzeniu pojazdów z perspektywy kamery skierowanej pod kątem jak na rysunku 4.4. Projekt działa w dwóch trybach: kalibracyjnym i pomiarowym. W tym pierwszym zostaje wybrany ROI (podrozdział 4.2) oraz określona zależność pomiędzy odległością na transformowanym obrazie (podrozdział 4.3), a

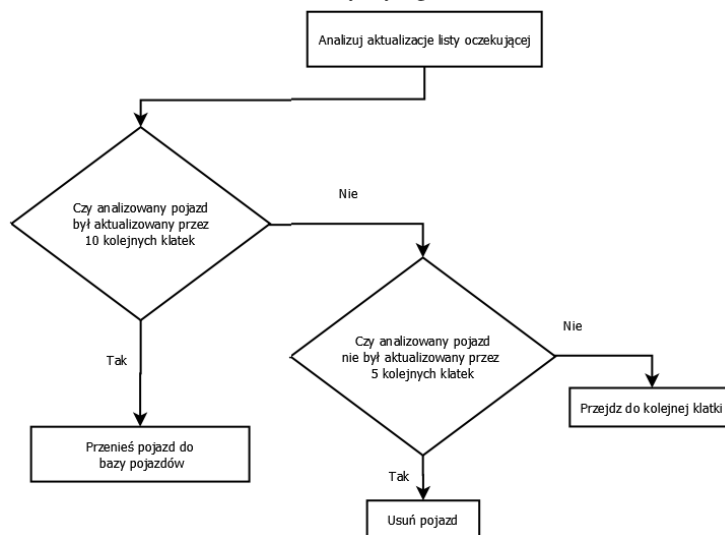


Rysunek 4.15: Wyniki pomiaru prędkości dla 2 przejeżdżających pojazdów - widoczne id oraz prędkość.

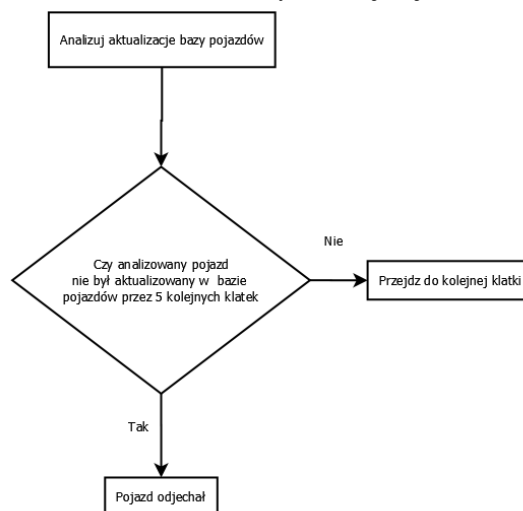
odległością w przestrzeni. Informacje odnośnie dystansu mogą być podane przez użytkownika, pobrane z mapy Google lub wyliczone na podstawie linii przerywanej (podrozdział 4.4). Następnie system przechodzi w tryb pomiarowy - nieskończoną pętlę wykonującą pomiar właściwy. Obiekty reprezentujące przejeżdżające pojazdy są wykrywane za pomocą algorytmu detekcji ruchu (podrozdział 4.5). Są śledzone na przestrzeni przebytej trasy wykorzystując namierzanie najbliższego centroidu lub algorytm przepływu obrazu. Wynikiem pracy systemu są prędkości: chwilowa i średnia, wyliczone dla każdego przejeżdżającego pojazdu.



(a) Analiza wykrytego obiektu



(b) Analiza listy oczekującej



(c) Analiza bazy samochodów systemu

Rysunek 4.16: Schemat działania algorytmu ekstrakcji tła.

## 5. Realizacja aplikacji

### 5.1. Użyte narzędzia i biblioteki

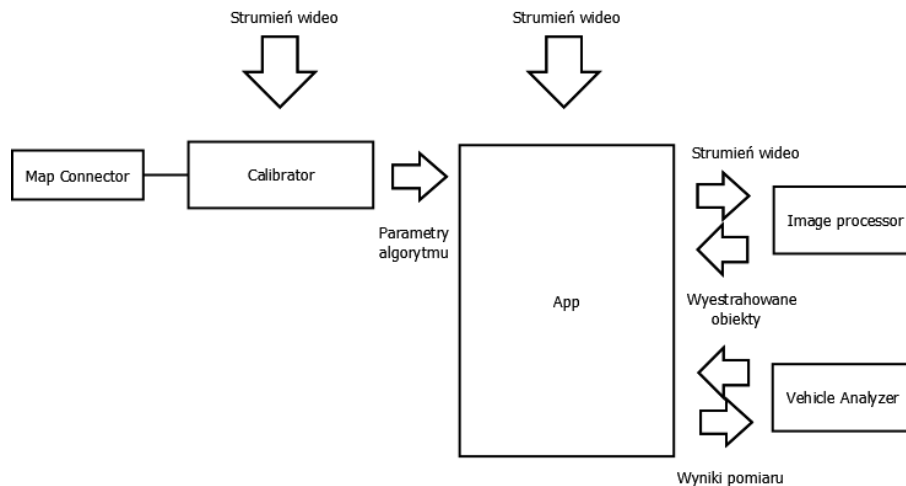
Aplikacja napisana na potrzeby poniższej pracy została zaimplementowana w językach python i javascript z wykorzystaniem następujących narzędzi:

- OpenCV (licencja BSD, wrapper dla języka python wykorzystujący moduły napisane w C/C++) - przechwytywanie i przetwarzanie obrazu
- Autobahn (licencja MIT) - obsługa protokołu WebSocket
- Twisted (licencja MIT) - obsługa protokołu WebSocket
- Google Maps JavaScript API (licencja GOOGLE MAPS API) - integracja funkcjonalności map Google

### 5.2. Architektura aplikacji

W skład aplikacji wchodzi następujące moduły: `calibrator`, `map_connector`, `image_processor` i `vehicle_tracer`. W dwóch pierwszych zrealizowano funkcje opisane w podrozdziałach 4.2, 4.3 i 4.4. Detekcja ruchu została zaimplementowana w module `image_processor`. Identyfikacja i śledzenie pojazdów (4.6) oraz końcowe obliczanie wyniku (4.7) zostały zawarte w komponencie `vehicle_analyzer`.

Komponent **calibrator** jest wywoływany z głównej aplikacji w celu semiautomatycznej kalibracji sceny. Na kadrze niezawierającym samochodów, przechwyconym z dostarczonej sekwencji wideo, użytkownik zaznacza obszar zainteresowania klikając w odpowiednie miejsca obrazu. Moduł ma zaimplementowane trzy metody wyliczenia współczynnika dystansu. Pierwsza metoda wymaga wybrania odcinka pomiarowego i wprowadzenia jego długości w metrach. Druga opcja, po wybraniu odcinka, korzysta z dodatkowego modułu **map\_connector** do pobrania odległości z mapy online. Trzecia metoda automatycznie wylicza okres występowania



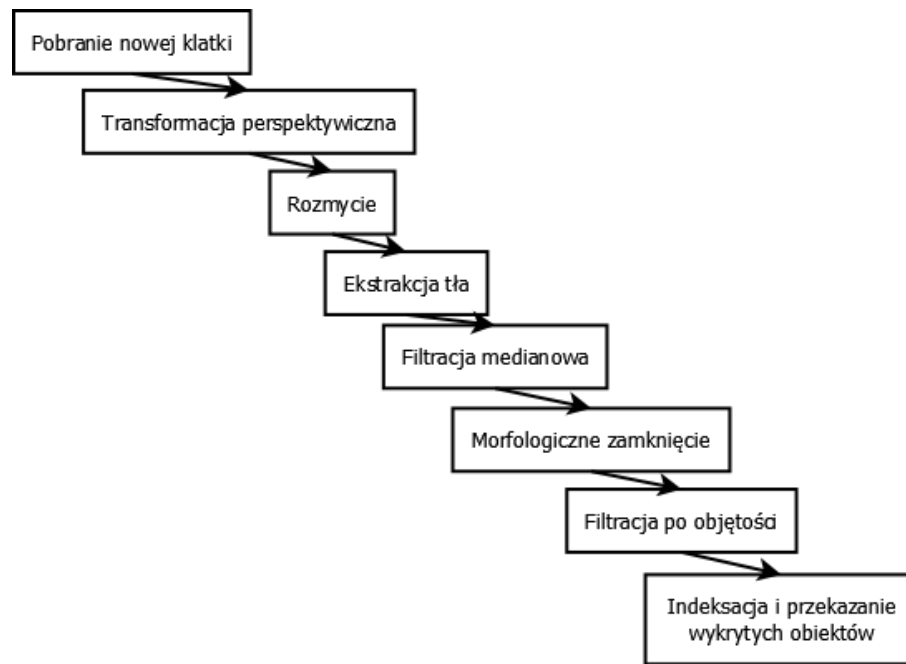
Rysunek 5.1: Wizualizacja architektury aplikacji.

pasów przerywanych widocznych na jezdni (w pikselach) i pobiera od użytkownika długość okresu w metrach. Wybór powyższych metod odbywa się w trakcie działania aplikacji wprowadzając wartość liczbową na ekranie kalibracyjnym.

Moduł **map\_connector** jest opcjonalny i zostaje użyty tylko w sytuacji, gdy użytkownik wybiera drugą metodę obliczania współczynnika odległości (podrozdział 4.4.2). Jego zadaniem jest wystartowanie WebSocket serwera i uruchomienie przeglądarki internetowej z przygotowaną stroną, w której ładuje się mapa Google w trybie satellite. Użytkownik ma możliwość wybrania na mapie dwóch punktów, pomiędzy którymi zostanie zmierzona odległość. Serwer oczekuje na wiadomości wysłane z przeglądarki zawierające długość odcinka. Po wybraniu zakończenia kalibracji w panelu webowym serwer zostaje zatrzymany, a pobrana odległość przekazana do modułu **calibrator**.

Moduł **image\_processor** odpowiada za cyfrowe przetworzenie każdej klatki sekwencji pomiarowej. Do przeprowadzenia obliczeń korzysta ze współrzędnych ROI otrzymanych w procesie kalibracji. Schemat działania przedstawiono na rysunku 5.2. Efektem końcowym jest lista obiektów reprezentujących poruszające się pojazdy. Zostaje ona przekazana do głównej aplikacji do dalszej analizy. W definicje każdego obiektu wchodzi: otoczka wypukła, jej centroid oraz wyliczone momenty geometryczne.

Komponent **vehicle\_analyzer** jest odpowiedzialny za zinterpretowanie dostępnych informacji i wyliczenie prędkości pojazdów. Jego dokładny algorytm przedstawiony jest na diagramie 4.16. Posiada dwie zaimplementowane metody do śledzenia pojazdów (podrozdziały 4.6.2 i 4.6.3). Wybór pomiędzy nimi odbywa się na etapie kalibracji. Implementacyjne rozwiązanie opiera się na wzorcu projektowym fabryka. W zależności od wybranej metody odszukiwania pojazdów, fabryka obiektów tworzy instancje `OpticalFlowVehicle` lub `CentroidTracedVehicle`,



Rysunek 5.2: Operacje wykonywane w module image\_processor.

które implementują minimalny interfejs klasy Vehicle. Rozwiązanie to sprzyja ewentualnemu rozszerzeniu aplikacji o nowe metody wykrywania pojazdów.

### 5.3. Ograniczenia aplikacji

Ograniczenia aplikacji są zbiorem wytycznych, które musi spełniać scena, aby wykonany pomiar był możliwy i dawał poprawne wyniki. Część z nich została już wspomniana na przestrzeni niniejszej pracy, wszystkie zostają zebrane i podsumowane w poniższym podrozdziale. Poniżej zamieszczono przykłady dobrych (rysunek 5.3) i złych (rysunek 5.4) ujęć w kontekście stworzonego systemu.

Lista ograniczeń:

– Kamera:

- rozdzielczość nie mniejsza niż 0.3Mpx
- umieszczona powyżej jezdni
- kąt nachylenia pomiędzy 20-45°

– Jezdnia:

- maksymalne nachylenia drogi do osi Y w układzie współrzędnych obrazu jest równe 30°



Rysunek 5.3: Przykład dobrego doboru sceny.

- droga pozbawiona jest zakrętów na odcinku przynajmniej 50 metrów od pozycji kamery
- powierzchnia jezdni jest płaska i nachylona pod stałym kątem
- Metoda kalibracji dystansu z użyciem map online:
  - znajomość współrzędnych geograficznych lokalizacji kamery
  - występowanie punktów charakterystycznych w obrębie kadru
- Metoda kalibracji dystansu na podstawie linii przerywanej:
  - dobra widoczność pasów
  - znajomość okresu występowania odcinków

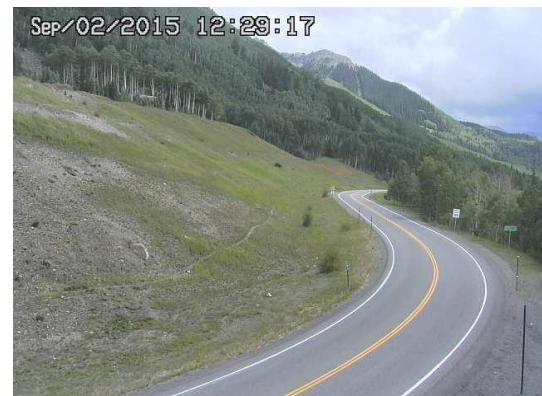
## 5.4. Konfiguracja testowa

Konfiguracje zapewniające płynne działanie aplikacji:

- Notebook - Procesor Intel Core i5, GeForce GT 550M, 8GB RAM
- Desktop - Procesor Intel Core i5, GeForce GTX 650TI, 8GB RAM



(a) Zbyt duże nachylenie jezdni do osi Y



(b) Zakręty na obszarze pomiarowym



(c) Zastłaniające obiekty



(d) Zbyt niskie położenie kamery

Rysunek 5.4: Przykład nieprawidłowego doboru sceny.



## 6. Testy

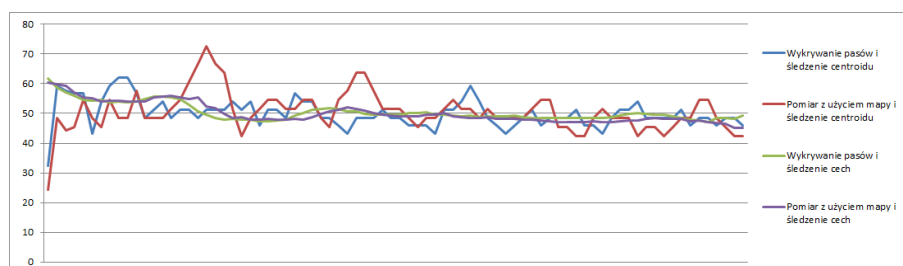
Nieodzownym etapem pracy badawczej jest walidacja osiągniętych wyników. Poniższy rozdział opisuje testy stworzonego systemu.

### 6.1. Integracja

Wstępem przed właściwą analizą zastosowanych algorytmów była weryfikacja działania aplikacji pod koniec tworzenia rozdziału 4. W celu porównania wyników stworzonego rozwiązania z rzeczywistą prędkością przejeżdżających pojazdów przeprowadzono serie pomiarów w trakcie trwania szczytu ruchu ulicznego. Na każdej z sekwencji wideo, prócz przypadkowych pojazdów, uchwycony został samochód z zamontowanym urządzeniem GPS dla uzyskania pomiaru kontrolnego.

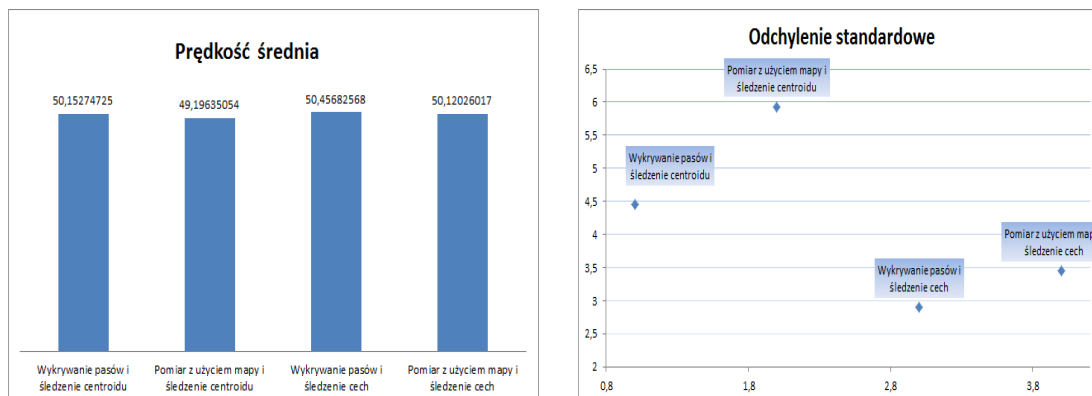
Na wszystkich z 8 prób pomiarowych algorytm wykazywał dokładność rzędu co najmniej 96% przy poprawnym skalibrowaniu sceny.

### 6.2. Porównanie metod śledzenia obiektów



Rysunek 6.1: Porównanie prędkości chwilowej.

Pierwszy przypadek testowy składa się z pomiarów prędkości pojedynczego pojazdu na 10 sekwencjach wideo. Prędkość rzeczywistą, podobnie jak w podrozdziale 6.1, zmierzono za pomocą urządzenia GPS zamontowanego w kontrolowanym samochodzie. Dla każdego zbioru danych wykonano pomiary w następujących kombinacjach metod:



Rysunek 6.2: Porównanie metody śledzenia obiektów

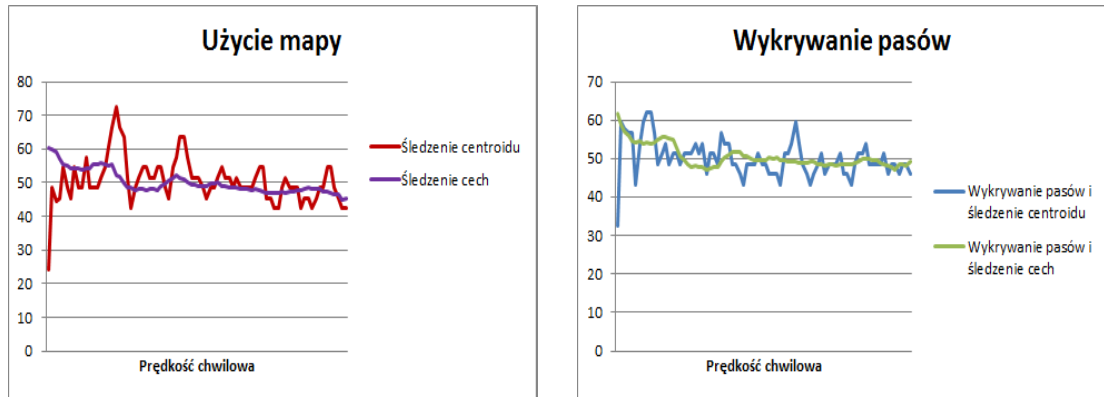
- pomiar dystansu: wykorzystanie linii przerywanej,  
identyfikacja: śledzenie centroidu
- pomiar dystansu: użycie mapy,  
identyfikacja: śledzenie centroidu
- pomiar dystansu: wykorzystanie linii przerywanej,  
identyfikacja: śledzenie cech
- pomiar dystansu: użycie mapy,  
identyfikacja: śledzenie cech

We wszystkich 10 próbach zaobserwowano podobne trendy wyników, dlatego w analizie skupiono się na pojedynczym reprezentacyjnym modelu (rysunek 6.1). Wykres przedstawia wartość prędkości chwilowej wyliczonej przez system na podstawie 5 ostatnich zarejestrowanych klatek.

Pojazd według pomiaru konwencjonalną metodą porusza się ze zmniejszającą się prędkością chwilową ( $V_{\Delta}$ ) od 53km/h do 48km/h i prędkością średnią ( $V$ ) z całego odcinka równą 50km/h.

Na rysunku 6.2a zaprezentowano wartości uzyskanej przez system prędkości średniej. Wszystkie metody mieszczą się w 2% granicy błędu, a dokładność trzech z nich sięga powyżej 99%.

Na podstawie powyższego rezultatu nie da się ocenić jakości funkcji śledzenia pojazdów, dlatego dalszej analizie poddano przebieg wartości chwilowych. Na podstawie odchylenia standardowego (rysunek 6.2b) widać, że zbiory uzyskanych danych różnią się od siebie, a metody wykorzystujące do namierzenia samochodu centroid cechują się większą rozbieżnością wyników  $V_{\Delta}$ .



Rysunek 6.3: Porównanie metody śledzenia obiektów

W celu uniezależnienia wpływu metody estymacji współczynnika dystansu na ocenę koncepcji z podrozdziału 4.4 pogrupowano wyniki w dwa klastry - rysunki 6.3a i 6.3b. Na obu - niezależnie - widać, że uzyskane wyniki chwilowe różnią się drastycznie pomiędzy wykorzystaniem śledzenia narożników, a śledzeniem środka masy wykrytego obiektu. Prędkość chwilowa uzyskana pierwszą metodą jest dużo bardziej zbliżona do rzeczywistej. Wyniki  $V_{\Delta}$  metody drugiej mocno oscylują od wyników otrzymanych na urządzeniu kontrolnym.

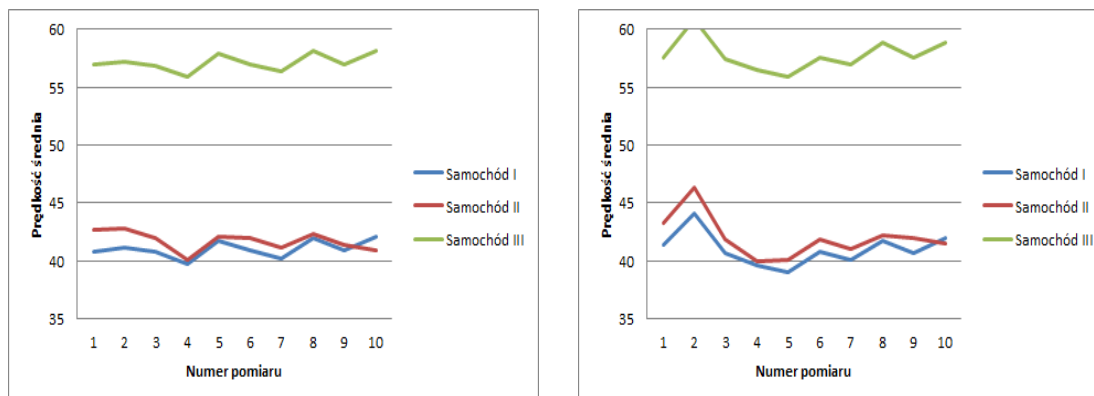
Pomimo, że śledzenie środka masy pozwala uzyskać wynik prędkości średniej o dostatecznej dokładności, co pokazano na rysunku 6.2a, metoda ta sprawdza się gorzej od śledzenia konkretnych elementów pojazdu. Wynika to z faktu, że wykryty na etapie detekcji ruchu zarys samochodu w każdej klatce zmienia swoje wymiary. Generuje to zmianę położenia centroidu i przekłada się na widoczne zaburzenia prędkości średniej. Im dłuższy jest odcinek pomiarowy, tym zaburzenia lokalne są mniej widoczne, ale nie zmienia to faktu, że występują. Proces ten zwiększa swój wpływ szczególnie na granicach ROI, gdzie przez transformacje perspektywy modyfikacje kształtów pojazdów są największe.

Metoda korzystająca z algorytmu przepływu i śledzenia konkretnych narożników nie jest obciążona tym problemem. Zysk z jej zastosowania jest szczególnie widoczny, gdy odcinek pomiarowy jest krótki.

### 6.3. Porównanie metod ewaluacji współczynnika dystansu

Drugi przypadek testowy ma za zadanie ocenę jakości metod dostarczania długości potrzebnej do obliczenia współczynnika dystansu. Na podstawie wniosków z podrozdziału 6.2 do wykonania pomiarów zostało użyte śledzenie pojazdów za pomocą koncepcji Optical Flow.

Podmiotem badań jest grupa trzech kontrolowanych pojazdów uchwyconych na jednej sekwencji wideo. W celu weryfikacji konsekwencji wyników pomiarowych algorytm został uru-



(a) Prędkość uzyskana z wykorzystaniem linii przerywanej

(b) Prędkość uzyskana z wykorzystaniem mapy

Rysunek 6.4: Porównanie metod wyliczania współczynnika dystansu.

chomiony 20 razy. Po 10 razy każdą metodą kalibracyjną. Jednym z założeń testu jest wybieranie innego ROI i obserwacja wpływu wielkości obszaru pomiaru na wynik.

Na wykresie 6.4a widać rezultaty uzyskane kalibrując aplikację z wykorzystaniem linii przerywanej. W zależności od doboru regionu działania algorytmu, współczynnik błędu się zwiększał, ale mimo wszystko był mniejszy niż w metodzie prezentowanej na wykresie 6.4b. Na uwagę zasługuje drugi pomiar przeprowadzony metodą wykorzystującą pobieranie dystansu za pomocą połączenia z interfejsem mapy online. Na podstawie faktu, że błąd tej samej skali wystąpił dla wszystkich kontrolowanych pojazdów można wywnioskować, że powodem problemu była kalibracja.

Metodą dającą bardziej wiarygodne i powtarzalne wyniki jest metoda automatycznie dobierająca współczynnik dystansu na podstawie okresowości pasów. Jej alternatywa wymagająca od użytkownika zaznaczenia odpowiednich markerów na obrazie i mapie obciążona jest dodatkowym błędem dokładności mapy i zaznaczenia odpowiedniego miejsca. Szczególnie dodawanie bardziej odległego znacznika jest utrudnione, ponieważ na polu widzenia kamery, nawet kilka pikseli w dalszym kadrze może reprezentować spore odległości w rzeczywistości.

W pierwszej kolejności powinna być stosowana metoda opisana w podrozdziale 4.4.3. Koncepcja wykorzystująca mapę jest dobrą alternatywą, gdy scena na której chcemy przeprowadzić pomiar nie spełnia wymagań do kalibracji sposobem automatycznym.

## 6.4. Wyniki testów

Przedstawione testy pokazały, że udało się stworzyć system wizyjny umożliwiający pomiar prędkości przejeżdżających pojazdów na podstawie sekwencji wideo. Porównano metody, któ-

rymi dysponuje zaproponowane rozwiązanie i wyłoniono 2 najlepsze konfiguracje, są to odpowiednio:

- 1) tryb automatycznej kalibracji dystansu z użyciem pasów widocznych na jezdni i śledzenia pojazdów za pomocą przepływu optycznego
- 2) kalibracja z wykorzystaniem mapy i śledzenia za pomocą przepływu optycznego w wypadku, gdy scena nie spełnia wymagań punktu pierwszego.



## **7. Podsumowanie**

### **7.1. Ocena realizacji celu**

W toku prac udało się stworzyć system spełniający wymagania funkcjonalne. System efektywnie mierzy prędkość przejeżdżających pojazdów. Zachowuje przy tym dostateczną dokładność wyników i akceptowalną skuteczność identyfikowania obiektów. Kalibracja została zoptymalizowana do minimum, a sam pomiar nie wymaga specjalnego przygotowania sceny.

### **7.2. Wnioski**

Szereg zastosowań pomiarów prędkości sięga począwszy od monitoringu ruchu ulicznego po nowatorskie rozwiązania autonomicznych samochodów. Koncepcja ta jest w bardzo dużej mierze opracowana i rozwiązana - o czym świadczy obecność komercyjnych systemów. Nic nie stoi jednak na przeszkodzie, aby je dalej rozwijać, dodawać nowe funkcjonalności oraz zwiększać wiarygodność i precyzję wyników. Temat pomiaru prędkości pojazdów na podstawie sekwencji wideo jest wiecznie żywy i każde możliwe udoskonalenie jest niewątpliwym plusem.

Coraz większą rolę w życiu ludzi (a także i miast) odgrywają inteligentne systemy asystencyjne. Kalibracja systemów sygnalizacji świetlnej, przewidywanie tendencji ruchu i występowania korków, identyfikacja zagrożeń - to wszystko systemy korzystające z informacji o prędkości pojazdów, które już teraz pomagają zwiększyć komfort i bezpieczeństwo użytkowników ruchu drogowego. Siłą rozwiązania opartego na analizie wideo w porównaniu do innych metod pomiarów jest jego niska cena - zaledwie kamera, łatwość instalacji i olbrzymie istniejące możliwości infrastrukturalne. Pomimo dużej sprawności takich systemów i dostępności rozwiązań, ich zastosowanie to zaledwie mały odsetek możliwości. W rękach departamentów zarządzających infrastrukturą drogową leży potencjał gotowy do wykorzystania.



## Bibliografia

- [1] R. Tadeusiewicz & P. Korohoda. *Komputerowa Analiza i przetwarzanie obrazów*. Wydawnictwo Fundacji Postępu Telekomunikacji, Krakow, 1997.
- [2] M. Gorgoń. *Układy FPGA w rekonfigurowalnych systemach wizyjnych czasu rzeczywistego*. Akademicka Oficyna Wydawnicza EXIT, Krakow, 2013.
- [3] M. Stencel. Wybrane zagadnienia pomiarów parametrów ruchu drogowego za pomocą wiropływowych, indukcyjnych czujników pętlowych, 2012.
- [4] Shan Du, M. Ibrahim, M. Shehata, and W. Badawy. Automatic license plate recognition (alpr): A state-of-the-art review. *Circuits and Systems for Video Technology, IEEE Transactions on*, 23(2):311–325, Feb 2013.
- [5] D.J. Dailey, F.W. Cathey, and S. Pumrin. An algorithm to estimate mean traffic speed using uncalibrated cameras. *Intelligent Transportation Systems, IEEE Transactions on*, 2000.
- [6] F.W. Cathey and D.J. Dailey. A novel technique to dynamically measure vehicle speed using uncalibrated roadway cameras. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 777–782, June 2005.
- [7] Sylvia Gil Milanese, Ruggero Milanese, and Thierry Pun. Comparing features for target tracking in traffic scenes. *Pattern Recognition*, 29(8):1285–1296, 1996. ID: unige:47445.
- [8] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik. A real-time computer vision system for measuring traffic parameters. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 495–501, Jun 1997.
- [9] Sedat Doğan, Mahir Serhan Temiz, and Sıtkı Külür. Real time speed estimation of moving vehicles from side view images from an uncalibrated video camera. *Sensors*, 10(5):4805, 2010.

- [10] H. Fradi and J. Dugelay. Robust foreground segmentation using improved gaussian mixture model and optical flow. In *Informatics, Electronics Vision (ICIEV), 2012 International Conference on*, pages 248–253, May 2012.
- [11] G.R. Bradski. Real time face and object tracking as a component of a perceptual user interface. In *Applications of Computer Vision, 1998. WACV '98. Proceedings., Fourth IEEE Workshop on*, pages 214–219, Oct 1998.
- [12] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981.