
Piotr Dalka

**Metody algorytmicznej analizy obrazu
wizyjnego do zastosowań w
monitorowaniu ruchu drogowego**

Rozprawa doktorska

Promotor:

prof. dr hab. inż. Andrzej Czyżewski
Wydział Elektroniki, Telekomunikacji
i Informatyki
Politechnika Gdańska

Gdańsk, 2014

Podziękowania

Autor pragnie wyrazić słowa podziękowania Promotorowi, Panu prof. dr. hab. inż. Andrzejowi Czyżewskiemu, za wszystkie sugestie i za pracę włożoną w konsultowanie rozprawy oraz za stworzenie warunków sprzyjających jej powstawaniu. Podziękowania należą się również partnerom projektów europejskich 7. Programu UE: „Indect” i „Addpriv” za owocną współpracę, która ułatwiła realizację praktycznych aspektów rozprawy. Szczególne podziękowania kieruję także do wszystkich pracowników Katedry Systemów Multimedialnych PG, za kreatywną atmosferę w miejscu pracy i za cenne rady w trakcie przygotowywania rozprawy.

Spis treści:

1. WPROWADZENIE	1
2. MONITOROWANIE RUCHU DROGOWEGO	7
2.1 HISTORIA ROZWOJU MONITOROWANIA RUCHU DROGOWEGO	7
2.2 WSPÓLczesne METODY ANALIZY RUCHU DROGOWEGO	8
3. WIZYJNA ANALIZA RUCHU DROGOWEGO	12
3.1 WYKRYWANIE OBIEKTÓW W STRUMIENIU WIZYJNYM	13
3.1.1 <i>Detekcja punktów</i>	13
3.1.2 <i>Segmentacja obrazu</i>	15
3.1.3 <i>Klasyfikatory uczące się z nadzorem</i>	17
3.1.4 <i>Przepływ optyczny</i>	18
3.1.5 <i>Odejmowanie tła</i>	18
3.1.6 <i>Podsumowanie</i>	22
3.2 ŚLEDZENIE OBIEKTÓW RUCHOMYCH	23
3.2.1 <i>Śledzenie kształtów geometrycznych</i>	24
3.2.2 <i>Śledzenie jądra</i>	32
3.2.3 <i>Śledzenie sylwetki</i>	37
3.2.4 <i>Podsumowanie</i>	40
3.2.5 <i>Śledzenie wielokamerowe</i>	42
3.3 KLASYFIKACJA OBIEKTÓW W NAGRANIACH WIZYJNYCH	43
3.3.1 <i>Klasyfikacja typu obiektu</i>	43
3.3.2 <i>Określanie rodzaju pojazdu</i>	45
3.4 WIZUALNA DETEKCCJA ZDARZEŃ W RUCHU DROGOWYM	53
3.4.1 <i>Wyznaczanie statystyk ruchu</i>	54
3.4.2 <i>Detekcja zdarzeń</i>	55
4. OPRACOWANIE ALGORYTMU WYKRYWANIA I ŚLEDZENIA OBRAZU RUCHOMYCH OBIEKTÓW	59
4.1 DETEKCCJA OBIEKTÓW RUCHOMYCH	59
4.1.1 <i>Modelowanie tła</i>	60
4.1.2 <i>Detekcja i usuwanie cienia</i>	66
4.1.3 <i>Przetwarzanie morfologiczne</i>	68
4.2 ŚLEDZENIE RUCHOMYCH OBIEKTÓW	70
4.2.1 <i>Zastosowanie filtrów Kalmana</i>	70
4.2.2 <i>Ustalanie powiązania między ruchomymi obiektami a wynikami odejmowania tła</i>	73
4.3 WALIDACJA DZIAŁANIA ALGORYTMÓW DETEKCCJI I ŚLEDZENIA OBIEKTÓW	82
4.3.1 <i>Detekcja obiektów</i>	82
4.3.2 <i>Śledzenie obiektów</i>	89
4.4 ŚLEDZENIE WIELOKAMEROWE	95
4.4.1 <i>Filtracja czasowo-przestrzenna</i>	96
4.4.2 <i>Parametry obrazu obiektów</i>	98
4.4.3 <i>Reidentyfikacja obiektu</i>	101
4.4.4 <i>Wyniki eksperymentów</i>	105
5. OPRACOWANIE ALGORYTMU KLASYFIKACJI RODZAJU POJAZDU	111
5.1 WYKRYWANIE POJAZDU	112
5.1.1 <i>Algorytm klasyfikacji obiektów</i>	112
5.1.2 <i>Walidacja poprawności działania</i>	116
5.2 DESKRYPTORY POJAZDÓW	118

5.2.1	<i>Parametry kształtu</i>	118
5.2.2	<i>Parametry obrazu</i>	120
5.3	OPIS ALGORYTMU KLASYFIKACJI	124
5.3.1	<i>Wykorzystywane klasyfikatory</i>	125
5.3.2	<i>Dobór wektorów uczących i testujących</i>	128
5.4	EKSPERYMENTY I ICH WYNIKI	129
5.4.1	<i>Charakterystyka zbioru danych</i>	129
5.4.2	<i>Dobór wektora cech i klasyfikatora</i>	132
5.4.3	<i>Wyniki klasyfikacji</i>	134
6.	EKSPERYMENTY Z ZAKRESU AUTOMATYCZNEJ ANALIZY RUCHU DROGOWEGO	140
6.1	WYKRYWANIE ZDARZEŃ NA PARKINGU	140
6.1.1	<i>Detekcja parkującego pojazdu</i>	141
6.1.2	<i>Detekcja pozostałych zdarzeń</i>	145
6.1.3	<i>Eksperymenty i wyniki detekcji zdarzeń parkingowych</i>	146
6.2	WYKRYWANIE ZDARZEŃ W RUCHU DROGOWYM	152
6.2.1	<i>Detekcja obiektów wraz z ich kierunkiem ruchu</i>	152
6.2.2	<i>Wykrywanie nieprawidłowej zmiany pasa ruchu</i>	154
6.2.3	<i>Eksperymenty i wyniki detekcji zmiany pasa</i>	160
7.	OPROGRAMOWANIE DO PRZETWARZANIA STRUMIENI WIZYJNYCH	163
7.1	PLATFORMA PROGRAMISTYCZNA	163
7.2	MODUŁY ANALIZY OBRAZU	165
7.3	WYDAJNOŚĆ DZIAŁANIA	167
7.3.1	<i>Prędkość przetwarzania i wykorzystanie procesora</i>	169
7.3.2	<i>Równoczesna analiza wielu strumieni wizyjnych</i>	171
7.3.3	<i>Przepustowość podsystemu pamięci i jej użycie</i>	171
7.3.4	<i>Analiza wydajności poszczególnych modułów przetwarzania obrazu</i>	173
8.	WNIOSKI I PODSUMOWANIE	177
9.	BIBLIOGRAFIA	180
10.	ZAŁĄCZNIKI	196
10.1	INFRASTRUKTURA BADAWCZA – ROZPROSZONY SYSTEM MONITORINGU	196
10.1.1	<i>Architektura systemu</i>	196
10.1.2	<i>Charakterystyka elementów składowych systemu</i>	197
10.1.3	<i>Warstwa komunikacyjna</i>	199
10.1.4	<i>Transmisja danych multimedialnych</i>	200
10.1.5	<i>Implementacja</i>	201
10.1.6	<i>Budowa Stacji Monitorującej</i>	202
10.2	PŁYTA DVD	204
10.3	SPIS RYSUNKÓW	205
10.4	SPIS TABEL	209

Lista najważniejszych oznaczeń stosowanych w rozprawie:

I	przetwarzana ramka obrazu lub jej podobszar
M	maska binarna zawierająca oznaczenie pikseli należących do ruchomych obiektów
$I_k(x, y)$	wartość piksela o współrzędnych (x, y) w k -tej ramce obrazu
\mathbf{x}_k	wektor stanu filtru Kalmana w czasie k
$\hat{\mathbf{x}}_k^-$	estymata <i>a priori</i> wektora stanu filtru Kalmana w czasie k
$\hat{\mathbf{x}}_k$	estymata <i>a posteriori</i> wektora stanu filtru Kalmana w czasie k
\mathbf{z}_k	wektor pomiarowy w czasie k
A	macierz przejścia filtru Kalmana
H	macierz wyjściowa filtru Kalmana
$p(\mathbf{x}_t)$	prawdopodobieństwo przyjęcia przez określony piksel wartości \mathbf{x} w czasie t
η	funkcja gęstości prawdopodobieństwa rozkładu normalnego
w_t^i	waga i -tego rozkładu gaussowskiego w czasie t
μ_t^i	wartość średnia i -tego rozkładu gaussowskiego w czasie t
$\sigma_t^{i^2}$	odchylenie standardowe i -tego rozkładu gaussowskiego w czasie t
Σ_t^i	macierz kowariancji i -tego rozkładu gaussowskiego w czasie t
K	liczba wykorzystywanych rozkładów gaussowskich
α	współczynnik determinujący szybkość adaptacji modelu tła
x^b, y^b, w^b, h^b	położenie (x, y) oraz wysokość i szerokość regionu w obrazie dane w pikselach
x, y, w, h	położenie (x, y) oraz wysokość i szerokość trackera w dane pikselach
dx, dy, dw, dh	zmiana położenia (dx, dy) trackera oraz jego wysokości i szerokości w stosunku do poprzedniej ramki obrazu
C	macierz współwystępowania
\mathbf{C}_N	symetryczna, unormowana macierz współwystępowania
$C_{N_{i,j}}$	element macierzy \mathbf{C}_N o współrzędnych (i, j)
P, R, A	miary precyzji, czułości i dokładności służące do oceny działania algorytmów

Wykaz najważniejszych akronimów:

- ANN (ang. *Artificial Neural Network*) – sztuczna sieć neuronowa
- CCTV (ang. *Closed-Circuit Television*) – telewizja przemysłowa
- GMM (ang. *Gaussian Mixtures Model*) – model mieszanin gaussowskich
- HMM (ang. *Hidden Markov Models*) – ukryte modele Markowa
- ICA (ang. *Independent Component Analysis*) – analizy składowych niezależnych
- ITU-R (ang. *International Telecommunication Union – Radiocommunication Sector*) – Międzynarodowa Unia Telekomunikacyjna – Sektor Radiokomunikacji
- JPDAF (ang. *Joint Probability Data Association Filter*) – filtr łącznego prawdopodobieństwa asocjacji danych
- KLT (ang. *Kanade–Lucas–Tomasi*) – detektor punktów charakterystycznych w obrazie
- LDA (ang. *Linear Discriminant Analysis*) – liniowa analiza dyskryminacyjna
- MHT (ang. *Multiple Hypothesis Tracking*) – metoda wielokrotnych hipotez
- NAT (ang. *Network Address Translator*) – translator adresu, element infrastruktury sieciowej
- PCA (ang. *Principal Component Analysis*) – analiza składowych głównych
- PTZ (ang. *Pan-Tilt-Zoom*) – kamera obrotowa
- RBF (ang. *Radial Basis Function*) – radialna funkcja bazowa
- ROI (ang. *Region of Interest*) – obszar zainteresowania w obrazie
- RPROP (ang. *Resilient Back-Propagation*) – algorytm wstecznej propagacji błędu służący do treningu ANN
- SIFT (ang. *Scale Invariant Feature Transform*) – deskrytor lokalnych parametrów obrazu
- SURF (ang. *Speded Up Robust Features*) – deskrytor lokalnych parametrów obrazu
- SVM (ang. *Support Vector Machines*) – maszyna wektorów nośnych
- XMPP (ang. *Extensible Messaging and Presence Protocol*) – protokół komunikacyjny służący do natychmiastowej transmisji wiadomości w sieci

1. Wprowadzenie

Monitoring wizyjny, telewizja przemysłowa (CCTV, ang. *Closed-Circuit Television*) lub telewizyjne systemy dozoru to praktycznie synonimy określające „zespoły środków technicznych i programowych przeznaczonych do obserwacji, wykrywania, rejestrowania oraz sygnalizowania warunków wskazujących na istnienie niebezpieczeństwa powstania szkód lub zagrożeń osób i mienia” [129]. Pierwsze zastosowanie kamer do obserwacji najbliższego otoczenia miało miejsce w czasie drugiej wojny światowej, kiedy to nazistowskie Niemcy użyły ich do nadzorowania startów rakiet V-2 [194]. Praktyczna ekspansja systemów monitoringu nastąpiła na świecie jednak dopiero w latach 70-tych ubiegłego wieku. Zapoczątkowała ją instalacja dwóch kamer na Trafalgar Square w Londynie w 1969 roku [80]. Od tego czasu liczba instalacji monitoringowych wzrosła niepomiaralnie. Szacuje się, że w samej Wielkiej Brytanii znajduje się obecnie ponad 1,85 miliona kamer, z których tylko ok. 150 tysięcy to kamery nadzorowane przez miasto, służby bezpieczeństwa i transport publiczny, a przeciętny mieszkaniec Londynu jest widziany dziennie przez 70 różnych kamer [89]. Na tym polu Polska osiąga znacznie gorsze metryki. Największą liczbą kamer monitoringu miejskiego dysponuje Warszawa (414 sztuk). Tylko 7 polskich miast posiada 100 i więcej kamer, oprócz stolicy są to: Poznań (356), Giżycko (144), Gdynia (136), Gdańsk (117), Białystok (103) i Gliwice (100). Systemy monitoringu wizyjnego mają wszystkie miasta wojewódzkie oraz 86% miast powiatowych [272].

Obecnie praktycznie nie sposób znaleźć na świecie publiczne miejsce, gdzie człowiek nie byłby pod stałą obserwacją kamer. Są one wykorzystywane przede wszystkim do zwalczania przestępczości, monitorowania ruchu drogowego, zwiększenia bezpieczeństwa w środkach transportu oraz nadzorowania procesów przemysłowych.

Aby instalacje kamerowe były skuteczne, obrazy z nich muszą być na bieżąco obserwowane i interpretowane. Niestety, ze względu na ogromną liczbę kamer nie jest praktycznie możliwe, żeby obrazy z każdej z nich były non-stop oglądane przez operatorów systemów monitoringu. W rezultacie większość strumieni jest jedynie rejestrowana, aby po fakcie móc prześledzić przebieg zdarzenia. Słabym ogniwem tego procesu jest także człowiek, który zwykle jest odpowiedzialny za jednoczesną obserwację strumieni z kilku do kilkunastu kamer (rys. 1.1). Naturalne w takim przypadku jest znużenie i zmęczenie, które prowadzi do spadku czujności. Badania wykazały, że obserwując tylko obraz z pojedynczej kamery, przeciętny operator już po upływie 20

minut na tyle traci swoją koncentrację, że nie dostrzega 90% aktywności w scenie [80]. Stąd konieczność wdrażania inteligentnych systemów monitoringu, które na drodze algorytmicznej analizy obrazu są w stanie wykrywać sytuacje warte uwagi i przedstawiać je operatorowi do oceny. W ten sposób poprawia się poziom bezpieczeństwa monitorowanych obszarów (aktywna analiza obrazu ze wszystkich kamer, brak wpływu zmęczenia na osiągnięte wyniki).



Rys. 1.1 Centrum monitoringu wizyjnego w Gdyni [269]

W ostatnim czasie na świecie realizowanych jest wiele projektów związanych z zagadnieniem inteligentnego monitoringu. W trzech z nich autor rozprawy wraz z macierzystą Katedrą brał aktywny udział. Pierwszym jest projekt badawczy Ministerstwa Nauki i Szkolnictwa Wyższego pod nazwą „Multimedialny system wspomagający identyfikację i zwalczanie przestępczości (w tym przemocy w szkołach) oraz terroryzmu (R00-O0005/3)” [46]. Projekt ten, realizowany w latach 2007-2010, był wspierany przez Polską Platformę Bezpieczeństwa Wewnętrznego. Istota projektu polegała na opracowaniu narzędzi teleinformatycznych, które uzupełnią istniejące i stale rozwijane systemy monitoringu wizyjnego i akustycznego. Uzupełnienie to polegało na wprowadzeniu funkcji automatycznego rozumienia dźwięków i obrazów, tak aby systemy komputerowe w sposób automatyczny mogły wykrywać potencjalne zagrożenia i informować o nich służby odpowiedzialne za bezpieczeństwo i porządek publiczny.

Dwa kolejne projekty mają zasięg międzynarodowy i są realizowane w ramach 7. programu ramowego Unii Europejskiej. Projekt INDECT (ang. *Intelligent Information System Supporting Observation, Searching and Detection for Security of Citizens in Urban Environment*) realizowany jest od 2009 roku z udziałem Policji polskiej, północno-irlandzkiej oraz czołowych uczelni technicznych i firm z Polski oraz Europy. Ma na celu wykorzystanie innowacyjnych algorytmów i metod z zakresu informatyki do wykrywania i walki z terroryzmem oraz innymi działaniami przestępczymi. Projekt ten zakłada powstanie zestawu rozwiązań do inteligentnej obserwacji i automatycznego wykrywania podejrzanych zachowań lub przemocy w środowisku miejskim [270].

Z kolei projekt ADDPRIV (ang. *Automatic Data Relevancy Discrimination for a Privacy-sensitive Video Surveillance*) realizowany jest od 2011 roku we współpracy z partnerami z Hiszpanii, Włoch, Wielkiej Brytanii i Irlandii. Jego celem jest poprawa bezpieczeństwa publicznego przez zapewnienie prawa prywatności osobom objętym monitoringiem. Zadanie to realizowane jest poprzez rozszerzenie funkcjonalności istniejących systemów o automatyczną dyskryminację nieistotnych nagrań z zastosowaniem automatycznych metod analizy obrazu. To z kolei, pozwala ograniczyć przestrzeń dyskową konieczną do przechowywania rejestracji, co tym samym prowadzi do zwiększenia prywatności osób znajdujących się na obszarze objętym monitoringiem [52][265].

Przedmiotem badań rozprawy jest opracowanie nowych i rozwinięcie istniejących metod przetwarzania i analizy obrazu z kamer wizyjnych systemów monitoringu mających na celu wykrywanie wybranych zdarzeń w ruchu ulicznym. Oznacza to konieczność opracowania, zbadania, implementacji i dostosowania do pracy w określonych warunkach wszystkich niezbędnych do tego celu algorytmów. Obejmują one detekcję i śledzenie obiektów w polu widzenia kamer, reidentyfikację tego samego obiektu w obrazie z różnych kamer, klasyfikację typu obiektu (w tym określenie typu pojazdu). Niezbędne jest także przetestowanie wdrożonych rozwiązań, czego dokonano na przykładzie wykrywania zdarzeń na parkingu oraz na ruchliwym skrzyżowaniu. W celu osiągnięcia satysfakcjonujących rezultatów, w procesie tym niezbędne jest zastosowanie algorytmów uczących się. Od strony praktycznej opracowane rozwiązania determinuje założona konieczność ich działania w czasie quasi-rzeczywistym. Jednocześnie, wymaganie to w dużym stopniu wpłynęło na potrzebę poszukiwania oryginalnych rozwiązań algorytmicznych, umożliwiających spełnienie tego warunku, tzn. aby detekcja

predefiniowanych zdarzeń mogła odbywać się w trakcie ich zachodzenia, a nie jedynie na podstawie analizy nagrań po ich wystąpieniu.

Przedstawiony wyżej zakres prac wiąże się z postawionymi w rozprawie tezami:

- 1. Opracowane i przebadane eksperymentalnie cechy dystynktywne obrazu wizyjnego pojazdów umożliwiają skuteczną identyfikację i klasyfikację pojazdów dokonywaną z użyciem algorytmów uczących się.**
- 2. Opracowane algorytmy detekcji i śledzenia ruchomych obiektów dostosowane do działania w warunkach rzeczywistych, pozwalają na szybkie i skuteczne z praktycznego punktu widzenia wykrywanie kluczowych zdarzeń związanych z parkowaniem pojazdów oraz istotnych zdarzeń zagrażających bezpieczeństwu ruchu drogowego.**

Układ treści rozprawy jest następujący: przegląd stosowanych technologii monitoringu ruchu drogowego, z uwzględnieniem metod innych, niż przetwarzanie obrazu, przedstawiono w rozdziale 2. Następny rozdział ukierunkowany jest na wizyjną analizę ruchu drogowego i zawiera przegląd rozwiązań stosowanych do wykrywania obiektów w obrazie, śledzenia ich w polu widzenia kamery, reidentyfikacji obiektów przemieszczających się pomiędzy polami widzenia kamer, klasyfikacji obiektów ruchomych pomiędzy pojazdy i osoby, rozpoznawania typu pojazdu oraz wykrywania zróżnicowanych zdarzeń w ruchu drogowym. W oparciu o zestawione elementy wiedzy, w rozdziale 4 przedstawiono szczegółowo zaimplementowane algorytmy detekcji i śledzenia obiektów ruchomych oraz ich reidentyfikacji oraz zbadano eksperymentalnie skuteczność ich działania. Kolejny rozdział prezentuje opracowane algorytmy klasyfikacji wszystkich obiektów pomiędzy trzy klasy: „pojazdy”, „osoby” i „inne” oraz pojazdów na: „samochody osobowe”, „furgonetki/busy” oraz „pojazdy ciężarowe”. Działanie algorytmów klasyfikacji zostało zweryfikowane z użyciem nagrań testowych. Rozdział 6 prezentuje eksperymenty związane z algorytmami automatycznego monitorowania ruchu drogowego, które dotyczą dwóch scenariuszy: detekcji zdarzeń na parkingu, takich jak zaparkowanie na określonym miejscu wraz z identyfikacją tego miejsca, wjazdu i wyjazdu z parkingu oraz wejścia i wyjścia osób z budynku oraz wykrywania przypadków zabronionej zmiany pasa ruchu przez pojazdy na skrzyżowaniu dróg o dużym natężeniu ruchu. Skuteczność detekcji zdarzeń została zweryfikowana za pomocą nagrań z rzeczywistych instalacji monitoringu umiejscowionych w odpowiednich lokalizacjach. W ramach realizacji rozprawy powstała uniwersalna platforma sprzętowo-programistyczna służąca do realizacji szeroko pojętych zadań przetwarzania obrazu. Platforma ta integruje

w sobie wszystkie niezbędne funkcjonalności związane z zagadnieniem monitorowania ruchu drogowego. Opracowane uniwersalne narzędzie do rozproszonego, wielowątkowego i modułowego przetwarzania obrazu w czasie rzeczywistym opisano w rozdziale 7. W rozdziale 8 przedstawiono wnioski i zamieszczono dowody postawionych tez.

Autor rozprawy wielokrotnie publikował wraz ze współpracującymi osobami prace dotyczące tematu rozprawy w czasopismach naukowych i w postaci referatów konferencyjnych. Tab. 1.1 prezentuje 23 spośród najważniejszych publikacji z podziałem na zagadnienia, których dotyczą. Opublikowane prace, w ich niektórych fragmentach, mogą stanowić uzupełnienie treści prezentowanych w rozprawie.

Tab. 1.1 Publikacje autora w tematyce rozprawy

<i>Detekcja ruchomych obiektów</i>	
[55]	Dalka, P., "Detection and segmentation of moving vehicles and trains using Gaussian mixtures, shadow detection and morphological processing", <i>Machine Graphics & Vision International Journal</i> , vol. 15, no. 3/4, pp. 339–348, 2006.
[50]	Czyżewski, A. and Dalka, P., "Visual Traffic Noise Monitoring in Urban Areas", <i>International Journal of Multimedia and Ubiquitous Engineering</i> , vol. 2, no. 3, pp. 91–102, 2007.
[49]	Czyżewski, A. and Dalka, P., "Teleinformatyczna akwizycja i przetwarzanie danych dla potrzeb diagnostyki środowiska aglomeracji miejskich", <i>Inteligentne wydobywanie informacji w celach diagnostycznych</i> , vol. 21, no. 5, Z. Kowalczyk and B. Wiszniewski (Ed.). Pomorskie Wydawnictwo Naukowo-Techniczne, pp. 241–256, 2007.
[231]	Szwoch, G. and Dalka, P., "Layered background modeling for automatic detection of unattended objects in camera images", <i>Proc. International Workshop on Image Analysis for Multimedia Interactive Services</i> , pp. 1–4, 2011.
<i>Śledzenie ruchomych obiektów</i>	
[48]	Czyżewski, A. and Dalka, P., "Moving object detection and tracking for the purpose of multimodal surveillance system in urban areas", <i>New Directions in Intelligent Interactive</i> , vol. 142, pp. 75–84, 2008.
[47]	Czyżewski, A. and Dalka, P., "Examining Kalman Filters Applied to Tracking Objects in Motion", <i>Proc. Ninth International Workshop on Image Analysis for Multimedia Interactive Services</i> , pp. 175–178, 2008.
[46]	Czyżewski, A. et al., "Multimedialny system wspomagający identyfikację i zwalczanie przestępczości oraz terroryzmu", <i>Praktyczne elementy zwalczania przestępczości zorganizowanej i terroryzmu: Nowoczesne technologie i praca operacyjna</i> , L. Paprzycki and Z. Rau, (Ed.). Wolter Kluwers Polska, pp. 211–227, 2009.
[227]	Szczodrak, M., Dalka, P., and Czyżewski, A., "Moving object tracking algorithm evaluation in autonomous surveillance system", <i>Proc. IEEE International Conference on Multimedia Communications, Services and Security</i> , pp. 219–223, 2010.
[64]	Dalka, P., Szwoch, G., Szczuko, P., and Czyżewski, A., "Video Content Analysis in the Urban Area Telemonitoring System", <i>Multimedia Services in Intelligent Environments</i> , vol. 3, pp. 241–261, Dec. 2010.
[235]	Szwoch, G., Dalka, P., and Czyżewski, A., "Resolving conflicts in object tracking for automatic detection of events in video", <i>Elektronika: konstrukcje, technologie, zastosowania</i> , vol. 52, no. 1, pp. 52–54, 2011.
<i>Śledzenie wielokamerowe i reidentyfikacja obiektów</i>	
[56]	Dalka, P., "Multi-camera Vehicle Tracking Using Local Image Features and Neural Networks", <i>Proc. Multimedia Communications, Services and Security</i> , pp. 58–67, 2012.
[52]	D'Arminio, P., Buendia-Iglesias, R., Cichowski, J., Dalka, P., Ellwart, D., and Orwell, J., "Technologies for granting balance between security and privacy in video-surveillance", <i>Proc. European Intelligence and Security Informatics Conference</i> , pp. 278–283, 2012.
<i>Klasyfikacja typu obiektu</i>	
[234]	Szwoch, G., Dalka, P., and Czyżewski, A., "Objects classification based on their physical sizes for detection of events in camera images", <i>Proc. Signal Processing Algorithms, Architectures, Arrangements, and Applications (SPA)</i> , pp. 15–20, 2008.
[230]	Szwoch, G. and Dalka, P., "Identification of regions of interest in video for a traffic monitoring system", <i>Proc. 1st International Conference on Information Technology</i> , pp. 1–4, 2008.
[233]	Szwoch, G., Dalka, P., and Czyżewski, A., "Estimation of object size in the calibrated camera image", <i>Elektronika: konstrukcje, technologie, zastosowania</i> , vol. 50, no. 3, pp. 10–14, 2009.
[60]	Dalka, P. and Czyżewski, A., "Vehicle classification based on soft computing algorithms", <i>Rough Sets and Current Trends in Computing</i> , pp. 70–79, 2010.
<i>Detekcja zdarzeń</i>	
[61]	Dalka, P., Ellwart, D., and Szwoch, G., "Camera orientation-independent parking events detection", <i>Proc. 12th Int. Workshop on Image Analysis for Mult. Interact. Services (WIAMIS)</i> , 2011.
[63]	Dalka, P., Szwoch, G., and Ciarkowski, A., "Distributed Framework for Visual Event Detection in Parking Lot Area", <i>Multimedia Communications, Services and Security</i> , vol. 149, A. Dziech and A. Czyżewski, (Ed.). Berlin Heidelberg: Springer, pp. 37–45, 2011.
[57]	Dalka P., Ciarkowski A., Szczuko P., Czyżewski A.: "Open standards-based communication system for distributed intelligent surveillance solution", <i>Przegląd Telekomunikacyjny</i> , no. 8–9, pp. 777–785, 2013.
<i>Wykorzystanie kamer obrotowych w systemie monitoringu</i>	
[58]	Dalka, P., Ciarkowski, A., Szczuko, P., Szwoch, G., and Czyżewski, A., "Surveillance Camera Tracking of Geo positioned Objects", <i>New Directions in Intelligent Interactive Multimedia Systems and Services - 2</i> , vol. 226, E. Damiani, J. Jeong, R. J. Howlett, and L. C. Jain, (Ed.). Berlin Heidelberg: Springer, pp. 21–30, 2009.
[229]	Szwoch, G. and Dalka, P., "Automatic detection of abandoned luggage employing a dual camera system". <i>Proc. IEEE International Conference on Multimedia Communications, Services and Security</i> , pp. 56–61, 2010.
[232]	Szwoch, G., Dalka, P., Ciarkowski, A., Szczuko, P., and Czyżewski, A., "Visual object tracking system employing fixed and PTZ cameras", <i>Intelligent Decision Technologies</i> , vol. 5, no. 2, pp. 177–188, 2011.
[236]	Szwoch G., Dalka P., Czyżewski A.: "Spatial Calibration of a Dual PTZ-Fixed Camera System for Tracking Moving Objects in Video", <i>Journal of Imaging Science and Technology</i> , vol. 57, no. 2, pp. 1–10, Mar. 2013.

2. Monitorowanie ruchu drogowego

Wraz z rozwojem motoryzacji i zwiększającą się liczbą pojazdów na drogach powstała konieczność efektywnego zarządzania ruchem drogowym w celu umożliwienia optymalizacji przepływu pojazdów oraz poprawy bezpieczeństwa uczestników ruchu. Rozwiązaniem praktycznym nie jest w tym przypadku jedynie budowa nowych dróg, gdyż przyrost ich liczby jest znacznie wolniejszy, niż wzrost liczby pojazdów na drogach (dla przykładu, w latach 1987-1997 liczba kilometrów przejechanych przez pojazdy po autostradach w Stanach Zjednoczonych wzrosła o 33% podczas, gdy długość dróg o niecałe 2% [66]). Dlatego istotnym i praktycznie realizowanym sposobem poprawy sytuacji na drogach jest optymalne zarządzanie ruchem drogowym. Jest ono możliwe tylko w przypadku wykonywania na bieżąco pomiarów natężenia ruchu i dokonywania detekcji zdarzeń w nim występujących.

2.1 Historia rozwoju monitorowania ruchu drogowego

Potrzeba automatycznego zbierania danych o ruchu drogowym narodziła się w latach 20 ubiegłego wieku, wraz z wdrożeniem automatycznej sygnalizacji świetlnej, która zastąpiła sygnalizatory sterowane ręcznie przez operatorów [143]. Pierwsze rozwiązanie temu służące zostało opracowane przez Charlesa Adlera Jr. i wdrożone w 1928 roku w Baltimore, w Stanach Zjednoczonych. Składało się z mikrofonu umieszczonego na słupie obok drogi, aktywowanego dźwiękiem klaksonu, do którego włączania w wyznaczonym miejscu zobowiązany był kierowca każdego przejeżdżającego samochodu. Umożliwiało to półautomatyczne sterowanie sygnalizacją świetlną. W 1931 roku Adler zaproponował inne rozwiązanie wykorzystujące puste, metalowe pudełka montowane w nawierzchni drogi. Ich zadaniem było wzmacnianie dźwięku przejeżdżających po nich kół samochodowych i transmitowanie go do mikrofonów, co zapewniało w pełni automatyczną detekcję pojazdów.

W tym samym czasie Henry A. Haugh opracował czujnik nacisku montowany na powierzchni drogi. Składał się on z dwóch metalowych płytek, które były ze sobą zwierane pod wpływem kół przejeżdżającego pojazdu, zamykając tym samym obwód elektryczny. Rozwiązanie to cieszyło się większym powodzeniem, niż czujnik dźwiękowy i było w praktyce stosowane przez ponad 30 lat do sterowania sygnalizacją świetlną.

Nie było to jednak rozwiązanie optymalne pod kątem ekonomicznym, nawet mimo podjętych prób zastąpienia czujnika kontaktowego czujnikiem elektro-pneumatycznym.

Często zdarzało się, że pługi odśnieżające drogę jednocześnie usuwały z niej pedał naciskowy powodując konieczność kosztownych napraw. Ponadto każde renowacje powierzchni drogi wymagały ponownej instalacji czujnika. Wszystko to oznaczało, że potencjalnie najłatwiej mierzalna cecha pojazdu, jaką jest jego waga, nie mogła być efektywnie wykorzystywana. W związku z tym podejmowano prace nad wykorzystaniem innych parametrów pojazdów, taki jak:

- dźwięk (czujniki akustyczne)
- nieprzezroczystość (kamery wizyjne i czujniki działające w podczerwieni)
- geomagnetyzm (czujniki magnetyczne)
- odbicie transmitowanej energii (radar laserowy w podczerwieni, sensory ultradźwiękowe, radar w paśmie mikrofalowym)
- indukcja elektromagnetyczna (pętle indukcyjne)
- wibracje (czujniki tryboelektryczne, sejsmiczne i bezwładnościowe)

2.2 Współczesne metody analizy ruchu drogowego

Nie wszystkie z powyższych koncepcji znalazły komercyjne zastosowanie. Obecnie największą popularnością cieszą się detektory w postaci pętli indukcyjnej. Do różnych celów związanych z zarządzaniem ruchem drogowym stosowane są również czujniki magnetyczne, radary w paśmie podczerwonym i mikrofalowym, czujniki ultradźwiękowe, akustyczne i pasywne w podczerwieni oraz kamery wizyjne [143].

Typowym zastosowaniem wymienionych czujników jest sterowanie sygnalizacją świetlną, detekcja zdarzeń drogowych i gromadzenie informacji ilościowych o ruchu drogowym z podziałem na kategorie pojazdów. Omawiane urządzenia montowane są w nawierzchni drogi, nad nią lub obok niej.

Pętla indukcyjna jest najpowszechniej stosowanym czujnikiem w systemach zarządzania ruchem drogowym. Jest montowana wewnątrz nawierzchni drogi i wykrywa przewodzący, metalowy obiekt poprzez indukowanie w nim prądu, czego następstwem jest zmniejszanie indukcyjności pętli. Pętle indukcyjne pozwalają na wykrywanie poruszającego się pojazdu, jak i samej jego obecności.

Pod powierzchnią drogi montowane są również czujniki magnetyczne. Są to pasywne urządzenia wykrywające obecność przedmiotów ferromagnetycznych poprzez zmiany, jakie wywołują one w ziemskim polu magnetycznym. Istnieją dwa typy tych czujników stosowane w monitoringu ruchu drogowego. Dwuosiowy magnetometr transduktorowy

jest wrażliwy na zmiany składowej poziomej i pionowej ziemskiego pola magnetycznego pod wpływem przejeżdżającego pojazdu, podczas gdy magnetometr indukcyjny wykrywa zakłócenia linii strumienia ziemskiego pola magnetycznego wywołane poruszającym się samochodem. Czujniki magnetyczne dostarczają jedynie informacji o ruchu pojazdu, a nie o jego obecności lub stopniu zajętości drogi, wobec czego ich praktyczne zastosowanie jest ograniczone.

Radar mikrofalowy jest urządzeniem aktywnym emitującym promieniowanie elektromagnetyczne o częstotliwościach od 10 do 30 GHz w kierunku obserwowanego odcinka drogi. Pod wpływem przejeżdżającego pojazdu część energii wiązki promieniowania odbijana jest z powrotem i wykrywana przez odbiornik umieszczony zwykle w pobliżu nadajnika. Radary mikrofalowe mogą być montowane ponad powierzchnią drogi lub z jej boku. W zależności od rodzaju użytej modulacji mogą wykrywać jedynie poruszające się pojazdy, przekazując tym samym informację o ich liczbie i prędkości lub być w stanie rozpoznać także samochody stojące, co pozwala uzyskać dodatkowo dane na temat zajętości drogi oraz długości pojazdów.

Czujniki pracujące w paśmie podczerwonym mogą być aktywne lub pasywne. Rozwiązania aktywne oświetlają monitorowany odcinek drogi promieniowaniem o niskiej mocy w paśmie dalekiej podczerwieni (o długości fali rzędu 850 μm) za pomocą diod laserowych lub (rzadziej) diod LED. Część energii promieniowania jest odbijana z powrotem lub rozpraszana, co jest wykrywane przez odbiornik. Czujniki pasywne nie dysponują własnym źródłem promieniowania i bazują na detekcji energii emitowanej bezpośrednio przez pojazdy i powierzchnię drogi oraz energii emitowanej poprzez atmosferę i odbitej od elementów wchodzących w skład monitorowanej sceny. Czujniki na podczerwień mogą być montowane nad drogą lub obok niej i dostarczają informacji o prędkości, liczbie i typie pojazdów [206].

Przetworniki ultradźwiękowe emitują falę akustyczną o częstotliwości od 25 do 50 kHz w formie impulsów. Pozwala to mierzyć odległość od powierzchni drogi i od pojazdów poprzez pomiar energii odbitej z powrotem do detektora. Fakt zmierzenia odległości innej, niż do powierzchni drogi interpretowany jest jako obecność pojazdu. Sensory ultradźwiękowe pozwalają wykryć obecność pojazdów, policzyć je i ocenić zajętość drogi. Odmiana przetworników ultradźwiękowych, emitująca ciągłą falę akustyczną i bazująca na zjawisku Dopplera jest w stanie dodatkowo mierzyć prędkość pojazdów.

Czujniki akustyczne wykrywają obecność, poruszanie się pojazdów i określają ich prędkość na podstawie pomiaru poziomu dźwięku emitowanego przez wiele źródeł w samym pojeździe, a także generowanego w wyniku interakcji opon pojazdu z nawierzchnią drogi [51][226]. Istnieją dwa rodzaje czujników, służące do pomiarów tylko na jednym pasie ruchu lub jednocześnie na wielu pasach. Oba wykorzystują do tego celu dwuwymiarową matrycę mikrofonów.

Kamery wizyjne trafiły do systemów zarządzania ruchem drogowym jako narzędzia do akwizycji i transmisji obrazu do centrów monitoringu w celu ich interpretacji przez operatorów systemów nadzoru wizyjnego. Prace związane z automatyczną analizą obrazu zostały zainicjowane w latach 70-tych ubiegłego wieku [115]. Obecnie dane wizyjne są przetwarzane przez systemy komputerowe w celu automatycznego określania parametrów ruchu drogowego (liczba pojazdów, ich prędkości oraz zajętość drogi dla każdego typu pojazdu niezależnie) oraz w celu detekcji zdarzeń, takich jak zawracanie lub jazda pod prąd. Systemy wykorzystujące kamery wizyjne są w stanie zastąpić wiele pętli indukcyjnych i zredukować koszty utrzymania systemu. Przegląd technik wizualnej detekcji zdarzeń w ruchu drogowym przedstawiono w rozdziale 3.4. Całkowicie odmiennym rozwiązaniem wykorzystującym kamery wizyjne jest monitorowanie ruchu drogowego z pokładu bezzałogowych statków latających (dronów) [133].

W tab. 2.1 przedstawiono porównanie technologii monitorowania ruchu drogowego ze względu na rodzaj mierzonych parametrów ruchu, uwzględnianie wielu stref pomiaru (pasów ruchu), zapotrzebowanie na pasmo do transmisji danych oraz koszty. Większość typów czujników mierzy liczbę pojazdów, wykrywa ich obecność i dostarcza informacji o zajętości drogi. Niektóre czujniki (ultradźwiękowe oraz część wykorzystujących podczerwień) nie mierzą prędkości pojazdów. Nie wszystkie radary mikrofalowe są w stanie wykrywać nieruchome lub wolno jadące samochody. Czujniki magnetyczne, pasywne czujniki pracujące w podczerwieni i sensory akustyczne nie dostarczają danych pozwalających na klasyfikację typu pojazdu.

Wysoka skuteczność działania czujników montowanych w nawierzchni drogi (pętla indukcyjne, czujniki magnetyczne) wiąże się, częściowo, z ich niewielką odległością od pojazdów, co zwiększa niezawodność pomiaru. Są przez to niewrażliwe na niekorzystne warunki pogodowe. Ich wadą jest instalacja wewnątrz drogi, co wymaga ingerencji w jej budowę w procesie montażu. Czujniki umieszczane nad drogą lub obok niej zwykle dostarczają więcej danych i obsługują wiele stref pomiaru. Największymi możliwościami charakteryzują się systemy bazujące na przetwarzaniu obrazu. Ich dodatkową zaletą jest

fakt, że często nie wymagają dodatkowych inwestycji w infrastrukturę, gdyż mogą wykorzystywać istniejące kamery monitoringu wizyjnego.

Wiele z technologii wymienianych w niniejszym rozdziale jest stale rozwijanych w celu zwiększenia zakresu mierzonych parametrów i ich dokładności. W szczególności, prace prowadzone są nad pętlami indukcyjnymi, radarami mikrofalowymi oraz systemami przetwarzania obrazu. Tworzone są również rozwiązania, które integrują wiele różnych technologii, takich jak analiza obrazu i pętla indukcyjna [114] lub ultradźwiękowy radar dopplerowski i kamera wizyjna [200].

Tab. 2.1 Porównanie technologii monitorowania ruchu drogowego [143]

Technologia	Mierzone parametry ruchu drogowego					Wiele stref pomiaru (pasów ruchu)	Pasma do transmisji wyników pomiarów	Koszt zakupu
	Liczba pojazdów	Obecność pojazdu	Prędkość	Zajętość drogi	Klasyfikacja typu pojazdu			
Pętla indukcyjna	+	+	+ ^a	+	+	-	12	1
Magnetometr tansdukturowy	+	+	+ ^a	+	-	-	1	2
Magnetometr indukcyjny	+	+	+ ^a	+	-	-	1	12
Radar mikrofalowy	+	+ ^b	+	+ ^b	+ ^b	+ ^b	2	12
Czujnik aktywny w podczerwieni	+	+	+ ^c	+	+	+	12	23
Czujnik pasywny w podczerwieni	+	+	+ ^c	+	-	-	12	12
Czujniki ultradźwiękowe	+	+	-	+	-	-	1	12
Czujniki akustyczne	+	+	+	+	-	+ ^c	12	2
Przetwarzanie obrazu	+	+	+	+	+	+	123	23

+ – tak

- – nie

1 – niskie, 2 – średnie, 3 – wysokie, 12 – niskie do średnich, 23 – średnie do wysokich, 123 – niskie do wysokich

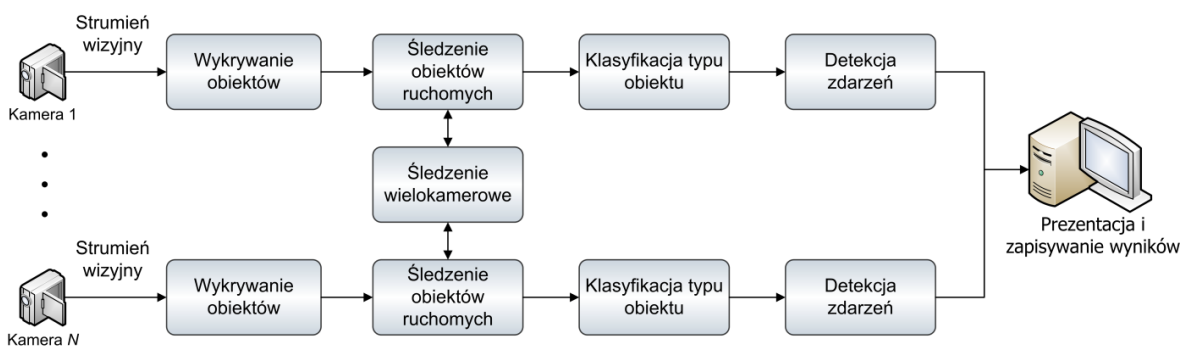
a – prędkość może być mierzona za pomocą dwóch sensorów o znanej odległości między nimi

b – w zależności od charakterystyki wykorzystywanego promieniowania

c – w zależności od typu czujnika

3. Wizyjna analiza ruchu drogowego

Analiza strumieni wizyjnych ze stacjonarnych kamer monitoringu pod kątem wykrywania zdarzeń w ruchu drogowym składa się z kilku etapów, pokazanych na rys. 3.1. W pierwszym z nich w każdej ramce obrazu z każdej kamery wykrywane są wszystkie obiekty ruchome. Obiekty te są następnie śledzone przez cały czas pobytu w polu widzenia kamery w celu umożliwienia analizy ich zachowania. Jednocześnie następuje proces klasyfikacji typu obiektu w celu uwzględnienia tej informacji w ostatnim module systemu, jakim jest detekcja zdarzeń. Przetawiony proces zachodzi równoległe dla wszystkich analizowanych strumieni wizyjnych obejmujących swym polem widzenia monitorowany teren. Wyniki analizy są zachowywane i prezentowane na ekranie.



Rys. 3.1 Elementy składowe wizyjnej analizy obrazu

Oddzielnym tematem, nie poruszonym szerzej w rozprawie, jest wykorzystanie kamer obrotowych (PTZ, ang. *Pan-Tilt-Zoom*) w systemie monitoringu. Autor rozprawy zaproponował w tym zakresie rozwiązanie, w którym kamery obrotowe są automatycznie nakierowywane na obiekty wybrane przez operatora w polu widzenia kamer stacjonarnych i mogą podążać swoim polem widzenia za poruszającym się obiektem [229]. W tym celu opracowano nowatorski algorytm wykonywania półautomatycznej, szybkiej i zdalnej kalibracji pary kamer: stacjonarnej i obrotowej tak, aby kamera ruchoma mogła być sterowana z dużą dokładnością [236]. Bazując na wynikach detekcji i śledzenia obiektów w polu widzenia wszystkich kamer stacjonarnych monitorujących określony obszar, kamery obrotowe samodzielnie śledzą, zatem, poruszające się obiekty. Możliwe jest również śledzenie obiektów, których pozycja określana się za pomocą odbiornika GPS [58][232].

3.1 Wykrywanie obiektów w strumieniu wizyjnym

Podstawowym elementem każdego systemu automatycznej analizy obrazu pod kątem wykrywania zdarzeń jest moduł detekcji obiektów ruchomych. Jego zadaniem jest znalezienie, zwykle niezależnie w każdej ramce obrazu, lokalizacji wszystkich obiektów pierwszoplanowych. Wyniki działania tego algorytmu stanowią dane wejściowe dla algorytmu śledzenia ruchomych obiektów w kolejnych ramkach strumienia wizyjnego.

Istnieje wiele różnych podejść do zagadnienia detekcji obiektów w obrazie. Można je podzielić na 5 grup, które zostaną omówione w kolejnych paragrafach tego rozdziału:

- Detekcja punktów charakterystycznych
- Segmentacja obrazu
- Klasyfikatory uczące się z nadzorem
- Przepływ optyczny
- Odejmowanie tła

3.1.1 Detekcja punktów

Detektory punktów służą do znalezienia w obrazie punktów charakterystycznych, które wyróżniają się miarami tekstury w porównaniu z otoczeniem. Zakłada się, że takie punkty znajdują się w obrębie obiektów pierwszoplanowych. Charakterystyka tych punktów i ich lokalizacja w sąsiednich ramkach obrazu jest następnie wykorzystana do śledzenia obiektów ruchomych. Dane te mogą również służyć zagadnieniom klasyfikacji i identyfikacji obiektów.

Pożądaną cechą punktu charakterystycznego jest jego niezależność od zmian warunków oświetleniowych oraz orientacji (pozy) wykrywanych obiektów. Do najczęściej wykorzystywanych detektorów punktów charakterystycznych należą operator Moraveca (ang. *Moravec's Interest Operator*), detektor Harrisa, detektor KLT (ang. *Kanade-Lucas-Tomasi*) i detektor SIFT (ang. *Scale Invariant Feature Transform*). Przykładowe wyniki detekcji punktów charakterystycznych wybranymi metodami pokazano na rys. 3.2. Przegląd i porównanie różnorodnych detektorów punktów charakterystycznych można znaleźć w artykule Mikołajczyka i Szmida [173].

W celu znalezienia punktów charakterystycznych, operator Moraveca [176] wyznacza wariancję jasności obrazu w oknach o rozmiarach 4×4 w czterech kierunkach: pionowym, poziomym oraz dla obu przekątnych i przyjmuje najmniejszą z nich, jako reprezentatywną dla całego okna. Punkt obrazu jest uznawany za charakterystyczny, jeśli

znaleziona dla niego minimalna wariancja stanowi maksimum lokalne w sąsiedztwie 12×12 pikseli.

Detektor Harrisa [101] wyznacza położenie punktów charakterystycznych w oparciu o macierz zawierającą pochodne obrazu w kierunkach poziomym i pionowym w najbliższym otoczeniu każdego punktu obrazu:

$$\mathbf{M} = \begin{bmatrix} \sum D_x^2 & \sum D_x D_y \\ \sum D_x D_y & \sum D_y^2 \end{bmatrix} \quad (3.1)$$

gdzie D_x i D_y oznaczają pochodne obrazu określonych kierunkach. Punkt charakterystyczny identyfikowany jest za pomocą maksimum lokalnego miary danej wzorem:

$$R = \text{DET}(\mathbf{M}) - k \cdot \text{TR}(\mathbf{M})^2 \quad (3.2)$$

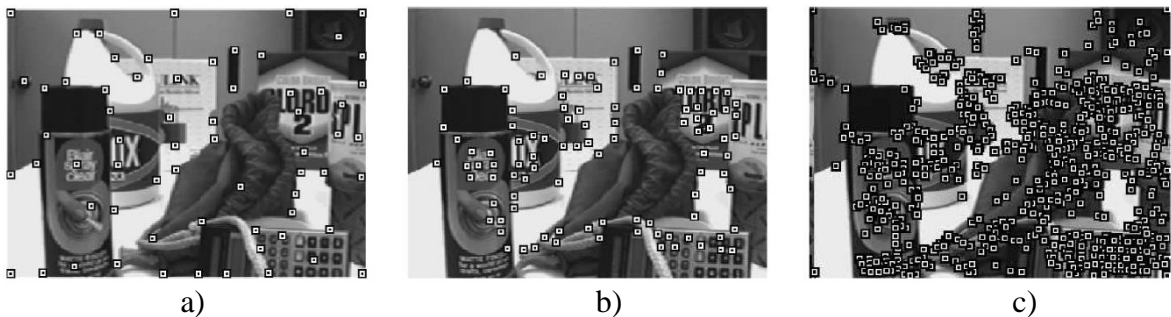
gdzie operatory DET i TR oznaczają odpowiednio wyznacznik i ślad macierzy, a k jest wartością stałą.

Detektor KLT [216] bazuje na tej samej macierzy \mathbf{M} danej wzorem (3.1), lecz jako miarę kwalifikującą dany punkt jako charakterystyczny przyjęto minimalną wartość własną macierzy. W praktyce, jako że oba te detektory, tzn. Harrisa i KLT, używają podobnych miar, znajdowane z ich wykorzystaniem punkty charakterystyczne są bardzo podobnie usytuowane. Jediną istotną różnicę stanowi występujący w detektorze KLT warunek na minimalną odległość przestrzenną sąsiednich punktów charakterystycznych.

W teorii macierz \mathbf{M} jest niezależna od obrotu i przesunięcia, jednak jest wrażliwa na przekształcenia afiniczne. Wady tej pozbawiony jest detektor SIFT [160], którego działanie składa się z 3 kroków. W pierwszym obraz filtrowany jest wielokrotnie jądrem gaussowskim o różnych wartościach odchylenia standardowego, a następnie obliczane są różnice między obrazami wyjściowymi w celu otrzymania obrazów przefiltrowanych pasmowo (DoG, ang. *Difference-of-Gaussians*). Kandydaci na punkty szczególnie znajdowani są wśród wartości minimalnych i maksymalnych w każdym z wyników filtracji. W drugim kroku korygowana jest lokalizacja każdego z kandydatów poprzez interpolację wartości pikseli leżących w jego sąsiedztwie. Ostatni krok prowadzi do eliminacji kandydatów charakteryzujących się niskim kontrastem lub leżących na krawędziach. Detektor SIFT, w porównaniu z dwoma pozostałymi, generuje znacznie

więcej punktów kluczowych ze względu na kumulację wyników uzyskanych dla każdego wyniku filtracji.

Detekcja punktów w procesie wykrywania obiektów ruchomych ma zastosowanie przede wszystkim w przypadkach, gdy obiekty te można modelować za pomocą punktów. Ma to miejsce, gdy obiekty zainteresowania są bardzo małe w stosunku do rozmiarów ramki obrazu lub gdy rozmiary obiektu nie mają znaczenia. Żaden z tych warunków nie jest w ogólności spełniony w przypadku detekcji obiektów w systemach automatycznego monitoringu.



Rys. 3.2 Wyniki detekcji punktów charakterystycznych za pomocą detektorów a) Harrisa, b) KLT i c) SIFT [258]

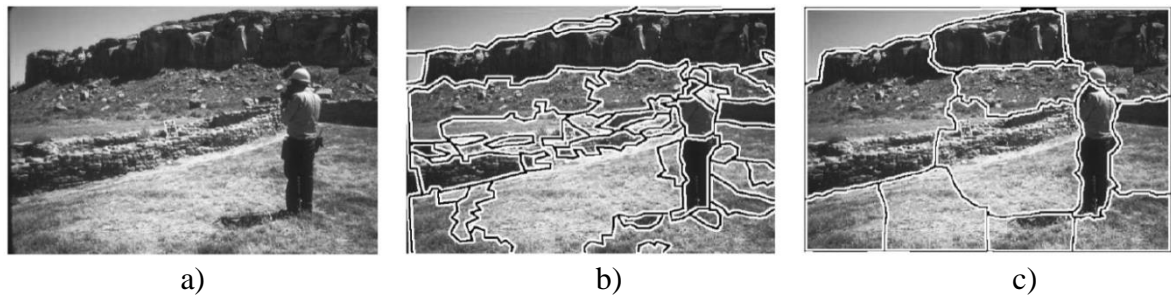
3.1.2 Segmentacja obrazu

Celem segmentacji obrazu jest podział obrazu na obszary, które się od siebie perceptualnie różnią. Wśród algorytmów segmentacji przydatnych w procesie detekcji obiektów można wyróżnić grupowanie Mean-Shift, algorytm Graph-Cuts oraz aktywne kontury.

Grupowanie Mean-Shift (ang. *Mean-Shift Clustering*) [39] polega na znalezieniu klastrów w łącznej przestrzeni uwzględniającej kolor piksela i jego położenie. Algorytm jest inicjalizowany dużą liczbą hipotetycznych, losowo wybranych środków poszczególnych klastrów. Następnie, każdy klaster przesuwany jest w kierunku wartości średniej elementów leżących wewnątrz wielowymiarowej elipsoidy zakotwiczonej w środku klastra. Wektor tego przesunięcia (zwany wektorem *mean-shift*), jest wyznaczany interaktywnie do czasu, gdy jego długość jest bliska zero (położenia klastrów się nie zmieniają). Algorytm ten dopuszcza możliwość łączenia się klastrów. Jednak w celu otrzymania satysfakcjonujących wyników segmentacji wymaga on dokładnego dostrojenia wartości wielu parametrów.

Algorytm Graph-Cuts może być zdefiniowany jako problem podziału grafu, gdzie wierzchołki grafu (piksele obrazu) są dzielone na wiele rozdzielnych podgrafów poprzez rozcinanie ważonych krawędzi grafu. Wagi krawędzi są obliczane zwykle oparciu o miary podobieństwa koloru, jasności lub tekstury poszczególnych węzłów grafu. Łączna waga wszystkich rozciętych krawędzi pomiędzy dwoma podgrafami jest nazywana cięciem (ang. *cut*). Najbardziej typowym sposobem podziału grafu jest podział minimalizujący wartości cięć [256]. Wadą tej metody jest skłonność do generowania zbyt dużej liczby segmentów ze względu na rosnący koszt cięcia w funkcji zwiększającej się liczby krawędzi. W związku z tym powstała idea cięcia znormalizowanego (ang. *Normalized Cut*) [215], w którym rozcinanie krawędzi uzależnione jest nie tylko od sumy wag w cięciu, ale również od stosunku sumy wag wszystkich krawędzi wewnątrz segmentu do sumy wag wszystkich krawędzi w grafie. Takie rozwiązanie jest stosunkowo kosztowne w aspekcie zapotrzebowania na moc obliczeniową i pamięć, ale jednocześnie wymaga ono ręcznego dobrania wartości znacznie mniejszej liczby parametrów, niż grupowanie Mean-Shift. Przykładowe wyniki segmentacji obiema metodami pokazano na rys. 3.3.

W metodzie aktywnych konturów (ang. *Active Contours*) segmentacja obrazu osiągnięta jest poprzez stopniowe przekształcanie zamkniętego konturu w taki sposób, aby ostatecznie przylegał on ściśle do granic wybranego obiektu. Przekształcanie konturu następuje w oparciu o wartości funkcji energii konturu, na którą składają się ograniczenia regulacyjne, związane z koniecznością zachowania ciągłości konturu i znalezieniem konturu najkrótszego, parametry wyglądu obrazu oraz ograniczenia dodatkowe. Wśród parametrów obrazu stosowane są zarówno cechy lokalne, zwykle dotyczące gradientu obrazu wyznaczanego w pobliżu konturu [23][135] oraz cechy globalne, takie jak kolor [197][259][264] i tekstura [187], obliczane wewnątrz i na zewnątrz konturu. Istotnym zagadnieniem rozwiązania jest inicjalizacja konturu. Dla rozwiązań bazujących na gradiencie obrazu zwykle wymaga się aby kontur ten znajdował się na zewnątrz segmentowanego obiektu, a następnie, w trakcie działania algorytmu podlega on kurczeniu. Dla metod wykorzystujących cechy globalne, kontur może być zainicjowany zarówno na zewnątrz obiektu (i wtedy się kurczy) jak i wewnątrz obiektu (wówczas się powiększa, aż wypełni cały obiekt). Jednak wszystkie te rozwiązania wymagają wcześniejszej wiedzy o położeniu obiektów w analizowanym obrazie. Wiedza ta może być gromadzona automatycznie, np. poprzez użycie algorytmów odejmowania tła do inicjalizacji konturów [186].



Rys. 3.3 Wyniki segmentacji obrazu (a) algorytmami *Mean-Shift* (b) i cięcia normalizowanego (c) [258]

Osobnym zagadnieniem jest sposób reprezentacji konturu: jawny, w postaci zbioru punktów lub niejawny, w którym kontur reprezentowany jest przez siatkę przestrzenną, w której wartość bezwzględna elementów siatki oznacza odległość od konturu, a znak elementu determinuje jego lokalizację jako wewnątrz lub na zewnątrz obiektu. Największą zaletą niejawnej reprezentacji konturu jest możliwość dowolnej zmiany topologii poprzez łączenie i rozdzielanie konturów.

3.1.3 Klasyfikatory uczące się z nadzorem

Odmiernym sposobem detekcji obiektów jest zastosowanie algorytmu, który na podstawie różnych obrazów obiektów nauczy się je automatycznie rozpoznawać w zbiorze testowym. Korzystając z przygotowanych wzorców, algorytmy uczące się tworzą funkcję mapującą wejściowe wektory (wartości) uczące na oczekiwane wektory (wartości) wyjściowe. Systemy uczące się przystosowane są do rozwiązywania problemu klasyfikacji, w którym to algorytm aproksymuje zachowanie funkcji poprzez generowanie wyników w postaci albo wartości ciągłych (regresja), albo w postaci dyskretnych etykiet (klasyfikacja). W kontekście detekcji obiektów, wzorce uczące składają się z par cech obiektów i właściwej etykiety. Oba te elementy muszą być określone ręcznie.

Wybór odpowiednich cech obiektów jest kluczowy z punktu widzenia skuteczności działania zastosowanego klasyfikatora, dlatego należy wybierać takie cechy, które pozwalają na jak najlepsze rozróżnienie obiektów różnych klas. Wśród cech wizualnych używanych w procesie detekcji obiektów zastosowanie mają przede wszystkim: kolor [188], krawędzie [15] oraz miary tekstury, takie jak na przykład macierze współwystępowania [98], falki [165] oraz piramida filtrów kierunkowych (ang. *Steerable Pyramid*) [92]. Zastosowanie znajdują również dodatkowe parametry statystyczne (np. wielkość obiektu) lub miary rozkładu wartości parametru, np. wykorzystujące histogram kierunków gradientów (ang. *Histogram of Oriented Gradients*, HOG) [208]. Jako

klasyfikator w systemie detekcji obiektów stosowane są m.in. takie rozwiązania, jak sztuczne sieci neuronowe [199], drzewa decyzyjne [93], maszyny wektorów wspomagających (SVM) [185][208] i kaskada BOOST (ang. *Adaptive Boosting*) [247]. Szczególnie ten ostatni klasyfikator jest powszechnie używany, szczególnie do detekcji twarzy w obrazie [59][247].

3.1.4 Przepływ optyczny

Gęsty przepływ optyczny jest zbiorem translacji (zdefiniowanych jako pole wektorowe), które przekształcają bieżącą ramkę w sekwencji wizyjnej w ramkę kolejną, zgodnie z zależnością:

$$I_{k-1}(x, y) - I_k(x + dx, y + dy) = 0 \quad (3.3)$$

gdzie $I_k(x, y)$ oznacza wartość piksela w miejscu o współrzędnych (x, y) w czasie k .

Metoda ta zakłada niezmiennosc jasności odpowiadających sobie pikseli w sąsiednich ramkach obrazu [105]. Na podstawie wynikowego pola wektorowego można wykryć grupy pikseli poruszające się spójnie (w tym samym kierunku i z tą samą prędkością), które są tożsame z ruchomymi obiektami. Istnieje wiele metod służących do wyznaczania pola wektorowego, wśród których można wymienić metody gradientowe (bazujące na analizie przestrzennych i czasowych pochodnych obrazów), widmowe (wykorzystujące filtrowanie obrazów w dziedzinie częstotliwości) oraz korelacyjne (badające podobieństwo obrazów) [5]. Do popularnych metod obliczania gęstego przepływu optycznego należą te zaproponowane przez Horna Schuncka [105], Lucasa i Kanade'a [161], Blacka i Anandana [13], Szeliskiego i Couglana [228] oraz Farnebacka [79].

Jednak większość metod prowadzących do wyznaczenia pola wektorowego jest nieefektywna obliczeniowo, zatem nie może być stosowana do przetwarzania w czasie rzeczywistym strumieni o wymaganej rozdzielczości bez pomocy dedykowanych rozwiązań sprzętowych.

3.1.5 Odejmowanie tła

Jest to najbardziej popularna kategoria metod służących do detekcji obiektów zainteresowania w poszczególnych ramkach obrazu. W swej najbardziej podstawowej formie polega na porównaniu ze sobą kilku sąsiednich ramek obrazu w celu określenia różniących się obszarów i znalezienia w ten sposób obiektów ruchomych. Jako parametry

obrazu mogą w tym wypadku być używane bezpośrednio wartości pikseli lub przekształcenia wykorzystujące pochodne pierwszego i drugiego rzędu w celu znalezienia konturów obiektów [140].

W ogólności odejmowanie tła polega na tworzeniu modelu tła analizowanej sceny i znajdowaniu różnic pomiędzy modelem, a każdą kolejną ramką obrazu. W wyniku uzyskuje się maskę binarną, w której zaznaczone są piksele różniące się od tła, czyli te należące do ruchomych obiektów.

Najprostszym sposobem uzyskania tła sceny jest uśrednienie w czasie kolejnych ramek obrazu. Niestety, metoda ta wymaga okresu treningu, w trakcie którego w analizowanej scenie obecnych jest niewiele ruchomych obiektów. Ponadto możliwości dynamicznej adaptacji wyznaczonego tła są bardzo ograniczone. Z tego powodu metoda ta nie nadaje się do praktycznych zastosowań w warunkach zmiennego oświetlenia.

Prace związane z określaniem różnic pomiędzy sąsiednimi ramkami obrazu prowadzone są już od późnych lat 70 ubiegłego wieku [119]. Jednak zyskały one na popularności od czasu pojawienia się rozwiązania zaproponowanego przez Wrena [254], w którym każdy piksel modelowany jest za pomocą pojedynczego, trójwymiarowego rozkładu gaussowskiego. Parametry tego modelu (wartość średnie i kowariancje) wyznaczone są na podstawie obserwacji wartości pikseli w określonym przedziale czasu. Następnie, dla każdego piksela określane jest prawdopodobieństwo zgodności koloru piksela z rozkładem i te piksele, które odbiegają od rozkładu, są oznaczane jako pierwszoplanowe. Jednak modelowanie tła sceny za pomocą pojedynczego rozkładu normalnego nie nadaje się do praktycznego zastosowania w warunkach zewnętrznych ze względu na fakt, że wiele różnych barw może występować w tym samym miejscu ze względu na powtarzalny ruch w tle, odbicia itd. [86]. W związku z tym istotnym udoskonaleniem modelowania tła jest użycie wielomodowych modeli statystycznych do opisu koloru piksela. Przykładem takiego rozwiązania jest wykorzystanie sumy ważonych rozkładów gaussowskich do modelowania koloru pikseli tła [217]. W rozwiązaniu tym bieżący kolor piksela jest porównywany z każdym rozkładem, dopóki pasujący rozkład nie zostanie znaleziony. W przypadku powodzenia, wartości średnie i wariancje pasującego rozkładu są uaktualniane, w przeciwnym wypadku do modelu wprowadzany jest nowy rozkład o wartości średniej równej reprezentacji koloru piksela. Piksel uznawany jest za część tła sceny, jeśli pasuje do jednej z dystrybucji uznawanych w danym momencie za reprezentujące tło sceny. Proces ten zilustrowano na rys. 3.4.

Innym podejściem do zagadnienia detekcji obiektów jest uwzględnienie w tym procesie również informacji przestrzennej, zamiast bazowania jedynie na kolorze każdego piksela. Przykładem takiego rozwiązania jest użycie nieparametrycznej estymacji gęstości jądra do modelowania tła sceny dla każdego piksela [77]. Podczas odejmowania tła piksel porównywany jest nie tylko z odpowiadającym mu ze względu na położenie pikselem modelu lecz również z pikselami sąsiednimi. W związku z tym metoda ta jest odporna na niewielkie drgania kamery lub na mały ruch w tle sceny. Z kolei w pracy [242] zaproponowano podejście trójwarstwowe, w którym obok danych na poziomie poszczególnych pikseli analizowane są również informacje na poziomie całej ramki obrazu jak i w odniesieniu do jej podobszarów. Na poziomie piksela zastosowano filtrację Wienera w celu uzyskania probabilistycznej predykcji wartości oczekiwanej koloru tła, na poziomie podobszarów regiony pierwszoplanowe charakteryzujące się jednorodnym kolorem są wypełniane, natomiast informacja o nagłej zmianie wartości pikseli na poziomie całej ramki jest używana do unieważnienia dotychczasowego modelu tła i do przeprowadzenia jego ponownej inicjalizacji.



Rys. 3.4 Przykład działania algorytmu odejmowania tła z wykorzystaniem sumy ważonych rozkładów gaussowskich a) oryginalna ramka obrazu, b) wartości średnie rozkładu o największej wadze, c) wartości średnie następnego w kolejności rozkładu, nie zaliczanego w tym momencie do modelu tła, d) wykryte piksele należące do obiektu ruchomego

Do reprezentowania tła sceny możliwe jest również zastosowanie algorytmu Codebook, w którym każdy piksel obraz modelowany jest za pomocą słów kodowych (ang. *Codewords*) [141]. Pojedyncze słowo jest wektorem opisującym zakres dopuszczalnych wartości koloru i jasności piksela, wraz z informacjami pomocniczymi, takimi jak czas dodania słowa kodowego do modelu, czas jego ostatniej aktualizacji i najdłuższy okres, przez jaki dane słowo pozostawało niedopasowane do analizowanego obrazu. Algorytm wymaga fazy treningu w celu przypisania odpowiednich słów kodowych do poszczególnych pikseli. Dzięki odpowiednim modyfikacjom algorytmu, możliwe jest

zarówno uzyskanie przybliżonych wyników odejmowania tła już w czasie trwania treningu, jak i adaptacja modelu do zmian w scenie [231].

Odmiernym rozwiązaniem zagadnienia odejmowania tła jest reprezentowanie różnorodności jasności pikseli obrazu jako dyskretnych stanów, odpowiadających zdarzeniom występującym w analizowanej scenie (np. stanami takimi są obiekty pierwszoplanowe, tło oraz cień obiektu). Do klasyfikacji niewielkich obszarów obrazu jako należących do każdego z tych trzech stanów można użyć ukrytych modeli Markowa (ang. *Hidden Markov Models*, HMM) [196]. Modele te są również przydatne do detekcji zdarzeń włączenia i wyłączenia świateł w pomieszczeniu na potrzeby odejmowania tła [219].

Zamiast modelowania zmienności każdego piksela obrazu indywidualnie, możliwe jest podejście holistyczne wykorzystujące rozkład według wartości własnych macierzy zawierającej k ramek obrazu, gdzie każdy wiersz zawiera wszystkie wartości pikseli jednej ramki obrazu. W ten sposób tło reprezentowane jest przez $i < k$ wektorów własnych, które całkowicie obejmują różnorodność wartości pikseli tła, co sprawia, że metoda ta jest mniej wrażliwa na zmiany oświetlenia. Obiekty pierwszoplanowe są wykrywane poprzez rzutowanie bieżącej ramki na wyznaczoną przestrzeń własną w oparciu o odnajdywanie różnic między obrazem zrekonstruowanym, a oryginalnym [182]. Ograniczeniem tej metody jest fakt, że wymaga ona statycznego tła sceny. Wynikający stąd problem można rozwiązać za pomocą modelowania obszarów obrazu jako procesu ARMA (ang. *Autoregressive Moving Average*), który pozwala poznać i estymować wzorce ruchu w analizowanej scenie [175][263]. W ten sposób możliwe jest modelowanie zmiennego w czasie tła sceny (np. fale na wodzie, przemieszczające się chmury itd.).

Obecnie większość stosowanych praktycznie metod detekcji obiektów bazuje na algorytmach odejmowania tła ze względu na to, że pozwalają modelować zmienne warunki oświetleniowe i okresowe zmiany w tle sceny, a przez to skutecznie wykrywać obiekty w rozmaitych warunkach. W praktyce, często się zdarza, że wyniki odejmowania tła są niekompletne, tzn. obiekt jest podzielony na kilka mniejszych lub w obiekcie pojawiają się obszary niepewne, ze względu na fakt, że nie ma gwarancji, iż cechy wyglądu obiektu będą się różnić od tła. Największym jednak ograniczeniem tych metod jest fakt, że wymagają one kamery stacjonarnej, gdyż ruch kamery zwykle zaburza model tła. Zostały podjęte pewne próby wykorzystania algorytmów odejmowania tła do detekcji obiektów z kamer mobilnych, poprzez ciągłą regenerację modelu w krótkich, kilkuramkowych oknach czasowych [131] lub poprzez kompensację ruchu z wykorzystaniem mozaiki tła

[116][198], jednak wszystkie te rozwiązania zakładają płaską scenę oraz niewielkie zmiany położenia kamery z ramki na ramkę.

3.1.6 Podsumowanie

W tab. 3.1 umieszczono opracowane przez autora rozprawy porównanie metod detekcji obiektów w strumieniu wizyjnym ze względu na ich przydatność do monitorowania ruchu drogowego. Na podstawie jej zawartości można stwierdzić, że do tego zastosowania nie nadają się metody wykrywania punktów charakterystycznych, gdyż nie pozwalają na uzyskanie informacji o położeniu całego obiektu, a jedynie wybranych punktów leżących na nim. Podobnie nieprzydatne są algorytmy segmentacji obrazu, gdyż nie uwzględniają one w procesie detekcji przesłanki związanej z ruchem obiektów. W rezultacie otrzymuje się lokalizacje regionów należące zarówno do pierwszoplanowych obiektów ruchomych, jak i do stacjonarnych obiektów będących częścią tła sceny.

Tab. 3.1 Porównanie metod detekcji obiektów w obrazie

Metoda	Detekcja położenia całego obiektu	Wykrywane tylko obiekty ruchome	Działanie z kamerami ruchomymi	Odporność na zmiany oświetlenia	Odporność na zmiany w tle sceny	Nie wymaga treningu	Nie wymaga wstępnej informacji o położeniu obiektów
Detekcja punktów	-	-	+	+	-	+	+
Segmentacja obrazu	+	-	+	+	-	+	o
Klasyfikatory uczące się z nadzorem	+	-	+	+	-	-	-
Przepływ optyczny	+	+	-	-	+	+	+
Odejmnowanie tła	+	+	-*	+	+	+	+

+ – tak

- – nie

o – w zależności od algorytmu

* – odporny tylko na niewielkie ruchy (drżania) kamery

Klasyfikatory uczące się z nadzorem nie są uniwersalnymi metodami detekcji obiektów, gdyż wymagają dostarczenia wzorcowych obrazów obiektów przeznaczonych do detekcji. W zadaniu monitorowania ruchu drogowego istnieje praktycznie nieograniczona dowolność w rodzaju i wyglądzie wykrywanych obiektów (m. in. pojazdy i osoby obserwowane pod różnym kątem i na całkowicie losowym tle), przez co nie sposób przygotować optymalny zbiór treningowy. Z kolei metody bazujące na przepływie

optycznym zakładają stałość wyglądu tych samych pikseli z ramki na ramkę, przez co nie są odporne na zmianę warunków oświetleniowych.

Z przedstawionej analizy wynika, że metody bazujące na odejmowaniu tła najlepiej nadają się do detekcji obiektów w procesie monitorowania ruchu drogowego. Ich praktyczny brak odporności na ruch kamery nie jest stanowi problemu, gdyż zakłada się działanie systemu monitoringu w oparciu o kamery stacjonarne.

3.2 Śledzenie obiektów ruchomych

Śledzenie obiektów ruchomych jest nierozłącznym elementem aplikacji z dziedziny komputerowego przetwarzania i automatycznej analizy obrazu. Śledzenie obiektów poruszających się w polu widzenia kamery bazuje na algorytmach detekcji obiektów i ma za zadanie przypisanie jednoznacznego i niezmiennego kodu identyfikacyjnego każdemu obiektowi w obrazie na cały czas jego istnienia w strumieniu wizyjnym. Oznacza to, że w idealnej sytuacji, obiekt powinien być bezbłędnie śledzony niezależnie od zmian w wyglądzie samego obiektu (np. wynikających z obrotu obiektu) i od zmian trajektorii jego ruchu (skręcanie, zmiana prędkości) oraz, że proces śledzenia powinien być odporny na wzajemne przesłanie się obiektów z elementami tła oraz obiektów między sobą. Efektem działania algorytmu śledzenia powinna być ciągła trajektoria ruchu każdego obiektu pojawiającego się w scenie.

W algorytmach śledzenia obiektów wyróżnia się dwa procesy [40]. Pierwszy z nich dotyczy reprezentacji i lokalizacji obiektów (ang. *Target Representation and Localization*) i stanowi podejście „od szczegółu do ogółu” zajmujące się opisem cech obiektu i na tej podstawie lokalizowaniem obiektu w kolejnych ramkach. Ten proces musi sobie radzić z możliwymi zmianami w wyglądzie obiektów. Drugim procesem jest filtracja i asocjacja danych (ang. *Filtering and Data Association*), stanowiąca podejście „od ogółu do szczegółu” i zajmuje się dynamiką obiektów, wydobywaniem informacji o scenie oraz testowaniem różnych hipotez. Sposób i stopień, w jaki te dwa procesy są wykorzystywane w konkretnej aplikacji śledzenia obiektów zależą od przeznaczenia danej aplikacji oraz determinują skuteczność jej działania.

Algorytmy śledzenia obiektów można podzielić na trzy grupy (rys. 3.5):

- śledzenie kształtów geometrycznych,
- śledzenie jądra (ang. *Kernel Tracking*),
- śledzenie sylwetki.



Rys. 3.5 Podział metod śledzenia obiektów ruchomych

3.2.1 Śledzenie kształtów geometrycznych

Algorytmy z tej kategorii modelują śledzone obiekty jako proste kształty geometryczne: punkty, prostokąty, elipsy itd. Można je podzielić na dwie grupy. Do pierwszej należą metody deterministyczne, które bazując na heurystycznych regułach dotyczących charakterystyki położenia, wymiarów i ruchu obiektów pozwalają na ustalenie powiązania między wynikami detekcji obiektów w bieżącej ramce, a śledzonymi obiektami. Do drugiej grupy należą metody probabilistyczne, które bazując na obserwacji obiektów i uwzględniając niepewności tego pomiaru pozwalają na ustanowienie powiązań.

3.2.1.1 Metody deterministyczne

Metody deterministyczne śledzenia obiektów opisanych za pomocą kształtów geometrycznych wykorzystują funkcję kosztu powiązania każdego obiektu z ramki poprzedniej z określonym obiektem w ramce bieżącej. Funkcja kosztu powiązania definiowana jest z wykorzystaniem takich ograniczeń jak:

- bliskość – lokalizacje obiektów nie mogą zmienić się istotnie z ramki na ramkę,
- maksymalna prędkość – górne ograniczenie prędkości poruszania się obiektów, zawężające obszar poszukiwań do kołowego sąsiedztwa wokół obiektu,
- mała zmiana prędkości – wektor prędkości (kierunek i moduł) obiektu nie zmienia się w sposób raptowny,
- małe zmiany rozmiarów – wielkość obiektu nie zmienia się w sposób nagły,

- wspólny wzorzec ruchu – wektory prędkości punktów bliskich sobie są zbliżone; ma zastosowanie w przypadku reprezentowania obiektów przez wiele punktów,
- sztywność – rzeczywiste obiekty są bryłami sztywnymi, więc wzajemnie położenie punktów tych obiektów w obrazie nie może się zmieniać.

Reguły te są również często stosowane w przypadku innych metod śledzenia obiektów (np. metod statystycznych).

Minimalizacja funkcji kosztu uzyskiwana jest metodami programowania matematycznego. Rozwiązanie w postaci powiązania jeden-do-jednego między obiektami w sąsiednich ramkach obrazu może być uzyskane za pomocą optymalnej asocjacji (np. metoda węgierska [148]) lub metodami zachłannego poszukiwania (ang. *Greedy Search*) [4].

Praktyczne implementacje śledzenia obiektów metodami z tej kategorii wykorzystują różne ograniczenia oraz metody minimalizacji funkcji kosztu. W rozwiązaniu przedstawionym w pracy [212] zastosowano podejście zachłannego poszukiwania w oparciu o reguły bliskości i sztywności, bazujące na porównaniu ze sobą dwóch sąsiednich ramek obrazu. Ponieważ zakłada ono stałą liczbę obiektów, zostało rozszerzone o możliwość wykrywania nowych obiektów i uwzględniania obiektów znikających (poprzez opuszczenie sceny lub zasłonięcie) za pomocą dodania obiektów hipotetycznych w oparciu o reguły małej zmiany prędkości i rozmiarów [205]. Zasada wspólnego wzorca ruchu została wykorzystana w eksperymentach opisanych w pracy [245], gdzie do optymalizacji funkcji kosztu użyto metody węgierskiej. Chociaż rozwiązanie to uwzględnia możliwość zasłaniania się obiektów, to jednak zakłada stałą ich ustaloną liczbę, wobec czego nie jest odporne na pojawianie się lub znikanie obiektów. Odmiennym podejściem jest wykorzystanie teorii grafów do analizy zachowania obiektów w wielu sąsiednich ramkach przy uwzględnieniu kryteriów małej zmienności w czasie prędkości i lokalizacji obiektów [213]. W metodzie tej, dla każdego obiektu znajdowana jest najlepsza, unikalna ścieżka w grafie skierowanym, która zawiera lokalizację obiektu w każdej z analizowanych ramek. W procesie poszukiwania powiązań wykorzystywany jest algorytm zachłanny. Metoda ta uwzględnia możliwość wzajemnego zasłaniania się obiektów trwającego krócej niż okno czasowe analizy.

3.2.1.2 Metody statystyczne

Statystyczne metody śledzenia obiektów uwzględniają fakt, że dane pomiarowe dotyczące położenia i kształtu obiektów obarczone są szumem. Ponadto sam ruch obiektów cechuje nieprzewidywalność, związana np. z zakręcaniem. Wykorzystanie metod statystycznych pozwala uwzględnić te niepewności w procesie estymacji bieżącego stanu obiektu.

Metody statystyczne bazują przede wszystkim na procesie filtracji i asocjacji danych [42]. Można go w najbardziej ogólny sposób zdefiniować wykorzystując podejście przestrzeni stanów w modelowaniu dynamicznych systemów dyskretno-czasowych [6][40]. Informacja charakteryzująca obiekt jest zdefiniowana za pomocą sekwencji stanów $\{x_k\}_{k=0,1,\dots}$, których zmiany w czasie są określone przez dynamiczne równanie $x_k = f_k(x_{k-1}, v_k)$. Zmierzone dane $\{z_k\}_{k=1,2,\dots}$ są powiązane z odpowiadającymi im stanami równaniem pomiaru $z_k = h_k(x_k, n_k)$. W ogólności, f_k i h_k są nieliniowymi, zmiennymi w czasie funkcjami odwzorowującymi rzeczywiste liczby na wektory. Zakłada się, że każda sekwencja szumu $\{v_k\}_{k=1,2,\dots}$ i $\{n_k\}_{k=1,2,\dots}$ jest niezależna.

Celem śledzenia obiektów jest oszacowanie stanu x_k z wykorzystaniem wszystkich dotychczas zmierzonych wielkości $\{z_k\}_{k=1,\dots,k}$ lub (równoważnie) określenie funkcji gęstości prawdopodobieństwa $p(x_k|z_{1,\dots,k})$. Teoretycznie optymalne rozwiązanie można otrzymać z wykorzystaniem rekursywnego filtru Bayesa, który rozwiązuje problem w dwóch krokach. W kroku predykcji dynamiczne równanie i wyznaczona uprzednio funkcja gęstości prawdopodobieństwa stanu $p(x_{k-1}|z_{1,\dots,k-1})$ są stosowane do wyznaczenia funkcji *a priori* gęstości prawdopodobieństwa stanu bieżącego $p(x_k|z_{1,\dots,k-1})$. Następnie, w kroku uaktualniania, funkcja wiarygodności $p(x_k|z_k)$ bieżącego pomiaru jest wykorzystywana do obliczenia funkcji *a posteriori* gęstości prawdopodobieństwa $p(x_k|z_{1,\dots,k})$.

Metody statystyczne można podzielić w zależności od liczby analizowanych obiektów. W przypadku śledzenia pojedynczego obiektu wyróżnia się filtry Kalmana i filtry cząsteczkowe. Z kolei do śledzenia wielu obiektów jednocześnie przystosowane są filtr łącznego prawdopodobieństwa asocjacji danych (ang. *Joint Probability Data Association Filter*, JPDAF) oraz metoda wielokrotnych hipotez (ang. *Multiple Hypothesis Tracking*, MHT).

Filtr Kalmana

Jedną z najpowszechniej stosowanych metod statystycznego śledzenia obiektów jest zastosowanie filtrów Kalmana [16][252]. Przy założeniu, że sekwencje szumowe $\{v_k\}_{k=1,2,\dots}$ i $\{n_k\}_{k=1,2,\dots}$ mają rozkład normalny oraz funkcje f_k i h_k są liniowe, optymalne rozwiązanie można uzyskać wykorzystując filtr Kalmana, w wyniku czego prawdopodobieństwo wynikowe *a posteriori* ma również rozkład normalny.

Filtr Kalmana [252] to algorytm rekurencyjnego wyznaczania minimalno-wariancyjnej estymaty wektora stanu modelu liniowego dyskretnego układu dynamicznego na podstawie pomiarów wyjścia tego układu. Przyjmuje się założenie, że zarówno pomiar, jak i proces przetwarzania wewnątrz układu są obciążone błędami o rozkładzie gaussowskim. Filtr Kalmana ma postać zbioru rekursywnych równań matematycznych.

W ogólności, filtr Kalmana pozwala na estymowanie stanu $\mathbf{x} \in R^n$ procesu dyskretno-czasowego, który może być opisany stochastycznym równaniem różnicowym:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \quad (3.4)$$

i wykorzystuje wektor pomiarowy $\mathbf{z} \in R^n$:

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_{k-1} \quad (3.5)$$

Zmienne losowe \mathbf{w}_k oraz \mathbf{v}_k reprezentują odpowiednio szum przetwarzania i szum pomiarowy. Przyjmuje się, że błędy te są niezależne statystycznie i mają rozkład normalny:

$$p(\mathbf{w}) \sim N(0, \mathbf{Q}) \quad (3.6)$$

$$p(\mathbf{v}) \sim N(0, \mathbf{R}) \quad (3.7)$$

W praktyce kowariancja szumu przetwarzania \mathbf{Q} oraz kowariancja szumu pomiarowego \mathbf{R} mogą się zmieniać w każdym kroku procesu lub pomiaru.

Macierz przejścia \mathbf{A} o wymiarach $n \times n$ w równaniu różnicowym (3.4) wiąże stan procesu x w kroku poprzednim $k-1$ ze stanem w bieżącym kroku k , przy zerowym stanie wejścia kontrolnego \mathbf{u} oraz braku szumu przetwarzania. Macierz wejściowa \mathbf{B} o wymiarach $n \times 1$ wiąże opcjonalne wejście kontrolne $\mathbf{u} \in R^1$ ze stanem procesu \mathbf{x} . Macierz wyjściowa \mathbf{H} o wymiarach $m \times n$ w równaniu (3.5) wiąże stan procesu \mathbf{x} z bieżącym wektorem pomiarowym \mathbf{z} . W praktyce macierze \mathbf{A} i \mathbf{H} mogą się zmieniać w każdym kroku procesu lub z każdym pomiarem.

Filtr Kalmana estymuje proces w określonym momencie czasu z wykorzystaniem zwrotnej informacji o bieżącym (i zaszumionym) wyniku pomiaru. W związku z tym równania filtru Kalmana dzielą się na dwie kategorie: równania aktualizacji czasu i równania aktualizacji pomiarów. Równania aktualizacji czasu dokonują predykcji stanu układu w przód w celu otrzymania estymaty *a priori* $\hat{\mathbf{x}}_k^-$ stanu układu w kolejnym kroku czasowym:

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} \quad (3.8)$$

$$\hat{\mathbf{P}}_k^- = \mathbf{A}\hat{\mathbf{P}}_{k-1}\mathbf{A}^T + \mathbf{Q} \quad (3.9)$$

gdzie $\hat{\mathbf{P}}_k^-$ jest estymatą *a priori* błędu kowariancji, a $\hat{\mathbf{P}}_k$ – estymatą *a posteriori* błędu kowariancji.

Równania aktualizacji pomiarów są odpowiedzialne za sprzężenie zwrotne, tj. za włączenie nowych wyników pomiarów do estymaty *a priori* stanu procesu w celu uzyskania estymaty *a posteriori* $\hat{\mathbf{x}}_k$:

$$\mathbf{K}_k = \hat{\mathbf{P}}_k^- \mathbf{H}^T (\mathbf{H}\hat{\mathbf{P}}_k^- \mathbf{H}^T + \mathbf{R})^{-1} \quad (3.10)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-) \quad (3.11)$$

$$\hat{\mathbf{P}}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \hat{\mathbf{P}}_k^- \quad (3.12)$$

Macierz \mathbf{K}_k jest wzmocnieniem filtru Kalmana. Wartość wzmocnienia decyduje o tym, czy filtr bardziej "ufa" ocenie stanu (*a priori*), czy informacji zawartej w pomiarach. Zerowe wzmocnienie oznacza, że zupełnie ignorujemy pomiary - są one wówczas obarczone nieskończenie wielkim błędem (wariancja szumu pomiarowego \mathbf{R} jest nieskończona). Im mniejsza wariancja szumu pomiarowego, tym większa wartość wzmocnienia filtru Kalmana i tym większy wpływ wartości pomiarowych na estymatę stanu procesu.

Warunki początkowe filtru Kalmana są zdefiniowane przez początkowe estymaty $\hat{\mathbf{x}}_0$ i $\hat{\mathbf{P}}_0 = \text{diag}(p_0)$, gdzie $\text{diag}(p_0)$ oznacza macierz diagonalną z wartościami p_0 na głównej przekątnej.

W przypadku, gdy funkcje f_k i h_k są nieliniowe, poprzez ich linearyzację za pomocą rozwinięcia w szereg Taylora otrzymuje się rozszerzony filtr Kalmana (ang. *Extended Kalman Filter*) [6][260], którego prawdopodobieństwo *a posteriori* ma wciąż rozkład

normalny. Alternatywą dla rozszerzonego filtru Kalmana jest bezśladowy filtr Kalmana (ang. *Unscented Kalman Filter*) [126], który może być stosowany do modeli nieliniowych i nie-gaussowskich. Wykorzystuje on zestaw próbkowanych dyskretnie punktów do parametryzacji wartości średniej i kowariancji funkcji *a posteriori* gęstości prawdopodobieństwa.

Filtr cząsteczkowy

Filtry cząsteczkowe (ang. *Particle Filters*) [142][237][251], inaczej nazywane *Bootstrap Filters* [90] mogą być stosowane do modelowania procesów, w których zmienne nie są opisane rozkładami normalnymi. W podejściu tym, bazującym na metodach całkowania Monte Carlo, bieżąca gęstość prawdopodobieństwa stanu jest reprezentowana przez zbiór losowy próbek wraz ze stowarzyszonymi wagami. Nowa gęstość prawdopodobieństwa jest obliczana na podstawie tych próbek i ich wag [1][72]. Bezśladowy filtr Kalmana może zostać wykorzystany do wygenerowania rozkładów dla filtru cząsteczkowego, który wówczas jest nazywany bezśladowym filtrem cząsteczkowym (ang. *Unscented Particle Filter*) [172].

W metodzie wykorzystującej filtrację cząsteczkową, warunkowa gęstość prawdopodobieństwa stanu $p(x_k | z_{1...k})$ w czasie k jest reprezentowana przez zbiór próbek (cząsteczek) $\{s_k^n : n = 1 \dots N\}$ z wagami π_k^n . Wagi determinują istotność próbki, czyli częstotliwość jej występowania w danych pomiarowych [117]. W celu zredukowania złożoności obliczeniowej, dla każdej pary uporządkowanej (s^n, π^n) przechowywana jest również skumulowana waga c^n , gdzie $c^N = 1$. Nowe próbki w czasie k są wybierane spośród zbioru próbek $\mathbf{S}_{k-1} = \{(s_{k-1}^n, \pi_{k-1}^n, c_{k-1}^n) : n = 1 \dots N\}$ w czasie $k - 1$ w oparciu o różnorodne sposoby próbkowania [164]. Najpopularniejszym z nich jest próbkowanie ważnościowe (ang. *Importance Sampling*), które polega na wylosowaniu N próbek ze zbioru \mathbf{S}_{k-1} z uwzględnieniem skumulowanej wagi każdej z nich, tzn. losuje się taką próbkę j , dla której $c_{k-1}^j > r$, gdzie r jest liczbą losową z przedziału $[0,1]$ i ustawia się $\hat{s}_k^n = s_{k-1}^j$. Następnie dokonuje się predykcji wylosowanych próbek \hat{s}_k^n poprzez wygenerowanie nowych próbek:

$$s_k^n = f(\hat{s}_k^n, W_k^n) \quad (3.13)$$

gdzie W_k^n jest błędem gaussowskim o zerowej wartości średniej a f jest funkcją nieujemną, np. $f(s) = s$. W ostatnim etapie następuje korekcja wag π_k^n każdej próbki w oparciu o pomiar z_k zgodnie z równaniem:

$$\pi_k^n = p(z_k | x_k = s_k^n) \quad (3.14)$$

gdzie prawdopodobieństwo p może być modelowane rozkładem normalnym. W oparciu o nowy zbiór próbek \mathbf{S}_k nowa pozycja ε_k śledzonego obiektu może być estymowana wg zależności:

$$\varepsilon_k = \sum_{n=1}^N \pi_k^n \cdot f(s_k^n, W) \quad (3.15)$$

Filtr cząsteczkowy może być zainicjowany albo w oparciu o pierwszy pomiar ($s_0^k \sim Z_0$), $\pi_0^n = \frac{1}{N}$ lub poprzez trening systemu za pomocą przykładowych nagrań.

Metoda JPDAF

Metoda JPDAF (ang. *Joint Probability Data Association Filter*) pozwala na łączną analizę statystyczną trajektorii wielu obiektów w celu znalezienia optymalnego dopasowania wyników odejmowania tła w bieżącej ramce do aktualnie istniejących obiektów w analizowanej scenie [42][192]. Ścieżka ruchu zdefiniowana jest jako sekwencja pomiarów pochodzących od tego samego obiektu. Zakładając, że w chwili k dostępnych jest N_k ścieżek i m_k pomiarów $\mathbf{Z}_k = \{\mathbf{z}_k^1 \dots \mathbf{z}_k^m\}$, celem jest przypisanie tych pomiarów do odpowiednich ścieżek. Zakłada się, że liczba ścieżek jest niezmienna w czasie. Niech $\mathbf{v}_{i,l}$ będzie innowacją skojarzoną ze ścieżką l w wyniku pomiaru \mathbf{z}^l :

$$\mathbf{v}_{i,l} = \mathbf{z}^l - \mathbf{H}^l(\mathbf{x}^l) \quad (3.16)$$

Metoda JPDAF kojarzy wszystkie pomiary z każdą ścieżką. Łączna ważona innowacja \mathbf{v}_l dla ścieżki l dana jest zależnością:

$$\mathbf{v}_l = \sum_{i=1}^{m_k} \beta_i^l \mathbf{v}_{i,l} \quad (3.17)$$

gdzie β_i^l jest prawdopodobieństwem *a posteriori*, że pomiar i należy do obiektu skojarzonego ze ścieżką l .

Głównym ograniczeniem metody JPDAF jest fakt, że nie uwzględnia ona nowych obiektów pojawiających się w polu widzenia kamery lub obiektów go opuszczających. W związku z tym i faktem, że asocjacja danych odbywa się w oparciu o stałą liczbę obiektów śledzonych w dwóch sąsiednich ramek obrazu, zmiana liczby obiektów może doprowadzić do powstania poważnych błędów.

Metoda MHT

Wspomnianej w poprzednim akapicie wady pozbawiona jest metoda śledzenia obiektów z wykorzystaniem wielokrotnych hipotez MHT (ang. *Multiple Hypothesis Tracking*), który szacuje prawdopodobieństwo, że dany obiekt wywołał określoną sekwencję pomiarów [193]. W metodzie tej decyzja o powiązaniu ze sobą ścieżek ruchu i danych obserwacyjnych jest opóźniona do czasu, aż analizie zostanie poddanych kilka ramek obrazu. W związku z tym, algorytm rozpatruje wiele hipotez powiązania każdego obiektu w każdej ramce obrazu. Wynikową ścieżką obiektu jest najbardziej prawdopodobny zbiór powiązań w trakcie trwania okna czasowego analizy.

MHT jest algorytmem iteracyjnym, który rozpoczyna się od zbioru hipotez; każda hipoteza stanowi zbiór rozłącznych ścieżek reprezentujących zachowanie poszczególnych obiektów. Dla każdej hipotezy dokonuje się predykcji położenia każdego obiektu w kolejnej ramce. Wyniki predykcji są następnie porównywane z rzeczywistymi wynikami pomiarów za pomocą miary odległościowej. Na tej podstawie tworzony jest zestaw nowych powiązań dla każdej hipotezy, które indukują nowe hipotezy dla kolejnej iteracji algorytmu; każda z nich reprezentuje nowy zbiór ścieżek bazujących na bieżących obserwacjach [26][43].

Dużą zaletą metody jest możliwość rozpoczynania nowych ścieżek dla obiektów pojawiających się w polu widzenia kamery i kończenie ścieżek dla obiektów znikających z kadru. Jest również w stanie rozwiązywać krótkookresowe konflikty śledzenia w przypadku, gdy niektóre obiekty przez określony czas nie są wykrywane w obrazie na skutek ich zasłonięcia.

Sposób tworzenia powiązań w omawianej metodzie jest deterministyczny i polega na listowaniu wszystkich możliwych powiązań. W celu ograniczenia złożoności obliczeniowej i wymagań pamięciowych tego procesu, zaproponowano probabilistyczny wariant algorytmu MHT (PMHT, ang. *Probabilistic Multiple Hypothesis Tracking*) [220]. W rozwiązaniu tym poszczególne powiązania traktowane są jako niezależne statystycznie zmienne losowe, co znosi konieczność analizy wszystkich możliwych wariantów.

3.2.2 Śledzenie jądra

Algorytmy z grupy metod bazujących na śledzeniu jądra (ang. *Kernel Tracking*) stanowią realizację procesu reprezentacji i lokalizacji obiektów. Polegają one w ogólności na estymacji ruchu obiektu (w sposób parametryczny jako wartości przekształceń lub nieparametryczny jako pole wektorowe), z ramki na ramkę, poprzez analizę wyglądu obiektów, które reprezentowane są przez proste kształty geometryczne (prostokąt, elipsa, itd.). Poszczególne rozwiązania różnią się sposobem reprezentacji wyglądu obiektów, liczbą jednocześnie śledzonych obiektów oraz sposobem estymacji ruchu obiektów. Ze względu na sposób opisu wyglądu obiektów metody wykorzystują:

- wzorce obiektów i gęstości prawdopodobieństwa określonych parametrów wyglądu obiektu,
- wiele widoków obiektu.

3.2.2.1 Śledzenie w oparciu o wzorce i cechy wyglądu obiektów

Wzorec śledzonego obiektu definiowany jest z wykorzystaniem prostych kształtów geometrycznych lub całej sylwetki obiektu [81]. Zaletą stosowania wzorca jest fakt, że łączy on informację o wyglądzie obiektu z relacjami przestrzennymi. Z kolei wygląd obiektu najczęściej jest reprezentowany przez estymaty gęstości prawdopodobieństwa określonych cech wyglądu (koloru, tekstury itp.) w obszarach obrazu zdefiniowanych kształtem obiektu. Gęstości prawdopodobieństwa mogą być wyrażone w sposób parametryczny (np. jako rozkład normalny [264] lub poprzez sumę ważonych rozkładów gaussowskich [187]) albo nieparametryczny, jako okna Parzena [77] lub histogramy [40].

Algorytmy śledzenia obiektów należące do tej grupy można podzielić ze względu na przystosowanie do śledzenia pojedynczych obiektów lub ich większej ilości.

Śledzenie pojedynczych obiektów

Najbardziej powszechną metodą należącą do tej kategorii jest dopasowanie wzorca (ang. *Template Matching*), które jest metodą siłowego poszukiwania w bieżącej ramce obszaru podobnego do wzorca zdefiniowanego w ramce poprzedniej. Pozycja obiektu w bieżącej ramce znajdowana jest w oparciu o miary podobieństwa wzorców, np. korelację lub błąd kwadratowy. Jako parametry obrazu używane są zwykle kolor, jasność pikseli lub gradient obrazu [11]. Zaletą tej ostatniej miary jest jej niewrażliwość na zmiany oświetlenia. Ze względu na podejście siłowe, metody dopasowania wzorca są złożone

obliczeniowo. Sposobem zwiększenia wydajności algorytmu jest wprowadzenie ograniczeń dotyczących ruchu obiektu (np. poszukiwanie obiektu w pobliżu jego położenia w ramce poprzedniej) oraz zastosowanie bardziej efektywnych algorytmów dopasowywania [210].

Zamiast stosowania wzorców możliwe jest użycie innych reprezentacji wyglądu obiektów, w szczególności histogramów koloru lub modeli mieszanych, wyznaczanych wewnątrz obszarów o prostych kształtach. Dla przykładu, w rozwiązaniu przedstawionym w pracy [81] modele obiektów są tworzone poprzez wyznaczenie średniej wartości koloru w prostokątnym obszarze. W celu zredukowania złożoności obliczeniowej, obiekt poszukiwany jest w ośmiu sąsiednich lokalizacjach, w porównaniu z ramką poprzednią i tą lokacją, której średnia barwa jest najbardziej zbliżona do koloru modelu jest uznawana za bieżące położenie obiektu.

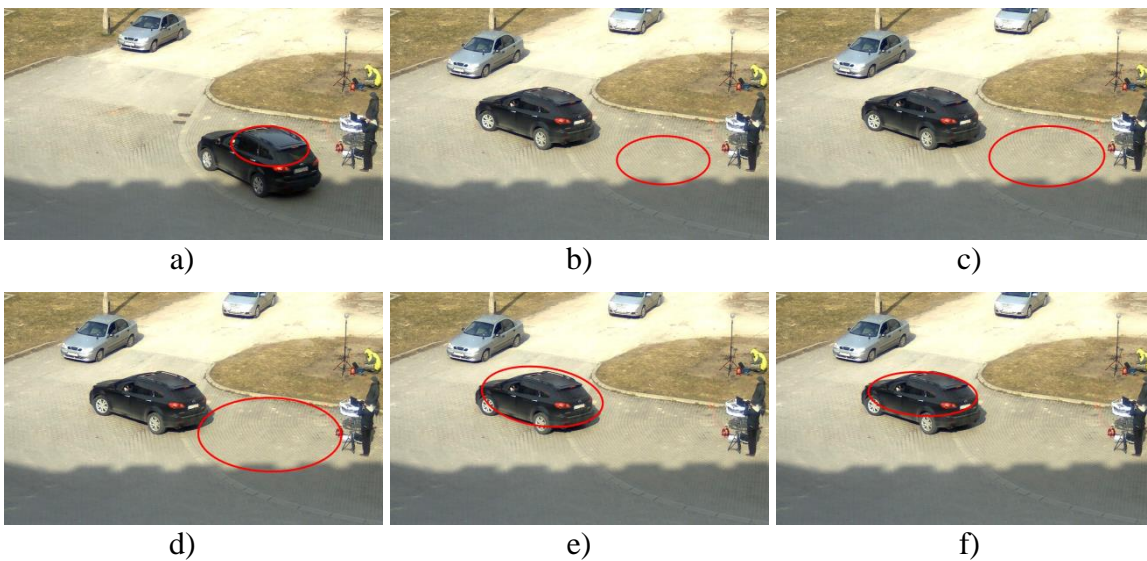
Innym sposobem realizacji śledzenia obiektów jest użycie ważonego histogramu obliczonego w okrągłych obszarach jako reprezentacji wyglądu obiektu i zastosowanie algorytmu Mean-Shift (podrozdział 3.1.2) do znalezienia położenia obiektu w bieżącej ramce [40]. Wartości pikseli służące do wyznaczenia histogramu są ważone w funkcji odległości ich położenia od środka okręgu; im punkt położony bliżej krawędzi, tym waga mniejsza. Algorytm śledzenia w sposób iteracyjny modyfikuje wektor przesunięcia *mean-shift* w celu zwiększania podobieństwa wyglądu poprzez porównanie histogramów (z wykorzystaniem miary Bhattacharya) obiektu i okna poszukiwania zlokalizowanego wokół hipotetycznej pozycji obiektu. Proces kończy się po uzyskaniu zbieżności, co zwykle wymaga kilku powtórzeń. Rozszerzeniem tego podejścia jest użycie w miejsce zwyczajnego histogramu rozkładu łącznego wykorzystującego zarówno wartości pikseli jak i ich lokalizację w przestrzeni [38]. Przykładowe wyniki śledzenia obiektów algorytmem Mean-Shift pokazano na rys. 3.6. Niewątpliwą zaletą zastosowania tej metody w miejsce dopasowywania wzorca jest wyeliminowanie poszukiwania „na ślepo” i uzyskanie wyniku zwykle po kilku iteracjach. Wadą jest konieczność inicjalizacji algorytmu poprzez wskazanie śledzonego obiektu w pierwszej ramce obrazu (elipsa reprezentująca obiekt musi zawierać przynajmniej fragment śledzonego obiektu).

Praca [122] zawiera opis algorytmu śledzenia obiektów modelowanych przez kombinację trzech rozkładów reprezentujących: część stabilną, obejmującą najbardziej wiarygodny wygląd obiektu na potrzeby estymacji ruchu, część zmienną, identyfikującą szybko zmieniające się piksele oraz część szumową, która zajmuje się elementami wyglądu obiektu wyraźnie odstającymi od reszty na skutek szumu. W celu wyznaczenia

parametrów rozkładów zastosowano algorytm maksymalizacji wartości oczekiwanej EM (ang. *Expectation Maximization*) [67]. Obiekty są reprezentowane przez elipsy, a jako parametry obrazu wykorzystano fazę odpowiedzi filtrów kierunkowych (ang. *Steerable Filters*) [83]. Ruch obiektu jest wyznaczany za pomocą transformacji deformującej obiekt z ramki poprzedniej do ramki bieżącej; parametrami tej deformacji są przesunięcie (t_x, t_y) , obrót (a, b) oraz skala s :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = s \begin{bmatrix} a & b \\ -b & a \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (3.18)$$

Do wyznaczenia parametrów przekształcenia wykorzystywana jest ważona kombinacja rozkładów reprezentujących części: stabilną i zmienną obiektu. Wykorzystanie obu części daje możliwość przyznania wyższej wagi cechom stabilnym obrazu, co pozwala skutecznie śledzić obiekty częściowo zmienne (np. twarz mówiącej osoby, gdzie część zmienna reprezentuje ruchy ust, a część stała nieruchomy nos i czoło).



Rys. 3.6 Ilustracja działania śledzenia ciemnego pojazdu algorytmem Mean-Shift: a) estymowana pozycja obiektu w ramce poprzedniej, b),c), d), e), f) wyniki lokalizacji obiektu w bieżącej ramce po przeprowadzeniu iteracji numer 1, 2, 3, 4 i 6

Zastosowanie znajduje także wyznaczanie przesunięcia obiektu opisanego prostym kształtem za pomocą metod obliczania przepływu optycznego (podrozdział 3.1.4), rozszerzonych na zagadnienie wyznaczania wektorów przesunięcia dla obszarów prostokątnych. Poprzez analizę przesunięcia regionów obrazu zaczepionych w punktach charakterystycznych znalezionych za pomocą detektora KLT (podrozdział 3.1.1)

wyznaczane są odpowiadające im punkty charakterystyczne w następnej ramce obrazu [216]. Jakość dopasowanego regionu obrazu badana jest z poprzez wyznaczenie parametrów przekształcenia afinicznego pomiędzy regionem źródłowym a dopasowanym, rzutowanie regionu źródłowego z wykorzystaniem tego przekształcenia i obliczenie błędu kwadratowego w porównaniu z regionem źródłowym. W przypadku niskiej wartości różnicy, śledzenie może być kontynuowane.

Śledzenie wielu obiektów

Przedstawione w poprzednim podrozdziale metody śledzenia są przystosowane do modelowania wyglądu tylko jednego, określonego obiektu, w związku z czym nie biorą pod uwagę możliwych interakcji tego obiektu z elementami tła i innymi obiektami. W związku z tym w pracy [239] przedstawiono podejście, w którym modelowany jest kompletny obraz jako zbiór warstw, na który składa się jedna warstwa dotycząca tła sceny i po jednej warstwie na każdy ruchomy obiekt. Na każdą warstwę składa się model kształtu w postaci elipsy, model ruchu jako wartości przesunięcia i obrotu oraz model wyglądu zdefiniowany jako pojedynczy rozkład normalny wartości luminancji. W pierwszej kolejności kompensowany jest ruch tła sceny, a następnie wyznaczane jest prawdopodobieństwo przynależności każdego piksela do określonej warstwy (obiektu) biorąc pod uwagę dotychczasowe parametry ruchu i kształtu obiektu, a także wygląd obiektu. Parametry modelu (kształt, ruch, wygląd) są uaktualniane iteracyjnie algorytmem maksymalizacji wartości oczekiwanej EM (ang. *Expectation Maximization*) [67] w celu maksymalizacji obserwacji określonej warstwy w bieżącej ramce obrazu.

Innym sposobem łącznego modelowania tła i obiektów pierwszoplanowych jest zastosowanie modelu łącznego, w którym zarówno wygląd tła, jak i wszystkich obiektów, modelowane są za pomocą sumy ważonych rozkładów gaussowskich (podrozdział 3.1.5) [118]. Zakłada się przy tym znajomość płaszczyzny, po której poruszają się obiekty, w związku z czym możliwe jest określenie ich pozycji 3D oraz modelowanie ich kształtu za pomocą walca. Śledzenie obiektów odbywa się za pomocą filtrów cząsteczkowych (rozdział 3.2.1.2), których wektor stanu uwzględnia lokalizację 3D, kształt i wektor prędkości wszystkich ruchomych obiektów w scenie. Poprzez zastosowanie zmodyfikowanych procedur predykcji i korekcji filtrów cząsteczkowych możliwa jest dynamiczna zmiana rozmiaru wektora stanu w celu uwzględniania obiektów pojawiających się jak i znikających. Wadą tej metody jest konieczność określenia z góry maksymalnej liczby obiektów możliwych do śledzenia, użycie tego samego modelu

wyglądu do śledzenia wszystkich obiektów i konieczność wcześniejszego treningu algorytmu w celu rozpoznawania obiektów pierwszoplanowych.

3.2.2.2 Śledzenie w oparciu o wiele widoków obiektu

Przedstawione dotychczas algorytmy opierają się na tworzonym dynamicznie, pojedynczym modelu wyglądu obiektu. Jest od dopasowany do wyglądu aktualnie poruszającego się obiektu, ale w przypadku zmiany jego pozy lub położenia w warunkach nierównomiernego oświetlenia, wygląd obiektu może ulec drastycznym zmianom, co zwykle skutkuje błędami śledzenia obiektów. Rozwiązaniem tego problemu może być przygotowanie wielu reprezentacyjnych obrazów obiektów i wcześniejsze nauczenie algorytmu wyglądu śledzonych obiektów np. przy użyciu analizy składowych głównych PCA (ang. *Principal Component Analysis*) [14] lub analizy składowych niezależnych ICA (ang. *Independent Component Analysis*) [179].

Przykładową realizacją tego pomysłu jest zastosowanie analizy PCA do utworzenia nowej podprzestrzeni reprezentującej wygląd obiektu. W trakcie śledzenia wyznaczane są iteracyjnie parametry przekształcenia afinicznego, które minimalizuje różnicę między bieżącym obrazem obiektu, a obrazem zrekonstruowanym przy użyciu wektorów własnych [14].

Rozszerzeniem zastosowania analizy PCA do śledzenia obiektów jest wykorzystanie aktywnych modeli obiektów. Modele takie powstają przez równoczesne modelowanie kształtu obiektu oraz cech jego wyglądu [76]. W ogólności, kształt obiektu zdefiniowany jest przez zbiór punktów, które zlokalizowane są w charakterystycznych miejscach obiektu rozmieszczonych na jego krawędziach lub w jego wnętrzu. W oparciu o punkty kształtu, wyznaczana jest reprezentacja wyglądu obiektu. Przykładowymi realizacjami tego podejścia są algorytmy Active Shape Model (ASM) [62] oraz jego bezpośredni następca, Active Appearance Model (AAM) [41][218]. Oba algorytmy wykorzystują taką samą definicję kształtu obiektu, jednak różnią się sposobem reprezentacji wyglądu obiektu. W metodzie ASM dla każdego punktu kształtu jest to wygląd obiektu w sąsiedztwie tego punktu, zwykle w postaci wektora zawierającego kolor, teksturę lub gradient obrazu. Z kolei AAM uwzględnia wszystkie piksele obiektu znajdujące się wewnątrz jego konturu. Analiza PCA w aktywnych modelach służy do wyznaczenia modelu obiektu w postaci typowych zmian kształtu i wyglądu obiektu. Oznacza to konieczność przygotowania treningowych obrazów obiektów, w których pozycja punktów kształtu została oznaczona ręcznie. Proces dopasowywania modelu do nowego obrazu jest iteracyjny i wymaga

rozwiązania problemu optymalizacyjnego, którego zdaniem jest minimalizacji różnicy pomiędzy wyglądem obiektu obliczonym na podstawie jego modelu, a wyglądem obiektu wyznaczonym bezpośrednio z bieżącej ramki obrazu dla różnych położeń obiektu w obrazie i deformacji kształtu obiektu [41][218]. Proces ten jest wrażliwy na warunki początkowe, co oznacza że estymacja położenia obiektu w pierwszej ramce analizowanego strumienia powinna być możliwie jak najbardziej zbliżona do rzeczywistego położenia obiektu.

Innym rozwiązaniem jest użycie dowolnego klasyfikatora uczącego się, np. przedstawionego w rozdziale 3.1.3, do nauczenia wyglądu obiektu. Jako wzorce pozytywne służą w tym przypadku obrazy obiektów, a jako wzorce negatywne – obrazy tła sceny, co do których istnieje największa szansa pomylenia ich z obiektami pierwszoplanowymi. W trakcie śledzenia następuje skanowanie ramki obrazu oknem ruchomym w celu znalezienia tego obszaru, który charakteryzuje się najwyższym wynikiem klasyfikacji. Przykład takiego podejścia, wykorzystującego klasyfikator SVM, przedstawiono w artykule [3]. Zaletą tej metody jest fakt, że poza wyglądem obiektu, wytrenowany klasyfikator zawiera w sposób jawny również wiedzę o tle sceny.

3.2.3 Śledzenie sylwetki

Podobnie jak metody śledzenia jądra, śledzenie sylwetki obiektu stanowi realizację procesu reprezentacji i lokalizacji obiektów. W przypadku obiektów o skomplikowanych kształtach (np. sylwetek ludzi), opisywanie ich za pomocą prostych kształtów geometrycznych (prostokąt, elipsa) może być zbyt mało dokładne. W takich przypadkach lepiej sprawdzają się metody oparte na śledzeniu całej sylwetki. Polegają one na znalezieniu w ramce bieżącej położenia obiektu na podstawie jego modelu wygenerowanego w oparciu o ramki poprzednie. Model ten może być w formie histogramu koloru, krawędzi obiektu lub jego konturów. Metody śledzenia sylwetki można podzielić na dwie grupy: dopasowywanie kształtu i śledzenie konturu. Pierwsza grupa dotyczy poszukiwania określonej sylwetki obiektu w bieżącej ramce. W ramach metod z drugiej grupy śledzenia konturu odbywa się poprzez ewolucyjne przemieszczaniu konturu obiektu na jego nową pozycję w bieżącej ramce.

3.2.3.1 *Dopasowywanie kształtu*

Dopasowywanie kształtu obiektu do bieżącej ramki obrazu może być przeprowadzone podobnie, jak w przypadku algorytmów bazujących na dopasowaniu wzorca (podrozdział 3.2.2.1). Poszukiwanie obiektu w bieżącej ramce odbywa się poprzez obliczenie podobieństwa obiektu do modelu wygenerowanego na podstawie sylwetki obiektu w ramce poprzedniej. Zakłada się, że sylwetka może ulec jedynie przesunięciu, wobec czego zmiany wielkości i obrót nie są uwzględniane w sposób jawny. Model obiektu jest reinicjalizowany dla każdej ramki tak, aby uwzględnić potencjalne modyfikacje wyglądu obiektu wynikające z jego obrotu, zmiany kształtu lub zmian oświetlenia. Przykładem realizacji takiego algorytmu jest użycie modelu obiektu bazującego na krawędziach i zastosowanie metryki Hausdorffa [104] do wyznaczenia płaszczyzny korelacji, w której minimalne wartości są wybierane jako nowa pozycja obiektu [113]. Odległość Hausdorffa przyjmuje w tym przypadku duże wartości w sytuacji braku dopasowania krawędzi, wobec czego usunięcie tych krawędzi (tożsamy np. z nogami i rękoma idącego człowieka) poprawia jakość procesu śledzenia. Podobne podejście opisano w literaturze [155], gdzie odległość Hausdorffa została wykorzystana do weryfikacji estymowanej trajektorii ruchu obiektu, uzyskanej za pomocą przepływu optycznego.

Inną metodą dopasowywania kształtu jest znalezienie odpowiadających sobie regionów wyznaczonych w dwóch sąsiednich ramkach obrazu. Proces ten jest analogiczny do zagadnienia śledzenia prostych kształtów geometrycznych (podrozdział 3.2.1), różnice dotyczą innej reprezentacji śledzonego obiektu. Detekcja sylwetek obiektów w porównywanych ramkach obrazu odbywa się zwykle metodami odejmowania tła (podrozdział 3.1.5). Następnie dopasowywanie znalezionych kształtów do siebie odbywa się poprzez obliczenie miar podobieństwa między modelami wyglądu obiektu skojarzonymi z każdym kształtem, opartych na histogramach krawędzi lub koloru, konturze obiektu, jego krawędziach lub kombinacji wszystkich tych cech [100][132]. Jako miara podobieństwa stosowana jest korelacja wzajemna, odległość Bhattacharya lub dywergencja Kullbacka-Leiblera [132].

W przeciwieństwie do poszukiwania odpowiadających sobie sylwetek w dwóch sąsiednich ramkach obrazu, możliwe jest znalezienie trajektorii ruchu obiektu poprzez wyznaczenie dominującego wektora ruchu pikseli wewnątrz konturów obiektu. W tym celu można posłużyć się transformacją Hougha w dziedzinie prędkości zastosowaną do

sylwetek obiektów w sąsiednich ramkach [207], znalezionych np. metodami odejmowania tła. Na tej podstawie uzyskuje się reprezentację czterowymiarową obrazu, w którym z każdą współrzędną piksela skojarzony jest ruch w kierunkach poziomym i pionowym. Taka reprezentacja zawiera informację o dominujących składowych ruchu obszarów w obrazie oraz ich podobieństwo w takim sensie, że zastosowanie operacji progowania pozwoli uzyskać obszary obrazu o podobnej charakterystyce ruchu. Podejście takie, w porównaniu z dopasowaniem kształtu na podstawie jego wyglądu, jest mniej wrażliwe na zmiany wyglądu obiektów, wynikające np. z jego obrotu.

3.2.3.2 Śledzenie konturu

Algorytmy bazujące na śledzeniu konturu w sposób iteracyjny modyfikują kontur obiektu znaleziony w poprzedniej ramce tak, aby odpowiadał on położeniu tego obiektu w ramce bieżącej. Podejście to wymaga, aby sylwetki obiektów w sąsiednich ramkach się na siebie nakładały. Istnieją dwa podejścia do zagadnienia śledzenia konturu. Pierwsze z nich wykorzystuje modele przestrzeni stanu do reprezentowania kształtu konturu i jego ruchu, drugie bezpośrednio przekształca kontur poprzez minimalizację określonej funkcji energii konturu, np. metodami gradientowymi.

Modele przestrzeni stanu

Stan śledzonego obiektu definiowany jest przy użyciu jego kształtu i parametrów ruchu konturu. Stan jest uaktualniany w każdej ramce obrazu tak, aby maksymalizować prawdopodobieństwo *a posteriori* dopasowania konturu. Zależy ono od poprzedniego stanu oraz bieżącego podobieństwa, które zwykle jest definiowane z wykorzystaniem odległości konturu od wykrytych krawędzi.

W praktycznych implementacjach stosuje się różne mechanizmy do modelowania stanu obiektu. W pracy [240] stan obiektu zdefiniowany jest poprzez dynamikę punktów kontrolnych, które poruszają się zgodnie z modelem sprężyny charakteryzowanej parametrem sztywności. Predykcja stanu następuje z wykorzystaniem filtru Kalmana. Innym sposobem jest modelowanie stanu za pomocą funkcji sklejaney reprezentującej kształt i afinicznych parametrów ruchu [117]. W tym przypadku stan jest uaktualniany za pomocą filtru cząsteczkowego. Po rozszerzeniu procedury próbkowania filtru cząsteczkowego o zasadę wyłączności możliwa jest również obsługa wzajemnego przesłaniania się dwóch obiektów. Wg tej zasady, jeśli dana cecha należy do obu obiektów, to wówczas zalicza się ona tylko do obiektu zasłaniającego.

Możliwe jest również zastosowanie ukrytych modeli Markowa HMM (ang. *Hidden Markov Models*) do modelowania stanu obiektu [32]. W tym przypadku kontur obiektu aproksymowany jest za pomocą elipsy. Każdy węzeł konturu jest powiązany z jednym HMM, a stan każdego HMM jest określony przez punkty leżące na prostej normalnej w stosunku do konturu. Bieżący stan konturu aproksymowany jest za pomocą algorytmu Viterbiego [249] na podstawie podobieństwa obserwacji konturu oraz prawdopodobieństwa zmiany stanu wyznaczonego za pomocą metody JPDAF (podrozdział 3.2.1.2).

Opisane w tym rozdziale metody bazują na jawnej reprezentacji konturu obiektu, która nie umożliwia zmian w topologii, takich jak: łączenie lub rozdzielanie się obszarów.

Bezpośrednia minimalizacja energii konturu

Wymienionej powyżej wady pozbawione są metody bazujące na minimalizacji energii konturu, w których kontur reprezentowany jest w sposób niejawni jako siatka przestrzenna. W siatce tej wartości bezwzględne punktów oznaczają odległość od konturu, a znak determinuje lokalizację jako wewnątrz lub na zewnątrz obiektu. Należące do tej grupy metody są analogiczne do metod detekcji obiektów za pomocą segmentacji, przedstawionych w rozdziale 3.1.2. Minimalizacja energii następuje albo metodami zachłannego poszukiwania albo gradientowymi [174]. Energia konturu definiowana jest w postaci czasowej jako gradient (przepływ optyczny) [10][44][166] albo jako statystyki wyglądu wygenerowane na podstawie obiektu oraz obszarów tła [197][259].

3.2.4 Podsumowanie

Przedstawione w powyższych rozdziałach metody śledzenia obiektów ruchomych charakteryzuje ogromna różnorodność podejścia do tego zagadnienia, która rzutuje na możliwość ich wykorzystania w konkretnych zastosowaniach. Ponadto powszechne jest stosowanie rozwiązań połączonych, w których użyte są metody należące do różnych kategorii. W tab. 3.2 przedstawiono porównanie metod śledzenia ruchomych obiektów pod kątem ich zastosowania w monitorowaniu ruchu drogowego.

Ze względu na praktyczną dowolność wyglądu i kształtu obiektów poruszających się w polu widzenia kamer monitoringu (ludzie, pojazdy itp.), wybrana metoda śledzenia obiektów musi być uniwersalna (niezależna od konkretnego typu obiektu) oraz nie może wymagać wcześniejszego treningu na w celu uzyskania skutecznego śledzenia. Z tego powodu nie nadają się do tego celu metody bazujące na aktywnych modelach kształtu i

algorytmach uczących się. Ponadto, w typowych scenariuszach monitoringu wizyjnego mamy do czynienia z obiektami, które można modelować z dużą dokładnością jako bryły sztywne (pojazdy) i których rozmiary są małe w porównaniu z wielkością ramki obrazu, wobec czego można pominąć zmiany wyglądu obiektu wynikające z pozycji i fazy ruchu (ludzie). Z tego założenia wynika praktyczna nieprzydatność metod z kategorii śledzenia sylwetki, których dokładność działania byłaby utrudniona ze względu na zwykle dość niską rozdzielczość obrazów poszczególnych obiektów, a związana z określaniem kształtu obiektu moc obliczeniowa, bardzo cenna w aplikacjach działających w czasie rzeczywistym, byłaby w praktyce marnowana.

Tab. 3.2 Porównanie metod śledzenia obiektów ruchomych

Kategoria	Metoda	Łączne śledzenie wielu obiektów	Uwzględnienie zmiany liczby obiektów	Odporność na zasłanianie	Nie wymaga treningu	Nie wymaga inicjalizacji
Śledzenie kształtów	Metody deterministyczne	+	o	c	+	+
	Filtr Kalmana	-	-	-	+	+
	Filtr cząsteczkowy	-	-	-	+	+
	Metoda JPDAF	+	-	-	+	+
	Metoda MHT	+	+	+	+	+
Śledzenie w oparciu o wzorce i cechy wyglądu	Dopasowanie wzorca	-	-	c	+	-
	Algorytm Mean-Shift	-	-	c	+	-
	Detektor KLT	-	-	c	+	-
	Łączne modelowanie tła i wszystkich obiektów	+	o	+	o	+
	Aktywne modele kształtu	-	-	c	-	-
	Klasyfikatory uczące	-	-	c	-	-
Śledzenie sylwetki	Dopasowywanie kształtu	-	-	-*	+*	+
	Śledzenie konturu za pomocą modeli przestrzeni stanu	-*	-	-*	-	+
	Śledzenie konturu z wykorzystaniem bezpośredniej minimalizacji energii konturu	-*	+	o	+*	+

+ – tak

- – nie

o – w zależności od algorytmu

* – w większości metod, choć istnieją rozwiązania, które posiadają przeciwną własność

c – częściowo

Z powyższej analizy wynika, że do śledzenia obiektów w systemie monitoringu należy zastosować metody bazujące na śledzeniu kształtów oraz bazujące na dopasowaniu wzorców i cech wyglądotów. Metody deterministyczne są najprostsze w swej implementacji, nie uwzględniają jednak niepewności danych pomiarowych. Z tego względu zdecydowanie lepsze wydają się być algorytmy statystyczne. Najpopularniejsze z nich (filtr Kalman i filtr cząsteczkowy) są jednak przystosowane do śledzenia tylko pojedynczego obiektu, w związku z tym niezbędne jest dodanie dodatkowej warstwy analizy pozwalającej rozwiązywać konflikty śledzenia. Takie własności mają algorytmy śledzenia w oparciu o wzorce i cechy wyglądotów, jednak ich wadą jest konieczność inicjacji procesu śledzenia, która zwykle może być przeprowadzona w oparciu o wyniki detekcji ruchomych obiektów w bieżącej ramce.

3.2.5 Śledzenie wielokamerowe

Oddzielnym zagadnieniem z zakresu śledzenia obiektów ruchomych jest śledzenie tych obiektów w środowisku wielokamerowym w taki sposób, aby zapewnić właściwą identyfikację tego samego obiektu w obrazach z różnych kamer, których pola widzenia nie pokrywają się. Wiedza taka jest niezbędna do poznania lokalizacji osoby lub pojazdu będącego obiektem zainteresowania w dłuższym okresie czasu, a przez to – do analizy jego zachowania i wykrywania określonych zdarzeń.

W literaturze można znaleźć wiele podejść do zagadnienia automatycznego, wielokamerowego śledzenia ruchomych obiektów. Wiele z nich wymaga skalibrowanego pola widzenia kamery i zakłada znajomość modelu całego monitorowanego obszaru [120]. Informacja o wzajemnych relacjach między polami widzenia poszczególnych kamer może być zdefiniowana w sposób manualny [21][37] lub wyznaczona automatycznie na podstawie obserwacji obiektów poruszających się po monitorowanym obszarze [139][153]. Dzięki temu dopasowywanie do siebie obiektów rejestrowanych przez różne kamery odbywa się na zbiorze kandydatów, który ograniczony jest na podstawie analizy fizycznego rozmieszczenia kamer. Oznacza to konieczność wprowadzenia założeń dotyczących możliwych ścieżek ruchu obiektów i ich prędkości [112][120][138]. W celu dopasowania lokalnego (w polu widzenia każdej kamery) oraz globalnego (na płaszczyźnie gruntu) stosuje się zwykle filtry cząsteczkowe [74] lub filtry Kalmana [35].

W procesie wielokamerowego śledzenia obiektów niezbędna jest analiza wyglądotów obiektów zarejestrowanych przez różne kamery w celu ich porównania i określenia obiektów sobie odpowiadających. W artykule [21] cechy geometryczne i barwa obiektu są

modelowane za pomocą wielowymiarowego rozkładu normalnego, a stopień podobieństwa wyznaczany jest w oparciu o odległość Mahalanobisa. Suma ważonych rozkładów Gaussa jest z kolei używana do reprezentowania wyglądu obiektu w rozwiązaniu opisanym w pozycji [121]. Modele obiektów są dopasowywane do siebie z wykorzystaniem funkcji maksymalizującej miarę niepodobieństwa między nimi.

3.3 Klasyfikacja obiektów w nagraniach wizyjnych

Klasyfikacja typu obiektu jest niezbędnym elementem zaawansowanych systemów automatycznego monitoringu wizyjnego. Pozwala ona na wykrywanie zdarzeń w odniesieniu do poszczególnych kategorii obiektów (np. łamanie przepisów drogowych) oraz umożliwia definiowanie interakcji pomiędzy różnymi klasami obiektów (np. przechodzenie przez jezdnię w miejscu niedozwolonym). Algorytmy klasyfikacji muszą być niezależne od pozycji obiektu i uwzględniać związany z tym fakt zmiany wyglądu obiektu obserwowanego przez stacjonarną kamerę.

Metody klasyfikacji obiektów można podzielić na dwie grupy. Do pierwszej z nich należą metody pozwalające określić ogólny typ obiektu (np. pojazd, osoba, przedmiot) [8]. Zostały one przedstawione w rozdziale 3.3.1. Druga grupa algorytmów pozwala na określenie podtypu w ramach znanej już kategorii. Z punktu widzenia niniejszej pracy istotne jest zagadnienie rozpoznawania typu pojazdu, więc to temu zagadnieniu poświęcono rozdział 3.3.2.

3.3.1 Klasyfikacja typu obiektu

Najbardziej oczywistą metodą rozróżnienia pojazdów od osób wydaje się porównanie rozmiarów obiektów. Jednak w praktyce metoda ta daje satysfakcjonujące wyniki tylko w określonych sytuacjach związanych z umieszczeniem kamery względem poruszających się obiektów, gdyż nie uwzględnia faktu zmian wielkości obiektów związanych z ich odległością od kamery. Niedogodność tą można zlikwidować poprzez kalibrację pola widzenia kamery, ale proces ten jest stosunkowo czasochłonny i musi być przeprowadzony niezależnie dla każdej kamery w systemie monitoringu [233][234]. Jednak nawet kalibracja nie pozwoli na uwzględnienie w procesie klasyfikacji zmian w wyglądzie obiektu związanych z jego obrotem (np. furgonetka widziana z boku ma istotnie różny współczynnik proporcji w porównaniu z jej ustawieniem przodem do kamery), na skutek

czego ten sam obiekt obserwowany przez różne kamery może charakteryzować się diametralnie różną sylwetką.

Opisywane w literaturze metody klasyfikacji typu obiektów można podzielić na trzy główne grupy w zależności od wykorzystywanych deskryptorów obrazu. Są to metody bazujące na parametrach kształtu obiektu, jego teksturze oraz rozwiązania analizujące charakter ruchu obiektu [22][244].

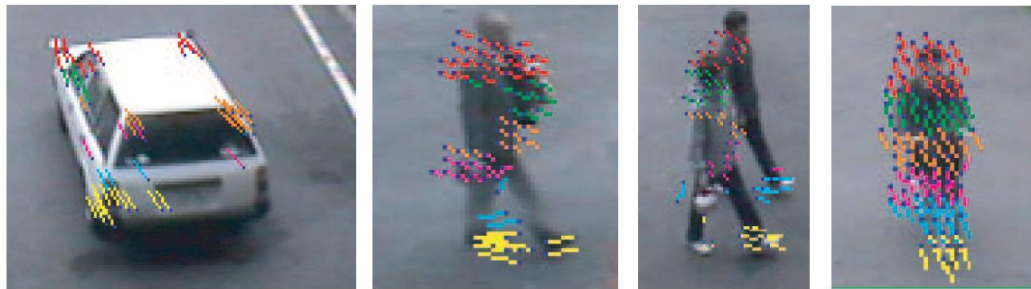
Najczęściej do zadań klasyfikacji obiektów wykorzystuje się kształt obiektu, który może być w praktyce reprezentowany przez ogromną liczbę parametrów geometrycznych i topologicznych. Do najpopularniejszych z nich należą współczynniki proporcji wysokości obiektu do jego szerokości, współczynniki wypełnienia (stosunek powierzchni prostokąta lub wielokąta wypukłego opisanego na kształcie obiektu do jego powierzchni), stosunki obwodu obiektu do jego powierzchni i inne parametry statystyczne kształtu [106]. Bardziej złożone metody wykorzystują parametryzację konturu obiektu w postaci kodu łańcuchowego (ang. *Chain Code*) lub histogramu geometrycznego (ang. *Pairwise Geometric Histogram*) [2][145] albo traktują kształt całościowo (np. momenty Zernicke’a [84][97]). Takie reprezentacje kształtu obiektu znalazły również zastosowanie w standardzie MPEG-7 [169].

Jako parametry tekstury obiektu na potrzeby jego klasyfikacji wykorzystywane są miary znajdujące krawędzie w sylwetce obiektu i grupujące je w zależności od ich kierunku. Można tego dokonać za pomocą wyznaczania gradientu [244] lub filtracji Gabora [94]. Prowadzone są także prace nad wykorzystaniem transformacji Fouriera obrazu obiektu do jego klasyfikacji [25].

Poza statyczną analizą obrazu obiektów, w procesie ich klasyfikacji można również wykorzystać parametry dynamiczne związane z charakterystyką i wzorcem ruchu obiektów danej klasy, szczególnie te związane z chodem człowieka [24][87], które pozwalają ponadto na identyfikację osób [157]. Do parametryzacji ruchu obiektu stosuje się zwykle metody wykorzystujące przepływ optyczny do znalezienia wariacji kierunków wektorów ruchu poszczególnych punktów charakterystycznych obrazu obiektu [125][244]. Przykładową ilustrację wektorów ruchu dla czterech kategorii obiektów pokazano na rys. 3.7.

Poza metodami klasyfikacji obiektów należącymi do wyżej wymienionych grup, spotyka się również inne rozwiązania. Do rozróżnienia osób od innych obiektów można wykorzystać fakt widoczności skóry, której kolor, szczególnie w połączeniu z innymi deskryptorami, może być efektywną cechą dystynktywną [12][99]. Ponadto łączenie

parametrów kształtu, tekstury i ruchu obiektów w jednym systemie pozwala uzyskać efekt synergii [54][124].



Rys. 3.7 Przykładowe wektory ruchu obiektów różnego typu, od lewej: pojazd, osoba, grupa osób oraz rowerzysta, obliczone w punktach charakterystycznych znalezionych za pomocą detektora KLT [244]. Punkt zaczepienia każdego wektora zilustrowano ciemną kropką, a kolor odpowiada lokalizacji wektorów w jednym z sześciu podobszarów, na które w pionie dzielony jest obraz w celu określenia stopnia zmienności wektorów w zależności od lokalizacji w obrazie

Klasyfikacja obiektów na podstawie ich wyglądu lub ruchu może być przeprowadzona w oparciu o reguły deterministyczne [12] lub za pomocą klasyfikatorów trenowanych obrazami wzorcowymi. W tym drugim przypadku zastosowanie znajdują sztuczne sieci neuronowe [37], maszyny wektorów nośnych SVM [53] lub klasyfikatory BOOST [248].

3.3.2 Określanie rodzaju pojazdu

Klasyczne rozwiązania zmierzające do wykrywania typu pojazdu bazują na zliczaniu liczby osi w pojeździe za pomocą pętli indukcyjnych lub innego rodzaju czujników montowanych w nawierzchni drogi lub w jej bezpośrednim sąsiedztwie [168]. Wadą takich rozwiązań jest jednak ich mała elastyczność (brak praktycznej możliwości przeniesienia detektora w inne miejsce) oraz niska skuteczność przy dużym natężeniu ruchu. Z tego względu popularność zyskują nieinwazyjne metody detekcji typu pojazdu za pomocą analizy obrazu z wykorzystaniem kamer wizyjnych. Umożliwiają one łatwy montaż, nie niszczą powierzchni drogi, mają stosunkowo duży zasięg działania oraz wysoką skuteczność.

Wszystkie obecnie spotykane metody klasyfikacji typu pojazdu na podstawie analizy obrazu bazują na takim samym, podstawowym schemacie. W ramach przetwarzania wstępnego dokonuje się detekcji i śledzenia ruchomych obiektów w obrazie kamery w celu uzyskania dokładnej lokalizacji wszystkich pojazdów. Na tej podstawie obliczane są

określone parametry obrazu pojazdów, które następnie z wykorzystaniem wybranego systemu decyzyjnego porównywane są ze wzorcami. Oznacza to również, że algorytmy takie wymagają treningu lub uprzedniego ręcznego zdefiniowania określonych zmiennych i warunków.

Metody klasyfikacji typu pojazdu mogą wymagać skalibrowanego pola widzenia kamery lub mogą działać bez konieczności przeprowadzenia kalibracji. Kalibracja wymaga zmierzenia szeregu parametrów związanych miejscem instalacji kamery i z jej kierunkiem patrzenia [33][95][152][222][253]. Pomiar tych parametrów jest podatny na błędy, a jednocześnie istotnie wpływa na końcowe wyniki klasyfikacji. Ponadto jakkolwiek zmiana w konfiguracji systemu wymaga ponownej kalibracji. Niekiedy możliwa jest autokalibracja, np. w oparciu o narysowane na powierzchni drogi linie oddzielające pasy ruchu [108]. Po określeniu parametrów kalibracyjnych, pierwszym krokiem działania algorytmu jest zwykle transformacja perspektywy obrazu z kamery [238], która pozwala (w ograniczonym stopniu) na zlikwidowanie różnic w wyglądzie pojazdów wynikających z ich odległości od kamery. Zwykle dąży się do tego, aby po przeprowadzeniu transformacji wszystkie pojazdy poruszały się na powierzchni gruntu, w kierunku prostopadłym do kierunku patrzenia kamery. Alternatywnie, dane kalibracyjne można wykorzystać do korekcji wyznaczonych uprzednio parametrów obrazu pojazdów.

Metody nie wymagające kalibracji pola widzenia kamery są wygodniejsze z punktu widzenia instalacji, ale jednocześnie wymagają stosowania innych, zazwyczaj bardziej zaawansowanych algorytmów.

Algorytmy klasyfikacji pojazdów można podzielić na dwie grupy. W skład pierwszej wchodzi rozwiązanie, które bezpośrednio bazują na wyznaczeniu wektora parametrów obrazu pojazdów i następnie na porównaniu go z bazą wzorców. Drugą grupę algorytmów stanowią metody wykorzystujące trójwymiarowe modele pojazdów. W trakcie klasyfikacji porównywane są parametry modelu zbudowanego w oparciu o bieżący obraz pojazdu z parametrami modeli referencyjnych. Najbardziej pasujący model determinuje klasę pojazdu. Te dwie grupy algorytmów są przedstawione w podrozdziałach 3.3.2.1 i 3.3.2.2.

Po wyznaczeniu wektora parametrów opisujących wygląd pojazdu konieczne jest porównanie tego wektora z wektorami wzorcowymi. Stosowane w tym celu algorytmy przedstawiono w podrozdziale 3.3.2.3. Osiągane wyniki klasyfikacji podsumowano w podrozdziale 3.3.2.4.

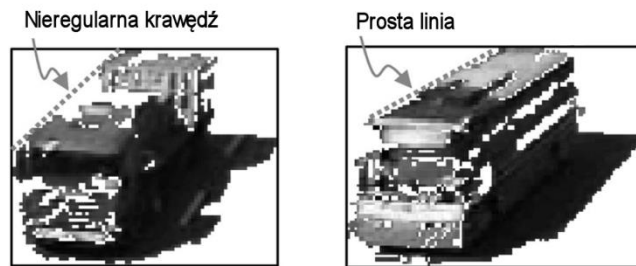
3.3.2.1 Metody bazujące na wyznaczeniu parametrów obrazu pojazdów

W literaturze pojawia się duża liczba parametrów, które są wykorzystywane do klasyfikacji typu pojazdu. Można je w ogólności podzielić na dwie grupy. W skład pierwszej wchodzi parametry wyznaczone jedynie w oparciu o maskę oznaczającą kształt pojazdu. Są to najczęściej:

- powierzchnia maski [102][177][178]
- wysokość i szerokość prostokąta opisanego na masce (może to być długość i wysokość pojazdu lub szerokość oraz kombinacja wysokości i długości pojazdu, w zależności od orientacji kamery) (rys. 3.8) [33][95][96][102][108][111][177][178]
- proporcja wysokości i szerokości prostokąta opisanego na masce [111]
- powierzchnia wielokąta wypukłego opisanego na masce [33][178]
- obwód wielokąta wypukłego opisanego na masce [33][177]
- parametry elipsy dopasowanej do maski (długość osi, ekscentryczność) [33][177][178]
- rozległość (stosunek powierzchni prostokąta otaczającego maskę do jej powierzchni) [33][111][177][178]
- masywność (stosunek powierzchni wielokąta wypukłego otaczającego maskę do jej powierzchni) [33][178]
- stosunek kwadratu obwodu maski do jej powierzchni [33][158]
- „liniowość” górnej krawędzi maski pojazdu [108] (rys. 3.9)
- pierwsze momenty statystyczne obrazu maski, w kierunkach poziomym i pionowym [102]
- niezmiennicze momenty statystyczne H_u maski pojazdu [156][191]



Rys. 3.8 W zależności od orientacji kamery względem pojazdów, wymiary prostokąta opisanego na masce pojazdu mogą oznaczać a) wysokość i długość pojazdu [177] lub b) szerokość oraz kombinację wysokości i długości pojazdu [191]

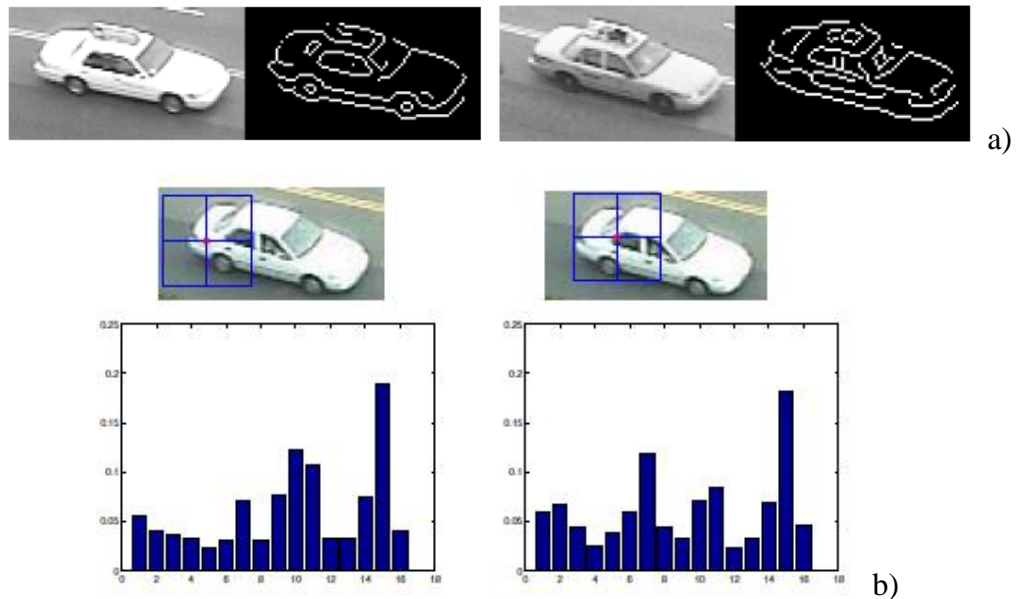


Rys. 3.9 Różnice w „liniowości” górnej krawędzi ciężarówki (z lewej) i autobusu (z prawej) [108]

Druga grupa parametrów obejmuje deskryptory bazujące na obrazie pojazdu i jego przekształceniach. Należą do nich:

- bezpośredni obraz pojazdu w postaci wartości kolejnych pikseli, zwykle znormalizowany do stałego rozmiaru [178][189]
- gradient obrazu luminancji (operator Sobela) w kierunku poziomym s_x i pionowym s_y [189]
- gradient znormalizowany bezpośrednio (ang. *Directly Normalized Gradient*)

$$\frac{s_x}{\sqrt{s_x^2 + s_y^2}}, \frac{s_y}{\sqrt{s_x^2 + s_y^2}}$$
 [189]
- gradient znormalizowany lokalnie (ang. *Locally Normalized Gradient*) $\frac{s_x}{g_L}, \frac{s_y}{g_L}$, gdzie g_L oznacza średnią wartość gradientu $\sqrt{s_x^2 + s_y^2}$ w sąsiedztwie $L \times L$ każdego piksela [189]
- gradient mapowany kwadratowo (ang. *Square Mapped Gradient*) $\frac{s_x^2 - s_y^2}{s_x^2 + s_y^2}, \frac{2s_x s_y}{s_x^2 + s_y^2}$ [189]
- wynik detekcji wierzchołków operatorem Harrisa [189]
- widmo fazowe transformacji Fouriera obrazu luminancji pojazdu [189]
- wektor elementów (x, y, o) , gdzie x i y oznaczają położenie punktu leżącego na krawędzi, a o – kierunek krawędzi; krawędzie i ich orientację znajduje się za pomocą progowania gradientu obrazu luminancji pojazdu [181]
- wektory zawierające sumy wartości gradientu obrazu pojazdu w wierszach i kolumnach [204]
- wektory SIFT (ang. *Scale Invariant Feature Transform*), zawierające histogramy orientacji krawędzi obrazu pojazdu w niewielkich podobszarach otaczających wybrane punkty charakterystyczne [71][163] (rys. 3.10)
- transformata obrazu pojazdu za pomocą filtru Gabora [75][123]



Rys. 3.10 Zastosowanie deskryptora SIFT do klasyfikacji typu pojazdu [18], a) obraz pojazdu i jego krawędzie wyznaczone w oparciu o detektor Canny'ego, b) histogramy orientacji krawędzi dla otoczenia dwóch różnych punktów charakterystycznych (oznaczonych na czerwono i położonych w środkach niebieskich kwadratów)

Z przedstawionego powyżej zestawienia wynika, że zdecydowanie częściej wykorzystywane są parametry obrazu pojazdu bazujące jedynie na opisie kształtu maski obiektu. Ponadto większość rozwiązań opisanych publikacjach wykorzystuje jedynie kilka wybranych parametrów.

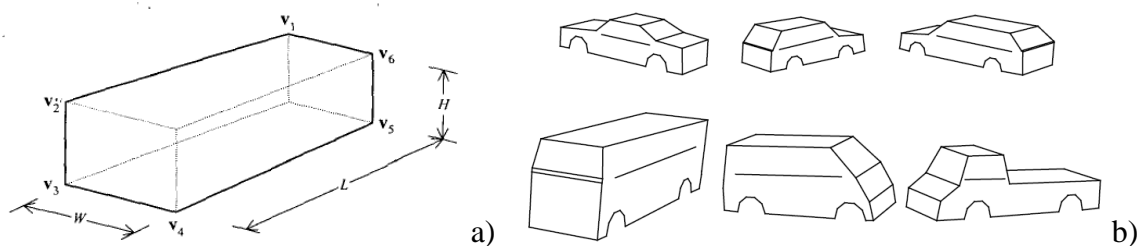
W celu ułatwienia działania algorytmów decyzyjnych, wektory parametrów są często poddawane redukcji liczby wymiarów za pomocą metod PCA lub LDA [178]. Metoda analizy składowych głównych PCA (ang. *Principal Component Analysis*) kładzie nacisk na zwiększanie separowalności pomiędzy wszystkimi wektorami pomiarowymi, bez względu na ich przynależność do poszczególnych klas. Z drugiej strony liniowa analiza dyskryminacyjna LDA (ang. *Linear Discriminant Analysis*) maksymalizuje wariancję pomiędzy wektorami należącymi do różnych klas względem wariancji wewnątrz każdej z klas. Z tego względu analiza LDA jest preferowana w zastosowaniach związanych z klasyfikacją.

Istotnym problemem w zadaniu rozpoznawania typów pojazdów jest fakt, że liczebność obiektów poszczególnych klas pojawiających się w jednostce czasu jest istotnie różna (np. samochodów ciężarowych jest zwykle znacznie mniej niż osobowych). Może to stanowić utrudnienie w procesie wyboru wektorów uczących, ponieważ liczebność obiektów każdej klasy powinna być zbliżona. Rozwiązaniem tego problemu może być

algorytm „fuzzy c means” [177], który służy w tym przypadku do normalizacji każdej klasy tak, aby miały one taką samą liczbę wektorów wzorcowych.

3.3.2.2 Metody bazujące na modelach 3D pojazdów

W odróżnieniu od metod bazujących na parametrach wyznaczanych na podstawie obrazu, w metodach bazujących na modelach pojazdów wiedza gromadzona jest nie w postaci zestawu przykładowych reprezentantów każdej klasy, ale w postaci jednego, ogólnego, trójwymiarowego modelu dla każdej klasy. Modele te mają zwykle postać symetrycznej bryły (rys. 3.11). W procesie rozpoznawania, dla bieżącego obrazu pojazdu wyznaczany jest model, który następnie porównywany jest z modelami wzorcowymi; najbardziej podobny model determinuje klasę pojazdu.



Rys. 3.11 Przykładowe modele 3D pojazdów: a) najprostszy [152], b) bardziej złożone [222]

Najprostszy model może mieć postać prostopadłościanu [152] (rys. 3.11a). Pozwala to na uzyskanie dokładniejszej estymacji rzeczywistej wysokości, szerokości i długości pojazdu w porównaniu z bezpośrednimi pomiarami maski obrazu pojazdu. Bardziej skomplikowane modele wykorzystują wielościenne bryły symetryczne wzdłuż osi podłużnej pojazdu [20][33][221][222][255] (rys. 3.11b). Model taki parametryzowany jest poprzez podanie położenia (względem środka ciężkości modelu) i orientacji wszystkich krawędzi modelu. Podczas dopasowywania modelu do bieżącego obrazu pojazdu, poszczególne wierzchołki mogą zmieniać położenie w ściśle zdefiniowanym zakresie.

3.3.2.3 Algorytmy porównywania wektora parametrów z wzorcami

W celu określenia, do jakiej klasy należy obserwowany pojazd, stosuje się różne algorytmy decyzyjne. Stosunkowo często używane są reguły heurystyczne [95][96][111][152]. Na podstawie wyznaczonych doświadczalnie warunków odnoszących

się do wartości parametrów, klasyfikują one obiekt do jednej z grup łączących kilka klas, a następnie w obrębie tej grupy – do konkretnej klasy.

Drugim co do popularności algorytmem klasyfikacji są różnego rodzaju metody odległościowe, wykorzystujące iloczyn skalarny lub metrykę euklidesową [102][108][158][189][204]. W jednym z rozwiązań opisanych w pracy [123] wykorzystany jest klasyfikator minimalno-odległościowy, który wywodzi się z teorii Bayesa. W celu sklasyfikowania wektora obliczane są najpierw odległości euklidesowe tego wektora od średniego wektora wzorcowego dla każdej klasy, a następnie wektor jest przypisywany do klasy charakteryzującej się minimalną odległością. Inną miarą podobieństwa może być również współczynnik korelacji [221].

Rozwinięciem metody odległościowej jest metoda wkNN (ważonych k najbliższych sąsiadów). Waga klasy jest rozmytą miarą przynależności próbki do klasy, co pozwala zwiększyć odporność algorytmu na szum i elementy „odstające” w zbiorze wzorcowym. Waga dla każdej klasy jest określana jako suma podobieństwa k najbliższych wzorców należących do tej klasy. Miarą podobieństwa jest odwrotność odległości euklidesowej między wektorami [177][178].

Na potrzeby klasyfikacji stosuje się również probabilistyczne modele konstelacyjne (ang. *Constellation Models*) [163], powiązane zwykle z ekstrakcją parametrów. Model konstelacyjny ma za zadanie reprezentowanie klasy obiektu jako zbioru wielu części objętych wspólnymi, geometrycznymi ograniczeniami. Poszczególne części reprezentowane są przez rozkłady Gaussa.

Zastanawiające jest rzadkie wykorzystywanie inteligentnych systemów decyzyjnych na potrzeby klasyfikacji typów pojazdów. W literaturze spotyka się praktycznie tylko wielowarstwową, jednokierunkową sztuczną sieć neuronową [75][156][191][255], o strukturze typowej dla klasyfikatora i sporadycznie maszynę wektorów nośnych SVM [33]. Wyjątkiem jest tu praca Zhanga [261], w której wykorzystano kaskadę wielu rozmaitych algorytmów uczących się (kNN, ANN, SVM, Random Forest).

3.3.2.4 Podsumowanie

Przestawiane w literaturze rozwiązania istotnie się różnią pod względem liczby rozpoznawanych klas pojazdów. W minimalnym przypadku system rozpoznaje tylko dwa typy pojazdów: ciężarówki i pozostałe [96]. Zwykle liczba rozpoznawanych klas pojazdów waha się od 3 do 7. Pojazdy dzielone są ze względu na wielkość na: samochody osobowe,

małe busy, autobusy, furgonetki, małe samochody ciężarowe, duże samochody ciężarowe oraz ze względu na typ nadwozia: sedan, hatchback, SUV.

Na tym tle wyróżniają się dwie publikacje, w których opisane jest rozpoznawanie 20 [181] oraz 77 [189] klas pojazdów. W tych przypadkach odpowiadają one konkretnym modelom samochodów osobowych. Taką skuteczność umożliwił fakt, że jako obraz pojazdu wykorzystywane są zbliżenia frontu samochodów robione z wysokości 1-2 metrów (rys. 3.12). W praktyce nie jest możliwe uzyskanie takiego ujęcia z typowych kamer monitoringu.



Rys. 3.12 Przykładowe zdjęcia wykorzystywane w systemie rozpoznającym 77 modeli samochodów [189]

Zdecydowana większość opisanych w literaturze systemów klasyfikacji pojazdów osiąga skuteczność około $90\% \pm 2\%$. Najlepsze wyniki, odpowiednio 95% [123] i 98% [75], osiągnięto w przypadku wykorzystania jako parametrów wyników filtracji Gabora gradientu obrazu pojazdów. Wyniki te jednak trudno bezpośrednio porównywać ze sobą w sytuacji, gdy liczba klas pojazdów i liczba przeprowadzonych prób rozpoznawania różnią się w poszczególnych eksperymentach. W pierwszym przypadku zastosowano minimalno-odległościowy klasyfikatora Bayesa, a w drugim sztuczną sieć neuronową. Pierwszy system rozpoznaje 5 typów pojazdów: sedan, hatchback, furgonetka, autobus, ciężarówka (rys. 3.13), a drugi 4 (samochody osobowe traktowane są jako jedna klasa).



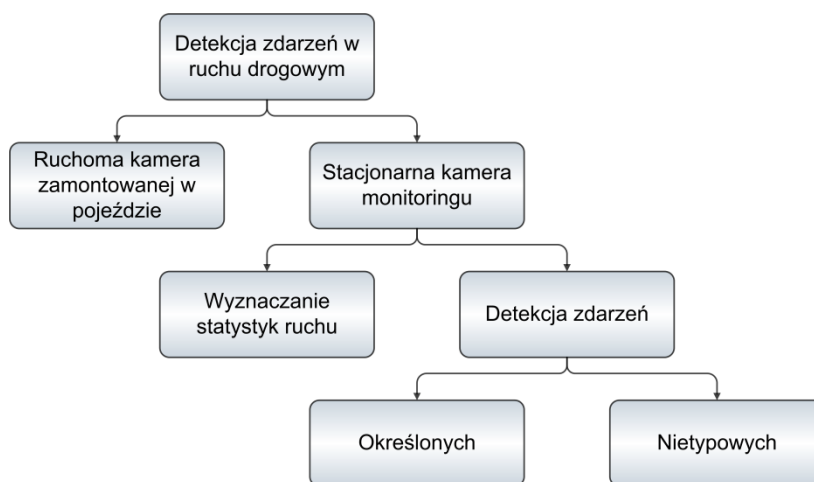
Rys. 3.13 Przykładowe zdjęcia pojazdów użyte w systemie, który osiągnął wysoką skuteczność klasyfikacji [123]

3.4 Wizualna detekcja zdarzeń w ruchu drogowym

Zagadnienie automatycznej detekcji zdarzeń w ruchu drogowym za pomocą analizy obrazu jest obszerne ze względu na dużą różnorodność zdarzeń możliwych do wykrywania oraz różne uwarunkowania praktyczne konkretnych implementacji [241], w tym konieczność analizy obrazu o niskiej jakości (wysokim stopniu kompresji) ze względu na jego transmisję kanałami o niskiej efektywnej przepływności [28].

W ogólności podziału metod analizy ruchu drogowego można dokonać w zależności od wykorzystywanej kamery: zamontowanej bezpośrednio w pojeździe lub stojącej przy drodze i będącej częścią systemu monitoringu (rys. 3.14). Automatyczna analiza obrazu drogi przed poruszającym się pojazdem sprowadza się do trzech głównych zastosowań: detekcji innych pojazdów poruszających się po drodze [223], wykrywania znaków drogowych [9] oraz śledzenia toru poruszania się pojazdu w odniesieniu do linii oddzielających pasy ruchu [170]. Istnieją również rozwiązania, w których kamera skierowana jest w stronę kierowcy w celu monitorowania jego skupienia uwagi [19]. Zagadnienia te, związane z wykorzystaniem kamery ruchomej wykraczają jednak poza zakres niniejszej rozprawy.

W dalszej części rozdziału przedstawiono rozwiązania dotyczące przetwarzania obrazu z kamer stacjonarnych. W tym zakresie można wyróżnić dwa typy zastosowań. Dotyczą one tworzenia statystyk ruchu drogowego na potrzeby optymalnego zarządzania nim służą do automatycznej detekcji różnorodnych zdarzeń (konkretnie określonych lub wszystkich nietypowych).



Rys. 3.14 Podział metod detekcji zdarzeń w ruchu drogowym

3.4.1 Wyznaczanie statystyk ruchu

Podstawowym przeznaczeniem algorytmów z tej grupy jest zliczanie pojazdów poruszających się po obserwowanym odcinku drogi, zwykle z podziałem na poszczególne pasy ruchu i kategorie pojazdów. Ma to na celu monitorowanie płynności ruchu drogowego poprzez pomiar jego średniej prędkości, natężenia oraz wykrywanie zatorów drogowych [107].

W związku z tym większość rozwiązań z tej grupy korzysta z dedykowanych metod. Powszechne jest stosowanie wirtualnych bramek na każdym z pasów ruchu w postaci czworokąta lub dwóch równoległych odcinków, w którym lub odpowiednio: pomiędzy którymi, zliczane są wszystkie przejeżdżające pojazdy i wyznaczana jest ich prędkość [30][150][151][154] (rys. 3.15). Detekcja pojazdów odbywa się zwykle za pomocą odejmowania tła [150][154][262] lub z wykorzystaniem modeli pojazdów zawierających informacje o geometrii ich kształtów oraz pozycji względem kamery [184][211], co oznacza konieczność wykonania jej kalibracji. W jednej z prac [171] odejmowanie tła uzupełnione jest detekcją krawędzi w celu ułatwienia usuwania cienia. Innym podejściem jest zastosowanie do wykrywania pojazdów wydajnego obliczeniowo algorytmu, który najpierw wyznacza ruchome punkty w obrazie za pomocą porównania za sobą trzech kolejnych ramek strumienia wizyjnego, znajduje te charakteryzujące się dużym kontrastem jako leżące na krawędziach obiektów oraz dokonuje przetwarzania morfologicznego w celu wyznaczenia sylwetek pojazdów [45]. Śledzenie pojazdów i określanie ich parametrów ruchu odbywa się z wykorzystaniem systemu opartego na wiedzy i na szeregu reguł.



Rys. 3.15 Zliczanie pojazdów za pomocą wirtualnej bramki [154]

Interesujące podejście zastosowano w eksperymentach opisanych w pracy [190], w których źródłem danych do analizy jest strumień wizyjny skompresowany algorytmem MPEG. Przedstawione rozwiązanie wykorzystuje bezpośrednio wektory ruchu zawarte w zakodowanym strumieniu w celu wyznaczania różnych faz płynności ruchu drogowego (od całkowitego zastoju, poprzez małą prędkość przy dużym natężeniu ruchu do dużej prędkości przy niewielkiej liczbie pojazdów).

Głównym problemem systemów wyznaczania statystyk ruchu drogowego jest wzajemne przesłanianie się poruszających się pojazdów. Dlatego w celu zredukowania tego problemu zwykle zalecane jest montowanie kamer wysoko nad jezdnią i kierowanie ich w dół na przejeżdżające pod nimi pojazdy; mogą się one poruszać wzdłuż pola widzenia kamery [30][154][159][171] lub w poprzek [151]. Przy mniej korzystnym umiejscowieniu kamery konieczne jest stosowanie dodatkowych rozwiązań algorytmicznych nakierowanych na śledzenie wzajemnie przesłaniających się pojazdów za pomocą pojedynczej kamery [184] lub z wykorzystaniem stereowizji [183].

Kluczową kwestią opisywanych systemów jest działanie w czasie rzeczywistym oraz minimalizacja zużycia energii i kosztów. Dlatego oprócz standardowych komputerów do przetwarzania obrazu stosuje się często procesory DSP [151][159].

3.4.2 Detekcja zdarzeń

Zdarzenia wykrywane w ruchu drogowym można podzielić według kryterium funkcjonalności na dwa rodzaje. Pierwsze z nich dotyczą detekcji konkretnych zdarzeń w określonych miejscach pola widzenia kamery, takich jak jazda pod prąd lub zawracanie w niedozwolonym miejscu. Drugą grupę obejmują wszystkie sytuacje, które są nietypowe dla obserwowanej sceny i które zakłócają normalny ruch pojazdów.

3.4.2.1 Wykrywanie określonych zdarzeń

Zdecydowana większość metod automatycznej detekcji zdarzeń w ruchu drogowym bazuje na analizie trajektorii poruszania się poszczególnych obiektów. Najczęściej wykrywanymi zdarzeniami w tej kategorii są zawracanie, zatrzymanie, jazda pod prąd, wjazd pojazdu na chodnik/wejście pieszego na ulicę, hamowanie, nadmierna prędkość i inne.

Wykrywanie poszczególnych zdarzeń realizowane jest różnymi metodami. W pracy [246] opisane jest wykrywanie przypadków zakręcania, zmiany pasa ruchu, zatrzymania pojazdu oraz zbyt wolnej lub za szybkiej jazdy. Śledzenie obiektów odbywa się za pomocą filtrów Kalmana, a zdarzenia definiowane są jako zbiór reguł czasowo-przestrzennych. Filtry Kalmana znajdują również zastosowanie w przypadku śledzenia toru ruchu pojazdów na skrzyżowaniu [34], do tworzenia semantycznego opisu poruszania się samochodów [127] oraz do analizy zachowań obiektów w oparciu o zbiór predefiniowanych reguł [149]. Ostatnie z wymienionych rozwiązań umożliwia wykrywanie niebezpieczeństwa kolizji pieszego i pojazdu oraz nieprawidłowego poruszania się pojazdu, związanego z zatrzymaniem się w niewłaściwym miejscu, ale wymaga uprzedniego zdefiniowania wszystkich rozpoznawanych scenariuszy oraz przygotowania informacji kontekstowej dotyczącej położenia obiektów zainteresowania w tle sceny (rys. 3.16).



Rys. 3.16 Położenie statycznych obiektów tła istotnych z punktu widzenia reguł detekcji zdarzeń [149]

Wykrywanie parkowania pojazdu, zatoru drogowego oraz kolizji przedstawiono w pracy [130], w której poszczególne zdarzenia definiowane są za pomocą ukrytych modeli

Markova (ang. *Hidden Markov Models*). Do wykrywania kolizji może być również wykorzystywana sztuczna sieć neuronowa [31], na której wejście podawane są ciągi czasowe z informacją o trajektorii ruchu obiektów. W podejściu tym sieć trenowana jest z wykorzystaniem informacji zwrotnej od operatora, który klasyfikuje przedstawiane mu zdarzenia jako istotne i nieistotne; kandydaci do prezentacji wyłaniani są za pomocą heurystycznego zestawu reguł.

W rozwiązaniu przedstawionym w pracy [201] zastosowano rzutowanie ramek wejściowych obrazu do przestrzeni jednowymiarowej w celu wykrywania zdarzeń zatrzymania się pojazdu. Podejście to ułatwia zagadnienie detekcji i śledzenia pojazdów oraz, przede wszystkim, znacznie redukuje zapotrzebowanie na moc obliczeniową.

Poza analizą sekwencji z typowego ruchu ulicznego, omawiane metody wykrywania zdarzeń znajdują również zastosowanie na lotnisku do analizy procesu transportu i ładowania/wyładowywania bagażu do/z samolotu [214] lub do wykrywania takich sytuacji jak osoba idąca po płycie lotniska, poruszający się samolot lub przekroczenie prędkości przez pojazd [147].

Wszystkie rozwiązania przedstawione w niniejszym fragmencie rozprawy wymagają fazy przygotowania przed ich praktycznym zastosowaniem w konkretnym miejscu. Przygotowanie to, w zależności od zastosowanego algorytmu, polega na dostarczeniu danych semantycznych dotyczących sceny obserwowanej przez kamery, zdefiniowaniu wartości parametrów dla reguł decyzyjnych lub przygotowaniu danych treningowych dla systemów uczących się.

3.4.2.2 Wykrywanie sytuacji nietypowych

Ogólniejszym podejściem w stosunku do wykrywania konkretnych zdarzeń drogowych jest detekcja wszelkich anomalii odbiegających od typowego ruchu pojazdów w analizowanej scenie. Metody z tej grupy wymagają wcześniejszego treningu poprzez obserwację sceny, lecz proces ten odbywa się automatycznie i nie wymaga wstępnych przygotowań oraz interakcji ze strony użytkownika.

W eksperymencie opisanym w pracy [70] zastosowano mapę zachowań bazującą na skumulowanych w czasie parametrach ruchu obiektów w analizowanej scenie i wykorzystującą algorytm GMM do detekcji ruchomych obiektów. Mapa ta jest tworzona automatycznie w fazie nauki. Po jej zakończeniu możliwe jest wykrywanie zdarzeń nietypowych, takich zdarzeń, jak: zawracanie lub jazda pod prąd w zachodzących w dowolnym miejscu pola widzenia kamery (rys. 3.17). Bardziej ogólne rozwiązanie

wykorzystuje ukryte modele Markowa i kryterium bayesowskie w celu wykrywania jakichkolwiek istotnych rozbieżności pomiędzy oczekiwanymi trajektoriami obiektów, a obserwowanymi [136]. Zbiór modeli Markowa jest również stosowany do wykrywania nietypowego ruchu pojazdów na skrzyżowaniu na podstawie analizy ich wektorów ruchu [137]. W tym przypadku brana jest pod uwagę faza sekwencji świateł ulicznych, co pozwala wykryć pojazdy łamiące zakaz wjazdu na skrzyżowanie.



Rys. 3.17 Przykładowy kadr obrazu (z lewej) i wygenerowana automatycznie mapa zachowań zawierająca skumulowane w czasie parametry ruchu obiektów (z prawej) [70]

4. Opracowanie algorytmu wykrywania i śledzenia obrazu ruchomych obiektów

Niniejszy rozdział przedstawia opracowane przez autora rozprawy algorytmy detekcji i śledzenia ruchomych obiektów wraz z oceną skuteczności ich działania.

4.1 Detekcja obiektów ruchomych

W rozdziale 3.1.6 przedstawiono porównanie metod detekcji obiektów ruchomych pod kątem ich zastosowania w monitorowaniu ruchu drogowego. Z przeprowadzonej analizy wynika, że najlepiej nadają się do tego zastosowania metody bazujące na odejmowaniu tła. Wobec tego na potrzeby detekcji ruchomych obiektów w obrazie z kamery zdecydowano się zaimplementować metodę modelującą tło sceny za pomocą ważonej sumy rozkładów gaussowskich. Metoda ta jest odporna na zmiany w oświetleniu i na bieżąco modyfikuje model tła tak, aby uwzględniał on wszelkie długoterminowe zmiany pojawiające się w tle sceny, np. obiekty pierwszoplanowe, które pozostają nieruchome przez dłuższy okres, stają się częścią tła. Ponadto model tła może mieć rozkład wielomodowy. Sprawia to, że kolory pikseli stanowiących część tła mogą podlegać regularnym, periodycznym zmianom, co umożliwia modelowanie takich zdarzeń jak kołysanie się gałęzi drzew na wietrze lub sekwencyjne zmiany barw świateł ulicznych.

Zaimplementowany algorytm bazuje w swoich podstawowych założeniach na rozwiązaniu zaproponowanym przez Stauffera i Grimsona [217], w stosunku do którego wprowadzono istotne modyfikacje dotyczące algorytmu adaptacji oraz estymowania modelu tła, które mają na celu uzyskanie lepszych wyników oraz zmniejszenie złożoności obliczeniowej. Ponadto dodano warstwę przetwarzania, której zadaniem jest kompensacja nagłej zmiany warunków oświetleniowych. Uzupełnieniem metody jest algorytm wykrywania i eliminacji cieni obiektów. Jego stosowanie jest praktycznie niezbędne dla uzyskania dokładnych rezultatów. Wynikowe maski binarne zawierające piksele ruchomych obiektów są przetwarzane morfologicznie w celu uzyskania zadowalającej dokładności segmentacji ruchomych obiektów. Kolejne etapy działania algorytmu detekcji ruchomych obiektów, prowadzące do ich ekstrakcji ze źródłowych ramek obrazu, zilustrowano na rys. 4.3.

4.1.1 Modelowanie tła

Każdy piksel obrazu jest modelowany niezależnie, za pomocą ważonej sumy K rozkładów gaussowskich [49][50][55], tzw. model mieszanin gaussowskich (ang. GMM - *Gaussian Mixtures Model*). Prawdopodobieństwo, że dany piksel ma wartość \mathbf{x}_t w czasie t jest dane wzorem:

$$p(x_t) = \sum_{i=1}^K w_t^i \eta(\mathbf{x}_t, \boldsymbol{\mu}_t^i, \boldsymbol{\Sigma}_t^i) \quad (4.1)$$

gdzie w_t^i oznacza wagę, $\boldsymbol{\mu}_t^i$ i $\boldsymbol{\Sigma}_t^i$ są wektorem wartości średnich i macierzą kowariancji i -tego rozkładu w czasie t , a η jest funkcją gęstości prawdopodobieństwa rozkładu normalnego zdefiniowaną wzorem:

$$\eta(\mathbf{x}_t, \boldsymbol{\mu}_t^i, \boldsymbol{\Sigma}_t^i) = \frac{1}{(2\pi)^{0,5 \cdot C} \sqrt{|\boldsymbol{\Sigma}_t^i|}} e^{-0,5 \cdot (\mathbf{x}_t - \boldsymbol{\mu}_t^i)^T \cdot \boldsymbol{\Sigma}_t^{i-1} \cdot (\mathbf{x}_t - \boldsymbol{\mu}_t^i)} \quad (4.2)$$

gdzie C oznacza liczbę składowych opisujących kolor piksela; dla wykorzystanej przestrzeni barw RGB C wynosi 3.

Liczba rozkładów K przyjmuje zwykle niewielką wartość z zakresu od 3 do 5 i zależy ona od dostępnej mocy obliczeniowej. Dla uproszczenia obliczeń i w celu zmniejszenia zapotrzebowania na pamięć przyjęto, że składowe RGB koloru piksela są niezależne. Jednak w przeciwieństwie do rozwiązania przedstawionego w [217] nie założono, że wariancje składowych koloru muszą być identyczne. W związku z tym macierz kowariancji $\boldsymbol{\Sigma}_i$ jest macierzą diagonalną o postaci:

$$\boldsymbol{\Sigma}_i = \begin{bmatrix} \sigma_{Ri}^2 & 0 & 0 \\ 0 & \sigma_{Gi}^2 & 0 \\ 0 & 0 & \sigma_{Bi}^2 \end{bmatrix} \quad (4.3)$$

gdzie σ_{Ri}^2 , σ_{Gi}^2 i σ_{Bi}^2 oznaczają wariancje składowych koloru RGB i -tego rozkładu.

Zakłada się, że każdy z rozkładów reprezentuje inny kolor piksela obrazu. Im częściej dany piksel przyjmuje określony kolor, tym większą wartość ma waga rozkładu związanego z tym kolorem.

Parametry w , $\boldsymbol{\mu}$ i $\boldsymbol{\Sigma}$ rozkładów każdego piksela są uaktualniane na bieżąco wraz z każdą kolejną ramką obrazu. Standardową metodą modyfikacji parametrów jest w takich przypadkach algorytm maksymalizacji wartości oczekiwanej EM (ang. *Expectation*

Maximization) [67]. Jednak jego implementacja byłaby bardzo nieefektywna i w praktyce niemożliwa, ponieważ wymaga on dostępu do wszystkich danych z przeszłości. Dlatego do uaktualniania parametrów rozkładów dla każdego piksela obrazu wykorzystano aproksymację wartości EM algorytm *on-line K-means approximation* [36].

W pierwszej kolejności, wszystkie rozkłady opisujące dany piksel obrazu są sortowane według malejącej wartości współczynnika r danego wyrażeniem:

$$r = \frac{w}{\sqrt{|\Sigma|}} \quad (4.4)$$

gdzie w jest wagą rozkładu, a $|\Sigma|$ oznacza wyznacznik macierzy kowariancji. Kolor piksela reprezentujący tło sceny zwykle częściej pojawia się w danych wejściowych, a zatem reprezentujący go rozkład charakteryzuje się niskimi wariancjami. W związku z tym, rozkład o największej wartości współczynnika r najwierniej reprezentuje kolor tła sceny.

Wraz z pojawieniem się nowej ramki wejściowej obrazu sprawdzane jest, który rozkład najbardziej pasuje do bieżącej wartości piksela. Rozkłady są sprawdzane w kolejności malejącej wartości współczynnika r . Bieżący kolor piksela pasuje do rozkładu, jeśli każda jego składowa RGB leży w granicach $\pm\Delta$ odchyłeń standardowych od odpowiedniej wartości średniej. Zwykle przyjmuje się $\Delta = 2,5$. Jeśli żaden rozkład nie pasuje do bieżącego koloru, rozkład o najmniejszej wartości współczynnika r jest zastępowany nowym rozkładem o małej wadze, dużych wariancjach składowych RGB i wartościach średnich równych bieżącemu kolorowi piksela.

Wagi rozkładów są uaktualniane wg równania:

$$w_t = (1 - \alpha)w_{t-1} + \alpha M_t \quad (4.5)$$

gdzie M_t wynosi 1 dla pierwszego rozkładu pasującego do bieżącego piksela, a 0 – dla pozostałych rozkładów. Po modyfikacji, wagi rozkładów są normalizowane. Można zauważyć, że waga pierwszego pasującego rozkładu jest zwiększana, natomiast wagi pozostałych rozkładów są zmniejszane. Współczynnik α determinuje szybkość adaptacji modelu tła. Im jest on większy, tym szybciej model może się dostosowywać do zmian w tle analizowanej sceny (np. spowodowanych stopniową zmianą warunków oświetleniowych). Jednak istnieje przy tym ryzyko, że pewne obiekty pierwszoplanowe, które pozostają nieruchome przez dłuższy okres (np. pojazdy oczekujące na skrzyżowaniu) staną się zbyt szybko częścią tła.

W przypadku, gdy został znaleziony rozkład pasujący do bieżącej wartości piksela, jego wartości średnie i wariancje są modyfikowane zgodnie z zależnościami:

$$\boldsymbol{\mu}_t = (1 - \alpha)\boldsymbol{\mu}_{t-1} + \alpha \cdot \mathbf{x}_t \quad (4.6)$$

$$\boldsymbol{\sigma}_t^2 = (1 - \alpha)\boldsymbol{\sigma}_{t-1}^2 + \alpha \cdot (\mathbf{x}_t - \boldsymbol{\mu}_t)^2 \quad (4.7)$$

Dla uproszczenia obliczeń, zamiast współczynnika bazującego na bieżącej wartości funkcji gęstości prawdopodobieństwa opisanego w literaturze [217], we wzorach (4.6) i (4.7) użyto współczynnik szybkości adaptacji α tożsamy ze współczynnikiem ze wzoru (4.5). Ponadto ograniczono od dołu wartości, jakie mogą przyjmować wariancje składowych koloru RGB; nie mogą być one mniejsze, niż $\boldsymbol{\sigma}_{\text{MIN}}^2$. Pozwala to uniknąć niestabilności w obszarach sceny, które pozostają statyczne przez dłuższy czas.

Tylko D pierwszych rozkładów i -tego piksela w czasie t , posortowanych wg malejącej wartości współczynnika r , jest wykorzystywanych do stworzenia modelu tła dla tego piksela. D jest zdefiniowane wg wzoru:

$$D_i^t = \arg \min_d \left(\sum_{j=1}^d w_j^t > T \right) \quad (4.8)$$

Próg T wyznacza, jaka część rozkładów modelujących kolor piksela zostanie uznana za bieżący model tła. Jeśli T jest małe to model tła ma przeważnie rozkład jednomodowy. Jeśli T jest większe, rozkład modelu tła może być wielomodowy co oznacza, że dany piksel tła może być opisany większą liczbą kolorów. Pozwala to prawidłowo modelować pewne dynamiczne zjawiska, jak np. flagi kołyszące się na wietrze, które z założenia stanowią część tła sceny.

Jeśli bieżący piksel nie pasuje do żadnego z pierwszych D rozkładów tworzących model tła, to jest on uznawany za część obiektu ruchomego. Na rys. 4.3a-c pokazano przykładowy wynik odejmowania tła.

Przedstawiony powyżej algorytm charakteryzuje się zadowalającą skutecznością działania w większości praktycznych przypadków. Jednakże jest on wrażliwy na nagłe zmiany warunków oświetleniowych. Sytuacja taka prowadzi do czasowego „oślepienia” algorytmu (praktycznie wszystkie piksele ramki są wówczas wykrywane jako należące do ruchomych obiektów). Stan ten trwa do czasu uaktualnienia modelu tła do nowego wyglądu sceny i może trwać od kilku do kilkudziesięciu sekund, w zależności od wartości

współczynnika szybkości adaptacji α i liczby ramek obrazu na sekundę w źródłowym strumieniu wizyjnym. Nagłe zmiany oświetlenia występują szczególnie w przypadku monitorowania terenów zewnętrznych i są powodowane głównie przez chmury czasowo przesłaniające słońce. Dlatego algorytm detekcji obiektów ruchomych został uzupełniony o dodatkowy poziom analizy mający na celu wykrywanie takich sytuacji i odpowiednie na nie reagowanie.

Błędy wykrywania obiektów ruchomych w obecności raptownych zmian oświetlenia wynikają z faktu, że proces adaptacji modelu tła przebiega zbyt wolno, aby mógł na bieżąco kompensować zachodzące zmiany. Jednym ze sposobów poradzenia sobie z tym problemem może być zastosowanie dynamicznie zwiększanej wartości współczynnika szybkości adaptacji α [224]. Jednak takie podejście może również prowadzić do niedokładności, gdyż parametr α wpływa na tempo adaptacji wszystkich parametrów (w , μ i Σ) pojedynczego rozkładu każdego piksela, co w przypadku długotrwałych zmian oświetlenia może doprowadzić do zakłócenia hierarchii najważniejszych rozkładów (wg wartości współczynnika r) ze względu na rosnące wartości wariancji. Problem ten można częściowo zniwelować poprzez zastosowanie odmiennych współczynników do adaptacji wartości średnich i wariancji rozkładów [29]. Inna modyfikacja algorytmu GMM zakłada uwzględnienie sąsiedztwa poszczególnych pikseli przy adaptacji parametrów ich rozkładów, co prowadzi do przyspieszenia procesu konwergencji [250]. Z kolei Kaewtrakulpong i Bowden [128] zaproponowali użycie różnych równań aktualizacji modelu tła dla innych etapów detekcji ruchomych obiektów, co pozwala na szybszą i dokładniejszą adaptację modelu tła do zmiennych warunków.

Proponowana modyfikacja algorytmu GMM ma zastosowanie do scen z częstymi zmianami warunków oświetleniowych i bazuje na obserwacji, że zmiany oświetlenia mają charakter płynny, biorąc pod uwagę różnice międzyramkowe. W takich przypadkach algorytm uaktualniania tła powinien się charakteryzować wysokim tempem adaptacji i, w związku z tym, mniejszą czułością związaną z wykrywaniem obiektów ruchomych. Osiągnięto to poprzez wprowadzenie dodatkowego etapu przetwarzania, w którym wybrane regiony modelu tła są modyfikowane. W celu wytypowania tych regionów, estymowane jest tempo zmiany wartości każdego piksela V_i , w porównaniu z ramką poprzedzającą, zgodnie z równaniem:

$$V_t = (1 - \gamma)V_{t-1} + \frac{1}{C}\gamma \sum_{c=1}^C |x_t(c) - x_{t-1}(c)| \quad (4.9)$$

gdzie C oznacza liczbę składowych koloru opisujących każdy piksel, $x_t(c)$ i $x_{t-1}(c)$ oznaczają wartość składowej koloru c dla piksela odpowiednio w ramce bieżącej i poprzedniej, a $\gamma \in (0,1)$ oznacza współczynnik nauki tempa zmian. Następnie wartości tempa zmian V_t dla każdego piksela są porównywane z założoną wartością progową T_V w celu wykrycia tych regionów obrazu, które charakteryzują się niewielką zmiennością. W rezultacie, dla wszystkich pikseli spełniających ten warunek i jednocześnie zakwalifikowanych jako część tła sceny, przeprowadza się dodatkową aktualizację ich modelu tła, polegającą na modyfikowaniu średniej wartości rozkładu charakteryzującego się największą wartością współczynnika r zgodnie ze wzorem:

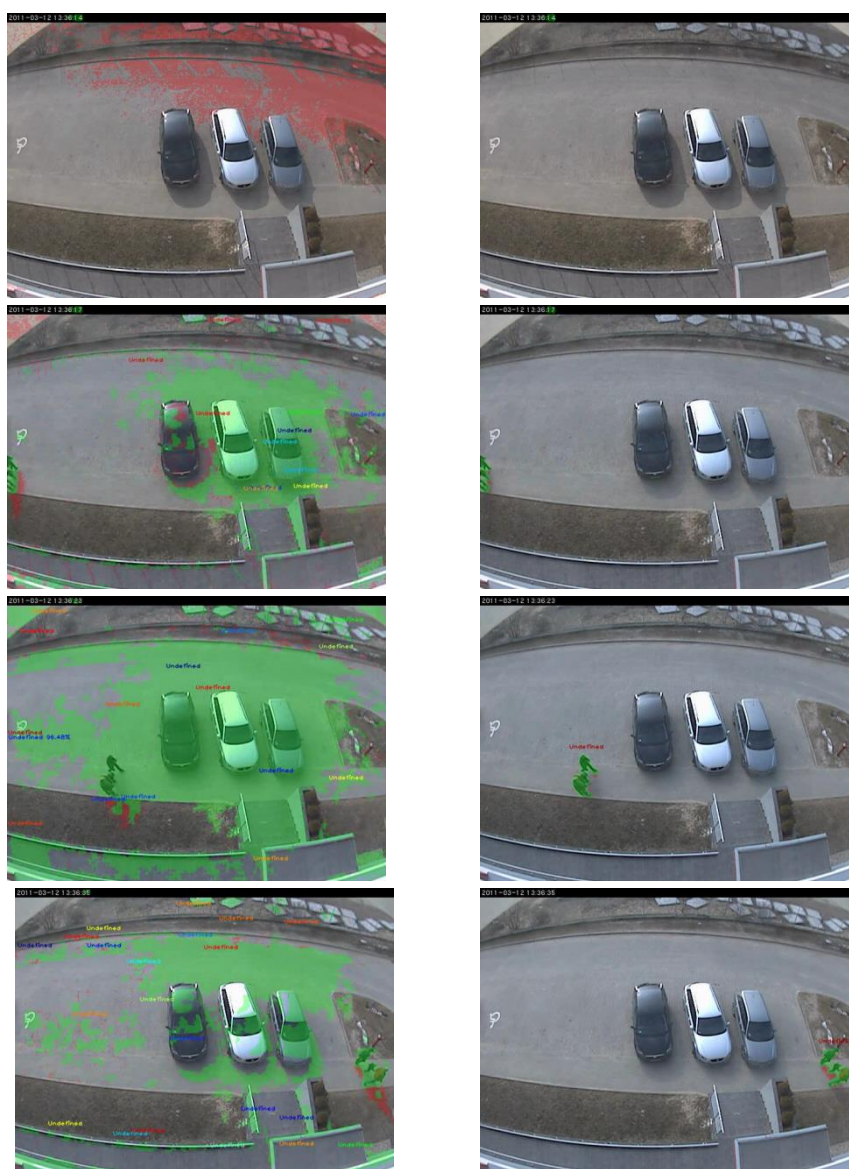
$$\boldsymbol{\mu}_t^{\max_r} = \boldsymbol{\mu}_{t-1}^{\max_r} + \delta(\mathbf{x}_t - \boldsymbol{\mu}_{t-1}^{\max_r}) \quad (4.10)$$

gdzie $\boldsymbol{\mu}_t^{\max_r}$ i $\boldsymbol{\mu}_{t-1}^{\max_r}$ oznaczają wektory wartości średnich rozkładu o największej wartości współczynnika r w bieżącej i poprzedniej ramce obrazu, \mathbf{x}_t oznacza bieżącą wartość piksela, a współczynnik $\delta \in (0,1)$ definiuje tempo dostosowywania wartości średniej najbardziej pasującego rozkładu do bieżącej wartości piksela. Ta operacja nie modyfikuje pozostałych parametrów żadnego z rozkładów, wobec czego wartości współczynników r dla wszystkich rozkładów piksela pozostają niezmiennie.

Przedstawione rozwiązanie pozwala na adaptację modelu tła do naturalnych zmian oświetlenia. Istnieje jednak ryzyko, że obiekt rzeczywisty, którego barwa jest na tyle zbliżona do modelu tła, że spełnia warunek niewielkiej zmienności wartości pikseli z ramki na ramkę, spowoduje rozstrojenie modelu. Skutkować to może wieloma błędami pierwszego rodzaju po ustąpieniu obiektu. W związku z tym, w toku eksperymentów prowadzonych przez autora rozprawy, wprowadzono dodatkowy, niezależny od pozostałych rozkład gaussowski, którego zadaniem jest dalsze zmniejszenie wrażliwości zmodyfikowanego algorytmu GMM. Rozkład ten jest uaktualniany zgodnie z równaniami (4.6) i (4.7), tylko w przypadku, gdy bieżący piksel został dopasowany do pierwszego rozkładu (charakteryzującego się największą wartością współczynnika r). W przypadku, gdy bieżący piksel nie zostaje dopasowany do żadnej z regularnych dystrybucji, sprawdzane jest, czy pasuje on do dystrybucji dodatkowej i tylko w przypadku negatywnej weryfikacji piksel jest klasyfikowany jako część obiektu ruchomego. W przeciwnym

przypadku, piksel traktowany jest jako należący do tła sceny, a pierwszy rozkład (o największej wartości współczynnika r) jest uaktualniany przy założeniu, że wartością obserwowaną piksela jest wartość średnia rozkładu dodatkowego.

Przykład działania oryginalnej wersji algorytmu GMM i wersji zmodyfikowanej pod kątem adaptacji do zmiennych warunków oświetleniowych zilustrowano na rys. 4.1. Ponadto na płycie DVD dołączonej do rozprawy umieszczono dwa filmy „gmm-mod-1.avi” i „gmm-mod-2.avi” porównujące działanie obu wersji algorytmu.



Rys. 4.1 Wybrane ramki z nagrania przetworzonego algorytmem GMM (lewa kolumna) oraz algorytmem z wprowadzoną przez autora rozprawę modyfikacją umożliwiającą reagowanie na nagłe zmiany oświetlenia (prawa kolumna); ramki pochodzą (odpowiednio od góry do dołu) z 6., 9., 15. oraz 27. sekundy nagrania testowego. Czerwony kolor oznacza regiony z wykrytym cieniem, a zielony – obszary wykryte jako zawierające piksele należące do ruchomych obiektów

W normalnych, stabilnych warunkach oświetleniowych, rozkład dodatkowy oraz najlepiej dopasowany rozkład spośród podstawowych zwykle są do siebie zbliżone, obejmując taki sam zakres wartości piksela. Jednak w przypadku zmian obie te dystrybucje się różnicują, co prowadzi do zmniejszenia wypadkowej czułości algorytmu.

4.1.2 Detekcja i usuwanie cienia

Istotnym elementem składowym wszystkich aplikacji z dziedziny monitoringu wizyjnego jest algorytm detekcji i eliminacji cieni obiektów. Cienie występują praktycznie zawsze, niezależnie od warunków pogodowych i jako poruszające się razem z obiektem pierwszoplanowym są klasyfikowane jako część tego obiektu. Wpływa to znacząco na pogorszenie dokładności segmentacji ruchomych obiektów.

Zaimplementowana metoda detekcji cieni wykorzystuje dwa kryteria. Przede wszystkim jasność zacienionego elementu tła znacząco maleje, podczas gdy jego barwa pozostaje praktycznie niezmienną. Po drugie, badana jest lokalna tekstura obrazu w otoczeniu zacienionych fragmentów sceny; powinna być ona niezmienna. Przykładowe wyniki detekcji i eliminacji cienia obiektów ruchomych pokazano na rys. 4.2 oraz rys. 4.3e. Ponadto na płycie DVD dołączonej do rozprawy zamieszczono filmy „S1-T1-C3-detekcja.avi” i „detekcja-ludzie.avi” ilustrujący wykrywanie ruchomych obiektów i ich cienia.

W celu określenia różnic w jasności i barwie pomiędzy bieżącym pikselem, a modelem tła autor zastosował w zaimplementowanym przez siebie algorytmie model koloru pozwalający na rozdzielenie wykorzystywanej przestrzeni barw RGB na składowe chrominancji i luminancji. Różnica w jasności b między bieżącym pikselem \mathbf{x} a kolorem \mathbf{e} opisywanym przez dany rozkład obliczana jest ze wzoru:

$$b = \arg \min_z (\mathbf{x} - z \cdot \mathbf{e})^2 \quad (4.11)$$

Różnica w jasności b jest wyrażona jako współczynnik skalujący wektora \mathbf{e} , który zapewnia minimalną odległość między obserwowanym pikselem \mathbf{x} i wektorem $b \cdot \mathbf{e}$ w przestrzeni kolorów RGB. Jeśli b jest mniejsze od jednościci to bieżący piksel jest ciemniejszy, w przeciwnym wypadku jest jaśniejszy od koloru modelowanego przez dany rozkład.

Różnica w barwie c jest zdefiniowana jako ortogonalna odległość między obserwowanym pikselem \mathbf{x} , a linią chrominancji, która przechodzi przez początek układu

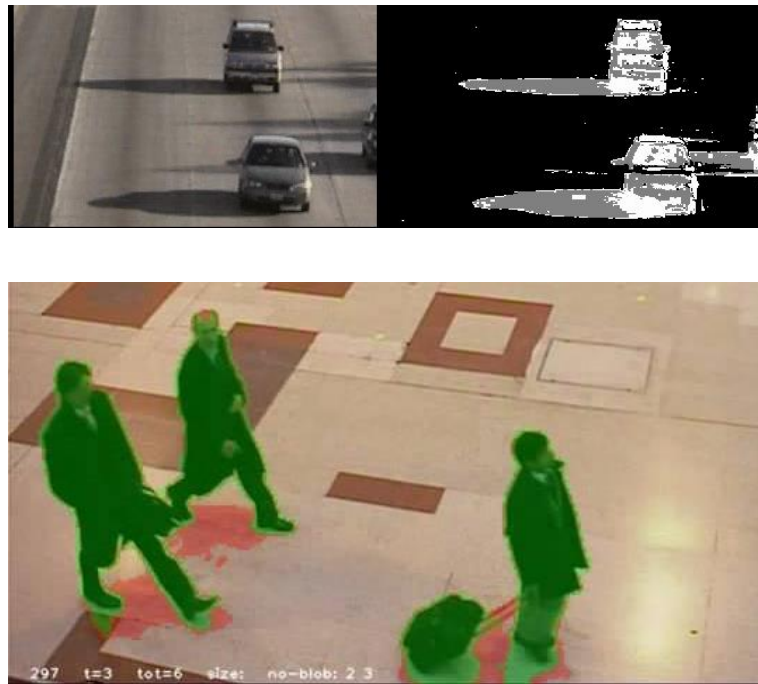
współrzędnych przestrzeni kolorów RGB i przez kolor modelowany przez dany rozkład. Różnica w barwie c jest dana wyrażeniem:

$$c = \|\mathbf{x} - b \cdot \mathbf{e}\| \quad (4.12)$$

W prezentowanym rozwiązaniu różnica w barwie może być zdefiniowana oddzielnie dla każdej składowej koloru jako:

$$c_{\text{RGB}} = \mathbf{x}_{\text{RGB}} - b \cdot \mathbf{e}_{\text{RGB}} \quad (4.13)$$

gdzie indeks RGB oznacza czerwoną, zieloną lub niebieską składową koloru.



Rys. 4.2 Przykładowe wyniki detekcji i eliminacji cienia obiektów ruchomych (szare obszary w prawej części górnego rysunku oraz czerwone obszary na dolnym rysunku)

Weryfikacji pod kątem obecności cienia podlegają tylko te piksele, które w wyniku odejmowania tła zostały sklasyfikowane jako należące do obiektu ruchomego. Jeśli dla przynajmniej jednego z D pierwszych rozkładów tworzących model tła oraz dla bieżącej wartości piksela:

- różnica w jasności b jest mniejsza od 1 (bieżący piksel jest ciemniejszy od koloru opisanego przez rozkład) oraz
- różnica w jasności jest większa niż próg T_b , oraz

- różnice w barwie c_{RGB} dla każdej składowej koloru mieszczą się w granicach 2.5 odchyłeń standardowych danego rozkładu,

to bieżący piksel jest uznawany za cień i jest klasyfikowany jako część tła sceny. Próg T_b będący minimalną wartością, jaką może przyjąć współczynnik b , ogranicza liczbę ciemnych pikseli błędnie rozpoznawanych jako cień. Ciemny piksel w przestrzeni kolorów RGB położony jest w pobliżu początku układu współrzędnych. Wszystkie linie chrominancji spotykają się również w początku układu współrzędnych. W związku z tym ciemny piksel znajduje się blisko każdej linii chrominancji.

Otrzymane wyniki detekcji cienia są walidowane poprzez porównanie tekstury dla bieżącej ramki obrazu oraz dla modelu tła. Jako miarę tekstury wykorzystano pochodną obrazu wyznaczoną za pomocą operatora Sobela. W pierwszym kroku dla całej ramki obrazu obliczany jest obraz różnicowy S zgodnie z zależnością:

$$S = |\text{sobel}(\ln(I)) - \text{sobel}(\ln(B))| \quad (4.14)$$

gdzie I oznacza bieżącą ramkę obrazu, a B – aktualny obraz modelu tła (wartości średnie rozkładu o największej wadze niezależnie dla każdego piksela).

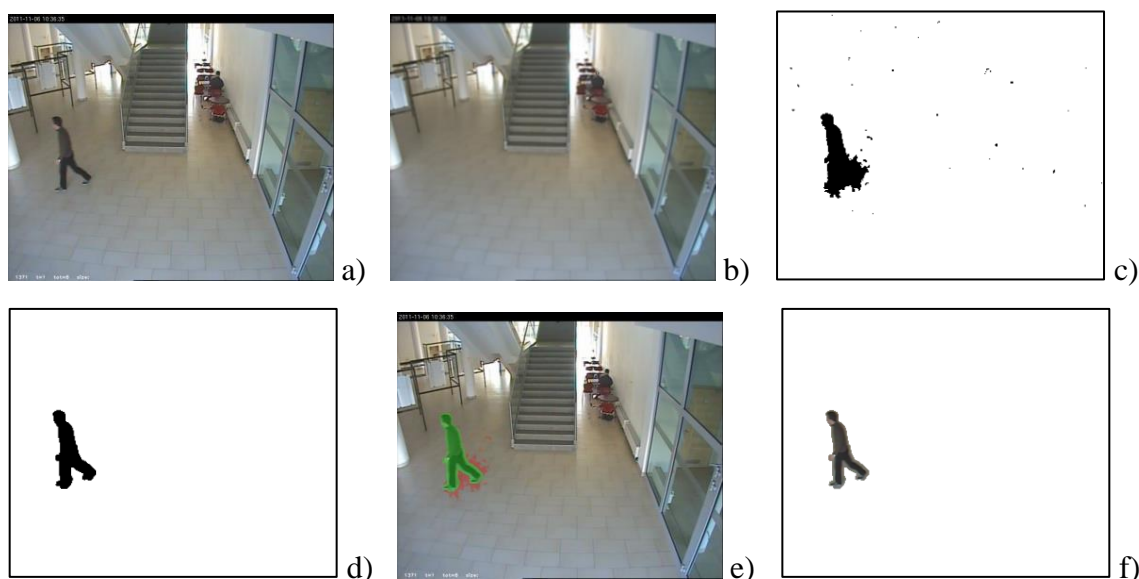
Dla każdego punktu obrazu zakwalifikowanego jako cień ruchomego obiektu obliczana jest miara podobieństwa między bieżącą ramką obrazu i modelem tła. Miara ta jest zdefiniowana jako suma wartości składowych koloru obrazu S dla wszystkich punktów w najbliższym sąsiedztwie analizowanego punktu, które zostały zakwalifikowane jako należące do ruchomych obiektów lub do ich cieni. Jeśli miara podobieństwa przekroczy założony próg, oznacza to, że detekcja cienia tylko na podstawie porównania jasności i barwy okazała się błędna, gdyż w tym samym czasie tekstura otoczenia analizowanego punktu zmieniła się w istotny sposób. W takim przypadku punkt obrazu, zakwalifikowany wcześniej jako cień, zostaje oznaczony jako należący do obiektu ruchomego.

4.1.3 Przetwarzanie morfologiczne

W wyniku modelowania tła otrzymuje się maskę binarną $\tilde{\mathbf{M}}$, której niezerowe elementy oznaczają piksele należące do wykrytych ruchomych obiektów znajdujących się w bieżącej ramce obrazu. Maskę tę musi być następnie przetworzona morfologicznie w celu umożliwienia dokładnej segmentacji obiektów [55][73]. Proces ten obejmuje znajdowanie wszystkich połączonych obszarów w masce, usuwanie obszarów zbyt małych (składających się z mniej, niż T_P pikseli), morfologiczne zamykanie oraz wypełnianie wszelkich „dziur” w znalezionych obszarach. Dodatkowo autor rozprawy

zaimplementował algorytm poprawiający skuteczność detekcji i usuwania cienia obiektów za pomocą morfologicznej rekonstrukcji [257] (rys. 4.3d).

Morfologiczna rekonstrukcja wykorzystuje dwa obrazy binarne: maskę \hat{M} i marker \tilde{M} ; marker \tilde{M} służy do oznaczenia obszarów do rekonstrukcji, znajdujących się w masce. Marker jest otrzymywany poprzez zastosowanie procedury eliminacji cienia, z wykorzystaniem algorytmu przedstawionego w poprzedniej sekcji. Operacja ta prowadzi do usunięcia wszystkich cieni, ale jednocześnie możliwe jest, że usuwana jest również część pikseli należących do ruchomych obiektów. Dodatkowo z markera kasowane są wszystkie pojedyncze, izolowane piksele. Postępowanie takie maksymalizuje jednak prawdopodobieństwo, że niezerowe elementy w markerze rzeczywiście odpowiadają położeniu pikseli należących do obiektów ruchomych.



Rys. 4.3 Kolejne etapy algorytmu detekcji obiektów ruchomych: a) oryginalna ramka obrazu z nagrania, b) model tła dla tej ramki (dla każdego piksela pokazano wartość średnią rozkładu charakteryzującego się najwyższą wagą), c) bezpośredni wynik odejmowania tła, bez przetwarzania morfologicznego i usuwania cienia, d) ostateczna maska binarna, e) wynik detekcji ruchomych obiektów: na zielono zaznaczono obiekt, na czerwono – jego cień, f) obraz ruchomego obiektu wyekstrahowany z ramki źródłowej

Maska \hat{M} powstaje jako suma logiczna markera \tilde{M} i maski S zawierającej jedynie piksele zakwalifikowanego jako cień przez algorytm jego detekcji, opisany w poprzednim rozdziale:

$$\hat{M} = \tilde{M} \cup S \quad (4.15)$$

Oznacza to, że niezerowe elementy maski $\hat{\mathbf{M}}$ tożsame są z położeniem w obrazie pikseli należących do ruchomych obiektów lub będących ich cieniem.

Maskę zrekonstruowaną \mathbf{M} uzyskuje się poprzez znalezienie części wspólnej maski $\hat{\mathbf{M}}$ i markera $\tilde{\mathbf{M}}$, który wcześniej poddano operacji dylatacji [73]:

$$\mathbf{M} = \hat{\mathbf{M}} \cap (\tilde{\mathbf{M}} \oplus \mathbf{E}) \quad (4.16)$$

gdzie \mathbf{E} jest jądrem filtracji, będącym kwadratowym elementem strukturalnym o rozmiarach 9×9 . W efekcie otrzymuje się wynikowy obraz binarny, w którym dokładnie usunięty jest cień obiektów, a uszkodzenia kształtu obiektów są zrekonstruowane.

4.2 Śledzenie ruchomych obiektów

Na podstawie porównania metod śledzenia obiektów ruchomych w strumieniu wizyjnym z pojedynczej kamery, przedstawionego w rozdziale 3.2.4, na potrzeby monitorowania ruchu drogowego zdecydowano się zaimplementować rozwiązanie łączące w sobie cechy metod pochodzących z różnych grup. Jako podstawę do śledzenia ruchu poszczególnych obiektów wybrano metodę statystyczną wykorzystującą filtry Kalmana. Cechuje ją niska złożoność obliczeniowa i wysoka skuteczność pod warunkiem, że nie następują konflikty śledzenia (wzajemne zasłanianie się obiektów lub ich częściowe znikanie za elementami tła sceny) [47][48]. Dlatego na potrzeby ich prawidłowego rozwiązywania zaimplementowano autorski algorytm z grupy metod dopasowania wzorca, który wykorzystuje cechy wyglądu obiektów do zachowania ciągłości śledzenia. W ten sposób bardziej złożone obliczenia, związane z wyglądem obiektów, ograniczone są przestrzennie i czasowo tylko do rozwiązywania konfliktów, a w sytuacjach jednoznacznych wykorzystywane jest bardziej wydajne śledzenie bazujące tylko na położeniu i rozmiarach obiektów. Dodatkowo wykorzystano szereg reguł deterministycznych dotyczących rozmiarów i prędkości ruchu obiektów w celu poprawy wyników działania. Kolejne kroki tej wieloetapowej procedury opisano w kolejnych podrozdziałach.

4.2.1 Zastosowanie filtrów Kalmana

W procesie śledzenia ruchomych obiektów, dla każdego wykrytego obiektu tworzony jest oddzielny filtr Kalmana, zwany nadążnikiem (ang. *tracker*), który ten obiekt reprezentuje. Filtr Kalmana jest wykorzystywany przede wszystkim do ustalania

poprawnych relacji pomiędzy wykrytymi obszarami należącymi do obiektów ruchomych w bieżącej ramce, a rzeczywistymi, ruchomymi obiektami znajdującymi się w analizowanej scenie. W przyjętym rozwiązaniu ruchomy obiekt jest reprezentowany przez prostokąt. W badaniach uwzględniono dwa rodzaje nadążników (filtrów Kalmana). W pierwszym z nich stan ruchomego obiektu (wektor \mathbf{x}^8) opisany jest ośmioma parametrami; stąd odpowiedni indeks w oznaczeniu wektora stanu:

$$\mathbf{x}^8 = [x \ y \ w \ h \ dx \ dy \ dw \ dh]^T \quad (4.17)$$

W drugiej wersji, wektor stanu \mathbf{x}^6 jest 6-cio elementowy:

$$\mathbf{x}^6 = [x \ y \ w \ h \ dx \ dy]^T \quad (4.18)$$

gdzie x i y oznaczają położenie obiektu (współrzędne środka prostokąta), w i h – szerokość i wysokość prostokąta, dx i dy – zmianę położenia obiektu (prostokąta) pomiędzy kolejnymi krokami czasowymi, a dw i dh – zmianę szerokości i wysokości obiektu (prostokąta) pomiędzy kolejnymi krokami czasowymi.

Nadążnik (*tracker*) z ośmioelementowym wektorem stanu różni się od trackera z wektorem sześćcioelementowym obecnością dodatkowych parametrów oznaczających dynamikę zmian rozmiarów obiektu. Oznacza to, że rozmiary obiektu (prostokąta) przy wykorzystaniu śledzenia z ośmioelementowym filtrem Kalmana mogą ulegać większym zmianom w krótszym czasie; w przypadku 6-ścio elementowego wektora stanu zmiana wymiarów obiektu jest możliwa jedynie na etapie aktualizacji tego wektora (równania aktualizacji pomiarów (3.10), (3.11) i (3.12)).

W obu przypadkach wektor pomiarowy \mathbf{z} ma postać:

$$\mathbf{z} = [x^b \ y^b \ w^b \ h^b] \quad (4.19)$$

Wektor ten zawiera dane o położeniu (x^b , y^b), szerokości w^b oraz wysokości h^b obszaru zawierającego piksele należące do ruchomego obiektu, który został powiązany z bieżącym trackerem.

W związku z tym, macierze przejścia \mathbf{A} i \mathbf{H} dla obu przypadków przedstawiają się następująco:

$$\mathbf{A}^8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{A}^6 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.20)$$

$$\mathbf{H}^8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{H}^6 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (4.21)$$

W zastosowanym modelu nie ma potrzeby korzystania z wejść kontrolnych, w związku z czym macierz wejściowa \mathbf{B} jest równa 0.

W eksperymentach macierze kowariancji szumu procesu \mathbf{Q} i kowariancji szumu pomiarowego \mathbf{R} były stałe i wynosiły:

$$\mathbf{Q} = \text{diag}(q) \quad \mathbf{R} = \begin{bmatrix} r_p & 0 & 0 & 0 \\ 0 & r_p & 0 & 0 \\ 0 & 0 & r_s & 0 \\ 0 & 0 & 0 & r_s \end{bmatrix} \quad (4.22)$$

gdzie operator $\text{diag}(q)$ oznacza macierz diagonalną z wartościami a na głównej przekątnej, a r_p i r_s są składowymi szumu wpływającymi na pomiar odpowiednio pozycji obiektu i jego rozmiarów.

Proces śledzenia obiektu ruchomego przebiega w kilku etapach. Po wykryciu nowego obiektu przypisywany jest mu nowy tracker, którego wektor stanu ustawiany jest w oparciu o zmierzone parametry obszaru, zgodnie z równaniem:

$$\mathbf{x}_{-1}^8 = [x_{-1}^b \quad y_{-1}^b \quad w_{-1}^b \quad h_{-1}^b \quad 0 \quad 0 \quad 0 \quad 0]^T \quad (4.23)$$

$$\mathbf{x}_{-1}^6 = [x_{-1}^b \quad y_{-1}^b \quad w_{-1}^b \quad h_{-1}^b \quad 0 \quad 0]^T \quad (4.24)$$

W następnym kroku czasowym (w kolejnej ramce) wektor stanu jest ponownie uaktualniany w oparciu o parametry obszaru odpowiadającego nowo powstałemu obiektowi:

$$\mathbf{x}_0^8 = [x_0^b \quad y_0^b \quad w_0^b \quad h_0^b \quad x_{-1}^b - x_0^b \quad y_{-1}^b - y_0^b \quad w_{-1}^b - w_0^b \quad h_{-1}^b - h_0^b]^T \quad (4.25)$$

$$\mathbf{x}_0^6 = [x_0^b \quad y_0^b \quad w_0^b \quad h_0^b \quad x_{-1}^b - x_0^b \quad y_{-1}^b - y_0^b]^T \quad (4.26)$$

W taki sposób przygotowany wektor \mathbf{x}_0 stanowi początkową estymatę wektora stanu $\hat{\mathbf{x}}_0$. Początkową estymatę błędu kowariancji \mathbf{P}_0 przyjmuje się równą:

$$\mathbf{P}_0 = p_0 \cdot \mathbf{ID} \quad (4.27)$$

gdzie \mathbf{ID} jest macierzą jednostkową.

W kolejnych krokach czasu, w pierwszej kolejności dokonuje się predykcji w przód wektorów stanu wszystkich filtrów Kalmana przypisanych do istniejących obiektów, w celu otrzymania estymaty *a priori* położenia obiektów w bieżącej ramce obrazu (równania aktualizacji czasu (3.8) i (3.9)). Następnie usuwane są wszystkie trackery, których estymata *a priori* wektora stanu zawiera niepoprawne wartości (zbyt małe lub ujemne wartości szerokości i wysokości obiektu – ta druga sytuacja możliwa w przypadku stosowania ośmioelementowego wektora stanu). Następnie odbywa się ustalanie, który region w bieżącej ramce obrazu jest powiązany z którym śledzonym obiektem. Na końcu następuje korekcja wektorów stanu filtrów Kalmana każdego obiektu w oparciu o pomiar parametrów odpowiadającego im obszaru zawierającego piksele należące do ruchomych obiektów, wykrytych w bieżącej ramce obrazu (równania aktualizacji pomiaru (3.10), (3.11) i (3.12)).

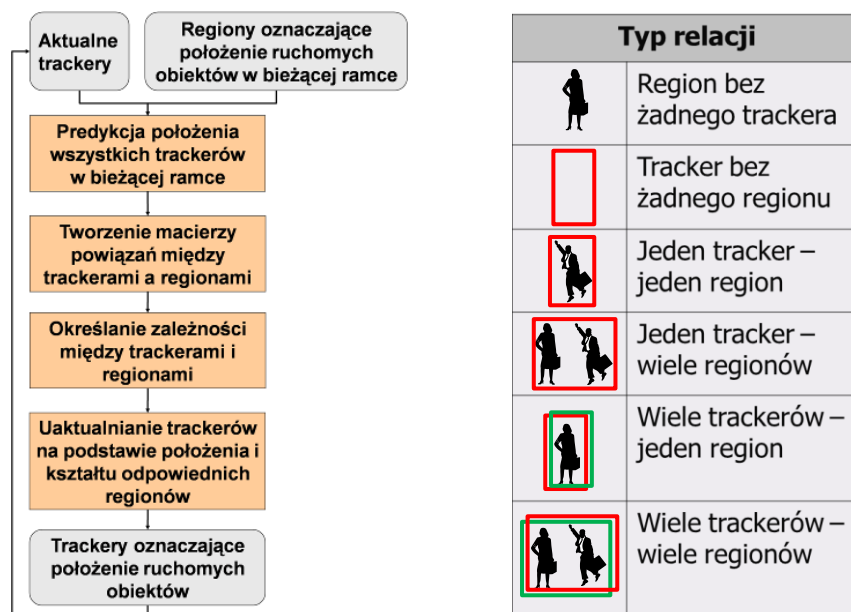
4.2.2 Ustalanie powiązania między ruchomymi obiektami a wynikami odejmowania tła

Kluczowym elementem algorytmu śledzenia ruchomych obiektów jest właściwe powiązanie istniejących trackerów z regionami będącymi wynikiem odejmowania tła w bieżącej ramce [230][235]. W tym celu tworzona jest binarna macierz powiązań \mathbf{L} , w której każdej parze tracker (estymata *a priori* stanu obiektu $\hat{\mathbf{x}}_k^-$) – region (wektor pomiarowy \mathbf{z}_k) przypisana jest wartość zero lub jeden, w zależności od tego, czy prostokąt otaczający region oraz pozycja obiektu estymowana przez tracker mają część wspólną (nachodzą na siebie). W efekcie dla i trackerów istniejących w bieżącej ramce oraz dla j wykrytych obszarów powstaje macierz powiązań \mathbf{L} o wymiarach $i \times j$.

Możliwych jest 6 typów relacji pomiędzy trackerami i regionami, które wymagają odmiennych działań. Zostały one zilustrowane na rys. 4.4.

Zastosowany sposób rozwiązywania konfliktów śledzenia wykorzystuje cechę filtrów Kalmana pozwalającą im estymować stan śledzonego obiektu, przy założeniu, że nie

zmieni on w sposób raptowny swojego wektora prędkości. W pierwszym kroku wydzielane są grupy wzajemnie ze sobą powiązanych trackerów i regionów; w każdej grupie znajdują się wszystkie regiony, które są skojarzone z przynajmniej jednym trackerem z grupy, i odwrotnie – wszystkie trackery, które są powiązane z przynajmniej jednym regionem z grupy. Osobne grupy tworzone są również dla każdego trackera nie powiązanego z żadnym regionem i dla każdego regionu nie skojarzonego z żadnym trackerem.



Rys. 4.4 Algorytm ustalania powiązań między trackerami a regionami będącymi wynikami odejmowania tła w bieżącej ramce (z lewej) oraz typy relacji uwzględniane w algorytmie powiązywania (z prawej)

Grupy te są następnie analizowane niezależnie. W przypadku, gdy w skład grupy wchodzi samotny tracker, to w bieżącej ramce nie jest wykonywana faza aktualizacji pomiaru tego trackera. Jeśli tracker ten nie zostanie powiązany z żadnym regionem przez pewną liczbę kolejnych ramek z rzędu (przyjęto wartość równą 5) to jest on usuwany. Predykcyjny charakter trackera sprawia, że nie jest „gubiony” obiekt ruchomy, którego detekcja metodą odejmowania tła jest czasowo niemożliwa (np. człowiek przechodzący za stałą przeszkodą). W przypadku, gdy w grupie znajduje się pojedynczy region, to jest tworzony nowy tracker inicjowany tym regionem pod warunkiem, że pole prostokąta opisanego na tym regionie jest większe niż T_R . Warunek ten zapobiega niepożądanemu tworzeniu się nadmiarowych trackerów w momencie, gdy śledzony obiekt czasowo (np. wskutek częściowego zasłonięcia przez obiekt tła) zostanie rozdzielony na wiele małych

regionów. Opracowaną metodę przedstawiono wcześniej w publikacjach opracowanych z udziałem autora rozprawy [64][235].

W obrębie każdej z pozostałych grup analizowane są kolejno wszystkie trackery. Sposób postępowania zależy od liczby regionów powiązanych z określonym trackerem:

- Jeśli tylko jeden region jest powiązany z trackerem:
 - Jeśli ten region jest skojarzony tylko z danym trackerem, to tracker ten jest uaktualniany w oparciu o parametry tego regionu; jest to najbardziej pożądany typ relacji (jeden-do-jednego)
 - Jeśli ten region jest skojarzony z więcej, niż jednym trackerem, to tracker jest uaktualniany w oparciu o jego estymowaną pozycję wewnątrz tego regionu (więcej na ten temat w następnym akapicie)
- Jeśli więcej, niż jeden region jest powiązany z danym trackerem:
 - Jeśli wszystkie te regiony są przypisane tylko do tego trackera, to tracker jest uaktualniany w oparciu o współrzędne położenia i wymiary prostokąta obejmującego łącznie wszystkie regiony; takie podejście ma zapewnić utrzymanie spójności śledzonego obiektu w przypadku, gdy w wyniku błędów w procesie odejmowania tła zostanie on rozdzielony na kilka regionów lub w sytuacji, gdy np. poruszający się samochód zostaje „rozdzielony” przez lampę uliczną.
 - Jeśli którykolwiek z tych regionów jest przypisany także do innego trackera, to tracker ten uaktualniany w oparciu o jego estymowaną pozycję wewnątrz połączonych regionów

Pozycja trackera wewnątrz regionu (lub grupy połączonych regionów) jest wyznaczana w oparciu o estymatę *a priori* stanu filtru Kalmana $\hat{\mathbf{x}}_k^-$ w bieżącej ramce obrazu oraz o stan filtru w poprzedniej ramce $\hat{\mathbf{x}}_{k-1}$, ponieważ w sytuacjach konfliktowych nie jest możliwy pomiar położenia i rozmiaru regionu. Wobec tego wektor pomiarowy \mathbf{z}_k jest konstruowany w oparciu o estymatę *a priori* pozycji obiektu w bieżącej ramce i o estymatę *a posteriori* rozmiaru obiektu w ramce poprzedniej, zgodnie z zależnością:

$$\mathbf{z}_k = [\hat{x}_k^- \quad \hat{y}_k^- \quad \hat{w}_{k-1} \quad \hat{h}_{k-1}] \quad (4.28)$$

Jeśli do konstrukcji wektora pomiarowego użyto by estymaty *a priori* rozmiarów obiektu w bieżącej ramce, mogłoby to doprowadzić do zniknięcia trackera lub jego ponadnormatywnego rozrośnięcia się w sytuacji, jeśli rozmiar obiektu ulegał zmianie w momencie zaistnienia sytuacji konfliktowej. Oznacza to, że założono niezmiennosc

rozmiarów obiektu w czasie trwania konfliktu. Założenie to jest zwykle nieprawdziwe w dłuższych okresach czasu, jednak dla większości sytuacji, w których konflikt trwa stosunkowo krótko (np. mijanie się grup osób) pozwala osiągnąć poprawne wyniki. Należy przy tym podkreślić, że założenie o niezmienności rozmiarów rzeczywistego obiektu nie oznacza, że rozmiar trackera pozostanie również niezmienny; może się on zmienić, gdyż w procesie aktualizacji pomiarów uwzględniany jest zarówno wektor pomiarowy, jak i bieżący stan filtru Kalmana.

Wyznaczony w przedstawiony sposób wektor pomiarowy \mathbf{z}_k jest korygowany w taki sposób, aby zapewnić, że znajduje się on całkowicie wewnątrz rozpatrywanego regionu (lub połączonej grupy regionów), a następnie wykorzystywany jest do uaktualnienia filtru Kalmana skojarzonego z trackerem. Na niezawodność opisanego rozwiązania wpływ mają: czas trwania sytuacji konfliktowej (im dłużej, tym gorzej), zachowanie poruszających się obiektów (nagle zmiany wektora prędkości prowadzą do błędnej estymacji pozycji trackera) i precyzja estymacji wektora stanu obiektu w momencie zaistnienia sytuacji konfliktowej (błędne warunki początkowe, np. uaktualnienie trackera w oparciu o częściowo widoczny region, prowadzą do niedokładności).

Przedstawiona do tej pory procedura rozwiązywania konfliktów może zawieść w przypadku rozdzielających się obiektów (np. osoba wysiadająca z samochodu), jeśli wartości szumu pomiarowego \mathbf{R} są wystarczająco małe (tracker mocno „ufa” pomiarom), a rozdzielenie obu obiektów ma charakter łagodny i powolny. W takim przypadku, oba obiekty będą stale reprezentowane przez jeden tracker o rozmiarach powiększających się w miarę, jak obiekty te oddalają się od siebie. Oczekiwanym zachowaniem jest w tym przypadku podążenie trackera za jednym z obiektów i stworzenie nowego trackera dla drugiego obiektu. Wobec tego, wprowadzono następującą modyfikację do algorytmu rozwiązywania konfliktów. Zgodnie z nim, w obrębie każdej grupy wzajemnie powiązanych ze sobą regionów i trackerów prowadzona jest kolejno analiza regionów powiązanych z każdym trackerem. Zmiana polega na podziale wszystkich tych regionów na podgrupy w zależności od ich wzajemnej odległości od siebie. Regiony tworzą jedną podgrupę, jeśli dla każdej pary regionów b_1 i b_2 w podgrupie spełniona jest nierówność:

$$\|\mathbf{c}_1 - \mathbf{c}_2\| \leq 0,25 \cdot \sqrt{\min(s_1, s_2)} \quad (4.29)$$

gdzie \mathbf{c}_1 i \mathbf{c}_2 oznaczają współrzędne środka prostokątów opisanych odpowiednio na regionach b_1 i b_2 , a s_1 i s_2 oznaczają pola powierzchni tych prostokątów.

Jeśli wszystkie regiony powiązane z określonym trackerem tworzą jedną podgrupę, dalsze postępowanie przebiega identycznie, jak w opisanej już procedurze. W przeciwnym wypadku zachodzi konieczność podziału trackera, czyli skojarzenia trackera z regionami jednej z podgrup, a następnie stworzenia nowych trackerów dla pozostałych regionów. W celu znalezienia podgrupy do powiązania z bieżącym trackerem wykorzystuje się cechy wizualne obiektu. Porównuje się w parach wygląd regionu w każdej z analizowanych podgrup z wyglądem obiektu zapisanym w trackerze. Wygląd regionu pochodzi z bieżącej ramki obrazu. Wygląd obiektu jest zapisywany w trackerze w momencie jego utworzenia i uaktualniany jest z każdą ramką obrazu, w której nie zachodzą konflikty śledzenia (występuje relacja jeden-do-jednego).

Jako stopień podobieństwa wyglądu obiektu zastosowano miarę integrującą dwuwymiarowy histogram koloru oraz parametry tekstury bazujące na macierzach współwystępowania.

Dwuwymiarowy histogram koloru wykorzystuje chromatyczną przestrzeń barw R_cG_c . W implementacji użyto histogram o wymiarach 64×64 , który pozwala uzyskać praktycznie taką samą precyzję działania, co pełen histogram o wymiarach 256×256 przy jednocześnie 16-krotnym zmniejszeniu ilości danych. Dla piksela o składowych koloru RGB, wartości R_c i G_c w chromatycznej przestrzeni barw można wyznaczyć następująco:

$$R_c = \frac{R}{R + G + B} \quad G_c = \frac{G}{R + G + B} \quad (4.30)$$

czyli wartości R_c i G_c reprezentują wielkość udziału składowej czerwonej i zielonej w oryginalnym kolorze. Łatwo też zauważyć, że wyznaczanie w analogiczny sposób udziału koloru niebieskiego jest zbędne, gdyż te trzy udziały dopełniają się do jedności.

Zdefiniowana w powyższy sposób przestrzeń barw jest stratna (nie można powrócić do pełnej przestrzeni barw RGB). Korzyścią jest fakt, że nie zawiera ona informacji o luminancji co sprawia, że obiekt poruszający się po nierównomiernie oświetlonej scenie ma w przestrzeni barw R_cG_c identyczny wygląd (np. dla $R_c = 1/3$ i $G_c = 1/3$ wiadomo, że udział każdej składowej koloru w oryginalnej przestrzeni barw RGB jest taki sam, nie sposób jednak powiedzieć, czy jest to kolor czarny, szary czy biały). Kolejną zaletą tej przestrzeni barw jest prostota jej wyznaczania oraz mniejszy rozmiar wyznaczonego na jej podstawie histogramu.

Stopień podobieństwa S_H pomiędzy wyglądem obiektu (histogramem R_cG_c) zapisanym w trackerze oraz wyglądem (histogramem R_cG_c) rozpatrywanego regionu jest wyznaczany za pomocą miary korelacji.

Drugą składową miary podobieństwa wyglądu obiektów stanowią parametry statystyczne macierzy współwystępowania (ang. *Co-Occurence Matrix*). Macierz współwystępowania C zdefiniowana dla obrazu I o rozmiarach $n \times m$ i dla przesunięcia $(\Delta x, \Delta y)$ dana jest wzorem:

$$C(i, j) = \sum_{p=1}^n \sum_{q=1}^m \begin{cases} 1, & \text{gdy } I(p, q) = i \text{ oraz } I(p + \Delta x, q + \Delta y) = j \\ 0, & \text{w przeciwnym wypadku} \end{cases} \quad (4.31)$$

gdzie $C(a, b)$ oraz $I(a, b)$ oznaczają element o współrzędnych (a, b) odpowiednio macierzy współwystępowania C oraz obrazu I .

Macierz współwystępowania określa rozkład występowania poszczególnych par kolorów obrazu w określonej odległości przestrzennej $(\Delta x, \Delta y)$. Macierz taką zamienia się następnie na symetryczną, poprzez dodanie do niej jej własnej transpozycji. Taka symetryczna macierz współwystępowania jest identyczna dla przesunięć symetrycznych względem $(0,0)$, np. $(-1,0)$ i $(1, 0)$, lub $(-1, 1)$ i $(1, -1)$. W ostatnim kroku macierz ta jest normowana poprzez podzielenie każdego jej elementu przez sumę wszystkich elementów.

Macierz współwystępowania jest zależna od obrotu ze względu na parametr przesunięcia $(\Delta x, \Delta y)$. Dlatego dla każdego obrazu wyznaczane są cztery macierze, których przesunięcia są skierowane pod kątami 0, 45, 90 oraz 135 stopni, co odpowiada wartościom przesunięć: $(0, 1)$, $(1,1)$, $(1,0)$, $(1,-1)$. Podane cztery macierze współwystępowania obliczane są dla każdej składowej przestrzeni barw YCbCr, do której konwertowany jest analizowany obraz. Głębokość koloru kwantuje się równomiernie za pomocą 16 wartości koloru, wobec czego macierze współwystępowania mają rozmiar 16×16 .

Macierz współwystępowania może być opisana za pomocą wielu parametrów statystycznych. Na potrzeby śledzenia ruchomych obiektów obliczanych jest pięć różnych parametrów dla każdej z macierzy współwystępowania. Zostały one dobrane w ten sposób, aby możliwie wszechstronnie charakteryzowały macierz, a jednocześnie, aby były jak najmniej wzajemnie skorelowane. Parametry te dla symetrycznej, unormowanej macierzy współwystępowania C_N dane są poniższymi zależnościami:

$$\text{kontrast} = \sum_{i,j} C_{N_{i,j}} (i - j)^2 \quad (4.32)$$

$$\text{energia} = \sqrt{\sum_{i,j} C_{N_{i,j}}^2} \quad (4.33)$$

$$\text{średnia} = \mu_i = \mu_j = \sum_{i,j} i \cdot C_{N_{i,j}} = \sum_{i,j} j \cdot C_{N_{i,j}} \quad (4.34)$$

$$\text{odchylenie standardowe} = \sigma_i = \sigma_j = \sqrt{\sum_{i,j} (i - \mu_i)^2 \cdot C_{N_{i,j}}} = \sqrt{\sum_{i,j} (j - \mu_j)^2 \cdot C_{N_{i,j}}} \quad (4.35)$$

$$\text{korelacja} = \sum_{i,j} C_{N_{i,j}} \frac{(i - \mu_i)(j - \mu_j)}{\sqrt{\sigma_i^2 \sigma_j^2}} \quad (\text{jeśli } \sigma_i = \sigma_j = 0 \text{ to korelacja} = 0) \quad (4.36)$$

gdzie $C_{N_{i,j}}$ oznacza element macierzy \mathbf{C}_N o współrzędnych (i, j) .

Wektor zawierający parametry statystyczne macierzy współwystępowania zawiera wobec tego 3 (obrazy) \times 4 (macierze) \times 5 (parametrów statystycznych) = 60 elementów.

Miara podobieństwa S_M dwóch wektorów parametrów statystycznych \mathbf{p}_1 i \mathbf{p}_2 wyznaczonych w oparciu o macierze współwystępowania dla dwóch obrazów wejściowych dana jest empiryczną zależnością, która przyjmuje wartości od 0 (najmniejsze podobieństwo) do 1 (największe podobieństwo):

$$s(\mathbf{p}_1, \mathbf{p}_2) = 1 - \frac{1}{N} \sum_i \frac{|\mathbf{p}_1(i) - \mathbf{p}_2(i)|}{\max(\mathbf{p}_1(i), \mathbf{p}_2(i))} \quad (4.37)$$

gdzie $\mathbf{p}(i)$ oznacza i -ty element wektora \mathbf{p} , a N jest liczbą parametrów w wektorach \mathbf{p}_1 i \mathbf{p}_2 .

Łączna miara podobieństwa dwóch obiektów S jest wyznaczana jako ważona średnia arytmetyczna obu miar składowych:

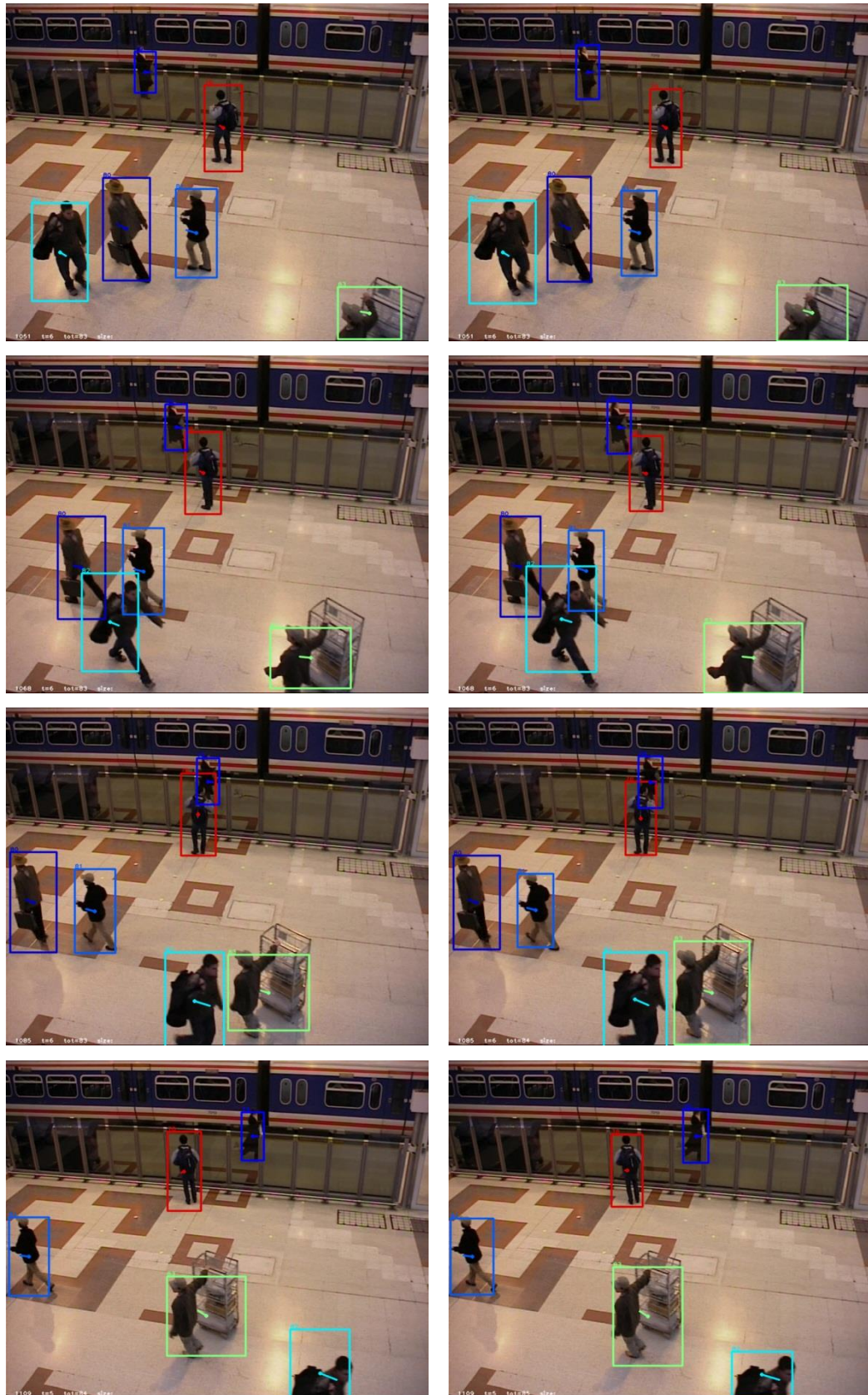
$$S = w_s \cdot S_H + (1 - w_s) \cdot S_M \quad (4.38)$$

gdzie $w_s = 0,5$ symbolizuje udział każdej z miar składowych w wynikowej wartości podobieństwa.

Ta podgrupa regionów, która zawiera region charakteryzujący się największą wartością miary podobieństwa S z analizowanym trackerem, jest wykorzystywana w dalszej części algorytmu do uaktualnienia trackera. Po skończonej analizie wszystkich grup powiązanych ze sobą regionów i trackerów, wszystkie pozostałe regiony są używane do zainicjowania nowych trackerów pod warunkiem, że pole prostokąta opisanego na regionie jest większe, niż T_R .

Przykładowe wyniki opracowanego algorytmu, służącego do rozwiązywania konfliktów, zilustrowano na rys. 4.5. dla przypadku równoczesnego mijania się wielu osób. Można zauważyć, że w żadnym przypadku nie nastąpiła pomyłka śledzenia, tzn. trackery cały czas podążały za właściwymi osobami. Różnice w wynikach działania pomiędzy 8- a 6-elementowym wektorem stanu filtru Kalmana nie są zbyt duże. Można jedynie dostrzec, że w przypadku 6-elementowego wektora stanu, tracker niekiedy nie nadąża ze zmianami rozmiarów obiektu wchodzącego w pole widzenia kamery co skutkuje tym, że pokrycie obiektu przez tracker jest niepełne. Przykładowy film „S1-T1-C3-konflikty.avi” ilustrujący działanie rozwiązywania konfliktów, przy wykorzystaniu 8-elementowego wektora stanu, umieszczono na płycie DVD dołączonej do rozprawy.

Dodatkowo przeprowadzana jest filtracja czasowa wyników śledzenia obiektów, która, poprzez wprowadzenie buforowania uzyskiwanych wyników, pozwala usunąć część fałszywie wykrytych obiektów. Za fałszywie wykryte obiekty uznaje się obiekty o „czasie życia” krótszym niż długość bufora (zwykle kilkanaście ramek obrazu). Filtracja poprawia uzyskiwane wyniki kosztem wprowadzenia w proces przetwarzania obrazu opóźnienia równego długości bufora.



Rys. 4.5 Przykładowe wyniki śledzenia obiektów w sytuacji konfliktowej; w lewej kolumnie użyto filtrów Kalmana z 6-cio elementowym wektorem stanu, w prawej – z 8-mio elementowym; w kolejnych wierszach znajdują się ramki nr 1051, 1068, 1085 i 1109 z nagrania S1-T1-C3 z zestawu nagrań referencyjnych Pets2006 [273]

4.3 Walidacja działania algorytmów detekcji i śledzenia obiektów

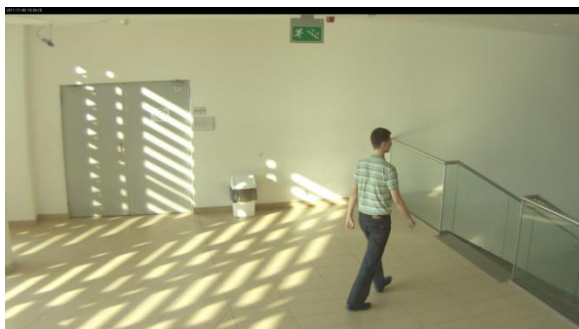
Walidacja działania opracowanych algorytmów detekcji i śledzenia ruchomych obiektów zwykle sprowadza się w literaturze do przedstawienia wybranych ramek obrazu prezentujących poprawne wyniki działania oraz ewentualnie przypadków, w których algorytmy zawiodły. Obiektywna ocena dokładności działania tych algorytmów jest jednak zadaniem nietrywialnym, gdyż wymaga zdefiniowania odpowiednich miar oraz wymaga dużego nakładu pracy w celu określenia danych wzorcowych (ang. *ground-truth*).

Na potrzeby oceny działania algorytmów detekcji i śledzenia obiektów ruchomych zostały wybrane cztery nagrania testowe (rys. 4.6), obejmujące zarówno wnętrza pomieszczeń jak i przestrzeń otwartą. Łączny czas trwania nagrań wynosił 10 minut. Nagrania I-1 oraz I-2 obejmują osoby chodzące po holu budynku oraz poruszające się schodami. Nagrania O-S i O-Ch zawierają rzeczywisty ruch osób i pojazdów po ulicy, jednakże w różnych warunkach oświetleniowych (odpowiednio: słoneczny oraz pochmurny dzień). Wszystkie nagrania testowe zostały przetworzone za pomocą algorytmów detekcji i śledzenia ruchomych obiektów.

Wyniki analizy wszystkich testowych nagrań przy użyciu wartości parametrów pokazanych w tab. 4.1 i tab. 4.3 umieszczono w postaci filmów na płycie DVD dołączonej do rozprawy („I-1.avi”, „I-2.avi”, „O-S.avi” i „O-Ch.avi”). Ponadto na płycie umieszczono kilka innych filmów („sledzenie-ludzie.avi”, „sledzenie-parking.avi”, „sledzenie-pojazdy-1.avi”, „sledzenie-pojazdy-2.avi”, „sledzenie-pojazdy-3.avi”, „sledzenie-deszcz.avi”) prezentujących wyniki detekcji i śledzenia obiektów w różnych sytuacjach; szczególnie ten ostatni dowodzi, że opracowane algorytmy są w stanie poradzić sobie również w trudnych warunkach atmosferycznych.

4.3.1 Detekcja obiektów

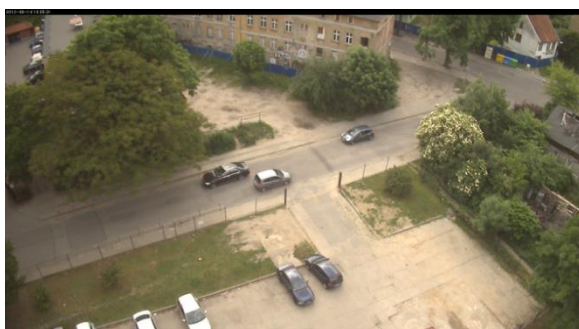
Do walidacji skuteczności detekcji ruchomych obiektów wykorzystano algorytmy: GMM oraz jego wersję rozszerzoną przez autora rozprawy (oznaczoną jako GMM-Mod), która umożliwia kompensację nagłych zmian warunków oświetleniowych. Parametry algorytmów użyte do analizy każdego z nagrań testowych pokazano w tab. 4.1. Różnice w wartościach, w zależności od nagrania wynikają z dopasowania algorytmów do parametrów technicznych poszczególnych nagrań i do charakterystyki pojawiających się w nich obiektów (np. minimalna wielkość obiektu T_P zależy od rozdzielczości ramki obrazu, jak i od wielkości widocznych w niej obiektów, które poruszają się w nagraniu).



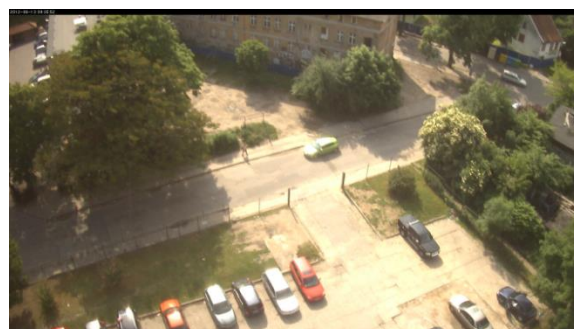
Nagranie I-1
1920x1080, 15 fps, 2296 ramek



Nagranie I-2
704x576, 15 fps, 1621 ramek



Nagranie O-Ch
1600x900, 10 fps, 1868 ramek



Nagranie O-S
1600x900, 10 fps, 1848 ramek

Rys. 4.6 Przykładowe ramki obrazu oraz charakterystyka nagrań testowych użytych do walidacji algorytmów detekcji i śledzenia ruchomych obiektów

Występujący w tab. 4.1 współczynnik skalowania oznacza wielkość zmiany szerokości i wysokości ramki obrazu na potrzeby tylko modelowania tła; detekcja cienia i przetwarzanie morfologiczne odbywają się już na ramkach o oryginalnym rozmiarze. Współczynnik skalowania mniejszy od 1 pozwala zmniejszyć złożoność obliczeniową algorytmu kosztem obniżonej dokładności uzyskiwanych wyników. Wstępne eksperymenty wykazały, że różnica pomiędzy wynikami uzyskanymi dla współczynników skalowania 0,5 (dwukrotnie zmniejszona wysokość i szerokość ramki obrazu) i 1 (czyli bez skalowania) jest niewielka z punktu widzenia działania pozostałych algorytmów przetwarzania obrazu. Ponieważ w rzeczywistych instalacjach testowych, których zarządzaniem zajmuje się autor niniejszej rozprawy w ramach realizacji projektów prowadzonych w Politechnice Gdańskiej, stosowany jest współczynnik skalowania 0,5 (co pozwala czterokrotnie zmniejszyć liczbę analizowanych pikseli obrazu), taką również wartość postanowiono wykorzystać do walidacji algorytmu.

Dane wzorcowe (odniesienia) do oceny jakości detekcji obiektów zostały przygotowane ręcznie dla co 30-tej ramki w każdym nagraniu testowym. Stanowi je

pojedyncza maska binarna dla ramki obrazu. W masce tej niezerowe piksele oznaczają położenie obiektów ruchomych. Maski binarne zostały ręcznie wyznaczone dla łącznie 257 testowych ramek obrazu.

Tab. 4.1 Wartości parametrów algorytmów detekcji obiektów ruchomych tła użyte do analizy nagrań testowych

Parametr	Nagranie testowe			
	I-1	I-2	O-Ch	O-S
Algorytm GMM				
Współczynnik skalowania	0,5	0,5	0,5	0,5
K	5	5	5	5
α	0,003	0,002	0,003	0,003
T_P	100	200	50	50
σ_{MIN}^2	9	16	5	5
Δ	2,5	2,5	2,5	2,5
T	0,5	0,5	0,5	0,5
T_b	0,7	0,8	0,65	0,65
Algorytm GMM-Mod				
T_V	5,0	5,0	5,0	5,0
γ	0,075	0,075	0,075	0,075
δ	0,1	0,1	0,1	0,1

W celu oszacowania działania algorytmów odejmowania tła, użyto trzech miar bazujących na liczbie błędów pierwszego i drugiego rodzaju w bieżącej ramce obrazu: precyzji P (ang. *Precision*), czułości R (ang. *Recall*) oraz dokładności A (ang. *Accuracy*), które dane są następującymi wzorami:

$$P = \frac{TP}{TP + FP} \quad (4.39)$$

$$R = \frac{TP}{TP + FN} \quad (4.40)$$

$$A = \frac{TP + TN}{TP + FP + FN + TN} \quad (4.41)$$

gdzie: TP – liczba pikseli poprawnie zakwalifikowanych jako należące do obiektów ruchomych (ang. *True Positives*), TN – liczba pikseli poprawnie zakwalifikowanych jako należące do tła sceny (ang. *True Negatives*), FP – liczba pikseli błędnie zakwalifikowanych jako należące do obiektów ruchomych sceny (ang. *False Positives*), FN – liczba pikseli błędnie zakwalifikowanych jako należące do tła sceny (ang. *False Negatives*).

Miara P oznacza liczbę poprawnie wykrytych pikseli ruchomych obiektów w odniesieniu do wszystkich pikseli obrazu rozpoznanych jako pierwszoplanowe. Miara R odzwierciedla stopień, w jakim każdy z ruchomych obiektów został rozpoznany w całości. Miara A ma charakter bardziej złożony i pokazuje stopień podobieństwa otrzymanych wyników i danych wzorcowych. Wszystkie te miary mają wartości z przedziału $\langle 0, 1 \rangle$ i ich wartości powinny być jak najwyższe. Wartości użytych miar dla przykładowych ramek obrazu pokazano w tab. 4.2.


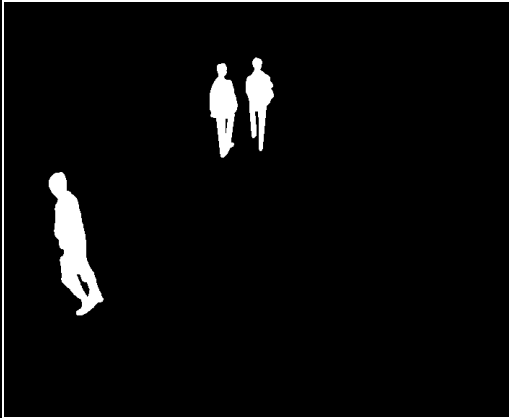
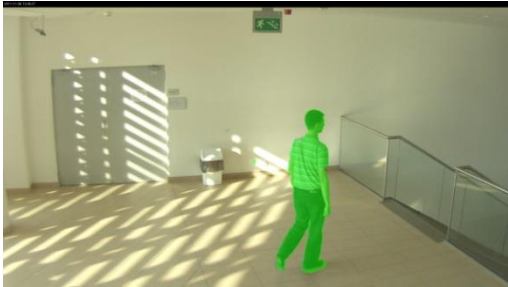
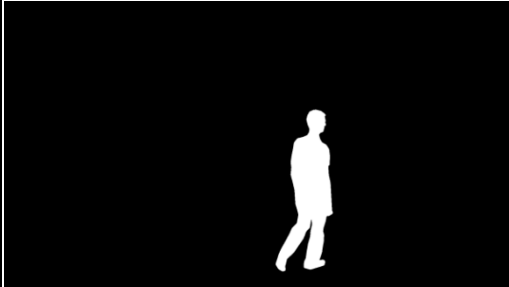




W analizie miar P i R pomięto te ramki wzorcowe, w których wielkości te są niezdefiniowane. Ma to miejsce, gdy algorytm nie wykryje żadnych pikseli należących do ruchomych obiektów (czy to wskutek błędu, czy też ich braku w bieżącej ramce) w przypadku miary P oraz gdy w ramce nie ma żadnych ruchomych obiektów, w przypadku miary R . Wszystkie miary były liczone dla ramek wzorcowych niezależnie, a następnie wyznaczano wartości średnie i odchylenia standardowe.

Wyniki walidacji algorytmów detekcji obiektów ruchomych pokazano na rys. 4.7. Precyzja A działania obu algorytmów jest bardzo wysoka. Lepsze wyniki dla nagrań wykonanych na zewnątrz wynikają z faktu, że, w porównaniu z nagraniami z wnętrza budynków, obiekty ruchome są znacznie mniejsze w odniesieniu do rozmiarów ramki obrazu, a co za tym idzie, udział wartości TN oznaczających poprawne wykrycie elementów tła jest znacznie większy. Porównując oba algorytmy można również zauważyć, że algorytm GMM radzi sobie lepiej w nagraniach z wnętrza, a GMM-Mod – w nagraniach ruchu ulicznego.

Wyniki precyzji P wykazują wyraźną przewagę algorytmu GMM-Mod w nagraniach zewnętrznych, czyli w warunkach występowania zmian warunków oświetleniowych. W tych przypadkach algorytm GMM-Mod generuje istotnie mniej błędów pierwszego rodzaju (FP), co oznacza, że większa część wykrytych pikseli należących do obiektów ruchomych znajduje pokrycie w rzeczywistości (rys. 4.8). W przypadku nagrań wykonanych wewnątrz, różnice między oboma rodzajami algorytmów są pomijalne.

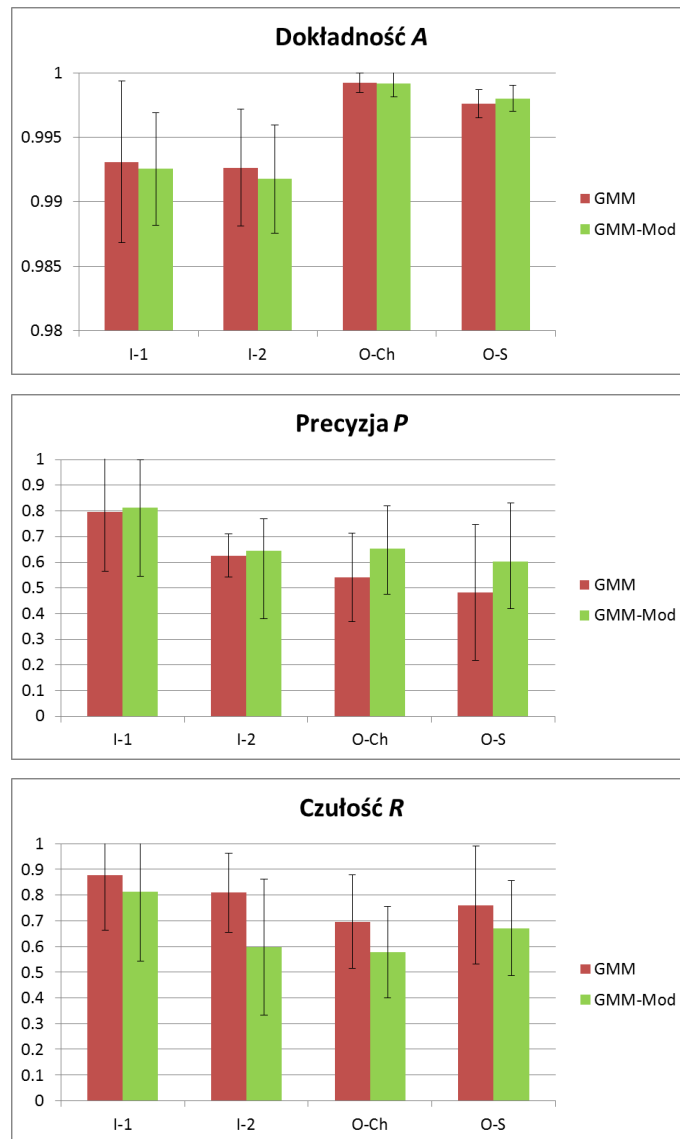
Porównując miary czułości R dla obu algorytmów można zauważyć, że w tym zakresie lepiej się spisuje algorytm GMM. Oznacza to, że przy jego zastosowaniu, możliwe jest dokładniejsze oznaczenie kształtu rzeczywistych obiektów. Mniejsza czułość algorytmu GMM-Mod jest zgodna z oczekiwaniami, gdyż wprowadzona w nim modyfikacja miała właśnie na celu obniżenie jego czułości w taki sposób, aby nie generował on fałszywych alarmów w warunkach zmiennego oświetlenia.

Tab. 4.2 Wartości miar oceny algorytmu detekcji obiektów dla przykładowych ramek obrazu; kolorem zielonym zaznaczono wykryte, ruchome obiekty

Ramka obrazu	Dane wzorcowe	Miary
		$A = 0,977$ $P = 0,616$ $R = 0,805$
		$A = 0,990$ $P = 0,840$ $R = 0,878$
		$A = 0,996$ $P = 0,728$ $R = 0,381$
		$A = 0,995$ $P = 0,574$ $R = 0,714$

Uśrednione wartości miar dla wszystkich nagrań testowych zilustrowano na rys. 4.9 Mimo, że wartości czułości R i precyzji P mogą się wydawać stosunkowo niskie, to w praktyce, oglądając nagrania wynikowe, nie widać żadnych istotnych błędów. Wynika to z definicji tych miar, które są ukierunkowane na badanie relacji liczby pikseli poprawnie wykrytych jako obiekty ruchome (TP); przy ich niewielkiej (lub zerowej) wartości, wystarczy bardzo niewielka liczba pikseli FP lub FN , aby miary te dla całej ramki obrazu przyjęły bardzo niskie wartości. W miarach precyzji i czułości można także zauważyć

wpływ zastosowania skalowania obrazu o połowę przy modelowaniu tła. W wyniku tego procesu kontury wynikowej maski binarnej mogą być o 1-2 piksele rozmięte z rzeczywistymi konturami obiektów; im obiekty mniejsze, tym większy udział tych błędów w całkowitej liczbie pikseli obiektów. Stąd dla nagrań wykonanych w warunkach zewnętrznych, charakteryzujących się obiektami o mniejszym rozmiarze, miary precyzji i czułości są gorsze.



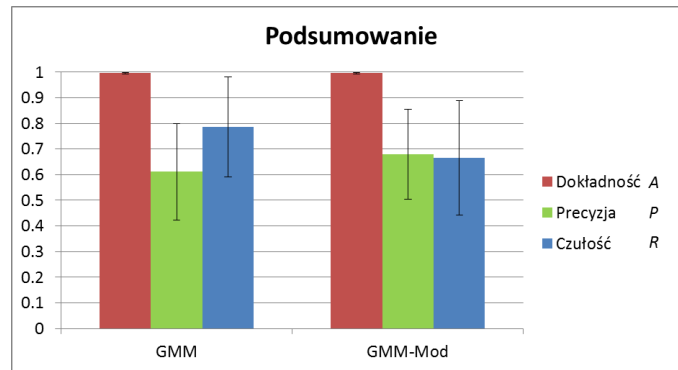
Rys. 4.7 Wartości średnie i odchylenia standardowe miar służących do oceny algorytmów detekcji obiektów ruchomych, wyznaczonych dla każdego każdej nagrania testowego i każdej ramki wzorcowej

Analiza wyników wskazuje, że w przypadkach stałych warunków oświetleniowych, czyli wewnątrz budynków, lepsze wyniki można uzyskać za pomocą algorytmu GMM. Jego zmodyfikowana wersja (GMM-Mod) okazuje się jednak niezastąpiona w warunkach

zewnętrznego, naturalnego oświetlenia, gdzie pozwala uzyskać znacznie większą precyzję. Uniknięcie fałszywych detekcji jest w tym przypadku znacznie cenniejsze, niż potencjalne niedokładności w określeniu kształtu rzeczywistych obiektów, wynikające z niższej czułości.



Rys. 4.8 Przykładowa ramka obrazu z nagrania O-S, analizowana algorytmem GMM (u góry) i GMM-Mod (u dołu). Zielone obszary oznaczają wykryte piksele należące do ruchomych obiektów (TP i FP), czerwone – wykryty cień. Dodatkowo kolorowymi prostokątami oznaczono wszystkie wykryte obiekty spełniające kryterium wielkości



Rys. 4.9 Miary oceny algorytmów detekcji ruchomych obiektów uśrednione dla wszystkich nagrań testowych

4.3.2 Śledzenie obiektów

W procesie oceny działania algorytmu śledzenia obiektów ruchomych wykorzystano dwa jego warianty: z sześć- i z ośmio-elementowym wektorem stanu filtru Kalmana (KF6 i KF8). Parametry algorytmu użyte do analizy każdego z nagrań testowych pokazano w tab. 4.3. Do detekcji obiektów ruchomych użyto optymalnej wersji algorytmu (GMM dla nagrań wewnątrz pomieszczeń i GMM-Mod dla nagrań zewnętrznych). Różnice w wartościach parametrów algorytmu śledzenia obiektów w zależności od nagrania wynikają z dopasowania algorytmów do parametrów technicznych poszczególnych nagrań i do charakterystyki pojawiających się w nich obiektów. Na przykład, minimalna wielkość obiektu T_R zależy od bezwzględnej wielkości występujących w scenie obiektów zainteresowania, a wartości parametrów reprezentujących szum pomiarowy r_p i r_s są ustalane w zależności od typowej prędkości obiektów, ich odległości od kamery i charakteru ruchu; w przypadku wolnych (lub odległych) obiektów poruszających się w sposób jednostajny ustawione poziomy szumu mogą być większe. Należy również zwrócić uwagę, że na skuteczność działania algorytmu śledzenia obiektów ruchomych istotny wpływ mają wyniki detekcji obiektów w każdej ramce obrazu.

Oceny algorytmów śledzenia obiektów ruchomych dokonano na podstawie porównania prostokątów opisujących każdy ruchomy obiekt w każdej ramce obrazu. Dane do walidacji, generowane przez algorytm, stanowiły prostokąty wyznaczone z wektora stanu filtru Kalmana skojarzonego z każdym ruchomym obiektem. Jako dane wzorcowe wykorzystano prostokąty otaczające ruchome obiekty we wszystkich ramkach nagrań testowych. Dane wzorcowe zostały przygotowane ręcznie z pomocą narzędzia ViPER [69].

Tab. 4.3 Wartości parametrów algorytmu śledzenia obiektów ruchomych użyte do analizy nagrań testowych

Parametr	Nagranie testowe			
	I-1	I-2	O-Ch	O-S
Algorytm detekcji obiektów	GMM	GMM	GMM-Mod	GMM-Mod
T_R	3000	10000	300	300
r_p	0,001	0,001	0,01	0,01
r_s	0,001	0,001	0,01	0,01
q	1e-5	1e-5	1e-5	1e-5
p_0	1,0	1,0	1,0	1,0

Do oceny algorytmu śledzenia obiektów użyto trzech miar [167][227] i związane z ich definicjami wielkości pomocnicze. Symbolem $G(t)$ oznaczono zbiór wzorcowy (odniesienia) zawierający wszystkie obiekty (prostokąty) występujące w ramce t . $D(t)$ oznacza wyniki działania algorytmu śledzenia ruchomych obiektów (zbiór wszystkich prostokątów) w ramce t . Liczebność obu tych zbiorów w ramce t oznaczono odpowiednio $N_{G(t)}$ oraz $N_{D(t)}$. Na tej podstawie, $U_{G(t)}$ oraz $U_{D(t)}$ oznaczają sumę przestrzenną wszystkich elementów (prostokątów) wchodzących w skład zbiorów $G(t)$ i $D(t)$:

$$U_{G(t)} = \bigcup_{i=1}^{N_{G(t)}} G_i(t) \quad U_{D(t)} = \bigcup_{i=1}^{N_{D(t)}} D_i(t) \quad (4.42)$$

Fragmentacja F jest pierwszą z wykorzystywanych miar. Jej celem jest wykrycie sytuacji, w których pojedynczy, rzeczywisty obiekt jest oznaczony przez wiele prostokątów i z punktu widzenia algorytmu śledzenia obiektów traktowany jak wiele obiektów. Miara ta uwzględnia zarówno sytuacje podziału obiektu rzeczywistego na kilka mniejszych (rozłączne prostokąty) jak i występowanie wielu podobnych, nakładających się prostokątów otaczających cały obiekt rzeczywisty. Fragmentacja pojedynczego obiektu $G_i(t)$ w ramce t dana jest zależnością:

$$F(G_i(t)) = (1 + \log(N_{D(t) \cap G_i(t)}))^{-1} \quad \text{dla } N_{D(t) \cap G_i(t)} > 0 \quad (4.43)$$

gdzie $N_{D(t) \cap G_i(t)}$ oznacza liczbę prostokątów w zbiorze wyjściowym $D(t)$, które mają część obszaru wspólną z prostokątem opisanym na obiekcie rzeczywistym $G_i(t)$.

Dla pojedynczej ramki obrazu miara fragmentacji $F(t)$ jest średnią miarą fragmentacji wszystkich obiektów w niej występujących (dla których miara $F(G_i(t))$ jest zdefiniowana). Fragmentacja F dla całego nagrania jest średnią miarą fragmentacji wszystkich obiektów rzeczywistych występujących we wszystkich ramkach nagrania testowego. Miara

fragmentacji pozwala oszacować, w jakim stopniu zachowana jest integralność rzeczywistych obiektów (im większa wartość, tym mniejsza fragmentacja). Pewną niedogodnością tej miary jest fakt, że pojedynczy prostokąt, który nawet o jeden piksel nachodzi na obiekt rzeczywisty jest traktowany jako efekt fragmentacji tego obiektu. Oznacza to, że w przypadku sytuacji konfliktowych, takich jak mijanie się grupy obiektów, miara fragmentacji będzie malała mimo braku faktycznych błędów.

Drugą wykorzystywaną miarą jest czułość detekcji obiektów R (ang. *Average Object Area Recall*). Czułość dla pojedynczego obiektu $G_i(t)$ w ramce t jest definiowana jako część powierzchni tego obiektu, jaka jest pokryta przez wynikowe prostokąty:

$$R(G_i(t)) = \frac{|G_i(t) \cap U_{D(t)}|}{|G_i(t)|} \quad (4.44)$$

gdzie $A \cap B$ oznacza część wspólną prostokątów A i B , a operator $|A|$ oznacza liczbę pikseli w obszarze A .

Miara czułości dla pojedynczej ramki obrazu dana jest zależnością:

$$R(t) = \frac{\sum_{G_i(t)} R(G_i(t))}{N_{G(t)}} \quad (4.45)$$

Całkowita miara czułości dla całego nagrania testowego zdefiniowana jest jako średnia ważona czułość w każdej ramce obrazu:

$$R = \frac{\sum_{t=1}^{N_f} N_{G(t)} \cdot R(t)}{\sum_{t=1}^{N_f} N_{G(t)}} \quad (4.46)$$

gdzie N_f oznacza liczbę ramek obrazu w analizowanym nagraniu.

Miara czułości określa stopień pokrycia powierzchni każdego obiektu wzorcowego przez wyniki działania algorytmu śledzenia obiektów. Każdy obiekt ma taki sam wpływ na ostateczną wartość miary, niezależnie od wielkości. Może to sprawić, że w skrajnym przypadku, gdy w ramce obrazu znajdują się dwa obiekty: bardzo duży wykryty w całości i bardzo niewielki, całkowicie pominięty, miara czułości $R(t)$ dla tej ramki będzie wynosić 0,5.

Ostatnią użytą miarą jest precyzja detekcji obiektów P (ang. *Average Detected Box Area Precision*). Miara ta jest uzupełnieniem miary czułości ukierunkowanym na analizę wyników śledzenia obiektów, zamiast na obiekty wzorcowe. Precyzja pojedynczego,

wynikowego prostokąta $D_i(t)$ w ramce t jest definiowana jako część powierzchni tego obiektu, jaka pokrywa rzeczywiste obiekty:

$$P(D_i(t)) = \frac{|D_i(t) \cap U_G(t)|}{|D_i(t)|} \quad (4.47)$$

Miary precyzji dla pojedynczej ramki obrazu i dla całego nagrania dane są wzorami analogicznymi, jak w przypadku miary czułości:

$$P(t) = \frac{\sum_{D_i(t)} P(D_i(t))}{N_{D(t)}} \quad (4.48)$$

$$P = \frac{\sum_{t=1}^{N_f} N_{D(t)} \cdot P(t)}{\sum_{t=1}^{N_f} N_{D(t)}} \quad (4.49)$$









Analogicznie jak w przypadku miary czułości, wpływ każdego prostokąta wynikowego na wartość miary precyzji jest niezależny od jego wielkości.

Wartości wszystkich stosowanych miar mieszczą się w przedziale $\langle 0,1 \rangle$. Wartości tych miar dla przykładowych ramek obrazu pokazano w tab. 4.4.

Wyniki oceny algorytmu detekcji śledzenia obiektów dla dwóch typów wektora stanu filtru Kalmana pokazano na rys. 4.10. Wartości miary fragmentacji są bardzo wysokie i zbliżone do siebie, niezależnie od nagrania testowego i wariantu algorytmu. Oznacza to, że w zdecydowanej większości przypadków jeden tracker odpowiada jednemu rzeczywistemu obiektowi i nie zachodzi podział rzeczywistego obiektu na kilka części. Różnice widoczne w zależności od nagrania wynikają w największym stopniu z charakteru ruchu występujących obiektów i częstości ich wzajemnego przesłaniania, co wpływa negatywnie na miarę fragmentacji.

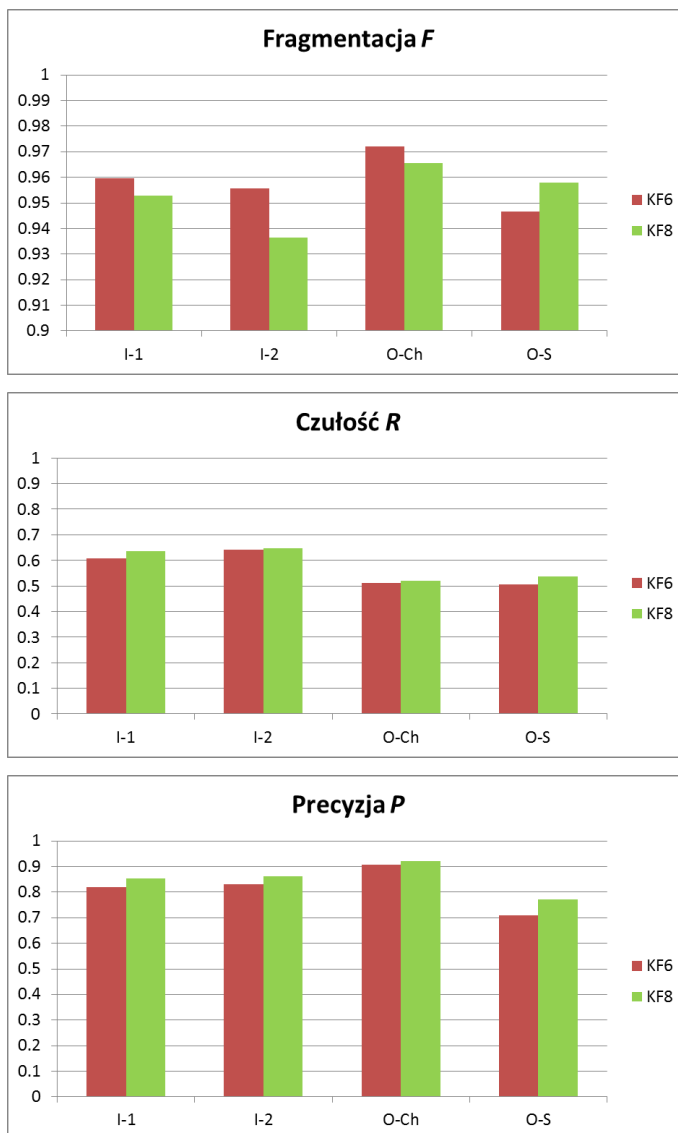
Miara czułości jest generalnie wyższa w przypadku nagrań wykonanych wewnątrz pomieszczeń. Wynika to przede wszystkim z zastosowanego algorytmu odejmowania tła, który w wersji użytej do nagrań zarejestrowanych wewnątrz pomieszczeń charakteryzuje się wyższą czułości (która, z kolei, w przypadku odejmowania tła ma analogiczne znaczenie, jak w przypadku śledzenia obiektów). W nagraniach zewnętrznych daje się okazjonalnie dostrzec, że np. tracker nie obejmuje całej sylwetki osoby, pomijając głowę lub nogi.

Tab. 4.4 Wartości miar oceny algorytmu śledzenia obiektów dla przykładowych ramek obrazu

Ramka obrazu	Dane wzorcowe	Miary
		$F(t) = 1,000$ $R(t) = 0,869$ $P(t) = 0,978$
		$F(t) = 1,000$ $R(t) = 0,609$ $P(t) = 0,865$
		$F(t) = 0,892$ $R(t) = 0,646$ $P(t) = 0,696$
		$F(t) = 1,000$ $R(t) = 0,904$ $P(t) = 0,668$

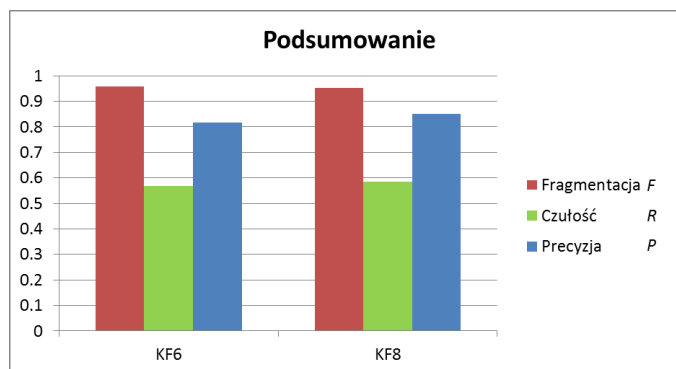
Miara precyzji jest zbliżona dla większości nagrań testowych i jej stosunkowo wysokie wartości wskazują, że rzadko w wynikach analizy pojawiają się fałszywe obiekty lub że tracker znacznie odstaje od rzeczywistego położenia śledzonego obiektu. Jedyne w przypadku nagrania O-S, realizowanego w najtrudniejszych warunkach oświetleniowych, wartość precyzji jest o kilkanaście procent niższa. W tym przypadku można również wyraźnie dostrzec zaletę stosowania autorskiego algorytmu GMM-Mod w procesie odejmowania tła, gdyż wartość precyzji P wyników śledzenia ruchomych obiektów

wyniosła 0,73 (dla KF8), podczas gdy ta sama miara przy zastosowaniu dla tego samego nagrania i algorytmu GMM wynosi tylko 0,18.



Rys. 4.10 Wartości miar służących do oceny algorytmu śledzenia obiektów ruchomych, wyznaczone dla każdego nagrania testowego

Porównując wyniki uzyskane przez obie wersje algorytmu śledzenia obiektów (KF6 i KF8), podsumowane na rys. 4.11, można zauważyć, że ich działanie jest zbliżone. Jednak w praktyce możliwość uzyskiwania szybszych zmian rozmiaru (mimo braku zmian rozmiaru rzeczywistego obiektu) zwiększa elastyczność trackera z 8-elementowym wektorem stanu filtru Kalmana i pozwala mu szybciej reagować, szczególnie w sytuacjach konfliktowych. Znajduje to odzwierciedlenie w nieznacznie większej wartości miary czulości i w wyraźnie lepszej wartości precyzji, zarówno dla każdego nagrania niezależnie, jak i średnio dla wszystkich nagrań testowych.



Rys. 4.11 Średnie wartości miar oceny algorytmu śledzenia ruchomych obiektów

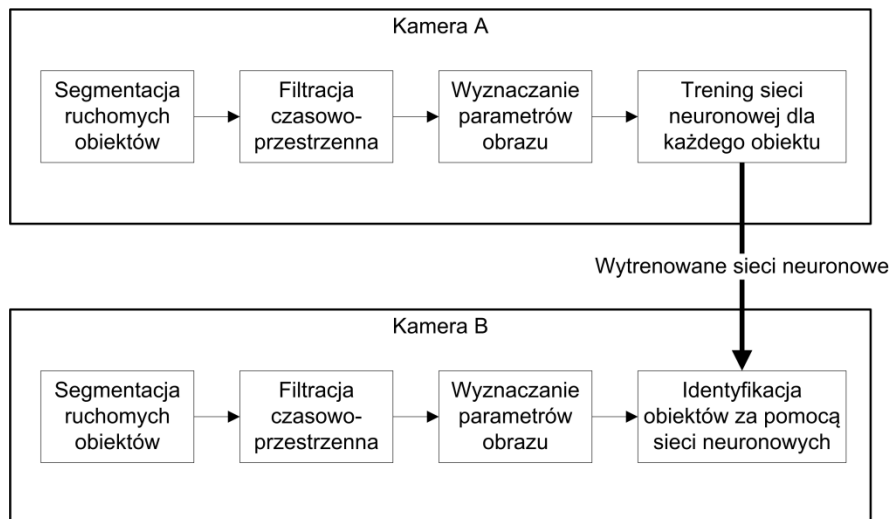
4.4 Śledzenie wielokamerowe

Algorytm reidentyfikacji obiektów opracowany na potrzeby wielokamerowego śledzenia obiektów nie wymaga kalibracji pola widzenia kamer. Niemniej pewne informacje dotyczące wzajemnego położenia kamer muszą być zdefiniowane. Algorytm wykorzystuje lokalne cechy obiektów oraz sztuczne sieci neuronowe, które są uczone wyglądu poszczególnych obiektów w określonej kamerze, a następnie użyte do ich rozpoznania w obrazach z innych kamer [56]. Takie podejście wydaje się innowacyjne, gdyż zdecydowana większość rozwiązań opisanych w literaturze (rozdział 3.2.5) bazuje na miarach odległościowych. Algorytm został opracowany (i przetestowany) dla ruchomych pojazdów, jednak rozwiązanie to jest uniwersalne, gdyż nie bazuje na żadnych specyficznych właściwościach pojazdów.

Schemat algorytmu wielokamerowego śledzenia obiektów pokazano na rys. 4.12. Jego celem jest identyfikacja tego samego obiektu w obrazach zarejestrowanych przez różne kamery.

W pierwszej kolejności strumień wizyjny z każdej kamery jest analizowany niezależnie w celu wykrycia i śledzenia ruchomych pojazdów (rozdziały 4.1 i 4.2). Następnie dokonywana jest filtracja czasowo-przestrzenna wykrytych obiektów, zarówno w obrębie pojedynczego kadru (w celu ograniczenia ilości danych do przetwarzania), jak i w aspekcie całego systemu monitoringu (z wykorzystaniem względnego położenia poszczególnych kamer w celu ograniczenia puli porównywanych ze sobą obiektów). W kolejnym kroku wyznaczane są parametry obrazu pojazdów, wykorzystywane do porównania obiektów pojawiających się w polu widzenia bieżącej kamery z obiektami widzianymi przez inne kamery. Porównywanie odbywa się z wykorzystaniem sztucznych sieci neuronowych. Każda sieć trenowana jest do rozpoznawania obrazów jednego

konkretnego pojazdu w polu widzenia konkretnej kamery. Następnie, sieć ta jest użyta do wykrycia obrazów tego samego pojazdu zarejestrowanych przez inne kamery. Wynika z tego, że całkowita informacja o wyglądzie obiektów przesyłana pomiędzy modułami analizującymi obrazy z różnych kamer jest zawarta w wytrenowanej sieci neuronowej w postaci jej wewnętrznych wag synaptycznych, a nie w formie wektora parametrów.



Rys. 4.12 Schemat algorytmu wielokamerowego śledzenia obiektów

4.4.1 Filtracja czasowo-przestrzenna

W wyniku odejmowania tła i śledzenia obiektów uzyskuje się zbiór obrazów poszczególnych obiektów w każdej ramce obrazu. Dalszej analizie poddawane są tylko te obrazy obiektów, które nie wykazują konfliktów śledzenia, zgodnie z informacjami pochodzącymi z algorytmu śledzenia ruchomych obiektów. Ponadto, nie ma praktycznej potrzeby przetwarzania wszystkich obrazów każdego pojazdu, ponieważ ich wygląd nie zmienia się w sposób istotny z ramki na ramkę. W związku z tym, konkretny obraz jest uwzględniany w analizie tylko wtedy, gdy upłynął określony czas t od pobrania poprzedniego obrazu danego obiektu lub gdy obiekt ten przemieścił się o zadany dystans d . Wartości parametrów t i d obliczane są wg poniższych równań wyznaczonych empirycznie:

$$t = \frac{3}{f} \quad (4.50)$$

$$d = 0,05 \cdot \frac{w + h}{2} \quad (4.51)$$

gdzie f oznacza liczbę ramek obrazu pobieranych z kamery w ciągu sekundy, a w i h reprezentują wysokość i szerokość ramki obrazu w pikselach. Takie podejście pozwala zredukować ilość danych do analizy od 3 do 10 razy, w zależności od charakterystyki ruchu obiektów w analizowanej scenie.

W aspekcie globalnym filtracja związana jest ze zdefiniowaniem w polu widzenia każdej kamery tzw. obszarów wejścia/wyjścia. W przypadku obiektu opuszczającego pole widzenia kamery w jednym z obszarów wyjściowych, jego parametry (wytrenowana sieć neuronowa) są przesyłane tylko do najbliższych kamer, w zależności od kierunku poruszania się obiektu i konkretnego obszaru wyjściowego. Analogicznie, aby obiekt mógł być rozpatrywany jako kandydat do identyfikacji, musi się on pojawić w polu widzenia kamery w jednym z obszarów wejściowych i jest identyfikowany przez sieci neuronowe nadesłane tylko z kamer powiązanych z tym konkretnym obszarem wejściowym. Ponadto analizowany jest czas przejścia pomiędzy polami widzenia kamer w celu dalszego zawężenia liczebności zbioru obiektów do porównywania.

Dodatkową zaletą uwzględniania w analizie jedynie obiektów pojawiających się w obszarach wejściowych i opuszczających pole widzenia kamery w obszarach wyjściowych jest pominięcie w procesie analizy obiektów postronnych (np. osób poruszających się innymi drogami, niż pojazdy) oraz potencjalnych błędów pierwszego rodzaju w działaniu algorytmu detekcji ruchomych obiektów, wynikających ze zmiennych warunków oświetleniowych lub zmian w tle analizowanej sceny.

Obszary wejścia/wyjścia są oznaczane ręcznie w polach widzenia poszczególnych kamer jako wielokąty, które są umiejscowione np. na drogach, na marginesie kadru (rys. 4.13). Ruchomy obiekt jest oznaczany jako pojawiający się w konkretnym obszarze wejściowym (i uwzględniany w dalszej analizie), jeśli wystąpiła seria poniższych zdarzeń:

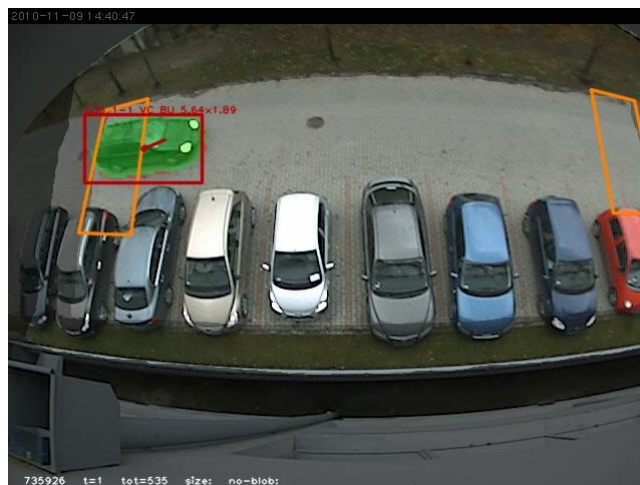
1. Obiekt pojawił się w obrazie w sąsiedztwie obszaru wejściowego. Warunek ten jest spełniony, jeśli istnieje część wspólna między prostokątem opisanym na obiekcie, a obszarem wejściowym
2. Przemieścił się do wnętrza obszaru – punkt zainteresowania zdefiniowany względem prostokąta opisanego na obiekcie dla tego obszaru (np. punkt centralny lub środek dolnej krawędzi) znalazł się w jego wnętrzu
3. Opuścił obszar wejściowy – punkt zainteresowania znalazł się poza obszarem
4. Porusza się w kierunku centrum ramki obrazu

Ostatni warunek sprawdzany jest poprzez wyznaczenie kąta φ między uśrednionym krótkoterminowo kierunkiem ruchu obiektu, a wektorem skierowanym do centrum ramki obrazu. Kąt φ dany jest wzorem:

$$\varphi = \arccos\left(\frac{(\mathbf{c} - \mathbf{p}) \cdot \mathbf{v}}{\|\mathbf{c} - \mathbf{p}\| \cdot \|\mathbf{v}\|}\right) \quad (4.52)$$

gdzie \mathbf{c} oznacza wektor współrzędnych środka ramki obrazu (w pikselach), \mathbf{p} – pozycję obiektu (środek prostokąta na nim opisanego), a \mathbf{v} jest wektorem zawierającym składowe prędkości obiektu wzdłuż osi pionowej i poziomej. Jeśli kąt φ jest mniejszy niż 90° , to przyjmuje się, że obiekt porusza się w kierunku środka kadru.

W analogiczny sposób znajdowane są obiekty opuszczające pole widzenia kamery w jednym z obszarów wyjściowych. Obiekt taki musi się znaleźć wewnątrz takiego obszaru, a następnie go opuścić, poruszając się w kierunku przeciwnym do środka kadru. W tym ostatnim warunku wartość kąta φ musi być większa niż 90° .



Rys. 4.13 Przykład pojazdu (zielona maska i czerwony prostokąt) pojawiającego się w polu widzenia w jednym z obszarów wejściowych (pomarańczowe wielokąty)

4.4.2 Parametry obrazu obiektów

Parametry obrazu obiektów przeznaczone do ich reidentyfikacji w polu widzenia różnych kamer powinny być odporne na zmiany orientacji obiektu w stosunku do kamery oraz na różnice w ustawieniach samych kamer, takich jak np. balans bieli. Z tego względu postanowiono wybrać deskryptory SURF, jako potencjalnie najlepsze do realizacji tego zadania.

Deskryptor SURF (ang. *Speded Up Robust Features*) [8] jest parametrem charakteryzującym obraz lokalnie, tj. w najbliższym sąsiedztwie wybranego punktu charakterystycznego. Jego zaletą jest niezależność względem operacji skalowania i obrotu, powtarzalność, dystynktywność oraz niska złożoność obliczeniowa w porównaniu z pierwowzorem, jakim jest deskryptor SIFT (ang. *Scale-Invariant Feature Transform*) [160]. Na skuteczność działania lokalnych deskryptorów obrazu wpływ mają dwa procesy: wyznaczanie położenia punktów charakterystycznych oraz ekstrakcja cech dystynktywnych w otoczeniu każdego punktu.

Do automatycznego znajdowania wszystkich punktów charakterystycznych w obrazie (ich położenia, rozmiaru i orientacji) wykorzystano algorytm bazujący na wyznacznikach macierzy Hessego. Detektor ten wyróżnia się w grupie detektorów, które są odporne na przekształcenia afiniczne, takich jak detektor Harrisa (ang. *Harris Affine Region Detector*), MSER (ang. *Maximally Stable Extremal Regions*), detektor Kadira-Brady’ a (ang. *Kadir-Brady Saliency Detector*), EBR (ang. *Edge-Based Regions*) oraz IBR (ang. *Intensity-Extrema-Based Regions*) [173].

Macierz Hessego jest dana zależnością [8]:

$$\mathbf{H}(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix} \quad (4.53)$$

gdzie $L_{xy}(\mathbf{x}, \sigma)$ oznaczają pochodne cząstkowe drugiego rzędu w punkcie obrazu $I(\mathbf{x})$ w określonych kierunkach, wygładzone za pomocą jądra Gaussowskiego z parametrem σ . Jako punkty charakterystyczne wybierane są te punkty obrazu, które stanowią jednocześnie maksimum lokalne wyznacznika oraz śladu macierzy Hessego danej wzorem (4.53). Wyznacznik DET i ślad TR macierzy Hessego wyrażone są równaniami:

$$\text{DET}(\mathbf{H}(\mathbf{x}, \sigma)) = \sigma^2 (L_{xx}(\mathbf{x})L_{yy}(\mathbf{x}) - L_{xy}^2(\mathbf{x})) \quad (4.54)$$

$$\text{TR}(\mathbf{H}(\mathbf{x}, \sigma)) = \sigma (L_{xx}(\mathbf{x}) + L_{yy}(\mathbf{x})) \quad (4.55)$$

Deskryptor SURF obliczany jest w kwadratowym sąsiedztwie wybranego punktu charakterystycznego. Sąsiedztwo to dzielone jest na 4×4 równe obszary. W każdym z obszarów obliczane są parametry wywiedzione z falki Haara [91] (ang. *Haar-Like Features*). Parametry te reprezentują różnice w sumie wartości pikseli między dwoma regionami obrazu sąsiadującymi w pionie lub poziomie (rys. 4.14). Suma parametrów w

zależności od kierunku filtracji (poziomego d_x i pionowego d_y w odniesieniu do orientacji punktu charakterystycznego) tworzy cztero-elementowy wektor dla każdego z obszarów w sąsiedztwie punktu charakterystycznego [8]:

$$\mathbf{v}_4 = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y| \right) \quad (4.56)$$

Po połączeniu wyników z każdego obszaru uzyskuje się 64-elementowy deskryptor SURF (SURF-64) dla każdego punktu charakterystycznego obrazu. Wyniki filtracji są niezależne od jasności obrazu (składowej addytywnej); niezależność względem kontrastu (składowej multiplikatywnej) uzyskuje się poprzez przekształcenie deskryptora w wektor jednostkowy.

Dodatkowo wykorzystano rozszerzoną wersję deskryptora SURF, w której dla każdego z 16 obszarów w sąsiedztwie punktu charakterystycznego wyznaczanych jest 8 parametrów, uwzględniając przy tym znak wyniku filtracji [8]:

$$\mathbf{v}_8 = \left(\sum_{d_x < 0} d_x, \sum_{d_x \geq 0} d_x, \sum_{d_x < 0} |d_x|, \sum_{d_x \geq 0} |d_x|, \right. \\ \left. \sum_{d_y \geq 0} d_y, \sum_{d_y < 0} d_y, \sum_{d_y < 0} |d_y|, \sum_{d_y \geq 0} |d_y| \right) \quad (4.57)$$

Daje to łącznie 128-elementowy deskryptor SURF (SURF-128) dla każdego punktu charakterystycznego.



Rys. 4.14 Ilustracja dwóch rodzajów parametrów (różnica sum wartości pikseli w obszarach jasnych i ciemnych) wykorzystywanych do wyznaczenia deskryptora SURF

Przewaga deskryptorów lokalnych SURF nad innymi rodzajami parametrów obrazu w zadaniu reidentyfikacji obiektów jest dwójaka. Po pierwsze, deskryptory SURF bazują na obrazach w odcieniach szarości, przez co są znacznie mniej wrażliwe na różnice w kolorze tych samych obiektów obserwowanych przez różne kamery. Ponadto, wykorzystanie wielu cech lokalnych wyznaczonych w sąsiedztwie automatycznie znalezionych punktów charakterystycznych zamiast jednego, globalnego parametru dla

całego obrazu wydaje się ułatwiać skuteczną identyfikację w sytuacji różnic w pozycji obiektu, o ile wystarczająca część obiektu jest widoczna w obrazach z obu kamer.

Liczba automatycznie znalezionych punktów charakterystycznych może się istotnie zmieniać w zależności od analizowanego obrazu. Dlatego zastosowano rozwiązanie mające na celu zredukowanie dynamiki tych zmian, którego główną cechą jest iteracyjny proces poszukiwania punktów charakterystycznych. Założono, że liczba tych punktów D_K musi mieścić się w przedziale $[D_K - 10\%, D_K + 10\%]$. Jeśli warunek ten nie jest spełniony, odpowiednio zmieniana jest czułość detektora Hessego i proces znajdowania punktów charakterystycznych jest powtarzany. Kończy się on po znalezieniu oczekiwanej liczby punktów lub po osiągnięciu granicznej liczby iteracji N_I . Na podstawie wstępnych eksperymentów przyjęto $D_K = 25$ i $N_I = 30$.

4.4.3 Reidentyfikacja obiektu

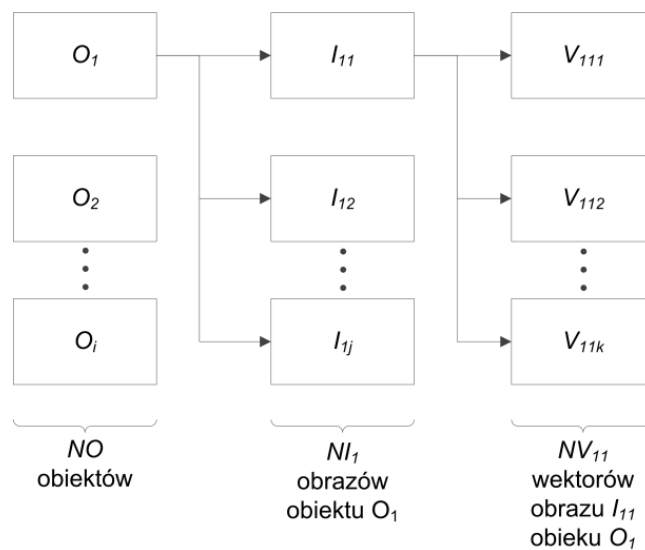
Do zidentyfikowania tego samego obiektu w polu widzenia innej kamery autor rozprawy zaproponował innowacyjną metodę wykorzystującą sztuczną sieć neuronową [56]. Sieć trenowana jest do rozpoznawania obrazów jednego, konkretnego obiektu w polu widzenia określonej kamery (źródłowej), a następnie jest ona użyta do wykrycia tego obiektu w strumieniu wizyjnym pochodzącym z innej kamery (docelowej). Pozytywne wzorce treningowe tworzone są w oparciu o parametry obrazu rozpatrywanego obiektu, opisane w rozdziale 4.4.2. Negatywne wzorce tworzą parametry obrazów innych obiektów, które pojawiły się w polu widzenia kamery źródłowej w ciągu ostatnich T minut. Oznacza to, że pojedyncza sieć neuronowa jest klasyfikatorem binarnym, uczonym do rozróżniania jednego, konkretnego obiektu od innych, potencjalnie podobnych obiektów, które znajdowały się w tym samym miejscu i czasie.

Na potrzeby reidentyfikacji obiektów wykorzystano jednokierunkową sieć neuronową z jedną warstwą ukrytą. Ponieważ liczba parametrów związanych z pojedynczym obrazem obiektu jest zmienna i zależy od liczby wykrytych punktów charakterystycznych, nie jest możliwe jednoczesne podanie na wejście sieci wszystkich parametrów. W związku z tym zdecydowano się podawać na wejścia niezależnie każdy wektor SURF, co rzutuje na liczbę wejść sieci (64 lub 128, w zależności o typu wykorzystywanego parametru SURF). Na podstawie wstępnych badań liczbę neuronów w warstwie ukrytej sieci ustalono na równą połowie liczby wejść. Wykorzystano sigmoidalną bipolarną funkcję aktywacji neuronów sieci.

Zaprojektowana przez autora rozprawy sieć ma dwa wyjścia. Oczekiwaną odpowiedzią sieci w przypadku, gdy na jej wejście zostanie podany wektor SURF należący do wybranego obiektu, jest wektor $[1, -1]$ albo $[-1, 1]$ w przeciwnym przypadku. Zastosowanie dwóch wyjść (zamiast jednego) umożliwia w procesie rozpoznawania uwzględnienie informacji nie tylko o wyniku klasyfikacji, ale pozwala również oszacować wiarygodność otrzymanego wyniku.

W celu zidentyfikowania obiektu S obserwowanego w kamerze źródłowej C_A , w ramach obrazu otrzymanych z kamery docelowej C_B , konieczne jest znalezienie najbardziej podobnego obiektu spośród wszystkich obiektów, które w wyniku filtracji przestrzenno-czasowej zostały zakwalifikowane jako kandydaci do dopasowywania w kamerze C_B . W związku z powyższym przyjęto następujący algorytm postępowania.

Niech $O_i, i = 1 \dots NO$ oznacza i -ty obiekt, wśród których poszukiwany jest obiekt S , w zbiorze NO obiektów, $I_{ij}, j = 1 \dots NI_i$ symbolizuje j -ty obraz i -tego obiektu O_i wśród NI_i obrazów tego obiektu, a $V_{ijk}, k = 1 \dots NV_{ij}$ oznacza k -ty wektor SURF obrazu I_{ij} przedstawiającego obiekt O_i w zbiorze NV_{ij} wektorów SURF znalezionych w tym obrazie. Zależności te zilustrowano na rys. 4.15.



Rys. 4.15 Zależności między obiektami O , ich obrazami I oraz wektorami parametrów lokalnych V występujące w opracowanym przez autora algorytmie reidentyfikacji obiektów

W procesie klasyfikacji, sieć neuronowa nauczona rozpoznawać obiekt S w polu widzenia kamery źródłowej C_A kolejno klasyfikuje wszystkie wektory SURF V_{ijk} każdego obrazu I_{ij} wszystkich obiektów O_i widocznych w kamerze docelowej C_B .

W celu uwzględnienia w dalszej analizie zarówno stopnia podobieństwa parametrów lokalnych obrazu analizowanego obiektu widocznego w kamerze C_B do obiekt S jak i wiarygodności odpowiedzi sieci, autor rozprawy proponuje zastosowanie dwóch miar skalarnych: odpowiedzi sieci r_{ijk} oraz wagi tej odpowiedzi w_{ijk} .

Odpowiedź sieci r_{ijk} na wektor wejściowy V_{ijk} jest obliczana zgodnie ze wzorem:

$$r_{ijk} = 0,5 \cdot (o_1 - o_2 + 1), \quad r_{ijk} \in [0, 1] \quad (4.58)$$

gdzie o_i oznacza wartość i -tego elementu wektora wyjściowego sieci przeskalowaną z przedziału $[-1, 1]$ do przedziału $[0, 1]$. Odpowiedź r_{ijk} reprezentuje stopień podobieństwa wektora V_{ijk} do parametrów lokalnych obiektu S , gdzie 0 oznacza podobieństwo minimalne, a 1 – maksymalne.

Do każdej odpowiedzi r_{ijk} przypisywana jest jej waga w_{ijk} , która reprezentuje wiarygodność wyniku klasyfikacji:

$$w_{ijk} = \begin{cases} \frac{\max(o_1, o_2)^2}{o_1 + o_2}, & \text{dla } o_1 + o_2 > 0 \\ 0, & \text{dla } o_1 + o_2 = 0 \end{cases}, \quad w_{ijk} \in [0, 1] \quad (4.59)$$

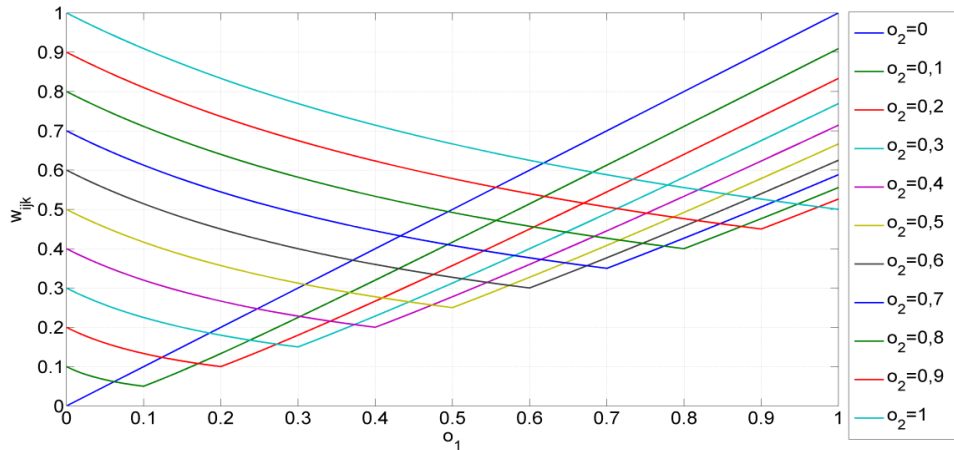
Rodzinę wartości funkcji $w_{ijk}(o_1, o_2)$ pokazano na rys. 4.16. Im większa różnica między wartościami na obu wyjściach sieci i im większa maksymalna wartość na wyjściu sieci, tym większa wiarygodność (pewność) wyniku klasyfikacji.

Odpowiedzi sieci r_{ijk} na wszystkie wektory opisujące wygląd pojedynczego obrazu I_{ij} są następnie agregowane jako średnia ważona w celu wyznaczenia skumulowanej odpowiedzi sieci R_{ij} na j -ty obraz i -tego obiektu:

$$R_{ij} = \frac{\sum_{k=0}^{NV_{ij}} w_{ijk} \cdot r_{ijk}}{\sum_{k=0}^{NV_{ij}} w_{ijk}} \quad (4.60)$$

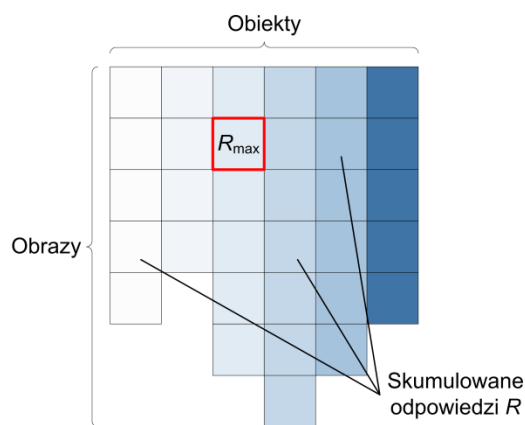
Obiekt O_i zarejestrowany w kamerze docelowej C_B jest uznawany za obiekt S obserwowany w kamerze źródłowej C_A wtedy i tylko wtedy, gdy istnieje skumulowana odpowiedź R dla jednego z jego obrazów, która jest nie mniejsza, niż dowolna inna skumulowana odpowiedź na obraz innego obiektu, zgodnie z regułą:

$$O_i = S \Leftrightarrow \exists_j \forall_{m \neq i} \forall_n (R_{ij} \geq R_{mn}) \quad (4.61)$$



Rys. 4.16 Wartości funkcji wagi w_{ijk} w zależności od wartości na obu wyjściach sieci neuronowej

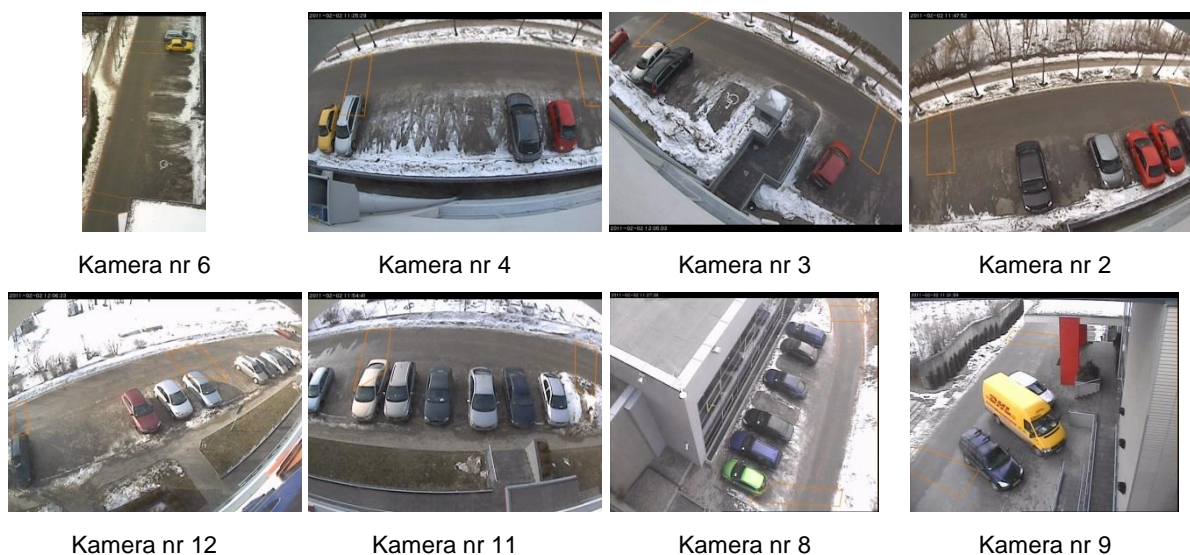
Warunek ten zilustrowano na rys. 4.17. W przypadku, gdy istnieje więcej, niż jeden obiekt spełniający warunek (4.61), obrazy z identycznymi wartościami skumulowanej odpowiedzi R są odrzucane. Następnie ponownie stosowane jest kryterium (4.61), jednak już tylko w stosunku do najbardziej podobnych obiektów, czyli tych których obrazy charakteryzowały się tą samą wartością skumulowanej odpowiedzi R i zostały usunięte z dalszej analizy. W przypadku wielu sieci neuronowych (wielu obiektów nadesłanych z kamer źródłowych), które zostały dopasowane do tego samego obiektu w kamerze docelowej, jako wynik klasyfikacji wybierany jest ten obiekt źródłowy, dla którego miara odpowiedzi skumulowanej jest najwyższa.



Rys. 4.17 Ilustracja procesu reidentyfikacji obiektu na podstawie wartości skumulowanej odpowiedzi sieci neuronowej

4.4.4 Wyniki eksperymentów

Eksperymenty związane z wielokamerowym śledzeniem obiektów zrealizowano z wykorzystaniem instalacji testowej zbudowanej w Gdańskim Parku Naukowo-Technologicznym, która została scharakteryzowana w rozdziale 6.1.3. Składa się ona z 8 kamer stacjonarnych, które są różnie zorientowane w stosunku do drogi, którą obserwują (rys. 4.18). Odległości między polami widzenia kamer wynoszą od kilkudziesięciu centymetrów do kilkudziesięciu metrów (np. zakręt w lewo o 90° między kamerami nr 8 i 11). Istnieje tylko jedna droga od wjazdu na parking do wyjazdu z monitorowanego obszaru; droga ta przecina pola widzenia wszystkich kamer. Wielokąty oznaczające położenie obszarów wejścia/wyjścia w polach widzenia wszystkich kamer zostały oznaczone ręcznie (rys. 4.18).



Rys. 4.18 Przykładowe ramki obrazu z kamer (wraz z ich numerami) użytych w eksperymentach związanych z wielokamerowym śledzeniem pojazdów; obszary wejścia/wyjścia w polu widzenia każdej kamery zaznaczono za pomocą pomarańczowych wielokątów

Na potrzeby eksperymentów przeanalizowano 10 godzin nagrań z każdej kamery, zarejestrowanych w czasie jednego dnia roboczego. Analizowane były wyłącznie obrazy pojazdów. Każdy pojazd był automatycznie śledzony w trakcie jego poruszania się w polu widzenia poszczególnych kamer, w celu zgromadzenia bazy danych obrazów pojazdów. Powiązanie ze sobą tych samych pojazdów widzianych przez różne kamery odbyło się półautomatycznie – ścieżki ruchu zostały wstępnie określone z wykorzystaniem globalnych reguł filtracji przestrzenno-czasowej (rozdział 4.4.1), a następnie zweryfikowane i poprawione ręcznie.

Zgromadzona baza danych zawiera 12274 obrazów 246 różnych pojazdów poruszających się po monitorowanym obszarze. Każdy z tych pojazdów przynajmniej raz przemieścił się pomiędzy polami widzenia dwóch sąsiednich kamer. Przykładowe obrazy dwóch pojazdów widzianych przez wszystkie kamery pokazano na rys. 4.19. Wyraźnie widoczne są duże różnice w pozie i w wyglądzie pojazdu obserwowanego przez różne kamery. Numeracja kamer nie jest ciągła, gdyż w skład instalacji testowej wchodzi też inne kamery, w tym obrotowe, które nie były użyte w eksperymentach.

W celu pełnej oceny algorytmu reidentyfikacji obiektów z wykorzystaniem ich cech wizualnych i sztucznych sieci neuronowych, oddzielna sieć neuronowa była trenowana dla każdego pojazdu pojawiającego się w każdej kamerze, a następnie wykorzystywana do rozpoznania tego pojazdu we wszystkich innych kamerach, w których się on pojawił, a nie tylko w kamerze kolejnej, zgodnie z kierunkiem poruszania się pojazdu. Pozytywne wzorce treningowe były tworzone z wykorzystaniem obrazów pojazdu S w kamerze źródłowej C_A . Obrazy wszystkich innych pojazdów, które pojawiły się w polu widzenia tej kamery w ciągu ostatnich $T = 10$ minut tworzyły wzorce negatywne. W procesie rozpoznawania, pozytywny zbiór wektorów został utworzony z obrazów pojazdu S w kamerze docelowej C_B , $C_A \neq C_B$. Aby pozyskać negatywne przykłady wektorów testowych, posłużono się obrazami wszystkich pojazdów, które zostały wykryte w kamerze C_B w ciągu kolejnych $T = 10$ minut. Jeśli nie został znaleziony co najmniej jeden pojazd, który mógłby zostać zaliczony do negatywnego zbioru uczącego lub testowego, obliczenia dla trójki uporządkowanej (S, C_A, C_B) były pomijane. Dodatkowo upewniono się, że negatywne wzorce treningowe i testowe nie zawierają obrazów tych samych pojazdów, w związku z czym w procesie rozpoznawania zadaniem sieci neuronowej było wykrycie poprawnego pojazdu spośród grupy innych, których obrazy na pewno nie były wykorzystane w procesie treningu.



Rys. 4.19 Przykładowe obrazy pojazdu wjeżdżającego (górny rząd) i wyjeżdżającego (dolny rząd) z monitorowanego terenu zarejestrowane przez wszystkie kamery użyte w eksperymentach; numer kamery pokazano pod obrazami

Wykorzystując zgromadzoną bazę obrazów pojazdów dokonano łącznie 5805 reidentyfikacji; w związku z niezalezieniem pojazdów, które mogłyby wejść w skład negatywnych przykładów zbioru uczącego lub testowego, nie było możliwe przeprowadzenie 409 zadań rozpoznawania. Tab. 4.5 prezentuje szczegóły dotyczące liczebności zbiorów użytych do treningu sieci neuronowych, a następnie do klasyfikacji. W skład negatywnych wzorców treningowych i testowych weszło średnio 6-7 pojazdów. Każdy pojazd był reprezentowany przez 9-10 obrazów. Oznacza to, że każda sieć była trenowana z wykorzystaniem znacznie większej liczby wzorców negatywnych, niż pozytywnych, a jej zadaniem było odnalezienie właściwego obiektu w zbiorze 9-10 pojazdów. Tak trudne warunki działania stanowią duże wyzwanie i w praktycznej instalacji nie mają szansy wystąpienia przy uwzględnieniu przestrzennych i czasowych relacji związanych z rozmieszczeniem kamer, pozwalają jednak znaleźć dolne oszacowanie możliwych do uzyskania wyników działania prezentowanego algorytmu.

Tab. 4.5 Średnie liczby pojazdów (P) i obrazów na jeden pojazd (O) w zbiorach uczącym i testowym przypadające na jedną sieć neuronową

Zbiór uczący				Zbiór testowy			
Wzorce pozytywne		Wzorce negatywne		Wzorce pozytywne		Wzorce negatywne	
P	O	P	O	P	O	P	O
1,0	9,4	6,1	9,8	1,0	9,4	7,5	9,7

Tab. 4.6 przedstawia zbiorcze wyniki reidentyfikacji pojazdów dla każdej kamery traktowanej jako źródłowa lub docelowa. Najgorsze wyniki (poniżej 50% poprawnych klasyfikacji) zanotowano dla kamer nr 6, 8, 9 i 12 potraktowanych jako kamery źródłowe i kamer 6, 8 i 9 użytych jako kamery docelowe. Analizując rys. 4.18 oraz rys. 4.19 można dostrzec, że kamery 6, 8 i 9 charakteryzują się zdecydowanie odmienną orientacją w stosunku do drogi w porównaniu z pozostałymi kamerami. Większość kamer obserwuje pojazdy z dużego kąta i z boku, podczas gdy oś optyczna kamer 6, 8 i 9 jest zorientowana wzdłuż osi jezdni. Ponadto, w przypadku tego samego pojazdu, kamery 6 i 9 rejestrują jego obraz od przodu podczas gdy kamera nr 8 obserwuje go od tyłu, i odwrotnie, w zależności od kierunku poruszania się pojazdu. I w końcu, kąt widzenia kamery nr 9 jest bardzo mały podczas gdy kamera 8 jest skierowana niemal prostopadle w dół. Takie różnice w ustawieniu kamer sprawiają, że obserwowane przez nie obrazy tych samych obiektów bardzo istotnie się różnią. Wyniki pokazane w tab. 4.6 dowodzą, że opracowany algorytm nie jest w stanie poradzić sobie z tak drastycznymi różnicami w wyglądzie

pojazdów. W związku z tym, wyniki uzyskane z użyciem kamer 6, 8 i 9 zostały wyłączone z dalszej analizy.

Tab. 4.6 Zbiorcze wyniki reidentyfikacji pojazdów dla każdej kamery użytej jako źródłowa lub docelowa (wykorzystano deskryptory SURF-64) z wyróżnionymi wynikami z liczbą poprawnych klasyfikacji ponad 50%

Numer kamery	Kamera wykorzystana jako źródłowa			Kamera wykorzystana jako docelowa		
	Liczba klasyfikacji	Poprawne klasyfikacje [liczba / %]		Liczba klasyfikacji	Poprawne klasyfikacje [liczba / %]	
2	912	681	74,7%	911	582	63,9%
3	937	635	67,8%	939	520	55,4%
4	923	644	69,8%	924	563	60,9%
6	868	340	39,2%	875	345	39,4%
8	533	166	31,1%	535	225	42,1%
9	704	275	39,1%	709	315	44,4%
11	454	245	54,0%	447	298	66,7%
12	376	134	35,6%	367	252	68,7%

W tab. 4.7 pokazano szczegółowe wyniki reidentyfikacji pojazdu dla każdej pary pozostałych kamer. Można dostrzec, że algorytm jest w stanie poprawnie poradzić sobie z obrotem obrazu pojazdu w kamerze nr 3, w porównaniu z kamerami sąsiednimi (2 i 4), pozwalając uzyskać skuteczność klasyfikacji dla każdej pary wśród tych trzech kamer wynoszącą ponad 85% (dla deskryptora SURF-64). Najgorsze wyniki uzyskano dla kamery 12, której orientacja w stosunku do drogi różni się w największym stopniu, w porównaniu do pozostałych kamer.

Wyniki uzyskane przy wykorzystaniu obu deskryptorów wyglądu obiektów (SURF-64 i SURF-128) są zbliżone (rys. 4.20). Jednak skuteczność klasyfikacji dla najgorszego przypadku jest niższa przy wykorzystaniu wektorów SURF-128. Biorąc pod uwagę, że deskryptory SURF-128 zawierają dwa razy więcej danych, co przekłada się na dłuższy czas trwania treningu sieci neuronowej i skutkuje większym rozmiarem struktury wytrenowanej sieci, do działania w warunkach rzeczywistych lepiej nadają się parametry obrazu bazujące na wektorach SURF-64.

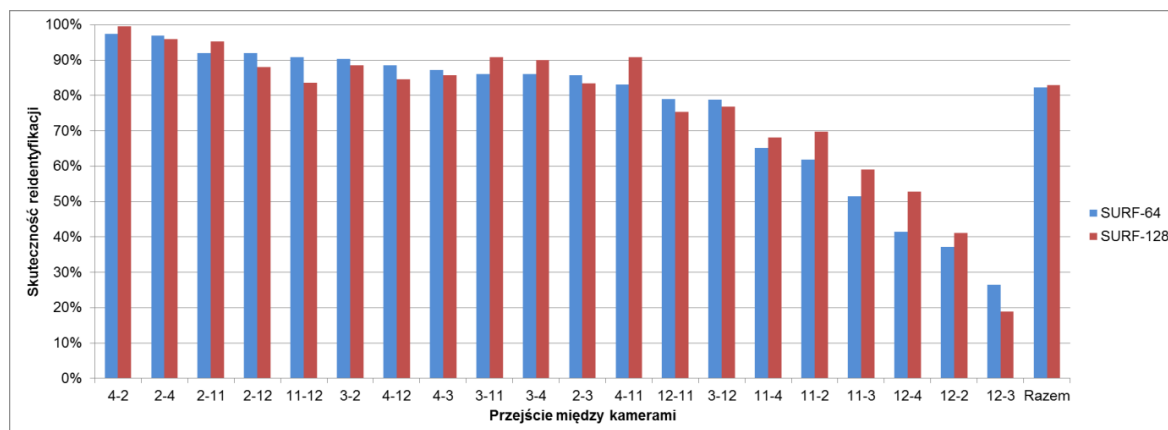
Przeprowadzone eksperymenty zakładają, że obrazy poszukiwanego pojazdu znajdują się w zbiorze testowym. Zadanie wykrywania brakujących obiektów (nieobecnych w polu widzenia kamery docelowej w czasie reidentyfikacji, np. wskutek przemieszczenia się obiektu do pola widzenia innej kamery docelowej) nie było przedmiotem prowadzonych badań. Jednakże zaobserwowano, że wartości skumulowanych odpowiedzi sieci neuronowej R dla pozytywnych wzorców testowych są wyraźnie wyższe, niż odpowiedzi dla wzorców negatywnych (tab. 4.8). Ta własność

mogłaby być wykorzystana w celu określenia minimalnego progu na wartość skumulowanej odpowiedzi R dla dopasowanego obiektu. Niestety, w oparciu o wykorzystaną instalację testową (jedna, liniowa droga łącząca pola widzenia wszystkich kamer), nie jest możliwe eksperymentalne zweryfikowanie tej tezy.

Tab. 4.7 Wyniki reidentyfikacji pojazdów dla każdej pary kamer i dwóch różnych deskryptorów obrazu z wyróżnionymi wynikami dla kamer nr 2, 3 i 4

Przejście między kamerami	Liczba klasyfikacji	SURF-64		SURF-128	
		Poprawne klasyfikacje [liczba / %]		Poprawne klasyfikacje [liczba / %]	
2 → 3	218	187	85,78	182	83,49
2 → 4	194	188	96,91	186	95,88
2 → 11	63	58	92,06	60	95,24
2 → 12	50	46	92,00	44	88,00
3 → 2	217	196	90,32	192	88,48
3 → 4	202	174	86,14	182	90,10
3 → 11	65	56	86,15	59	90,77
3 → 12	52	41	78,85	40	76,92
4 → 2	195	190	97,44	194	99,49
4 → 3	203	177	87,19	174	85,71
4 → 11	65	54	83,08	59	90,77
4 → 12	52	46	88,46	44	84,62
11 → 2	63	39	61,90	44	69,84
11 → 3	66	34	51,52	39	59,09
11 → 4	66	43	65,15	45	68,18
11 → 12	55	50	90,91	46	83,64
12 → 2	51	19	37,25	21	41,18
12 → 3	53	14	26,42	10	18,87
12 → 4	53	22	41,51	28	52,83
12 → 11	57	45	78,95	43	75,44
All	2040	1679	82,30	1692	82,94

Skuteczność algorytmu reidentyfikacji obiektów widzianych przez różne kamery monitoringu jest silnie zależna od różnic w ustawieniu i orientacji przestrzennej obu kamer wykorzystywanych w procesie dopasowywania. W przypadku istotnych różnic (np. wzajemnie prostopadłe kierunki widzenia kamer) opracowany algorytm zawodzi z powodu zbyt dużych rozbieżności w wyglądzie obserwowanych obiektów. Jednak przy zbliżonym ustawieniu kamer i przy uwzględnieniu zależności czasowo-przestrzennych między sąsiednimi kamerami, przedstawione rozwiązanie nadaje się do stosowania warunkach rzeczywistych.



Rys. 4.20 Wyniki reidentyfikacji pojazdów dla każdej pary kamer i dwóch różnych deskryptorów obrazu

Tab. 4.8 Statystyka wartości skumulowanej odpowiedzi sieci neuronowej R dla pojazdów pozytywnych (poprawnych) i negatywnych (niepoprawnych) znajdujących się w zbiorze testowym

Pary kamer	Pozytywne wzorce testowe		Negatywne wzorce testowe	
	Wartości R [średnia / odchylenie standardowe]		Wartości R [średnia / odchylenie standardowe]	
Wszystkie	0,3061	0,1223	0,1670	0,0660
Bez kamer nr 6, 8, 9	0,4268	0,1065	0,1654	0,0796

5. Opracowanie algorytmu klasyfikacji rodzaju pojazdu

Niniejszy rozdział przedstawia algorytm klasyfikacji typu pojazdu nie wymagający skalibrowanego pola widzenia kamery. Zasadniczym celem algorytmu jest podział wszystkich pojazdów na trzy kategorie:

- samochody osobowe,
- furgonetki/busy,
- samochody ciężarowe (w tym z otwartą lub zabudowaną skrzynią oraz ciągniki siodłowe z naczepami lub bez nich).

Przykładowe ujęcia pojazdów każdego typu pokazano na rys. 5.1.



Rys. 5.1 Przykładowe obrazy pojazdów rozpoznawanych typów

W niniejszej pracy nacisk położono na zagadnienia związane z określeniem typu pojazdu, jednak ze względu na konieczność działania algorytmów w warunkach rzeczywistych, w rozdziale 5.1 opisano wykorzystywany algorytm wykrywania pojazdów spośród wszystkich obiektów poruszających się w polu widzenia kamery. Kolejne rozdziały poświęcone są już rozpoznawaniu typu pojazdu: wykorzystywane parametry obrazu pojazdów omówiono w rozdziale 5.2, rozdział 5.3 przedstawia wykorzystane klasyfikatory, a rozdział 5.4 dokumentuje przeprowadzone eksperymenty i ich wyniki.

5.1 Wykrywanie pojazdu

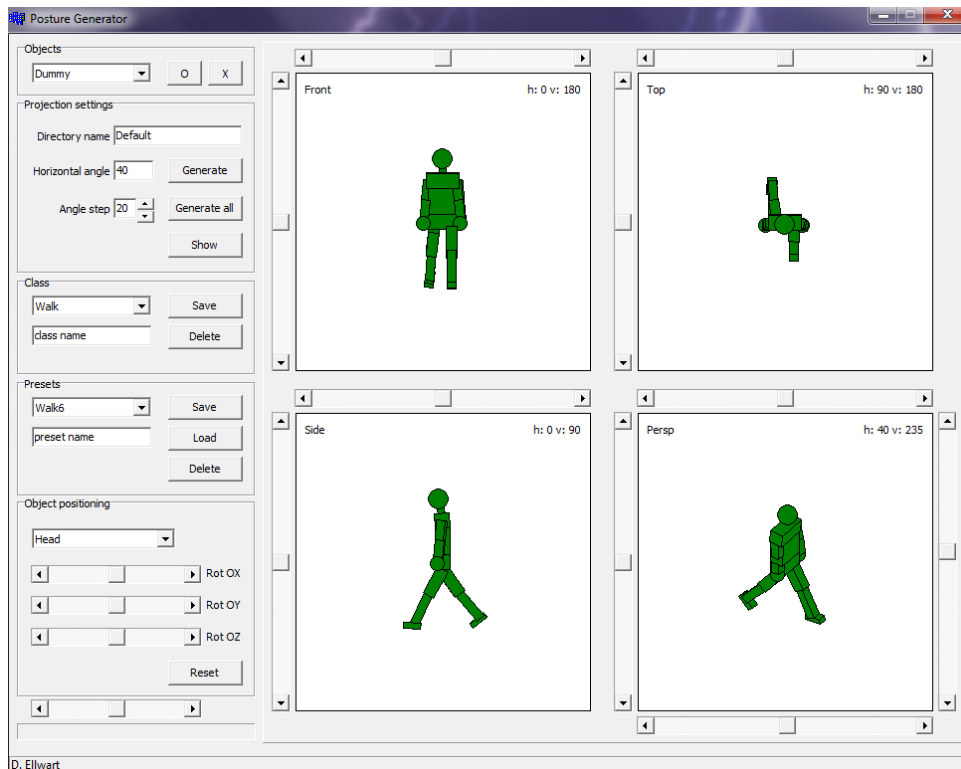
W rozdziale 3.3.1 przedstawiono przegląd stosowanych rozwiązań mających na celu klasyfikację typu obiektu w strumieniu wizyjnym, pozwalających na odróżnienie pojazdów od innych ruchomych obiektów, w szczególności od osób. Najbardziej popularne metody bazują na parametryzacji kształtu obiektu, których główną zaletą jest uniwersalność i niezależność od zmian wyglądu obiektu, związanych z warunkami oświetleniowymi. Jednak metody te muszą uwzględniać różnice w sylwetce rozpoznawanych obiektów wynikające z fazy ruchu i ustawienia w stosunku do kamery.

Niniejszy rozdział prezentuje opracowany w Katedrze Systemów Multimedialnych, przy udziale autora rozprawy, algorytm klasyfikacji obiektów za pomocą systemu decyzyjnego, który w procesie treningu wykorzystuje trójwymiarowe modele obiektów do wygenerowania różnorodnych sylwetek obiektów i wyznaczenia parametrów ich kształtów [78]. Kolejne dwa rozdziały prezentują opracowane rozwiązanie oraz przedstawiają wyniki eksperymentów dotyczących oceny jego działania.

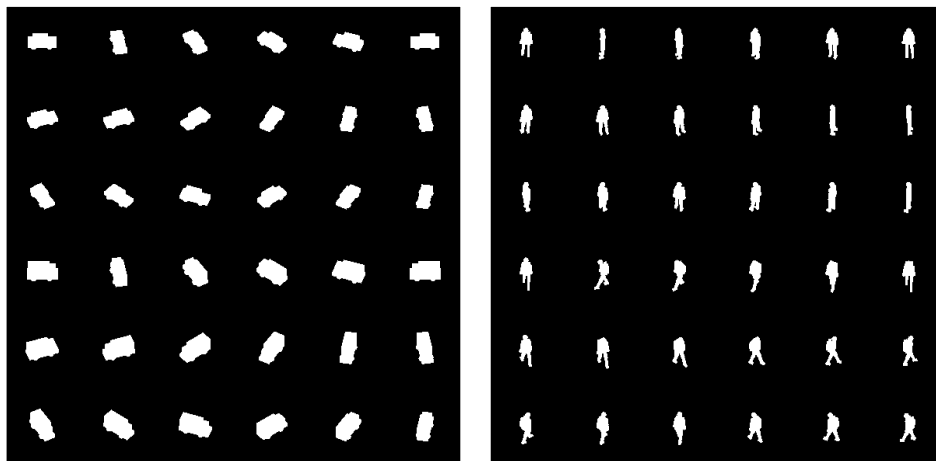
5.1.1 Algorytm klasyfikacji obiektów

Algorytm klasyfikacji obiektów wykorzystuje ich trójwymiarowe modele, które są następnie używane do wygenerowania odpowiednich wzorców treningowych dla różnych wariantów obrotu obiektu wokół osi pionowej i kąta widzenia kamery (kąt między osią optyczną kamery, a powierzchnią ziemi). Stworzono modele 3D reprezentujące dwie klasy bazowe: osoby i pojazdy (rys. 5.2). W przypadku pojazdów użyto 6 różnych modeli uwzględniających różne typy samochodów. Sylwetki osób mogą ulegać znacznie większym zmianom, dlatego w tym przypadku zastosowano 30 modeli reprezentujących osoby w różnych pozach, fazach marszu i niosących rozmaite przedmioty (np. torba, plecak). Wszystkie modele były następnie rzutowane na płaszczyznę pola widzenia kamery dla 24 różnych kątów obrotu obiektu wokół osi pionowej (co 15°) i dla wybranych kątów widzenia kamery. Daje to łącznie 144 różnych kształtów pojazdów i 615 sylwetek osób dla jednego kąta widzenia kamery; w przypadku osób usunięto te sylwetki, które były praktycznie identyczne. Przykładowe kształty obiektów pokazano na rys. 5.3. Wstępne eksperymenty wykazały, że dla pokrycia wszystkich, typowo stosowanych kątów widzenia kamery wystarczające jest przygotowanie bazy danych dla kątów 20° , 40° i 60° . Dodatkowo, na potrzeby treningu klasyfikatora, przygotowano obrazy 290 kształtów nie

należących do żadnej z dwóch rozróżnianych klas, które zostały wybrane spośród wyników generowanych przez algorytm detekcji ruchomych obiektów w obrazie.



Rys. 5.2 Okno aplikacji do generowania dwuwymiarowych sylwetek wybranych trójwymiarowych modeli obiektów dla dowolnych kątów widzenia



Rys. 5.3 Przykładowe sylwetki pojazdów (z lewej) i osób (z prawej) wygenerowane na podstawie modeli trójwymiarowych dla różnych kątów widzenia

Wygenerowane maski obiektów (na potrzeby treningu klasyfikatora) lub maski uzyskane automatycznie z wykorzystaniem algorytmów detekcji i śledzenia obiektów ruchomych (w procesie klasyfikacji prowadzonej w czasie rzeczywistym) są

parametryzowane w celu zmniejszenia nadmiarowości danych. Proces parametryzacji zilustrowano na rys. 5.4. W pierwszym etapie, maska binarna będąca wynikiem detekcji obiektów ruchomych jest skalowana do stałej rozdzielczości w celu uzyskania niezależności od wielkości obiektu, przy zachowaniu oryginalnych proporcji w taki sposób, aby dłuższy wymiar wynosił 50 pikseli. Następnie wyznaczany jest środek ciężkości maski (c_x, c_y) , a maska dzielona jest na przedziały o określonej szerokości kątowej i odległości od środka ciężkości. Użycie 10 przedziałów kątowych (z krokiem $\Delta_\alpha = 36^\circ$) i 10 przedziałów odległościowych (z krokiem $\Delta_r = 2$ pikseli) skutkuje podziałem maski na 100 podobszarów. Pixel maski (x, y) należy do przedziału zdefiniowanego przez parę uporządkowaną (r_{bin}, α_{bin}) , $r_{bin} = 0, 1 \dots 9$, $\alpha_{bin} = 0, 1 \dots 9$, jeśli spełnione są zależności:

$$r_{bin} = \begin{cases} \left\lfloor \frac{\left(\sqrt{(x - c_x)^2 + (y - c_y)^2} \right)}{\Delta_r} \right\rfloor & \text{dla } \sqrt{(x - c_x)^2 + (y - c_y)^2} \leq 9 \cdot \Delta_r \\ 9 & \text{dla } \sqrt{(x - c_x)^2 + (y - c_y)^2} > 9 \cdot \Delta_r \end{cases} \quad (5.1)$$

$$\alpha_{bin} = \left\lfloor \frac{\text{atan2}(y - c_y, x - c_x)}{\Delta_\alpha} \right\rfloor \quad (5.2)$$

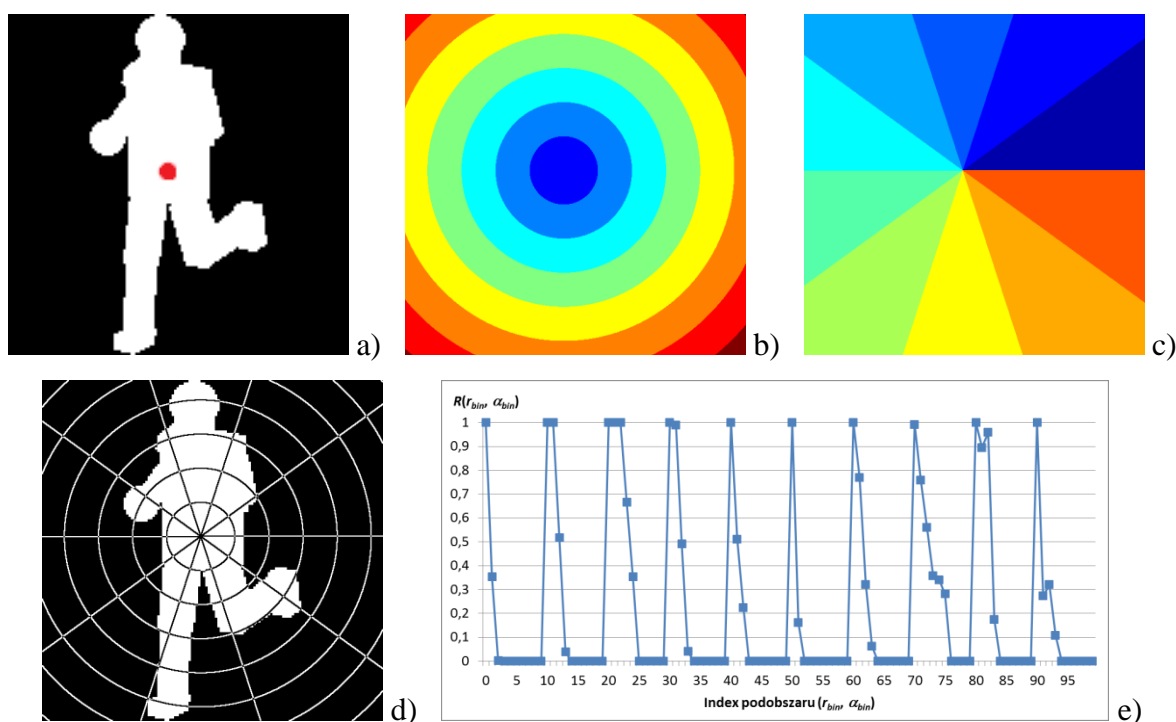
gdzie atan2 oznacza funkcję arcus tangens czteroćwiartkowy.

Następnie w każdym podobszarze (r_{bin}, α_{bin}) liczony jest udział $R(r_{bin}, \alpha_{bin})$ pikseli należących do maski w stosunku do wszystkich pikseli w obszarze. Udziały w poszczególnych podobszarach serializowane są wg schematu: najpierw wszystkie przedziały radialne odnoszące się do jednego przedziału kątowego. Przedziały radialne liczone są od COG obiektu, a przedziały kątowe w kierunku przeciwnym do wskazówek zegara, zaczynając od dodatniej półosi X. W ten sposób otrzymywany jest 100-elementowy wektor definiujący kształt analizowanego obiektu.

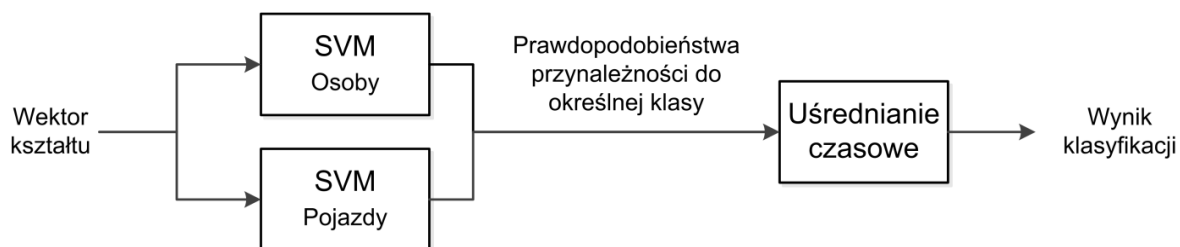
Jako algorytm decyzyjny, w procesie klasyfikacji wykorzystano maszynę wektorów nośnych (ang. *Support Vector Machines*, SVM), która w tym zastosowaniu charakteryzowała się największą stabilnością i powtarzalnością wyników, w porównaniu z innymi klasyfikatorami (ANN, Random Forest). Zadaniem klasyfikatora SVM jest znalezienie hiperpłaszczyzny, która oddziela dwa zbiory danych z możliwie największym marginesem między nimi. Wykorzystany klasyfikator bazuje na jądrze w postaci radialnej funkcji bazowej (ang. *Radial Basis Function*, RBF) danej wzorem [109]:

$$K(\mathbf{x}, \mathbf{x}') = e^{-\gamma \|\mathbf{x} - \mathbf{x}'\|} \quad (5.3)$$

gdzie $K(\mathbf{x}, \mathbf{x}')$ jest funkcją jądra definiującą przestrzeń parametrów, a γ definiuje szerokość radialnej funkcji bazowej. Optymalną wartości dla parametrów kosztu i γ klasyfikatora SVM znajdowane są z wykorzystaniem metody *Grid-Search* [109]. Proces klasyfikacji odbywa się zgodnie z zasadą jeden kontra wszystkie (ang. *One vs All*, OVA), co oznacza, że dla dwóch głównych rozróżnianych klas (osoby, pojazdy) trenowane są dwa klasyfikatory binarne (rys. 5.5). Równoległe budowany jest model statystyczny w celu uzyskania wartości prawdopodobieństw, że dany obiekt należy do określonej klasy [27].



Rys. 5.4 Ilustracja procesu parametryzacji sylwetki obiektu: a) przykładowy kształt do parametryzacji z oznaczonym środkiem ciężkości, b) przedziały radialne r_{bin} , c) przedziały kątowe α_{bin} , d) podobszary (r_{bin}, α_{bin}) do wyznaczenia wartości deskryptora, e) wartości $R(r_{bin}, \alpha_{bin})$ stanowiące wektor definiujący kształt obiektu



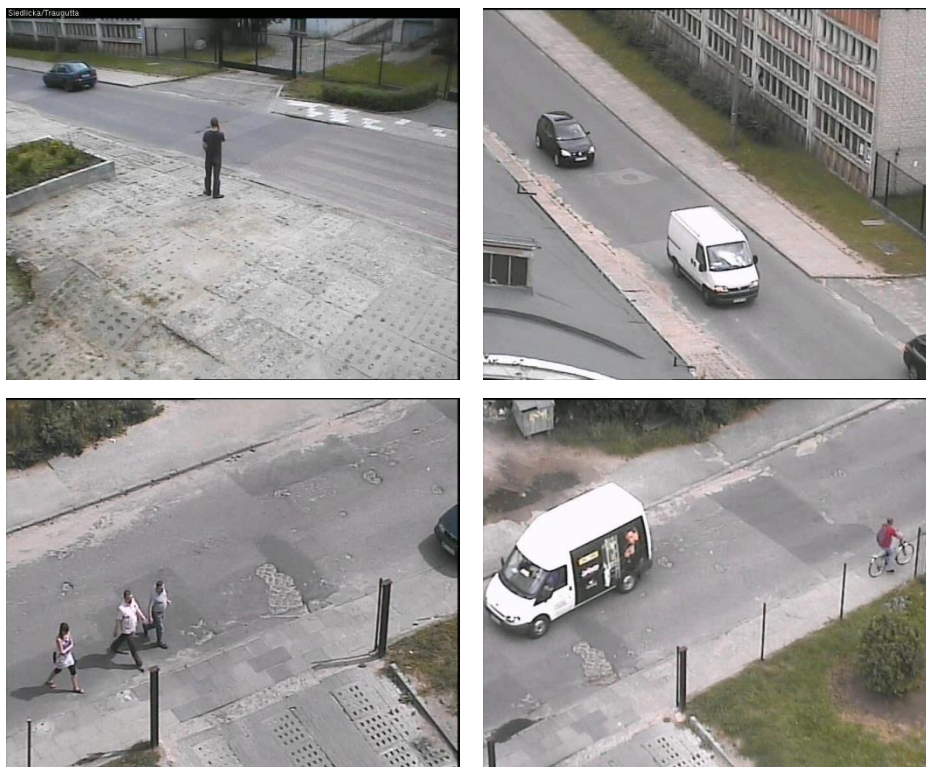
Rys. 5.5 Schemat procesu klasyfikacji

Proces klasyfikacji obiektu składa się z dwóch etapów. W pierwszym, obliczane są wektory cech wszystkich obiektów znajdujących się w bieżącej ramce obrazu, które następnie są rozpoznawane przez każdy z klasyfikatorów. W efekcie uzyskuje się wektor prawdopodobieństw przynależności konkretnego obiektu do każdej z rozpoznawanych klas. Wektory prawdopodobieństw są przechowywane przez czas życia obiektu, a jako wynik klasyfikacji w bieżącej ramce obrazu zwracana jest klasa z największą (jak dotychczas) średnią wartością prawdopodobieństwa. Jeśli nie przekracza ona progu 0,85, to obiekt przypisywany jest do kategorii „inne”. Zastosowanie uśredniania czasowego pozwala na uodpornienie wyników klasyfikacji na chwilowe zniekształcenia uzyskiwanych masek wywołane np. wzajemnym przesłanianiem się obiektów. Dodatkowo wprowadzono wymóg związany z czasem obserwacji obiektu: wynik jest dostępny w sytuacji, gdy obiekt znalazł się w całości w polu widzenia kamery (analiza częściowego kształtu obiektu prowadziła by do błędnych wyników) i przebywał w nim przez co najmniej jedną sekundę.

5.1.2 Walidacja poprawności działania

Przeprowadzone eksperymenty miały na celu sprawdzenie skuteczności klasyfikacji typu obiektów dla różnych kątów widzenia kamery. W tym celu dokonano analizy 6 godzin materiału wizyjnego. Nagrania zawierały samochody, pojedyncze osoby, grupy osób oraz rowerzystów. Oczekiwany wynik klasyfikacji to przydzielenie wszystkich cztero-kołowych obiektów do klasy „pojazd”. Do klasy „osoba” powinny być przydzielone tylko sylwetki pojedynczych osób, gdyż w ten właśnie sposób zdefiniowano obiekty tej klasy w zestawie treningowym. Do wykrycia grupy osób wymagane byłoby zastosowanie oddzielnych algorytmów segmentacji i rekonstrukcji obrazu, co biorąc pod uwagę przeznaczenie tego algorytmu klasyfikacji (rozdzielenie pojazdów od innych ruchomych obiektów) nie było konieczne. Obiekty nie należące do żadnej z dwóch zdefiniowanych klas powinny być klasyfikowane jako „inne”. Przykładowy film ilustrujący wyniki detekcji „klasyfikacja.avi” umieszczono na płycie dołączonej do rozprawy.

Ekspertymenty zostały przeprowadzone dla nagrań charakteryzujących się dwoma różnymi kątami widzenia kamery: 20° i 60° (rys. 5.6). Liczebności obiektów poszczególnych klas pokazano w tab. 5.1.



Rys. 5.6 Przykładowe ramki obrazu z eksperymentalnych nagrań dla kątów widzenia 20° (górny wiersz) i 60° (wiersz dolny)

Tab. 5.1 Liczebności obiektów poszczególnych klas w dwóch typach nagrań użytych w eksperymentach

Kąt widzenia kamery w nagraniach	Liczba osób	Liczba pojazdów	Liczba innych obiektów	Łączna liczba obiektów
20°	382	300	684	1366
60°	130	195	187	512

W tab. 5.2 i tab. 5.3 przedstawiono uzyskane wyniki klasyfikacji obiektów. Wynika z nich, że skuteczność rozpoznawania osób i pojazdów waha się w przedziale 85-95%. Można przy tym zauważyć, że różnica w uzyskiwanych wynikach w zależności od kąta widzenia kamery jest stosunkowo niewielka i wynosi ok. 2 punkty procentowe w przypadku osób i ok. 8 punktów procentowych dla pojazdów. Ponadto korzystny jest fakt, że zdecydowana większość błędów klasyfikacji polega na zidentyfikowaniu osoby lub pojazdu jako inny obiekt; rzadko pojazdy i osoby mylone są wzajemnie ze sobą. Niestety, zwiększanie kąta widzenia kamery prowadzi do ogólnego pogorszenia uzyskiwanych wyników i istotnego zwiększenia błędów klasyfikacji obiektów nie będących ani pojazdami, ani osobami. Wynika to z faktu, że podobieństwo sylwetek obiektów obserwowanych pod dużym kątem znacznie się zwiększa.

Tab. 5.2 *Macierz błędu klasyfikacji typu obiektu dla kąta widzenia kamery 20°*

Wynik klasyfikacji Rzeczywisty obiekt	Pojazd	Osoba	Inny
Pojazd	94,33%	0,59%	5,08%
Osoba	0,30%	94,20%	5,50%
Inny	1,75%	6,29%	91,96%

Tab. 5.3 *Macierz błędu klasyfikacji typu obiektu dla kąta widzenia kamery 60°*

Wynik klasyfikacji Rzeczywisty obiekt	Pojazd	Osoba	Inny
Pojazd	86,32%	1,89%	11,79%
Osoba	1,13%	92,72%	6,15%
Inny	2,14%	18,18%	79,68%

Uzyskane wyniki eksperymentów wskazują, że algorytm klasyfikacji typu obiektu bazujący na wzorcach sztucznie wygenerowanych za pośrednictwem trójwymiarowych modeli obiektów jest odpowiedni do stosowania w systemach automatycznej analizy obrazu. Zaletą rozwiązania jest brak konieczności kalibracji pola widzenia kamery – należy jedynie oszacować w przybliżeniu kąt widzenia kamery i wybrać odpowiedni klasyfikator z przygotowanej uprzednio bazy wytrenowanych klasyfikatorów. Metoda ta wymaga jednak odpowiednio dużych obrazów sylwetek ruchomych obiektów (zalecane kamery megapikselowe) oraz obrazu niezniekształconego przez obiektyw kamery (np. dystorsje beczkowate wprowadzane często przez obiektywy szerokokątne).

5.2 Deskryptory pojazdów

W celu osiągnięcia maksymalnie skutecznej klasyfikacji rodzaju pojazdu zdecydowano się wykorzystać do tego celu zarówno parametry kształtu pojazdu, jak i jego wyglądu. Wejściem do procedury wyznaczania parametrów jest obraz pojazdu \mathbf{I} oraz tej samej wielkości maska binarna \mathbf{M} , która definiuje kształt pojazdu (piksele należące do pojazdu w obrazie \mathbf{I}) [60].

5.2.1 Parametry kształtu

Na podstawie przedstawionej w rozdziale 3.3.2.1 analizy parametrów opisujących kształt pojazdu do badań wybrano następujące parametry:

- stosunek szerokości prostokąta opisanego na masce pojazdu do jego wysokości
- mimośród elipsy dopasowanej do maski

- stosunek pola prostokąta opisanego na masce do powierzchni maski
- stosunek pola wielokąta wypukłego opisanego na masce do powierzchni maski
- stosunek kwadratu obwodu maski do jej powierzchni
- 10 momentów zwykłych (do 3 rzędu włącznie) $m_{00}, m_{10}, m_{01}, m_{11}, m_{20}, m_{02}, m_{21}, m_{12}, m_{30}, m_{03}$:

$$m_{ij} = \sum_{(x,y) \in M} M(x,y) \cdot x^i \cdot y^j \quad (5.4)$$

- 7 momentów centralnych do 3 rzędu włącznie $mc_{11}, mc_{20}, mc_{02}, mc_{21}, mc_{12}, mc_{30}, mc_{03}$ (momenty $mc_{00} = m_{00}, mc_{10} = mc_{01} = 0$ zostały pominięte):

$$mc_{ij} = \sum_{(x,y) \in M} M(x,y) \cdot (x - \bar{x})^i \cdot (y - \bar{y})^j \quad (5.5)$$

gdzie (\bar{x}, \bar{y}) oznacza środek ciężkości maski:

$$(\bar{x}, \bar{y}) = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (5.6)$$

- 7 unormowanych momentów centralnych do 3 rzędu włącznie $mu_{11}, mu_{20}, mu_{02}, mu_{21}, mu_{12}, mu_{30}, mu_{03}$ (momenty $mu_{00} = 1, mu_{10} = mu_{01} = 0$ zostały pominięte):

$$mu_{ij} = \frac{mc_{ij}}{m_{00}^{1+\frac{i+j}{2}}} \quad (5.7)$$

- zbiór siedmiu momentów niezmienniczych H_u , które są niezależne od przesunięcia, skali i obrotu [82][110]:

$$H_1 = mu_{20} + mu_{02} \quad (5.8)$$

$$H_2 = (mu_{20} - mu_{02})^2 + (2mu_{11})^2 \quad (5.9)$$

$$H_3 = (mu_{30} - 3mu_{12})^2 + (3mu_{21} - mu_{03})^2 \quad (5.10)$$

$$H_4 = (mu_{30} + mu_{12})^2 + (mu_{21} + mu_{03})^2 \quad (5.11)$$

$$H_5 = (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] + (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \quad (5.12)$$

$$H_6 = (\mu_{20} - \mu_{02})[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03}) \quad (5.13)$$

$$H_7 = (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] - (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})[3(\mu_{30} - \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \quad (5.14)$$

Rozmiary obrazów pojazdów poruszających się w polu widzenia kamery zależą od ich odległości od kamery. Ponieważ pole widzenia kamery jest z założenia nieskalibrowane, zniekształcenia te nie mogą być automatycznie korygowane. Dlatego w zestawie parametrów opisujących maskę obiektu brak bezwzględnych wymiarów pojazdu. Jednak rozmiar obiektu jest istotny z punktu widzenia klasyfikacji pojazdów (samochody osobowe są istotnie mniejsze od ciężarowych), dlatego został on niejawnie uwzględniony w momentach statystycznych.

Zbiór wszystkich parametrów opisujących kształt pojazdu zawiera 36 elementów i został oznaczony jako *kształt*.

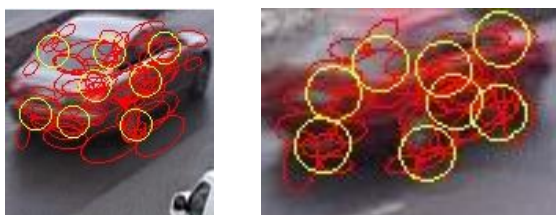
5.2.2 Parametry obrazu

Druga grupa deskryptorów bazuje na wyglądzie pojazdów. Ze względu na brak korelacji między typem pojazdu, a jego barwą, do wyznaczania parametrów obrazu pojazdów wykorzystano tylko ich luminancję. Konwersji z przestrzeni kolorów *RGB* (wejściowej) do obrazu w odcieniach szarości *Y* dokonano za pomocą zależności zdefiniowanej w standardzie Międzynarodowej Unii Telekomunikacyjnej ITU-R (ang. *International Telecommunication Union – Radiocommunication Sector*) BT.601 (inne nazwy: Rec. 601, BT.601, CCIR 601) definiującym standardy w dziedzinie cyfrowego kodowania obrazu [271]:

$$Y = 0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B \quad (5.15)$$

Wykorzystano cztery rodzaje parametrów wyglądu pojazdów; dwa z nich bazują na deskrytorach SURF, a kolejne dwa opierają się na gradiencie obrazu i filtracji Gabora.

Deskryptory SURF (ang. *Speded Up Robust Features*) zostały już przedstawione w rozdziale 4.4.3. W badaniach wykorzystano dwa rodzaje parametrów wyglądu pojazdu bazujących na deskrytorach SURF. W pierwszym z nich punkty charakterystyczne znajdowane są automatycznie z użyciem detektora bazującego na wyznacznikach macierzy Hessego (wzory (4.53) – (4.55)). Dla każdego punktu obliczono deskryptor SURF-64. Ponieważ liczba automatycznie znalezionych punktów charakterystycznych może się istotnie różnić w zależności od analizowanego obrazu (w zakresie od kilku do ponad 100), zachodzi konieczność wyrównania ich liczby. W tym celu dokonano klasteryzacji wszystkich punktów kluczowych za pomocą algorytmu centroidów (ang. *K-Means Algorithm*) [134]. Algorytm ten służy do podziału n wektorów obserwacji na k zbiorów ($k < n$) w taki sposób, aby zminimalizować wewnątrz-klasowy rozrzut zdefiniowany jako suma kwadratów odległości euklidesowych każdego wektora od środka klastra. Wykorzystując algorytm centroidów, wszystkie punkty charakterystyczne dzielone są na 8 klastrów na podstawie ich położenia w obrazie (rys. 5.7). Następnie obliczane są średnie wartości wektorów SURF dla wszystkich punktów charakterystycznych w obrębie każdego klastra. Średnie wektory są uzupełnione o dwie wartości oznaczające położenie środka klastra (współrzędne x i y znormalizowane w stosunku do szerokości i wysokości obrazu pojazdu). Następnie średnie wektory są łączone ze sobą wg położenia środka klastrów (od prawej do lewej i od góry do dołu). W ten sposób uzyskuje się deskryptor wyglądu pojazdu oznaczony jako *surf-8-kmeans*, zawierający $(64 + 2) \times 8 = 528$ elementów.



Rys. 5.7 Położenie automatycznie znalezionych punktów charakterystycznych (czerwone elipsy) oraz środki ośmiu klastrów wyznaczone algorytmem centroidów (żółte okręgi) dla dwóch przykładowych obrazów pojazdów o różnej rozdzielczości

Drugi rodzaj deskryptorów obrazu pojazdu bazujących na lokalnych parametrach wykorzystuje 128-elementowy wektor SURF (SURF-128). Jednak algorytm automatycznego znajdowania punktów charakterystycznych nie jest używany. Zamiast

tego, 4 punkty charakterystyczne umiejscowione są „z góry” w środkach czterech prostokątnych, rozdzielnych i przyległych obszarów, na które dzielony jest obraz pojazdu; obszary te są symetryczne względem środka ciężkości maski pojazdu (rys. 5.8). Rozmiar punktu charakterystycznego jest równy wysokości lub szerokości obszaru, w zależności od tego, która wartość jest większa. Ostatecznie otrzymuje się wynikowy deskryptor zawierający $128 \times 4 = 512$ elementów, oznaczony jako *surf-4*.



Rys. 5.8 Środek ciężkości maski pojazdu (żółty punkt) oraz położenie (i rozmiar) czterech punktów charakterystycznych (czerwone okręgi)

Ostatni rodzaj parametrów wyglądu pojazdów bazuje na filtracji gradientu obrazu przy użyciu banku filtrów Gabora. Gradient obrazu pojazdów liczony jest niezależnie w kierunku poziomym i pionowym za pomocą operatora Sobela z jądrem o rozmiarze 3×3 . Ostateczny obraz gradientu pojazdu I_S uzyskuje się poprzez zsumowanie kwadratów gradientu poziomego i pionowego, a następnie poprzez przeskalowanie wynikowego obrazu do wymiarów 100×80 .

Filtr Gabora jest filtrem liniowym, którego reprezentacje częstotliwościowe i kierunkowe odpowiadają zmysłowi widzenia człowieka [65]. Dwuwymiarowy filtr Gabora ma postać funkcji Gaussa modulowanej przez sinusoidę, zgodnie z wzorami [94]:

$$g_{\lambda, \theta, \varphi, \sigma, \gamma}(x, y) = e^{-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}} \cdot \cos\left(2\pi \frac{x'}{\lambda} + \varphi\right) \quad (5.16)$$

$$x' = x \cdot \cos\theta + y \cdot \sin\theta \quad y' = -x \cdot \sin\theta + y \cdot \cos\theta \quad (5.17)$$

gdzie λ oznacza długość fali składnika harmonicznego, θ reprezentuje orientację prążków filtru Gabora, φ jest wartością przesunięcia fazy, σ charakteryzuje głębokość obwiedni gaussowskiej, a γ jest współczynnikiem eliptyczności funkcji Gabora. W badaniach przyjęto $\gamma = 1$ oraz $\sigma = 0,56 \lambda$. Wymiary $g^x \times g^y$ filtru Gabora wyznaczone są z zależności:

$$g^x = 2 \cdot \max \left(\left| s \cdot \sigma \cos \Theta \right|, \left| s \cdot \frac{\sigma}{\gamma} \sin \Theta \right| \right) + 1 \quad (5.18)$$

$$g^y = 2 \cdot \max \left(\left| s \cdot \sigma \sin \Theta \right|, \left| s \cdot \frac{\sigma}{\gamma} \cos \Theta \right| \right) + 1 \quad (5.19)$$

gdzie $s = 3$ określa rozmiar jądra.

W eksperymentach wykorzystano bank $F = 8$ filtrów Gabora charakteryzowanych przez dwie różne długości fali λ (2,5 i 4) oraz przez 4 różne kierunki Θ (0° , 45° , 90° and 135°). Gradient obrazu pojazdu \mathbf{I}_S filtrowany jest każdym filtrem z banku g przy wykorzystaniu dwóch wariantów przesunięcia fazy γ , zgodnie ze wzorem:

$$\mathbf{I}_G = \sqrt{(g_{\varphi=0} * \mathbf{I}_S)^2 + (g_{\varphi=\pi/2} * \mathbf{I}_S)^2} \quad (5.20)$$

gdzie $x * y$ oznacza operację splotu.

W wyniku obliczeń uzyskuje się $F = 8$ przefiltrowanych obrazów \mathbf{I}_G (rys. 5.9). Każdy z tych obrazów jest następnie dzielony równomiernie w pionie i w poziomie na T równych podobszarów. Przyjęto T równe 16, co w efekcie daje podział każdego obrazu \mathbf{I}_G na 4×4 podobszary. W efekcie każdy podobszar ma rozmiar 25×20 pikseli. Liczba podobszarów jest kompromisem pozwalającym osiągnąć wystarczającą rozdzielczość przestrzenną przy jednocześnie akceptowalnej wymiarowości wynikowego wektora cech. W każdym podobszarze liczona jest suma wartości odpowiednich pikseli obrazu \mathbf{I}_G . Wyniki zapisywane są w macierzy \mathbf{G} o wymiarach $T \times F$, gdzie wiersz oznacza określony filtr Gabora, a kolumna definiuje podobszar obrazu \mathbf{I}_G .

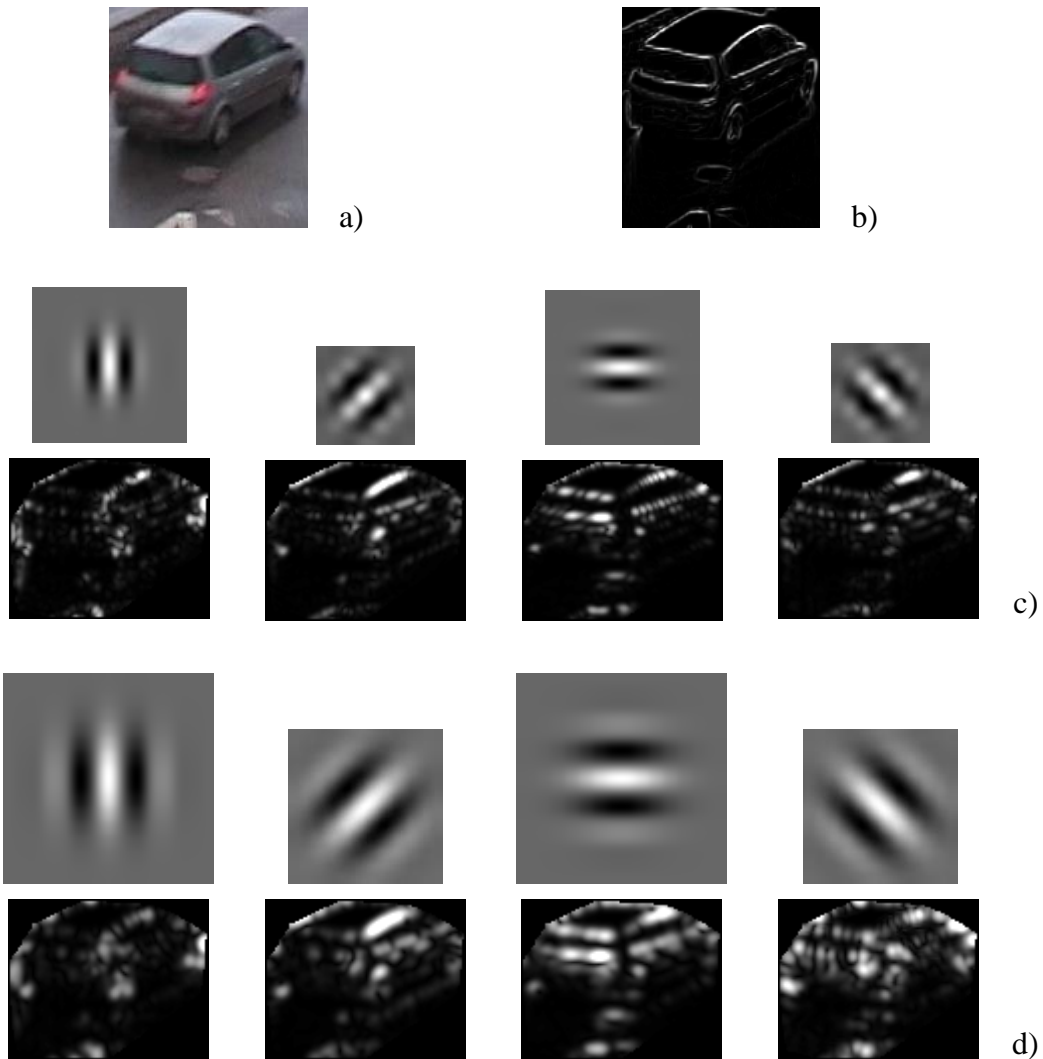
Następnie obliczane są udziały każdego elementu macierzy \mathbf{G} w sumie wszystkich elementów leżących w tej samej kolumnie oraz w tym samym wierszu, co prowadzi do uzyskania dwóch macierzy \mathbf{GO} oraz \mathbf{GF} o rozmiarze $T \times F$, których elementy dane są zależnościami:

$$GO_{m,n} = \frac{G_{m,n}}{\sum_{i=1}^T G_{i,n}} \quad (5.21)$$

$$GF_{m,n} = \frac{G_{m,n}}{\sum_{j=1}^F G_{m,j}} \quad (5.22)$$

Elementy macierzy \mathbf{GO} definiują, niezależnie dla każdego filtru Gabora, udział przefiltrowanych krawędzi w każdym podobszarze obrazu \mathbf{I}_G w odniesieniu do całego

obrazu. Z kolei elementy macierzy \mathbf{GF} zawierają, niezależnie dla każdego podobszaru, udział krawędzi przetworzonych przez określony filtr Gabora w odniesieniu do wszystkich zastosowanych filtrów. Elementy te tworzą, odpowiednio, dwa wektory parametrów nazwane *Gabor-obraz* i *Gabor-filtr*, z których każdy składa się ze 128 elementów.



Rys. 5.9 Obliczanie parametrów obrazu pojazdu z wykorzystaniem filtru Gabora: a) przykładowy obraz pojazdu, b) jego obraz gradientu \mathbf{I}_S oraz wyniki filtracji bankiem filtrów Gabora z parametrem c) $\lambda = 2,5$ i d) $\lambda = 4$; w górnej części c) i d) ilustracja filtru Gabora dla kierunków Θ równych $0^\circ, 45^\circ, 90^\circ$ i 135° , niżej: przefiltrowany obraz gradientu pojazdu \mathbf{I}_G

5.3 Opis algorytmu klasyfikacji

Po ustaleniu parametrów opisujących wygląd pojazdów, konieczne jest wybranie klasyfikatorów oraz sposobu podziału wszystkich wektorów cech na uczące i testujące.

Ten ostatni element jest niezwykle istotny dla jakości wyników eksperymentów ze względu na dużą różnicę liczebności obiektów poszczególnych klas [60].

5.3.1 Wykorzystywane klasyfikatory

Na potrzeby algorytmu klasyfikacji wykorzystano i zbadano przydatność czterech algorytmów decyzyjnych: algorytm K najbliższych sąsiadów (ang. *K Nearest Neighbors*, KNN), sztuczną sieć neuronową (ang. *Artificial Neural Network*, ANN), Random Forest (RF) oraz kaskadę BOOST. Wybrane klasyfikatory charakteryzują się odmienną zasadą działania i użycie ich większej liczby ma na celu empiryczne sprawdzenie, który z nich najlepiej nadaje się do rozwiązania postawionego problemu.

Algorytm najbliższych sąsiadów (KNN) [146] jest najprostszym z użytych klasyfikatorów, dzięki czemu uzyskiwane za jego pomocą wyniki najłatwiej podlegają interpretacji. KNN w procesie treningu zapamiętuje wszystkie wzorce, a następnie dokonuje klasyfikacji nieznanego wektora poprzez znalezienie K najbardziej podobnych wzorców w sensie odległości euklidesowej. Wynikiem klasyfikacji jest kategoria, która pojawia się najczęściej wśród K najbliższych sąsiadów; w przypadku remisu priorytet ma sąsiad najbliższy. W czasie eksperymentów użyto $K = 3$, co pozwala wyeliminować negatywny wpływ pojedynczych, nietypowych obserwacji w zbiorze treningowym, tj. takich próbek, które wyraźnie odstają od pozostałych próbek tej samej kategorii.

Sztuczna sieć neuronowa (ANN) [146] jest zaawansowanym klasyfikatorem, którego najmocniejszą stroną zdolności generalizacyjne. Pozwalają one sieci na wyodrębnienie ze zbioru treningowego w sposób niejawni tylko tych przesłanek, które są istotne z punktu widzenia oczekiwanego wyniku klasyfikacji, co pozwala poprawić skuteczność rozpoznawania wektorów nieznanymi. Sieć neuronowa składa się z określonej liczby warstw neuronów, które są ze sobą połączone synapsami o odpowiednich wagach. W procesie treningu sieci neuronowej (algorytm wstecznej propagacji błędu) wagi wszystkich połączeń są iteracyjnie uaktualniane poprzez analizę gradientu błędu odpowiedzi sieci w funkcji wag sieci w taki sposób, aby zminimalizować błąd odpowiedzi sieci. Liczba wejść sieci i_{ANN} jest równa liczebności wektora cech. Liczba wyjść o_{ANN} odpowiada liczbie rozpoznawanych klas obiektów; wartością oczekiwaną odpowiedzi sieci jest bowiem wektor, w którym jeden element ma wartość maksymalną, a pozostałe – minimalną. Jako wynik klasyfikacji przyjmuje się kategorię skojarzoną z wyjściem sieci o największej wartości.

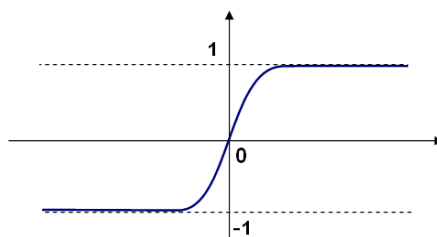
W eksperymentach wykorzystano jednokierunkową sieć neuronową, w pełni połączoną, o jednej warstwie ukrytej. Liczba neuronów w tej warstwie wyznaczana jest zgodnie z zależnością:

$$h_{ANN} = \sqrt{i_{ANN} \cdot (o_{ANN} - 1)} \quad (5.23)$$

Liczba warstw ukrytych i formuła na liczbę neuronów w warstwie ukrytej zostały dobrane empirycznie na podstawie wstępnych badań w taki sposób, aby sieć była w stanie nauczyć się z dużą dokładnością rozpoznawać wektory w zbiorze treningowym, a jednocześnie tak, aby nie zatraciła zdolności generalizacyjnych, co objawiałoby się bardzo małą skutecznością rozpoznawania nieznanymi próbek (sieć neuronowa zaczęłaby działać jak pamięć). Na zjawisko przeuczenia sieci ma duży wpływ właściwy dobór algorytmu treningu sieci i jego parametrów. W badaniach wykorzystano modyfikację algorytmu wstecznej propagacji błędów o nazwie RPROP (ang. *Resilient Back-Propagation*) [195]. W przeciwieństwie do tradycyjnego algorytmu wstecznej propagacji błędów, bierze on pod uwagę jedynie znak każdej składowej gradientu (wartość jest ignorowana), a ponadto współczynnik szybkości adaptacji jest modyfikowany w zależności od stałości znaku gradientu w kolejnych krokach (zwiększany – gdy znaki zgodne, zmniejszany – gdy różne). Pozwala to na znaczne przyspieszenie procesu treningu sieci, szczególnie w przypadku, gdy powierzchnia błędów charakteryzuje się niewielkim nachyleniem. Trening sieci neuronowej był kończony w momencie, gdy wartość błędów treningu odpowiedzi sieci osiągnęła wartość 10^{-6} lub po 5000 iteracjach. Jako funkcję aktywacji neuronów wykorzystano funkcję sigmoidalną bipolarną (tangens hiperboliczny, rys. 5.10) daną wzorem:

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (5.24)$$

Ze względu na stochastyczny proces inicjalizacji wag sieci neuronowej w procesie treningu, sieć neuronowa była trenowana i testowana pięciokrotnie i jako wynik końcowy przyjęto wartości średnie skuteczności klasyfikacji.



Rys. 5.10 Sigmoidalna bipolarna funkcja aktywacji neuronów sieci (tangens hiperboliczny)

Random Forest (RF) jest klasyfikatorem złożonym z wielu drzew decyzyjnych [17]. Pojedyncze drzewo decyzyjne [18] jest modelem predykcyjnym, który łączy dostępne obserwacje z wartościami docelowymi (etykietami) w postaci struktury drzewiastej; liście drzewa odpowiadają etykietom, a kolejne gałęzie prowadzące do określonego liścia tożsame są z koniunkcją warunków, jakie muszą spełniać wartości konkretnych deskryptorów w wektorze parametrów. Drzewo decyzyjne uczone jest w procesie rekurencyjnym, w którym zbiór treningowy dzielony jest wielokrotnie na podzbiory w zależności od spełnienia (lub nie) przez określony deskryptor warunku na jego wartość. Proces dzielenia dla określonego węzła jest kończony, gdy dalsze podziały nie wnoszą już dodatkowej informacji i nie wpływają na wyniki. Każde drzewo jest trenowane niezależnie z losowym przydziałem rozpatrywanych parametrów w każdym węźle drzewa, a wynik końcowy klasyfikacji jest wyznaczany w oparciu o cząstkowe wyniki uzyskane przez każde z drzew. Ze względu na stochastyczny proces treningu lasu losowego, algorytm RF był trenowany i testowany pięciokrotnie i jako wynik końcowy przyjęto wartości średnie skuteczności klasyfikacji.

Kaskada BOOST jest ostatnim użytym algorytmem decyzyjnym i bazuje ona na metodzie wzmacniania klasyfikatorów (ang. *Boosting*) [103]. Koncepcja ta pozwala łączyć ze sobą wiele klasyfikatorów słabych w celu uzyskania złożonego klasyfikatora silnego. Każdy klasyfikator słaby charakteryzowany jest przez skuteczność klasyfikacji niewiele większą, niż losowe odgadywanie i dzięki temu może być on bardzo prosty i niezłożony obliczeniowo. Jako implementację klasyfikatora słabego zastosowano drzewo decyzyjne składające się z jednego węzła, czyli opierającego swój wynik na pojedynczej decyzji dotyczącej wartości parametrów w wektorze wejściowym. W taki sposób skonstruowany klasyfikator złożony może skutecznie konkurować z klasyfikatorami monolitycznymi, jak np. sztuczne sieci neuronowe. Klasyfikator BOOST trenowany jest wg algorytmu AdaBoost (ang. *Adaptive Boost*) [85], zgodnie z którym każdy kolejny człon kaskady

przywiązuje większą wagę do poprawnej klasyfikacji tych wektorów ze zbioru treningowego, które zostały błędnie rozpoznane przez człony wcześniejsze.

Kaskada BOOST jest klasyfikatorem binarnym. Aby móc ją wykorzystać do rozpoznawania trzech i więcej klas, posłużono się zbiorem klasyfikatorów BOOST, których liczba jest równa liczbie klas. Każda kaskada jest trenowana do odróżnienia jednej klasy pojazdów od wszystkich pozostałych klas. Jako wynik końcowy przyjmuje się rezultat klasyfikacji uzyskany przez tę kaskadę BOOST, której suma wyjść składowych klasyfikatorów słabych była największa.

5.3.2 Dobór wektorów uczących i testujących

Dane wejściowe do systemu klasyfikacji stanowią wektory parametrów obliczone dla każdego obrazu pojazdu w bazie danych; każdy pojazd jest reprezentowany przez wiele obrazów, zgromadzonych w trakcie poruszania się pojazdu w polu widzenia kamery. Tymczasem wartością oczekiwaną wyniku klasyfikacji jest etykieta skojarzona z typem pojazdu, do którego należy rozpoznawany obraz. Dlatego szczególnie istotny jest taki sposób podziału wszystkich wektorów parametrów obrazów pojazdów na zbiory: uczący i testowy, który będzie uwzględniał dwa fakty:

- istnieje duża różnica w liczebności pojazdów określonych klas w bazie danych (samochodów osobowych jest znacznie więcej, niż pojazdów wszystkich pozostałych klas)
- istnieje duża różnica w liczbie obrazów każdego pojazdu

W związku z powyższym, podział wektorów jest dwuetapowy: w pierwszym z nich wybierane są pojazdy, które trafią do określonych zbiorów, a następnie wybierane są konkretne obrazy tych pojazdów. W każdym przypadku wybór pojazdów i obrazów ma charakter losowy.

W zbiorze uczącym, każdy typ pojazdów reprezentowany jest przez taką samą liczbę S pojazdów. Wartość S wyznaczana jest na podstawie równania:

$$S = 0,5 \cdot \min_i(N_i) \quad (5.25)$$

gdzie N_i oznacza liczbę pojazdów typu i w bazie danych. Wynika z tego, że o liczbie pojazdów każdego typu w zbiorze uczącym decyduje klasa reprezentowana przez najmniejszą liczbę pojazdów. Wszystkie pozostałe pojazdy zostają przydzielone do zbioru

testowego. Oznacza to, że liczba pojazdów określonego typu w zbiorze testowym jest różna.

Wybór konkretnych obrazów pojazdów do obu zbiorów odbywa się niezależnie dla każdego zbioru, lecz wg identycznego schematu. Każdy typ pojazdu jest reprezentowany przez taką samą liczbę obrazów wybranych spośród obrazów wszystkich pojazdów przydzielonych w poprzednim kroku; wartość ta jest równa najmniejszej licznie obrazów pojazdów każdego typu. W rezultacie zbiory uczące i testowe posiadają, niezależnie od siebie, taką samą liczbę obrazów pojazdów każdego typu.

Wyniki klasyfikacji można przedstawiać na dwa sposoby. W pierwszym z nich, każdy obraz w zbiorze testowym klasyfikowany jest niezależnie. Pozwala to ocenić skuteczność użytych deskryptorów i algorytmów decyzyjnych, ale nie uwzględnia faktu, że interesuje nas typ pojazdu, do którego należy analizowany obraz, a nie typ obrazu *per se*. Dlatego drugi sposób prezentacji wyników uwzględnia fakt, że każdy pojazd w zbiorze testowym jest reprezentowany przez wiele obrazów i sposób ten polega na agregacji wyników klasyfikacji każdego obrazu dla konkretnego pojazdu. W ten sposób jako ostateczne rezultaty przedstawiane są wyniki rozpoznawania pojazdów, a nie ich obrazów niezależnie. Agregacja polega na głosowaniu większościowym: do pojazdu przypisuje się tę klasę, która najczęściej pojawiła się w wynikach rozpoznawania obrazów tego pojazdu; w przypadku remisu założono, że klasyfikacja się nie powiodła.

Ze względu na losowy charakter doboru obrazów pojazdów wchodzących w skład zbiorów uczącego i testowego, wszystkie eksperymenty były powtarzane pięciokrotnie, za każdym razem z innym doбором zbiorów uczących i testowych, a przedstawione wyniki dotyczą uzyskanych wartości średnich.

Dla jednokrotnie ustalonych postaci zbioru treningowego i testowego wyznaczone były wektory parametrów, a następnie przeprowadzana była klasyfikacja za pomocą wszystkich algorytmów decyzyjnych. Pozwala to bezpośrednio porównywać wyniki klasyfikacji uzyskane za pomocą różnych postaci wektora parametrów i różnych klasyfikatorów, gdyż dotyczyły one każdorazowo tego samego zbioru danych.

5.4 Eksperymenty i ich wyniki

5.4.1 Charakterystyka zbioru danych

Eksperymenty z zakresu automatycznej klasyfikacji typu pojazdu przeprowadzono w oparciu o nagrania zrealizowane w różnych miejscach, przy zmiennej orientacji kamery w

stosunku do obserwowanych pojazdów w celu wykazania, że prezentowane rozwiązanie jest niezależne od warunków, w jakich dokonuje się akwizycji obrazu.

Wejście do systemu stanowią obrazy pojazdów wraz z maskami, które oznaczają faktyczne położenie pojazdu w obrazie. Dane te zostały uzyskane automatycznie za pomocą algorytmów detekcji i śledzenia ruchomych obiektów, przedstawionych w rozdziałach 4.1 i 4.2. Każdy pojazd reprezentowany jest przez zbiór obrazów pozyskanych w trakcie przemieszczania się tego pojazdu w polu widzenia kamery. Wzięto pod uwagę tylko te obrazy, które nie były przesłonięte przez inne obiekty ruchome, zgodnie z informacjami pochodzącymi z algorytmu śledzenia ruchomych obiektów. Dodatkowo odrzucono te obrazy, które zawierały poniżej 2000 pikseli (np. o rozdzielczości poniżej 50x40 pikseli), jako zawierające zbyt mało szczegółów. Następnie zebrane dane zostały zweryfikowane pod kątem poprawności i każdemu pojazdowi została nadana etykieta oznaczająca jego typ.

W eksperymentach wykorzystano cztery różne nagrania zrealizowane w 2 różnych lokacjach; jedno nagranie wykonano w pierwszej lokalizacji, a trzy – w drugiej. Nagrania w tej samej lokalizacji różniły orientacją kamery względem kierunku ruchu pojazdów. Charakterystykę wykonanych nagrań i zgromadzonych w oparciu o nie obrazów pojazdów przedstawiono w tab. 5.4, a przykładowe ramki obrazu pokazano na rys. 5.11. Nagranie A zostało wykorzystane we wstępnych eksperymentach polegających na doborze optymalnej postaci wektora cech i wyborze typu klasyfikatora. Uzyskane wyniki były walidowane w oparciu o nagrania B1, B2 i B3. W każdym przypadku klasyfikator był trenowany niezależnie dla każdego nagrania i każdego kierunku ruchu pojazdów (w stronę kamery i w stronę przeciwną) w oparciu o zasady przedstawione w podrozdziale 5.3.2.

Tab. 5.4 Charakterystyka nagrań użytych w eksperymentach

	Nagranie A	Nagranie B1	Nagranie B2	Nagranie B3
Czas trwania [mm:ss]	30:28	50:59	61:14	87:44
Rozdzielczość i liczba ramek na sekundę	720 x 576 x 25 fps	1920 x 1080 x 25 fps	1920 x 1080 x 25 fps	1920 x 1080 x 25 fps
Przybliżony kąt widzenia kamery względem kierunku poruszania się pojazdów	45°	10°	45°	80°
Lokalizacja	Droga krajowa wewnątrz aglomeracji miejskiej	Droga szybkiego ruchu	Droga szybkiego ruchu	Droga szybkiego ruchu
Liczba pasów ruchu*	2 + 3	3 + 2	3 + 2	3 + 3
Liczba pojazdów	1210	2808	1661	4694
– pojazdy osobowe	936	2403	1411	3974
– furgonetki/busy	154	201	131	392
– ciężarowe	120	204	119	328
Liczba obrazów pojazdów	50227	84156	60427	130728
Liczba obrazów na jeden pojazd (średnia i odchylenie standardowe)	41,5 (27,6)	30,0 (11,3)	36,4 (20,6)	27,9 (17,0)

*pierwsza wartość oznacza liczbę pasów ruchu na których pojazdy poruszają się w kierunku kamery, a druga odnosi się pasów z ruchem oddalającym się od kamery



Nagranie A



Nagranie B1



Nagranie B2



Nagranie B3

Rys. 5.11 Przykładowe ramki obrazu z poszczególnych nagrań

5.4.2 Dobór wektora cech i klasyfikatora

Ostateczny dobór parametrów wchodzących w skład wektora cech użytego do klasyfikacji typu pojazdu został wykonany na podstawie nagrania A z tab. 5.4. Równocześnie zbadano zachowanie różnego typu klasyfikatorów w celu sprawdzenia, który z nich najlepiej nadaje się do rozwiązania postawionego problemu. Wyniki pokazano w tab. 5.5.

Tab. 5.5 *Sumaryczne wyniki klasyfikacji typu pojazdów na podstawie nagrania A z wykorzystaniem różnych klasyfikatorów i parametrów bez agregacji wyników (każdy obraz pojazdu klasyfikowany niezależnie)*

Parametr	Ruch w kierunku kamery				Ruch oddalający się od kamery			
	ANN	kNN	RF	BOOST	ANN	kNN	RF	BOOST
	[%]							
<i>kształt</i>	80,2	78,8	79,7	79,6	76,4	77,3	76,1	76,5
<i>surf-4</i>	82,7	79,9	81,5	84,5	79,5	76,1	76,7	80,0
<i>surf-8-kmeans</i>	73,7	44,0	79,0	80,0	73,7	42,9	73,4	75,5
<i>Gabor-obraz</i>	82,1	80,1	77,2	84,0	78,1	73,1	74,5	78,6
<i>Gabor-filtr</i>	85,9	84,6	84,5	87,6	82,9	83,4	78,3	83,7

Analiza skuteczności określania typu pojazdu w zależności od użytego klasyfikatora wykazuje, że w zdecydowanej większości przypadków algorytm K najbliższych sąsiadów, jako najbardziej prymitywny spośród użytych klasyfikatorów, radzi sobie najgorzej. Różnice pomiędzy pozostałymi klasyfikatorami są już mniej wyraźne, aczkolwiek można dostrzec, że RF często ustępuje pozostałym dwóm klasyfikatorom (ANN i BOOST). Dlatego dalsze badania postanowiono zawęzić do grupy dwóch klasyfikatorów: ANN, BOOST, jako pozwalających osiągnąć najlepsze wyniki i jednocześnie charakteryzujących się diametralnie różną zasadą działania.

Analiza wyników w zależności od zastosowanych parametrów pokazuje, że deskryptory kształtu pojazdu pozwalają samodzielnie uzyskać wysoką skuteczność klasyfikacji typu pojazdu, sięgającą 80%. Spośród dwóch wektorów bazujących na deskryptorach SURF zdecydowanie lepsze wyniki uzyskano dla parametru *surf-4*, niezależnie od klasyfikatora i kierunku ruchu pojazdów. Automatyczne znajdowanie punktów charakterystycznych i ich klasteryzacja na 8 przedziałów (dla parametru *surf-8-kmeans*) okazały się mniej skuteczne niż stałe określenie położenia punktów charakterystycznych (dla *surf-4*). Wynika to zapewne z dużej różnorodności wyglądu pojazdów należących do jednej grupy (szczególnie w przypadku pojazdów ciężarowych), a co za tym idzie, dużym rozrzutem lokalizacji punktów charakterystycznych w zależności od obrazu pojazdu. Różnice pomiędzy dwoma wektorami należącymi do drugiej grupy

parametrów obrazu (bazujących na filtracji Gabora) są znacznie mniejsze, aczkolwiek i w tym przypadku jeden deskryptor (*Gabor-filtr*) jest lepszy niż drugi (*Gabor-obraz*). Wynika z tego, że z punktu widzenia klasyfikacji, cenniejsza jest informacja o udziale konkretnych rodzajów krawędzi (kierunek, rozmiar) w każdym z podobszarów obrazu pojazdu niezależnie, niż dane na temat udziału określonego podobszaru w posiadaniu krawędzi, niezależnie dla każdego ich typu.

W celu wybrania grup parametrów wchodzących w skład ostatecznego wektora cech, zbadano dodatkowo skuteczność określania typu pojazdu w parach (osobowe-furgonetki, osobowe-ciężarowe, furgonetki-ciężarowe). Wyniki pokazano w tab. 5.6. Są one analogiczne do tych uzyskanych w przypadku analizy klasyfikacji pojazdów na trzy kategorie: najgorszy jest *surf-8-kmeans* oraz parametr *Gabor-obraz* ustępuje parametrowi *Gabor-filtr*. W ostatnim wierszu tab. 5.6 pokazano dodatkowo wyniki binarnej klasyfikacji z użyciem łącznie obu parametrów opartych na filtrze Gabora. Można dostrzec, że w każdym przypadku wyniki łączne są lepsze, niż dla każdego z parametrów niezależnie. Różnice są tym większe, im bardziej podobne są do siebie oba rozróżnianie typy pojazdów, co widać szczególnie w przypadku separacji furgonetek i pojazdów ciężarowych. Oznacza to, że choć niezależnie *Gabor-filtr* pozwala uzyskać lepsze wyniki, niż *Gabor-obraz*, to ich kombinacja, mimo dużej wzajemnej korelacji, tworzy wartość dodaną pozwalającą w szczególnych, trudnych sytuacjach uzyskać prawidłowe rezultaty. Dlatego zdecydowano się na ostateczną postać wektora parametrów składającą się z deskryptorów: *kształt*, *surf-4*, *Gabor-obraz* i *Gabor-filtr*. Wektor ten ma łącznie 804 elementy.

Tab. 5.6 Sumaryczne wyniki binarnej klasyfikacji typu pojazdów na podstawie nagrania A z wykorzystaniem dwóch klasyfikatorów i różnych parametrów, bez agregacji wyników (każdy obraz pojazdu klasyfikowany niezależnie)

Parametr	osobowe – ciężarowe		osobowe – furgonetki		furgonetki – ciężarowe	
	ANN	BOOST	ANN	BOOST	ANN	BOOST
	<i>Ruch do / od kamery [%]</i>					
<i>kształt</i>	95,0 / 93,7	95,6 / 95,5	89,8 / 86,4	89,9 / 86,9	82,8 / 79,3	83,6 / 79,2
<i>surf-4</i>	94,2 / 95,9	95,1 / 94,8	91,8 / 91,5	91,9 / 91,1	83,5 / 80,9	82,7 / 79,4
<i>surf-8-kmeans</i>	92,6 / 93,3	95,2 / 93,8	85,5 / 86,3	87,9 / 87,1	74,5 / 78,7	78,4 / 78,7
<i>Gabor-obraz</i>	93,6 / 93,6	93,8 / 93,5	89,4 / 92,7	90,1 / 91,9	79,6 / 78,6	83,8 / 79,4
<i>Gabor-filtr</i>	96,9 / 96,6	96,7 / 95,9	95,0 / 93,6	94,1 / 92,4	83,0 / 84,2	85,9 / 84,6
<i>Gabor-obraz + Gabor-filtr</i>	96,7 / 96,8	97,1 / 96,7	95,1 / 94,5	94,3 / 93,2	85,7 / 86,0	87,5 / 87,6

5.4.3 Wyniki klasyfikacji

W tab. 5.7 pokazano szczegółowe wyniki klasyfikacji każdego obrazu pojazdu niezależnie z wykorzystaniem wektora cech dobraneo w rozdziale 5.4.2 i z użyciem dwóch klasyfikatorów. Niecałkowite wartości liczbowe w kolumnie z liczbą przeprowadzonych klasyfikacji wynikają z faktu, że w tabelach zawarto wyniki średnie pięciu sesji klasyfikacyjnych przeprowadzonych w oparciu o różne, losowo dobrane zbiory treningowe i testujące (rozdział 5.3.2). Klasyfikator BOOST pozwolił uzyskać w każdym przypadku skuteczność klasyfikacji o kilka punktów procentowych wyższą, niż algorytm ANN, osiągając łączny wynik w zakresie 88% - 95%, w zależności od nagrania i kierunku ruchu pojazdów.

Tab. 5.8 przedstawia analogiczne wyniki rozpoznawania typu pojazdów, ale tym razem z uwzględnieniem agregacji wyników, co pozwala uzyskać wynik klasyfikacji dla pojedynczego pojazdu, a nie dla każdego jego obrazu niezależnie. Należy podkreślić, że procentowe skuteczności klasyfikacji dla wszystkich typów pojazdu (wiersze „Razem”) zawierają średnią arytmetyczną procentowych skuteczności rozpoznawania poszczególnych klas; zastosowanie tradycyjnej formuły (iloraz liczby poprawnych klasyfikacji do liczby wszystkich pojazdów) prowadziłoby do zafałszowania oceny klasyfikacji ze względu na ogromną przewagę liczby samochodów osobowych w porównaniu z pozostałymi typami pojazdów. Łatwo zauważyć, że agregacja umożliwia, niemal w każdym przypadku, zwiększenie procentowej skuteczności klasyfikacji o 1 do 5 punktów procentowych.

Łączna skuteczność klasyfikacji pojazdów pomiędzy trzy kategorie wyniosła średnio 94,8% dla klasyfikatora BOOST i kierunku ruchu pojazdów w kierunku kamery oraz 92,5% dla przeciwnego kierunku ruchu, przy czym dla żadnego nagrania nie okazała się niższa, niż 88%. Średnie wyniki uzyskane za pomocą klasyfikatora ANN są o ok. 1,5 punktu procentowego niższe. Obserwowalna różnica w wynikach w funkcji kierunku poruszania się pojazdów wynika zapewne z faktu, że przednia część typowego pojazdu (kabina kierowcy, nachylenie czoła pojazdu) posiada więcej cech dystynktywnych, niż jego część tylna, która w dodatku często charakteryzuje się większą różnorodnością wewnątrzklasową.

Tab. 5.7 Szczegółowe wyniki klasyfikacji pojazdów pomiędzy trzy typy, bez agregacji wyników, z wykorzystaniem dwóch rodzajów klasyfikatorów i różnych nagrań

Typ pojazdu	Ruch w kierunku kamery			Ruch oddalający się od kamery		
	Liczba obrazów pojazdów	ANN [%]	BOOST [%]	Liczba pojazdów	ANN [%]	BOOST [%]
Nagranie A						
Osobowe	2402,2	95,7	96,4	557,4	95,3	95,5
Furgonetki	2402,2	80,3	88,9	557,4	75,6	83,4
Ciężarowe	2402,2	87,7	91,1	557,4	82,8	84,1
Razem	7206,6	87,9	92,1	1672,2	84,6	87,7
Nagranie B1						
Osobowe	1521,2	96,9	98,2	1484,6	95,3	96,3
Furgonetki	1521,2	88,9	91,6	1484,6	89,5	94,1
Ciężarowe	1521,2	94,9	95,4	1484,6	91,2	91,8
Razem	4563,6	93,5	95,1	4453,8	92,0	94,1
Nagranie B2						
Osobowe	1441,8	94,3	96,7	415,0	94,5	98,1
Furgonetki	1441,8	80,4	86,8	415,0	83,4	92,9
Ciężarowe	1441,8	91,6	90,8	415,0	73,1	75,6
Razem	4325,4	88,7	91,4	1245,0	83,7	88,9
Nagranie B3						
Osobowe	2586,0	95,2	95,9	1875,8	96,1	97,7
Furgonetki	2586,0	89,9	92,6	1875,8	85,4	90,9
Ciężarowe	2586,0	93,2	94,1	1875,8	86,9	88,7
Razem	7758,0	92,8	94,2	5627,4	89,5	92,4

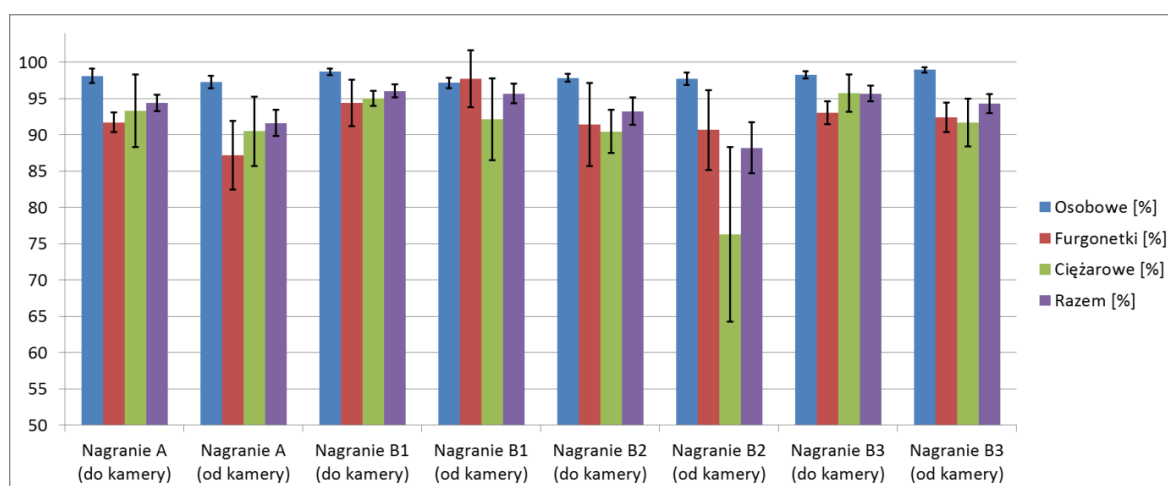
Analizując szczegółowo wyniki rozpoznawania poszczególnych klas można dostrzec, że są one najlepsze dla pojazdów osobowych i wynoszą co najmniej 96%. Wynika to z przede wszystkim z bardzo podobnego wyglądu wszystkich pojazdów osobowych i stosunkowo małej różnorodności ich sylwetek w zależności od marki. W tym zakresie furgonetki (do których zaliczono również busy), a szczególnie pojazdy ciężarowe (ciągniki siodłowe luzem oraz z różnego rodzaju naczepami, samochody ciężarowe skrzyniowe itd.) charakteryzują się ogromną różnorodnością sylwetek i kształtów. Fakt ten, w połączeniu ze znacznie mniejszym udziałem pojazdów nie-osobowych w bazie danych spowodował uzyskanie gorszych wyników klasyfikacji w tych grupach pojazdów. Zdecydowanie najgorszy wynik procentowy osiągnięto dla pojazdów ciężarowych w nagraniu B2 i dla ruchu oddalającego się od kamery, ale w tym przypadku niewielka liczba pojazdów tej kategorii (w zbiorze testowym tylko 16) wydaje się być zbyt mała dla wyciągnięcia wiążących wniosków. Potwierdza to wykres średniej skuteczności klasyfikacji z nałożonymi słupkami błędów, obrazującymi odchylenie standardowe wyników uzyskanych w pięciu sesjach eksperymentalnych, różniących się podziałem wszystkich wektorów na treningowe i testowe (rys. 5.12) – w tym przypadku wartość odchylenia

standardowego jest ponad dwukrotnie większa, niż dla pozostałych typów pojazdów i nagrań.

Tab. 5.8 Szczegółowe wyniki klasyfikacji pojazdów pomiędzy trzy typy, z agregacją wyników, z wykorzystaniem dwóch rodzajów klasyfikatorów i różnych nagrań

Typ pojazdu	Ruch w kierunku kamery			Ruch oddalający się od kamery		
	Liczba pojazdów	ANN [%]	BOOST [%]	Liczba pojazdów	ANN [%]	BOOST [%]
Nagranie A						
Osobowe	325,0	97,7	98,1	548,0	96,7	97,3
Furgonetki	41,0	88,6	91,7	53,0	81,0	87,2
Ciężarowe	39,0	92,1	93,3	21,0	89,0	90,5
Razem	405,0	92,8*	94,4*	622,0	88,9*	91,6*
Nagranie B1						
Osobowe	1338,0	97,9	98,7	969,0	97,0	97,2
Furgonetki	61,0	91,8	94,4	44,0	92,5	97,7
Ciężarowe	52,0	95,1	95,0	56,0	92,5	92,1
Razem	1451,0	94,9*	96,0*	1069,0	94,0*	95,7*
Nagranie B2						
Osobowe	908,0	96,2	97,8	442,0	96,2	97,7
Furgonetki	42,0	86,1	91,4	30,0	84,8	90,7
Ciężarowe	44,0	92,8	90,5	16,0	81,8	76,3
Razem	994,0	91,7*	93,2*	488,0	87,6*	88,2*
Nagranie B3						
Osobowe	2068,0	98,1	98,2	1728,0	98,2	98,9
Furgonetki	135,0	92,3	93,0	92,0	89,7	92,4
Ciężarowe	80,0	94,7	95,8	84,0	90,8	91,7
Razem	2283,0	95,0*	95,7*	1904,0	92,9*	94,3*

*Średnia arytmetyczna procentowej skuteczności klasyfikacji poszczególnych typów pojazdów



Rys. 5.12 Wyniki klasyfikacji typu pojazdu, z agregacją wyników przy wykorzystaniu klasyfikatora BOOST; słupki błędów ilustrują wartości odchylenia standardowego wyników uzyskanych w pięciu seriach eksperymentalnych różniących się doбором wektorów do zbioru treningowego i testowego

Analiza wartości odchyłeń standardowych wyników klasyfikacji wykazuje, że są one najmniejsze (poniżej jednego punktu procentowego) dla kategorii pojazdów osobowych i sięgają pięciu punktów procentowych (pomijając opisany wyżej wariant najgorszy) w niektórych nagraniach dla furgonetek i pojazdów ciężarowych. Większy rozrzut wyników dla pojazdów nie-osobowych wydaje się być ponownie spowodowany dużą nierównomiernością liczby pojazdów poszczególnych typów w bazie danych. Jednak bezwzględne wartości odchyłeń są stosunkowo niewielkie w niemal każdym przypadku, co oznacza, że skuteczność opracowanej metody jest niezależna od doboru konkretnych obrazów pojazdów do zbiorów treningowych i testowych.

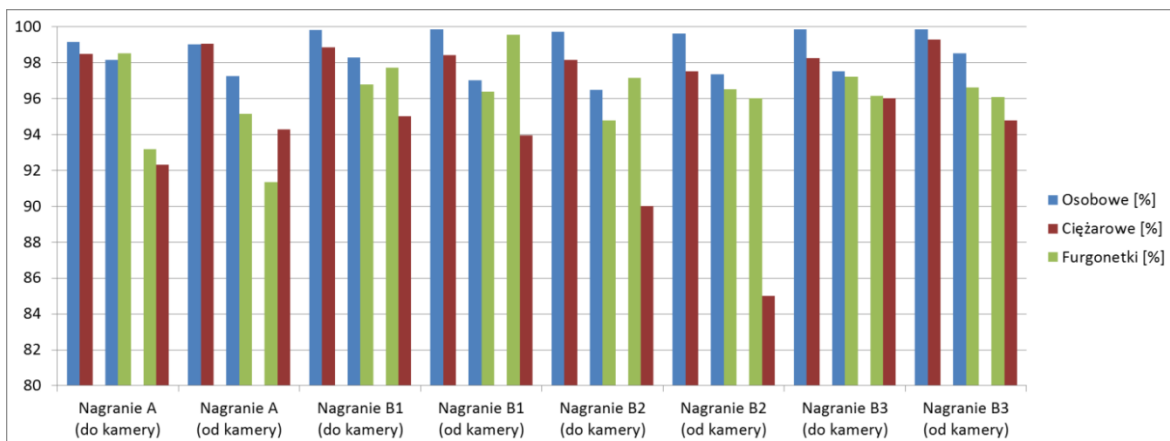
Analiza macierzy błędów dla pojedynczego klasyfikatora i nagrania (tab. 5.9) wskazuje, że pojazdy ciężarowe zdecydowanie częściej mylone są z furgonetkami, niż z pojazdami osobowymi, furgonetki – z pojazdami ciężarowymi, a pojazdy osobowe – z furgonetkami. Analogiczne macierze błędów uzyskano dla pozostałych nagrań i klasyfikatorów. Oznacza to, że w przypadku błędów, najczęściej zwracany jest typ pojazdu, który jest najbardziej podobny do typu pojazdu źródłowego. Dowodzi to poprawnego wytrenowania i działania algorytmu decyzyjnego.

Tab. 5.9 Macierz błędów dla klasyfikatora BOOST i nagrania A

Wyjście Wejście	Ruch w kierunku kamery			Ruch oddalający się od kamery		
	Osobowe	Furgonetki	Ciężarowe	Osobowe	Furgonetki	Ciężarowe
Osobowe	98,1%	1,3%	0,6%	97,3%	2,6%	0,2%
Furgonetki	2,9%	91,7%	5,4%	3,8%	87,2%	9,1%
Ciężarowe	1,0%	5,6%	93,3%	0,0%	9,5%	90,5%

W ramach ostatniego eksperymentu postanowiono sprawdzić skuteczność rozróżniania pojazdów dwóch określonych typów w następujących parach: osobowe – furgonetki, furgonetki – ciężarowe oraz osobowe – ciężarowe (rys. 5.13). Wyniki pokrywają się z tymi uzyskanymi dla rozpoznawania trójklasowego. Najmniej problemów sprawia separacja pojazdów osobowych od ciężarowych, gdzie dla każdego nagrania uzyskano skuteczność powyżej 97,5%, dla każdego typu pojazdu. Nieco gorsze rezultaty uzyskano dla separacji pojazdów osobowych i furgonetek, gdzie najgorszy wynik to 94,8%. Najbardziej problematyczna jest separacja pojazdów najbardziej podobnych do siebie (furgonetek i samochodów ciężarowych), choć i w tym przypadku (pomijając z wymienionych wyżej powodów nagranie B2 w dla ruchu oddalającego się od kamery), uzyskano skuteczność klasyfikacji 90% i wyższą.

Uzyskane wyniki klasyfikacji typu pojazdów wydają się być satysfakcjonujące, biorąc pod uwagę różnorodność sylwetek i wyglądu klasyfikowanych pojazdów oraz fakt zmiany ich wyglądu w trakcie poruszania się w polu widzenia kamery, związanej ze zniekształceniami perspektywicznymi. Zastosowanie czterech nagrań charakteryzujących się innym ustawieniem kamery w stosunku do drogi, wykonanych w dwóch różnych miejscach pozwoliło sprawdzić uniwersalność zaproponowanego rozwiązania. Uzyskane wyniki różnią się maksymalnie kilkoma punktami procentowymi, co pozwala stwierdzić, że parametry sylwetki pojazdu oraz cechy jego wyglądu uzyskane na podstawie parametrów lokalnych SURF oraz filtracji Gabora pozwalają na skuteczną klasyfikację typu pojazdu za pomocą inteligentnego algorytmu decyzyjnego, niezależnie od miejsca ustawienia kamery i jej orientacji względem drogi.



Rys. 5.13 Wyniki klasyfikacji typu pojazdu w parach osobowe-ciężarowe, osobowe-furgonetki i furgonetki-ciężarowe, z agregacją wyników i z wykorzystaniem klasyfikatora BOOST

W praktycznych instalacjach wskazane jest orientowanie kamery w ten sposób, aby mogła rejestrować pojazdy poruszające się w jej kierunku, co pozwoli zmaksymalizować uzyskiwaną dokładność klasyfikacji. Przedstawiona metoda ma jednak praktyczną wadę, którą jest konieczność przeprowadzenia treningu klasyfikatora dla konkretnego miejsca pracy, co oznacza potrzebę ręcznego zebrania bazy obrazów pojazdów i oznaczenia ich odpowiednimi etykietami kategorii, co może być kłopotliwe, ze względu na pracochłonność tych operacji. Teoretycznie można raz wytrenowany klasyfikator wykorzystać w innym miejscu o zbliżonej konfiguracji, ale takie postępowanie prowadziłyby do pogorszenia uzyskiwanych wyników. W celu maksymalnego ułatwienia treningu klasyfikatora przygotowano oprogramowanie, które automatycznie i w czasie

rzeczywistym tworzy na dysku bazę obrazów pojazdów na podstawie analizy strumienia wizyjnego z kamery zainstalowanej w docelowym miejscu, a następnie pozwala użytkownikowi na zweryfikowanie wyników, przypisanie etykiety typu do obrazów każdego pojazdu oraz wytrenowanie wybranych klasyfikatorów. Manualne oznaczanie typu pojazdów przebiega w praktyce czasie rzeczywistym, więc trwa tak długo jak przedział czasu, w którym gromadzono obrazy pojazdów.

6. Eksperymenty z zakresu automatycznej analizy ruchu drogowego

Eksperymenty związane z automatycznym wykrywaniem zdarzeń związanych z ruchem pojazdów zostały przeprowadzone w dwóch aspektach. W związku z pierwszym z nich skupiono się na wielokamerowej analizie ruchu pojazdów na terenie parkingu w celu detekcji rozmaitych zdarzeń, z których najważniejszym jest wykrycie momentu zaparkowania pojazdu i identyfikacja miejsca, na którym się on zatrzymał. Funkcjonalność taka stanowi potencjalnie istotną część większego systemu do automatycznego zarządzania parkingiem. Drugi aspekt eksperymentów dotyczył ruchu drogowego na zatłoczonych ulicach w celu wykrywania przypadków łamania często ignorowanego przepisu dotyczącego zakazu zmiany pasa ruchu w określonych miejscach. Wszystkie eksperymenty były prowadzone z wykorzystaniem istniejących instalacji monitoringowych i w rzeczywistych warunkach.

6.1 Wykrywanie zdarzeń na parkingu

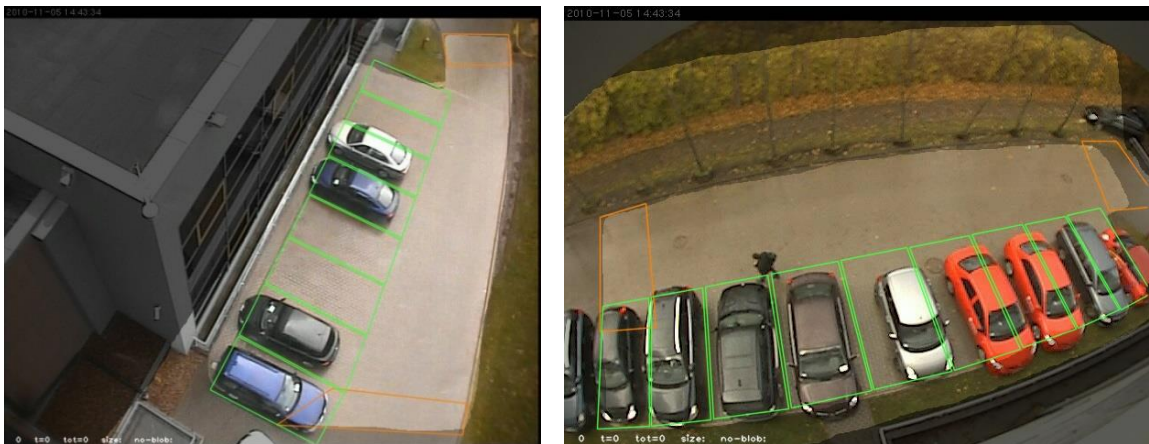
Najważniejszym zdarzeniem wykrywanym w strumieniach wizyjnych pochodzących z parkingu jest moment zatrzymania pojazdu na miejscu parkingowym oraz identyfikacja tego miejsca. Zachowana jest również ciągłość śledzenia pojazdu od wjazdu na parking do momentu zatrzymania w miejscu docelowym, co ma na celu powiązanie obu tych zdarzeń i potencjalną reidentyfikację parkującego pojazdu (np. w sytuacji, gdy brama/szlaban wjazdowy otwierany jest zindywidualizowanym pilotem lub automatycznie za pomocą systemu rozpoznawania tablic rejestracyjnych). Dodatkowo wykrywane są następujące zdarzenia:

- zatrzymanie się pojazdu poza miejscem parkingowym (tj. na drodze)
- wykrywanie i zliczanie pojazdów wjeżdżających na teren parkingu i wyjeżdżających z niego
- detekcja i zliczanie osób wchodzących do budynku, przy którym zlokalizowany jest parking

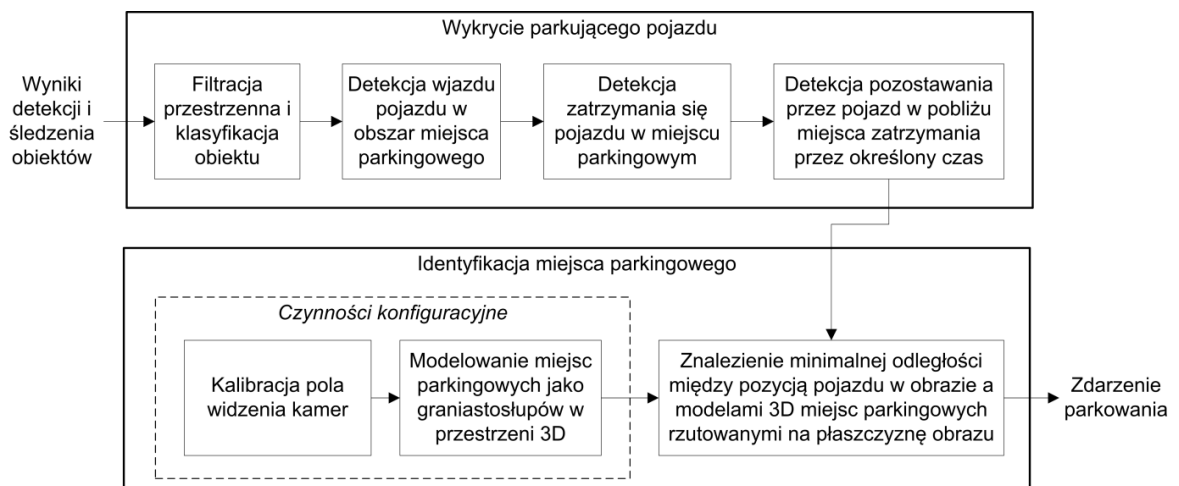
W procesie detekcji zdarzeń wykorzystywane są wyniki działania algorytmów detekcji i śledzenia ruchomych obiektów, w tym śledzenia wielokamerowego (rozdział 4) oraz klasyfikacji typu obiektu (rozdział 5).

6.1.1 Detekcja parkującego pojazdu

Detekcja parkującego pojazdu odbywa się w dwóch etapach. W pierwszym z nich wykrywany jest moment zatrzymania się pojazdu na dowolnym z miejsc parkingowych. W drugim następuje identyfikacja tego miejsca. Miejsca parkingowe są zdefiniowane za pomocą wielokątów, których położenie w polach widzenia poszczególnych kamer określone jest manualnie (rys. 6.1). Schemat algorytmu detekcji parkującego pojazdu przedstawiono na rys. 6.2; jego poszczególne elementy są szczegółowo omówione w dwóch kolejnych podrozdziałach.



Rys. 6.1 Miejsca parkingowe (zielone wielokąty) oznaczone w polu widzenia dwóch przykładowych kamer



Rys. 6.2 Schemat algorytmu wykrywania zdarzeń parkowania pojazdu

6.1.1.1 Wykrycie momentu zatrzymania

Wykrycie momentu zatrzymania się pojazdu na miejscu parkingowym zaimplementowano jako zestaw zdarzeń, które muszą nastąpić kolejno po sobie [61].

Przede wszystkim, analizowane są tylko te obiekty, które weszły w pole widzenia kamery z zewnątrz (tj. w jednym z obszarów wejścia/wyjścia położonych na granicy kadru i wykorzystywanych w algorytmie wielokamerowego śledzenia obiektów – rozdział 4.4), co stanowi filtrację przestrzenną obiektów. Ponadto, rozpatrywany obiekt musi zostać rozpoznany jako pojazd przez algorytm klasyfikacji typu obiektów. Uznaje się, że pojazd zaparkował jeśli wjechał na miejsce parkingowe, zatrzymał się będąc wciąż wewnątrz tego miejsca oraz pozostał w pobliżu miejsca zatrzymania przez określony czas T_S . Jeśli pojazd opuści miejsce zatrzymania, odliczanie czasu zostaje anulowane i ww. seria warunków musi wystąpić ponownie.

Wjazd i opuszczenie miejsca parkingowego wykrywane są w momencie, gdy środek prostokąta opisanego na pojeździe przekroczy krawędź wielokąta oznaczającego miejsce parkingowe, w odpowiednim kierunku. Detekcja zatrzymania pojazdu bazuje na krótko-okresowej średniej wartości prędkości pojazdu. Zatrzymanie ma miejsce, gdy wartość ta przez kilka kolejnych ramek obrazu (np. pięć) jest mniejsza, niż założony próg. Po wykryciu zatrzymania pojazdu, opisany na nim prostokąt definiuje jego obszar zatrzymania; aby mogło być wykryte zdarzenie parkowania, środek pojazdu nie może opuścić obszaru zatrzymania przez czas T_S .

Okres T_S , przez jaki pojazd musi pozostawać w jednym miejscu nie może być zbyt mały, w celu uniknięcia fałszywych detekcji pojawiających się np. w sytuacji, gdy pojazd zawraca na jednym z miejsc parkingowych. Z drugiej strony, im większa wartość T_S , tym większe opóźnienie w wykryciu zdarzenia parkingowego i wygenerowaniu odpowiedniego powiadomienia dla operatora systemu. W drodze wstępnych eksperymentów przyjęto wartość T_S równą 20 sekund, jako odpowiedni kompromis między szybkością reakcji, a dokładnością wyników.

Przedstawiony algorytm może wydawać się nadmiernie skomplikowany, ale został on dopracowany w celu zminimalizowania liczby błędów pierwszego i drugiego rodzaju powodowanych przez przesłaniające się wzajemnie obiekty oraz uwarunkowania praktyczne (np. kierowca opuszczający pojazd po zatrzymaniu wpływa chwilowo, ale istotnie na położenie prostokąta oznaczającego pojazd).

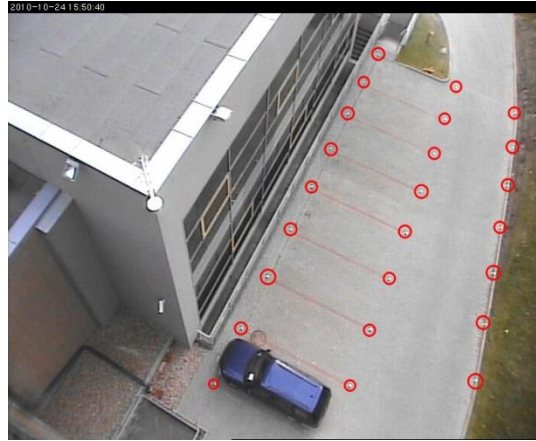
6.1.1.2 Identyfikacja miejsca parkingowego

Po wykryciu zdarzenia zatrzymania się pojazdu na miejscu parkingowym następuje proces identyfikacji tego miejsca [61]. Zadanie to jest trywialne w przypadku, gdy kamera skierowana jest prostopadłe w dół (kąt między kątem widzenia kamery, a powierzchnią

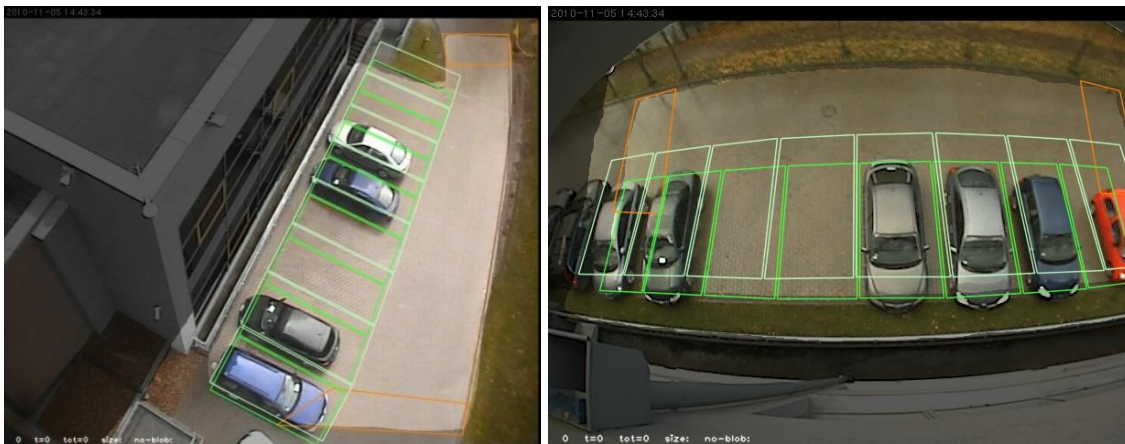
gruntu jest niemal prosty). Jednak wraz ze zmniejszającym się kątem widzenia kamery rośnie trudność związana z określeniem miejsca parkingowego na podstawie tylko położenia prostokąta otaczającego pojazd i wielokąta definiującego miejsce na poziomie gruntu, ze względu na rosnący wpływ wysokości pojazdu na jego rzut perspektywiczny na płaszczyznę drogi. Dodatkowo w przypadku szerokokątnych obiektywów kamer, istotne zniekształcenia są widoczne również w obszarze pojedynczej ramki obrazu w pobliżu jej krawędzi (rys. 6.1).

Aby umożliwić skuteczną identyfikację miejsca parkingowego niezależnie od orientacji kamery w przestrzeni, zdecydowano się przeprowadzić proces kalibracji pola widzenia wykorzystywanych kamer. Kalibracja ma na celu znalezienie jednoznacznego przekształcenia pozwalającego na konwersję współrzędnych trójwymiarowych dowolnego punktu w przestrzeni na współrzędne dwuwymiarowe piksela w ramce obrazu z kamery. Odwrotna konwersja nie jest możliwa bezpośrednio, gdyż wymaga podania wartości jednej ze współrzędnych opisujących położenie punktu w przestrzeni; zwykle podaje się w tym przypadku wartość z oznaczającą wysokość punktu. Parametry przekształcenia są wyznaczane w oparciu o model kalibracji, którego wejście stanowią pary współrzędnych (3D w przestrzeni i 2D w obrazie), dla wielu punktów kalibracyjnych rozmieszczonych w polu widzenia kamery (rys. 6.3). W eksperymentach wykorzystano model kalibracji opracowany przez Tsai [64][243]. Wykorzystując współrzędne punktów kalibracyjnych, metoda Tsai pozwala wyznaczyć 11 parametrów przekształcenia definiujących cechy wewnętrzne kamery (ogniskowa, zniekształcenia wprowadzane przez obiektyw) oraz zewnętrzne związane z jej umiejscowieniem w przestrzeni (macierze obrotu i przesunięcia).

Każde miejsce parkingowe, na potrzeby jego identyfikacji, modelowane jest za pomocą graniastosłupa prostego (zwykle prostopadłościanu) w przestrzeni. Jego dolna podstawa pokrywa się z wielokątem wyznaczającym położenie miejsca parkingowego w obrazie z kamery, który jest konwertowany do współrzędnych 3D przy założeniu, że znajduje się on na poziomie gruntu. Górną podstawę graniastosłupa tworzy jego podstawa dolna podniesiona o h metrów nad poziom gruntu. Wysokość graniastosłupa odpowiada typowej wysokości pojazdu; przyjęto $h = 1,5$. Na rys. 6.4 zilustrowano dolne i górne podstawy graniastosłupów (rzutowane na ramkę obrazu) dla przykładowych miejsc parkingowych.



Rys. 6.3 Położenie punktów kalibracyjnych dla jednej z kalibrowanych kamer



Rys. 6.4 Dolne (kolor ciemnozielony) i górne (kolor jasnozielony) podstawy graniastoslupów modelujących położenie miejsc parkingowych w polu widzenia dwóch kamer

W celu identyfikacji miejsca, na którym zatrzymał się pojazd, obliczana jest odległość pojazdu od wszystkich miejsc parkingowych w polu widzenia kamery, zgodnie ze wzorem:

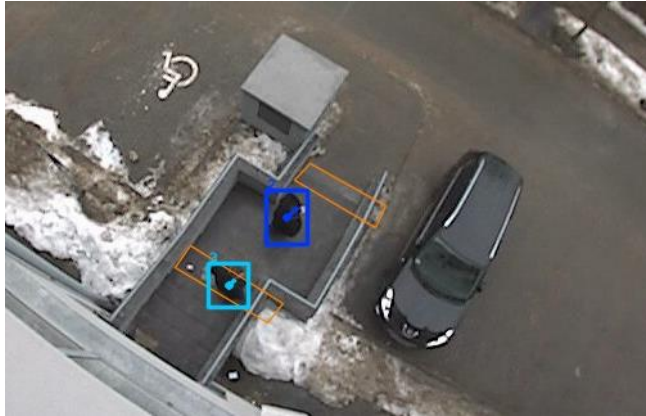
$$d_i = \|\mathbf{u}_i - \mathbf{p}\| + \|\mathbf{l}_i - \mathbf{p}\| \quad (6.1)$$

gdzie d_i oznacza miarę odległości pojazdu od i -tego miejsca parkingowego, \mathbf{p} symbolizuje pozycję pojazdu jako środek opisanego na nim prostokąta w momencie wykrycia zdarzenia parkowania, a \mathbf{u}_i i \mathbf{l}_i oznaczają rzutowane na płaszczyznę obrazu współrzędne ekranowe środków odpowiednio górnej i dolnej podstawy graniastoslupa modelującego i -te miejsce parkingowe. Przyjmuje się, że pojazd zaparkował na i -tym miejscu, jeśli odległość d_i do tego miejsca jest najmniejsza spośród odległości do wszystkich pozostałych miejsc parkingowych.

Na płycie DVD dołączonej do rozprawy umieszczono filmy „parkowanie-1.avi”, „parkowanie-2.avi” oraz „parkowanie-3.avi” ilustrujące działanie algorytmu detekcji parkujących pojazdów.

6.1.2 Detekcja pozostałych zdarzeń

Poza detekcją parkujących pojazdów, możliwe jest również wykrywanie kilku uzupełniających zdarzeń występujących na parkingu [63]. Przede wszystkim, wykrywane są wszystkie pojazdy, które na dłużej zatrzymały się na drodze (tj. poza miejscem parkingowym). Zastosowany algorytm jest analogiczny do algorytmu detekcji parkowania; taki sam jest również wymagany czas pozostawania w bezruchu (20 sekund). Pozwala to zapobiec np. niepoprawnemu parkowaniu, co może prowadzić do powstania zatorów drogowych. Ponadto wszystkie pojazdy wjeżdżające na parking i go opuszczające są wykrywane i zliczane, co pozwala na powiadamianie operatora systemu o każdym nowym pojeździe oraz służy do określania zmiany liczby pojazdów znajdujących się na parkingu. Funkcjonalność ta jest zaimplementowana jako pojawianie się w polu widzenia kamery lub jego opuszczanie przez pojazd w obszarze wejścia/wyjścia umiejscowionym w miejscu wjazdu na parking, który to obszar wykorzystywany jest w algorytmie wielokamerowego śledzenia obiektów (rozdział 4.4). Dodatkowo wykrywane są wszystkie osoby, które wchodzi do budynku biurowego zlokalizowanego przy parkingu lub z niego wychodzą. Funkcjonalność tę osiągnięto poprzez zaznaczenie w polu widzenia kamery dwóch obszarów (wielokątów) zlokalizowanych w pobliżu drzwi wejściowych (rys. 6.5). Pierwszy z obszarów (podstawowy) zlokalizowany jest przy samych drzwiach, a drugi (dodatkowy) – w pewnej odległości, na drodze prowadzącej do drzwi. Zdarzenie wejście do budynku jest sygnalizowane w momencie, gdy osoba przejdzie przez obszar dodatkowy (środek opisanego na niej prostokąta znajdzie się wewnątrz tego obszaru, a następnie poza nim) oraz zniknie z pola widzenia kamery będąc w obszarze podstawowym. Zdarzenie wyjścia z budynku wykrywane jest w analogiczny sposób: osoba musi pojawić się w polu widzenia kamery w obszarze podstawowym, a następnie „przejść” przez obszar dodatkowy. Zastosowanie dwóch obszarów do detekcji wejść i wyjść z budynku pozwala na istotną redukcję liczby fałszywych alarmów (wywołanych np. przez osoby stojące przez drzwiami).

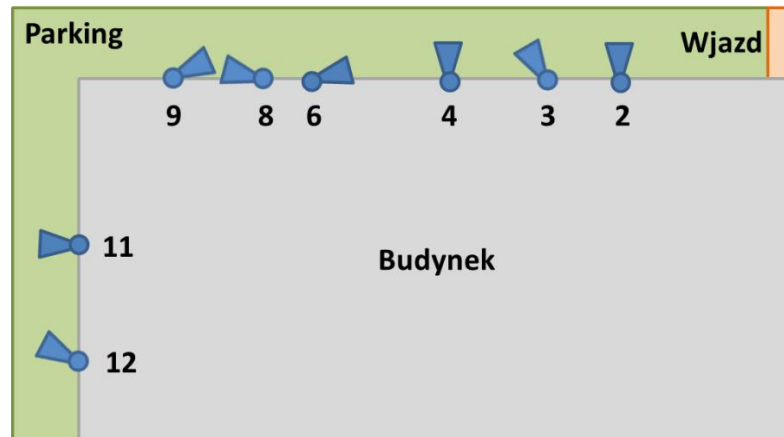


Rys. 6.5 Fragment ramki obrazu, związanej z wykrywaniem wejścia i wyjścia osób z budynku; kolorem pomarańczowym zaznaczono obszary podstawowy i dodatkowy wykorzystywane przez algorytm, a prostokątami w kolorach jasno- i ciemnoniebieskim algorytm zaznaczył dwie osoby wychodzące z budynku

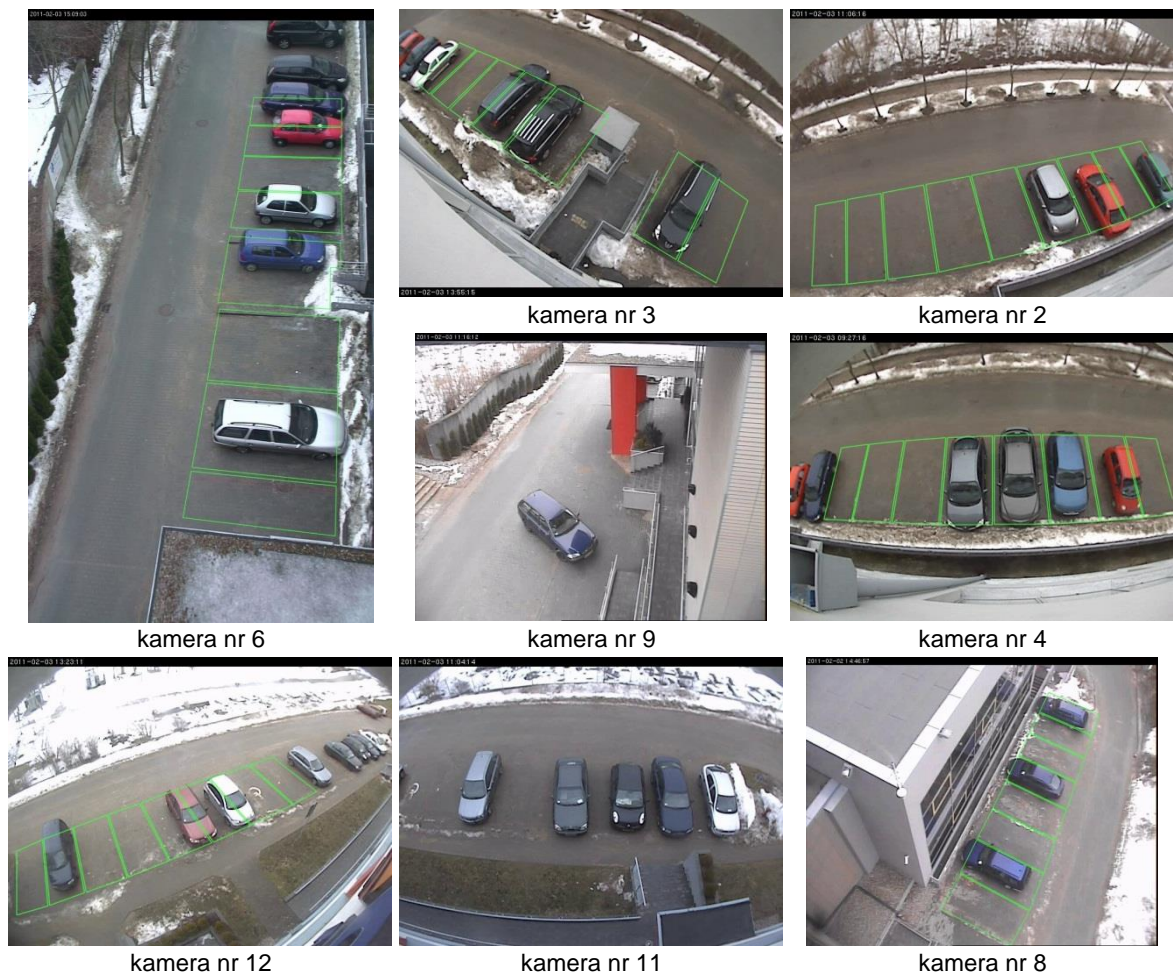
6.1.3 Eksperymenty i wyniki detekcji zdarzeń parkingowych

Na potrzeby tworzenia i walidacji algorytmów przetwarzania obrazu i detekcji zdarzeń zbudowano w Gdańskim Parku Naukowo-Technologicznym instalację testową, składającą się z zestawu kamer i serwerów komputerowych wykorzystanych do analizy strumieni wizyjnych. Na potrzeby detekcji zdarzeń zachodzących na parkingu wykorzystano strumienie z 8 kamer stacjonarnych obserwujących miejsca parkingowe zlokalizowane wokół budynku biurowego; tylko część wszystkich miejsc parkingowych znajduje się w polu widzenia kamer. Schemat rozmieszczenia kamer przedstawiono na rys. 6.6, a przykładowe ramki obrazu z każdej wykorzystywanej kamery pokazano na rys. 6.7. W tab. 6.1 przedstawiono modele wszystkich użytych kamer wraz z parametrami akwizycji obrazu.

Dane na potrzeby walidacji algorytmów detekcji zdarzeń na parkingu zgromadzono w trakcie dwóch dni roboczych. Obejmują one łącznie 16 godzin materiału wizyjnego dla każdej kamery, zarejestrowanego w zmiennych warunkach oświetleniowych i pogodowych (poranek, środek dnia i zmierzch, słoneczna pogoda i duże zachmurzenie). Analiza obrazu i detekcja zdarzeń odbywała się w czasie rzeczywistym na czterech serwerach. Wszystkie wyniki były zapisywane w obszernych dziennikach działania. Jednocześnie archiwizowano poszczególne, oryginalne strumienie wizyjne w celu walidacji wyników uzyskanych automatycznie.



Rys. 6.6 Schemat rozmieszczenia kamer stacjonarnych (wraz z ich numerami) wykorzystywanych do detekcji zdarzeń na parkingu; numeracja kamer nie jest ciągła ze względu na obecność w systemie innych kamer (w tym obrotowych, nie używanych w eksperymentach omawianych w rozprawie)



Rys. 6.7 Przykładowe ramki obrazu ze wszystkich kamer stacjonarnych użytych w eksperymentach z zakresu wykrywania zdarzeń na parkingu wraz z naniesionym położeniem miejsc parkingowych (zielone wielokąty); w polu widzenia kamery nr 9 nie ma zdefiniowanych żadnych miejsc parkingowych

Tab. 6.1 Modele kamer użytych w eksperymentach dotyczących detekcji zdarzeń na parkingu wraz z parametrami akwizycji obrazu

Numer kamery	Model kamery	Rozdzielczość obrazu	Liczba ramek na sekundę
2	Axis 221	640x480	15
3	Axis 221	640x480	15
4	Axis 221	640x480	15
6	Axis Q1755	720x1280	15
8	Axis 233D*	704x576	15
9	Axis 233D*	704x576	15
11	Axis 221	640x480	15
12	Axis 221	640x480	15

* kamera obrotowa PTZ działająca jako stacjonarna

Analiza zgromadzonych danych odbywała się w dwóch krokach. W pierwszej kolejności, sprawdzono poprawność automatycznie wykrytych zdarzeń (tzn. zweryfikowano, czy każde z nich rzeczywiście miało miejsce poprzez sprawdzenie zawartości zarchiwizowanych strumieni wizyjnych) w celu wykrycia fałszywych alarmów (błędów pierwszego rodzaju). Następnie, wszystkie nagrania archiwalne zostały dokładnie przejrane w celu znalezienia zdarzeń, które nie zostały automatycznie wykryte (błędy drugiego rodzaju). Dodatkowo w przypadku zdarzeń polegających na wykryciu pojazdu parkującego na miejscu parkingowym lub poza nim (na drodze) zweryfikowano, czy pojazd ten był poprawnie śledzony w systemie wielokamerowym od momentu wjazdu na parking do momentu jego zatrzymania. Dokonano tego poprzez manualne odszukanie tego samego pojazdu w obrazie z kamery obejmującej wjazd na parking (kamera numer 2) i porównanie, czy automatycznie nadawane przez system unikatowe numery identyfikacyjne (UUID) dla pojazdu w kamerze wjazdowej i w kamerze docelowej są identyczne.

Zdarzenia wjazdu na parking i wyjazdu z parkingu występują tylko w kamerze 2, która obejmuje swoim polem widzenia teren przy szlabanie wjazdowym. Analogicznie, zdarzenia wejścia i wyjścia z budynku występują tylko w kamerze 3, która „widzi” teren przy drzwiach wejściowych. Pozostałe zdarzenia (parkowanie oraz zatrzymanie się poza miejscem parkingowym) wykrywane są w strumieniach ze wszystkich kamer z wyjątkiem zdarzenia zatrzymania się na miejscu parkingowym w kamerze 9, w której polu widzenia nie znajdują się oznaczone miejsca parkingowe.

W tab. 6.2 przedstawiono wyniki detekcji zdarzeń na parkingu. Symbolem *TP* (ang. *True Positives*) oznaczono liczbę poprawnie wykrytych zdarzeń. Liczbę zdarzeń nie wykrytych przez algorytmy symbolizuje *FN* (błędy drugiego rodzaju, ang. *False*

Negatives). Z kolei liczbę fałszywych alarmów oznaczono jako *FP* (błędy pierwszego rodzaju, ang. *False Positives*). Do oceny działania detektorów poszczególnych typów zdarzeń użyto miar czułości *R* (ang. *Recall*), precyzji *P* (ang. *Precision*) i dokładności *A* (ang. *Accuracy*), zdefiniowanych w następujący sposób:

$$R = \frac{TP}{TP + FN} \quad (6.2)$$

$$P = \frac{TP}{TP + FP} \quad (6.3)$$

$$A = \frac{TP}{TP + FP + FN} \quad (6.4)$$

Tab. 6.2 Miary oceny skuteczności detekcji zdarzeń na parkingu

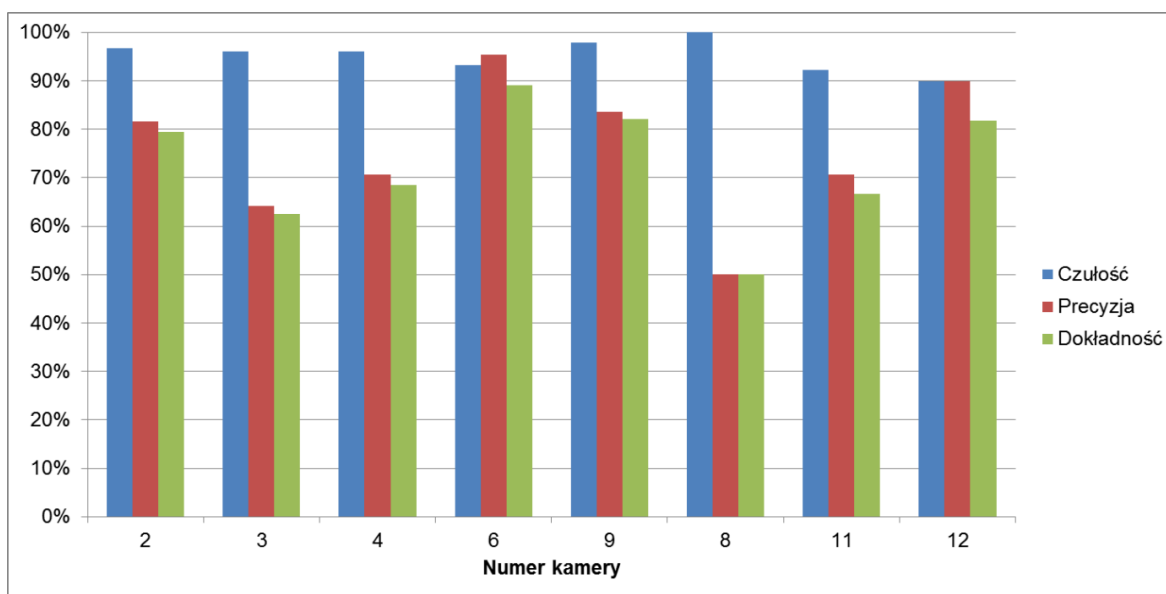
Rodzaj zdarzenia	Liczba wystąpień zdarzenia	<i>TP</i>	<i>FN</i>	<i>FP</i>	Czułość <i>R</i>	Precyzja <i>P</i>	Dokładność <i>A</i>
Parkowanie	152	143	9	32	94,1%	81,7%	77,7%
Zatrzymanie na drodze	89	87	2	29	97,8%	75,0%	73,7%
Wjazd na parking	271	265	6	8	97,8%	97,1%	95,0%
Wyjazd z parkingu	231	217	14	31	93,9%	87,5%	82,8%
Wejście do budynku	208	205	3	17	98,6%	92,3%	91,1%
Wyjście z budynku	146	146	0	0	100,0%	100,0%	100,0%
Razem	1097	1063	34	117	96,9%	90,1%	87,6%

Wyniki detekcji zdarzeń charakteryzują się względnie małą liczbą błędów drugiego rodzaju i większą liczbą błędów pierwszego rodzaju. Około 97% spośród ponad 1000 zdarzeń zostało wykrytych prawidłowo. Zdecydowana większość niewykrytych zdarzeń występuje w przypadku parkowania samochodu i jest powodowana głównie przez algorytm śledzenia ruchomych obiektów w przypadku zatłoczonej sceny, co sprawia, że prostokąt opisany na pojeździe nie był w stanie spełnić warunku o pozostawianiu nieruchomym przez zadany okres czasu.

Fałszywe zdarzenia stanowią około 10% wszystkich raportowanych zdarzeń. Powstają one głównie w przypadku błędów w śledzeniu obiektów, gdy wiele trackerów zostanie przypisanych do tego samego obiektu oraz w wyniku błędów w klasyfikacji typu

obiektu (np. osoba zatrzymująca się na miejscu parkingowym). Największy współczynnik błędów pierwszego rodzaju występuje dla zdarzeń parkowania oraz zatrzymania się poza miejscem parkingowym. Fałszywe alarmy stanowią istotną trudność dla operatora systemu i powinny być minimalizowane. Można to osiągnąć za pomocą dodatkowych etapów przetwarzania (np. poprzez wstrzymanie raportowania tych samych zdarzeń wykrytych w tym samym miejscu w ciągu określonego czasu).

Czułość detekcji zdarzeń w zależności od kamery waha się od 90% do 100% (rys. 6.8), co dowodzi, że opracowane rozwiązanie jest w stanie wykrywać poprawnie zdarzenia niezależnie od orientacji kamery względem gruntu.



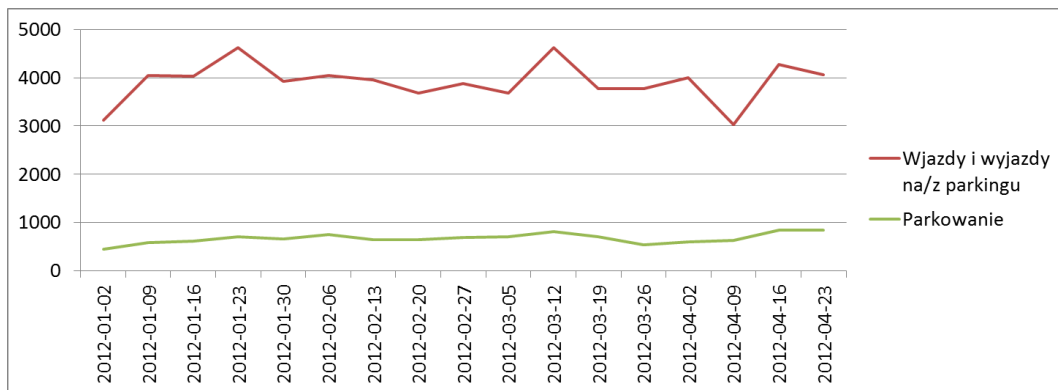
Rys. 6.8 Łączna czułość, precyzja i dokładność detekcji zdarzeń parkowania i zatrzymania się poza miejscem parkingowym w zależności od kamery

Wyniki pokazane w pierwszym wierszu w tab. 6.2 odnoszą się jedynie do wykrycia momentu parkowania; miejsce parkingowe zostało zidentyfikowane poprawnie we wszystkich przypadkach, bez względu na orientację kamery. Dla zdarzeń parkowania i zatrzymania się poza miejscem parkingowym zweryfikowano również poprawność śledzenia pojazdu od wjazdu na parking do miejsca zatrzymania; powiodło się to w 85% przypadków. W zależności od docelowego miejsca zatrzymania, wymagało to zachowania ciągłości śledzenia pojazdu w trakcie poruszania się obiektów w polu widzenia 8 różnych kamer oraz pomiędzy nimi. Zdecydowana większość błędów w powiązywaniu pojazdu zatrzymującego się z pojazdem wjeżdżającym spowodowana jest pomyłkami w procesie śledzenia obiektów ruchomych w polu widzenia pojedynczej kamery oraz na skutek

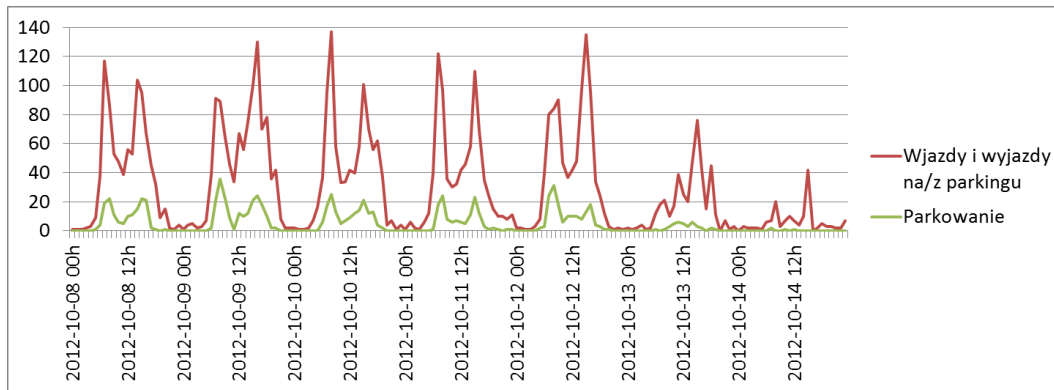
nieprzewidzianego sposobu zachowania się kierowców (np. długie postoje poza polami widzenia kamer).

System detekcji zdarzeń na parkingu działa w Gdańskim Parku Naukowo-Technologicznym od wielu miesięcy. Pozwala to na przekrojową analizę występowania poszczególnych rodzajów zdarzeń w dłuższych okresach czasu. Histogram zdarzeń wykrytych w jedno-tygodniowych przedziałach czasu na przestrzeni czterech pierwszych miesięcy 2012 roku pokazano na rys. 6.9. Wynika z niego, że rozkład występowania zdarzeń jest w miarę jednolity, z wyjątkiem wielkanocnej przerwy świątecznej w tygodniu rozpoczynającym się 9 kwietnia, kiedy to odnotowano niewielki spadek liczby wykrytych zdarzeń (zmniejszenie ruchu pojazdów). Z kolei rozkład występowania zdarzeń w czasie jednego tygodnia pokazano na rys. 6.10. Widoczny jest wyraźny spadek liczby wykrytych zdarzeń w czasie weekendu. Ponadto można wyróżnić dwa maksima lokalne w ciągu każdego dnia, odpowiadające przyjazdom i wyjazdom osób do i z pracy. Takie korelacje wyników detekcji zdarzeń z ogólną wiedzą dotyczącą godzin i dni pracujących potwierdzają poprawność działania algorytmów detekcji zdarzeń na parkingu.

Na płycie DVD dołączonej do rozprawy umieszczono przykładowy film „zdarzenia-parking.avi” pokazujący detekcję zdarzeń parkingowych. Na filmie zarejestrowano działanie Konsoli Operatora (załącznik 10.1.2), aplikacji opracowanej w Katedrze Systemów Multimedialnych i pozwalającej m.in. na wizualizację działania algorytmów analizy obrazu w wielokamerowym systemie monitoringu.



Rys. 6.9 Histogram występowania zdarzeń na parkingu w tygodniowych przedziałach czasu



Rys. 6.10 Histogram występowania zdarzeń na parkingu w godzinnych przedziałach czasu

6.2 Wykrywanie zdarzeń w ruchu drogowym

W ramach detekcji zdarzeń w ruchu drogowym skupiono się na wykrywaniu sytuacji nieprawidłowej zmiany pasa drogowego w obrębie skrzyżowania, tj. na przecinaniu przez kierowców linii ciągłej. W przeciwieństwie do wielu innych wykroczeń drogowych (np. jazda pod prąd), zdarzenie to w określonych miejscach występuje bardzo często, co pozwala na zebranie dużej ilości pozytywnych wzorców oraz na wiarygodną ocenę skuteczności działania opracowanego detektora.

Zmiana pasa w zabronionym miejscu zwykle jest wywołana warunkami drogowymi. W godzinach szczytu, przy dużym natężeniu ruchu drogowego, często się zdarza, że kierowcy omijają kolejkę pojazdów czekających na sąsiednim pasie (np. na możliwość skrętu na skrzyżowaniu), a następnie wjeżdżają przed nie, co jest jawnym łamaniem przepisów o ruchu drogowym. Aby móc skutecznie wykryć tego typu zdarzenia w obecności dużego ruchu pojazdów, konieczne było opracowanie algorytmu, który łączy detekcję ruchomych obiektów w oparciu o model tła z przepływem optycznym, w celu oszacowania kierunku ruchu każdego obiektu na poziomie poszczególnych pikseli obrazu. Następnie analizowany jest ruch poszczególnych regionów, symbolizujących poszczególne pojazdy w celu wykrycia nieprawidłowej zmiany pasa ruchu.

6.2.1 Detekcja obiektów wraz z ich kierunkiem ruchu

Analiza obrazu na potrzeby detekcji zmiany pasa ruchu ograniczona jest do prostokątnego podobszaru **I** wejściowej ramki obrazu, w którym to obszarze może wystąpić oczekiwane zdarzenie. Obszar **I** zawiera oba pasy ruchu oraz granicę między nimi, która definiuje zabronione kierunki przejazdu (tylko jeden, wybrany kierunek lub oba kierunki). Oba pasy i granica między nimi muszą być oznaczone ręcznie w obrazie z

kamery (rys. 6.11a). W obszarze **I** wykrywane są wszystkie piksele należące do ruchomych obiektów (rys. 6.11b). Wykorzystano w tym celu algorytm GMM-Mod (rozdział 4.1.1). W wyniku otrzymuje się macierz binarną **M**, której niezerowe elementy oznaczają położenie pikseli należących do ruchomych obiektów.

Równocześnie, w obszarze **I** obliczany jest gęsty przepływ optyczny wg algorytmu zaproponowanego przez Farnebacka [79], wykorzystany wielokrotnie i adoptowany w Katedrze Systemów Multimedialnych [144][225]. Przepływ optyczny znaleziony w bieżącej ramce obrazu stanowi punkt wyjścia do znalezienia przepływu optycznego w ramce kolejnej. Ważnym parametrem algorytmu wyznaczania przepływu optycznego jest wielkość okna analizy W_o , która musi być odpowiednio dobrana do wielkości i prędkości poruszania się obiektów w analizowanym obszarze obrazu, aby zapewnić wystarczającą zakładkę pomiędzy obrazami obiektów w dwóch sąsiednich ramkach. W wyniku obliczenia przepływu optycznego otrzymuje się dwuwymiarowe pole wektorowe **O** dla ramki obrazu t , które zdefiniowane jest zależnością:

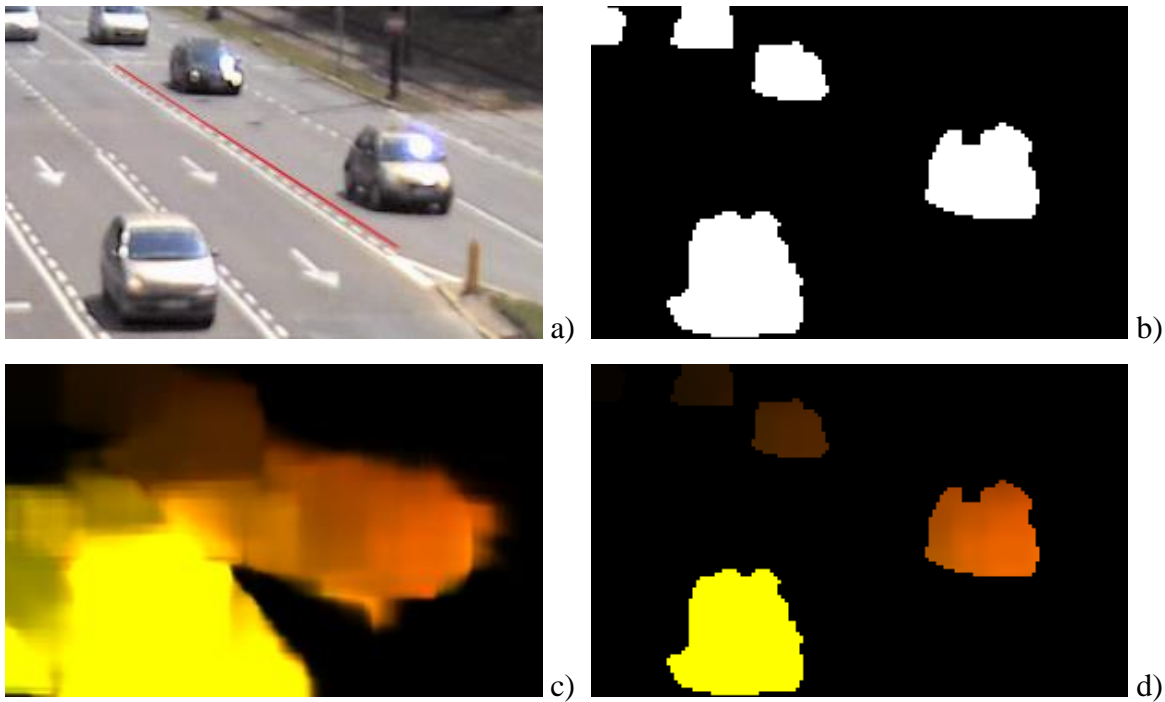
$$I_{t-1}(x, y) \cong I_t(x + O(x, y, 0), y + O(x, y, 1)) \quad (6.5)$$

gdzie $O(x, y, n)$ oznacza n -ty element pola wektorowego **O** w punkcie obrazu o współrzędnych (x, y) . Wynika z tego, że pole **O** zawiera chwilowy wektor prędkości (v_x, v_y) dla każdego piksela obszaru **I_t** (rys. 6.11c).

W następnym kroku tworzona jest maska binarna **F**, która łączy wyniki detekcji ruchomych obiektów za pomocą algorytmu GMM-Mod i przepływu optycznego:

$$F(x, y) = \begin{cases} 1, & \text{jeśli } M(x, y) > 0 \text{ oraz } \sqrt{v_x^2 + v_y^2} > V_{\min} \\ 0, & \text{w przeciwnym wypadku} \end{cases} \quad (6.6)$$

Niezerowe elementy maski **F** oznaczają położenie pikseli należących do ruchomych obiektów, których chwilowy moduł wektora prędkości jest nie mniejszy, niż V_{\min} . Kombinacja obu metod detekcji obiektów ruchomych pozwala uzyskać przybliżony wektor prędkości dla każdego piksela należącego do ruchomych obiektów (przepływ optyczny) oraz jednocześnie dostarcza informacji o dokładnej lokalizacji tych obiektów (GMM-Mod) (rys. 6.11d); w przypadku posługiwania się tylko przepływem optycznym, im większa wartość okna W_o , tym większe rozmycie granic obiektów.



Rys. 6.11 Detekcja ruchomych obiektów uzupełniona o kierunek ruchu: a) obszar obrazu \mathbf{I} z granicą między pasami ruchu zaznaczoną na czerwono, b) wyniki detekcji obiektów algorytmem GMM-Mod \mathbf{M} , c) wynik detekcji obiektów za pomocą przepływu optycznego \mathbf{O} , d) iloczyn logiczny obrazów b) i c); barwa pikseli na rysunkach c) i d) oznacza kierunek ruchu, podczas gdy jasność pikseli jest wprost proporcjonalna do modułu wektora prędkości

6.2.2 Wykrywanie nieprawidłowej zmiany pasa ruchu

Obszar ramki obrazu \mathbf{I} dzielony jest na rozdzielne, sąsiadujące ze sobą, kwadratowe regiony o boku długości s pikseli każdy. W związku z tym, każdy region zawiera $N_s = s^2$ pikseli. Regiony te są położone centralnie w obszarze \mathbf{I} , a ewentualne marginesy są wyłączane z dalszej analizy. Wielkość regionu jest zależna od rozmiarów analizowanych obiektów i powinna być od nich kilka razy mniejsza. W każdym regionie obliczany jest współczynnik r_v odzwierciedlający udział pikseli o oczekiwanym kierunku ruchu w stosunku do wszystkich pikseli w regionie, zgodnie z równaniem:

$$r_v = \frac{N_v}{N_s} \quad (6.7)$$

gdzie N_v oznacza liczbę pikseli w regionie, których wektor prędkości \mathbf{v} spełnia warunek:

$$D^a(\text{atan2}(v_y, v_x), \theta_B) \in [\Delta\theta_1, \Delta\theta_2], \quad \Delta\theta_1, \Delta\theta_2 \in [-\pi, \pi] \quad (6.8)$$

gdzie θ_B definiuje kierunek granicy między oboma pasami ruchu, a $\Delta\theta_1, \Delta\theta_2$ stanowią dolną i górną granicę dopuszczalnych odchyłek kierunku ruchu pikseli obiektów w

odniesieniu do kierunku granicy θ_B . Funkcja $D^a(\theta, \varphi)$, obliczająca różnicę ze znakiem między dwoma dowolnymi kierunkami (kątami) θ i φ , dana jest wzorem:

$$D^a(\theta, \varphi) = \text{atan2}(\sin(\theta - \varphi), \cos(\theta - \varphi)) \in [-\pi, \pi] \quad (6.9)$$

Wartości współczynników r_v dla każdego regionu tworzą macierz \mathbf{H} (rys. 6.12a). Równocześnie, w każdym regionie obliczany jest średni wektor prędkości pikseli spełniających warunek (6.8). Wyniki zapisywane są w macierzy \mathbf{V} .

Następnie zawartość macierzy \mathbf{H} jest analizowana w celu znalezienia lokalnych maksimów \mathbf{H}_M , zgodnie z zależnością:

$$H_M(x, y) = \begin{cases} 0, & \text{jeżeli } H(x, y) \neq H_D(x, y) \text{ lub } H(x, y) = 0 \\ 1, & \text{jeżeli } H(x, y) = H_D(x, y) \text{ oraz } H(x, y) \neq 0 \end{cases} \quad (6.10)$$

gdzie macierz \mathbf{H}_D zawiera wynik dylatacji macierzy \mathbf{H} przeprowadzonej za pomocą prostokątnego elementu strukturalnego o wielkości 3×3 . Przykładową ilustrację zawartości macierzy \mathbf{H}_M pokazano na rys. 6.12b.

W dalszej kolejności wyznaczane jest położenie środków dolnych krawędzi wszystkich połączonych obszarów, znalezionych w macierzy \mathbf{H}_M . Wyniki zapisywane są w macierzy \mathbf{H}_C (rys. 6.12c). Niezerowe elementy macierzy \mathbf{H}_C reprezentują położenie obiektów ruchomych na potrzeby analizy ich ruchu i detekcji zmiany pasa. Na rys. 6.12d pokazano łącznie wyniki detekcji ruchomych obiektów z nałożoną zawartością macierzy \mathbf{H}_C .

W następnym etapie, niezerowe elementy macierzy \mathbf{H}_C , symbolizujące położenie ruchomych pojazdów są śledzone z ramki na ramkę w celu dopasowania do siebie elementów macierzy \mathbf{H}_C w sąsiednich ramkach obrazu i znalezienia ścieżek ruchu \mathbf{P} zapisywanych w zbiorze \mathbf{C} (rys. 6.14). Stosowny algorytm dla bieżącej ramki obrazu przedstawia się następująco:

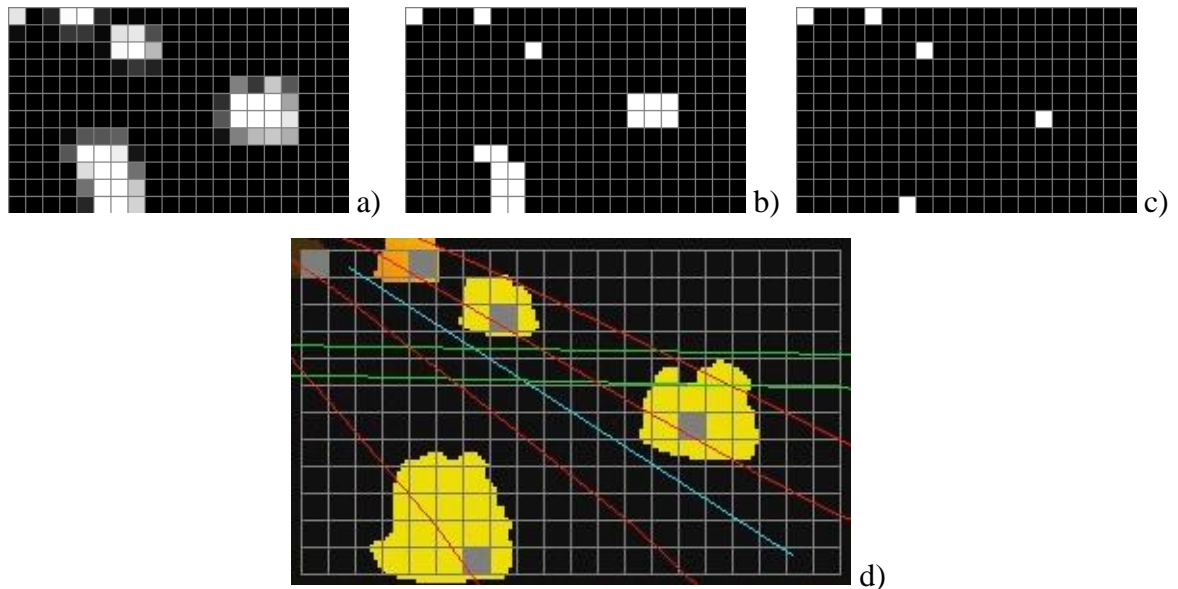
1. Na początku analizy strumienia wizyjnego zbiór \mathbf{C} jest pusty (brak ścieżek ruchu \mathbf{P}).
2. Niech punkty p_H oznaczają współrzędne niezerowych elementów macierzy \mathbf{H}_C .
3. Dla każdej ścieżki \mathbf{P} w zbiorze \mathbf{C} :
 - a. Niech p_L oznacza ostatni punkt w ścieżce \mathbf{P} .
 - b. Dla każdego punktu p_H z macierzy \mathbf{H}_C :
 - i. Jeśli $p_H = p_L$, to znaleziono dopasowanie punktu p_H do ścieżki \mathbf{P} .

- ii. W przeciwnym razie, analizowana jest zależność pomiędzy punktem p_L , a połączonym obszarem macierzy \mathbf{H}_M zawierającym punkt p_H . Jeśli p_L jest częścią tego obszaru lub sąsiaduje z nim krawędzią lub wierzchołkiem (sąsiedztwo Moore’a), to znaleziono dopasowanie, a punkt p_H dodawany jest do tymczasowego zbioru \mathbf{N} .
- iii. W przeciwnym razie, sprawdzane jest, czy punkty p_H i p_L należą do tego samego połączonego obszaru w macierzy \mathbf{H} ; jeśli tak, to znaleziono dopasowanie, a punkt p_H dodawany jest do tymczasowego zbioru \mathbf{N} .
- c. Jeśli żaden punkt macierzy \mathbf{H}_C nie został dopasowany do ścieżki \mathbf{P} , to ścieżkę tę uznaje się za zakończoną i zostaje ona przeniesiona do zbioru zakończonych ścieżek \mathbf{E} .
- d. W przeciwnym razie znaleziono dopasowanie. W przypadku dopasowania z punktu 3b(i), żadne dodatkowe kroki nie są wykonywane. Dla pozostałych typów dopasowań (3b(ii) oraz 3b(iii)), pierwszy z punktów w zbiorze \mathbf{N} dodawany jest do ścieżki \mathbf{P} , a wszystkie pozostałe punkty ze zbioru \mathbf{N} – do osobno utworzonych kopii ścieżki \mathbf{P} , dla każdego z tych punktów. Dodawanie punktu ze zbioru \mathbf{N} do ścieżki polega na dodaniu do ścieżki wszystkich punktów leżących na linii łączącej punkt p_L i punkt ze zbioru \mathbf{N} , przy założeniu metryki taksówkowej (kolejne punkty ścieżki stykają się tylko krawędziami).

4. Wszystkie punkty p_H macierzy \mathbf{H}_C niedopasowane do żadnej ze ścieżek w punkcie 3 algorytmu są użyte do zainicjowania nowych ścieżek, dodawanych do zbioru \mathbf{C} .

Wszystkie zakończone ścieżki ze zbioru \mathbf{E} są analizowane pod kątem wykrycia pojazdów zmieniających pas ruchu w niedozwolonym miejscu. Zadanie to wymaga wiedzy na temat położenia w obrazie obu pasów zainteresowania oraz granicy między nimi. Pasy ruchu są definiowane ręcznie w ramce obrazu jako linie łamane umiejscowione na środku (w osi) każdego z pasów. Granica jest definiowana poprzez linię łamaną, umieszczoną pomiędzy oboma pasami w miejscu, gdzie zmiana pasa jest zabroniona (rys. 6.12d). Jeden pas ruchu traktowany jest jako źródłowy (L_S), a drugi jako docelowy (L_D). Pas źródłowy oznacza pas na którym rozpoczyna się zabroniona zmiana pasa, a pas docelowy to pas, na

którym manewr ten się kończy (np. dla zabronionego przejazdu na lewy pas, patrząc w kierunku ruchu pojazdu, pasem źródłowym jest pas prawy, a docelowym – lewy). Granica między pasami oznaczona jest jako B i dodatkowo niesie ze sobą informację o zabronionym kierunku zmiany pasa (z lewego pasa na prawy, z prawego na lewy lub w obie strony). W przypadku obustronnego zakazu zmiany pasa cała analiza powtarzana jest dwukrotnie.

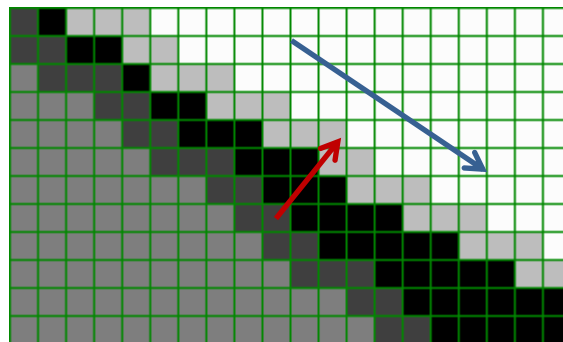


Rys. 6.12 Ilustracja macierzy wykorzystywanych do analizy ruchu obiektów w obrazie I ; im mniejsza wartość elementu macierzy, tym ciemniejszy kolor na rysunku: a) udział pikseli spełniających warunek dotyczący kierunku ruchu H , b) lokalne maksima H_M , c) środków dolnych krawędzi połączonych obszarów H_C , d) łączny wynik detekcji ruchomych obiektów z nałożoną zawartością macierzy H_C wraz z zaznaczonymi środkami pasów ruchu (zielone i czerwone linie) i granicą między dwoma pasami zainteresowania (niebieska linia)

Linie łamane oznaczające oba pasy ruchu skalowane są w dół do rozdzielczości macierzy H_C co sprawia, że macierz ta zostaje podzielona na 5 rozłącznych obszarów oznaczających: położenie obu pasów ruchu L_S i L_D , granicę między tymi pasami B oraz dwa obszary leżące na zewnątrz obu analizowanych pasów O_S i O_D (rys. 6.13).

Aby zakończona ścieżka P ze zbioru E mogła zostać zaklasyfikowana jako reprezentująca zdarzenie nieprawidłowej zmiany pasa ruchu, muszą być spełnione cztery warunki. W ramach pierwszego z nich analizowany jest przebieg ścieżki, punkt po punkcie. Analiza ta bazuje na histerezie; ścieżka musi przeciąć odpowiednie obszary w obrazie (rys. 6.13) w odpowiedniej kolejności, z opcjonalnym powrotem do poprzedniego obszaru. Analiza ścieżki P rozpoczyna się od pierwszego punktu leżącego na pasie L_S lub

w obszarze O_S na zewnątrz od niego (w zależności od tego, który punkt znajduje się bliżej początku ścieżki) z zastrzeżeniem, że do tego momentu analizy żaden punkt ścieżki nie może znajdować się na pasie docelowym L_D lub w obszarze O_D na zewnątrz od niego. Następnie ścieżka musi przeciąć pas L_S i wejść w obszar granicy B między pasami. Od tego momentu ścieżka nie może wrócić do obszaru O_S . W dalszej kolejności, ścieżka musi wejść w obszar pasa L_D i od tej pory nie może już wrócić na pas L_S lub obszar O_S . I jako ostatni warunek, ścieżka musi się znaleźć w obszarze O_D na zewnątrz pasa L_D i już nie może powrócić do obszaru granicy B lub obszarów wcześniejszych (L_S i O_S). W trakcie tej analizy znajdowany jest transjent \mathbf{T}_P ścieżki \mathbf{P} jako jej wycinek od momentu, gdy ścieżka ostatni raz znajdowała się w obszarze pasa L_S do momentu, gdy ścieżka weszła po raz pierwszy w obszar pasa L_D .



Rys. 6.13 Ilustracja pięciu obszarów, na jakie dzielony jest obraz dla przykładowego kierunku ruchu pojazdów zaznaczonego niebieską strzałką i zabronionego kierunku zmiany pasa (czerwona strzałka); od lewego dolnego rogu rysunku do prawego górnego: obszar na zewnątrz pasa źródłowego O_S , pas źródłowy L_S , granica między pasami B , pas docelowy L_D oraz obszar na zewnątrz pasa docelowego O_D

Drugi warunek, który musi być spełniony na potrzeby wykrycia zmiany pasa, dotyczy transjentu \mathbf{T}_P (o ile został znaleziony). Dla każdego punktu odpowiadającego transjentowi odczytywany jest wektor prędkości z macierzy \mathbf{V} , a następnie obliczany jest kierunek $\theta_{\bar{v}}$ średniego wektora prędkości dla wszystkich punktów należących do transjentu \mathbf{T}_P . Kierunek ten musi spełniać zależność:

$$\begin{aligned}
 D^a(\theta_{\bar{v}} - \theta_B) &> 0 \quad \text{dla zmiany pasa z prawego na lewy} \\
 &\text{lub} \\
 D^a(\theta_{\bar{v}} - \theta_B) &< 0 \quad \text{dla zmiany pasa z lewego na prawy}
 \end{aligned}
 \tag{6.11}$$

co oznacza, że kierunek $\theta_{\bar{v}}$ musi być odchyłony w stronę pasa docelowego L_D , a nie źródłowego L_S .

Trzeci warunek także dotyczy punktów wchodzących w skład transjentu \mathbf{T}_p . Jednak tym razem są one aproksymowane za pomocą odcinka przy użyciu algorytmu bazującego na M-estymatorach. Algorytm ten w sposób iteracyjny dopasowuje odcinek do zbioru punktów za pomocą metod ważonych, najmniejszych kwadratów [88]. Kierunek (zwrot) dopasowanego odcinka θ_T wybierany jest jako bliższy kierunkowi wektora łączącego pierwszy i ostatni punkt transjentu \mathbf{T}_p . Kierunek θ_T musi spełniać warunek:

$$D^a(\theta_T, \theta_B) \in [\Delta\theta_{T1}, \Delta\theta_{T2}], \quad \Delta\theta_{T1}, \Delta\theta_{T2} \in [-\pi, \pi] \quad (6.12)$$

gdzie wartości graniczne różnicy kierunku $\Delta\theta_{T1}, \Delta\theta_{T2}$ są ustawione w taki sposób, by pozwolić na mniejszą odchyłkę kierunku θ_T w stronę pasa źródłowego L_S , a większą w stronę pasa docelowego L_D .

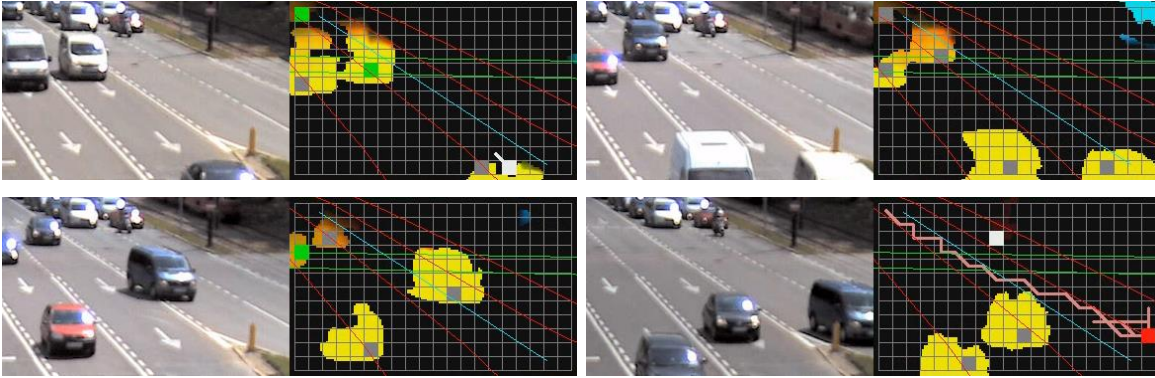
Ostatni warunek, który musi być spełniony na potrzeby wykrycia zmiany pasa, dotyczy wszystkich punktów wchodzących w skład ścieżki \mathbf{P} . Ścieżka ta jest aproksymowana za pomocą odcinka i znajdowany jest jego kierunek θ_p wg metody przedstawionej w poprzednim akapicie. Kierunek θ_p musi spełniać zależność:

$$D^a(\theta_p, \theta_B) \in [\Delta\theta_{p1}, \Delta\theta_{p2}], \quad \Delta\theta_{p1}, \Delta\theta_{p2} \in [-\pi, \pi] \quad (6.13)$$

gdzie wartości graniczne różnicy kierunku $\Delta\theta_{p1}, \Delta\theta_{p2}$ są ustawione w taki sposób, by pozwolić na mniejszą odchyłkę kierunku θ_p w stronę pasa L_S , a większą w stronę pasa L_D .

Jeśli zakończona ścieżka spełnia wszystkie wyżej wymienione warunki, to raportowane jest wykrycie zdarzenia zmiany pasa drogowego w niedozwolonym miejscu. Przykładową ilustrację detekcji pojedynczego zdarzenia zmiana pasa ruchu pokazano na rys. 6.14.

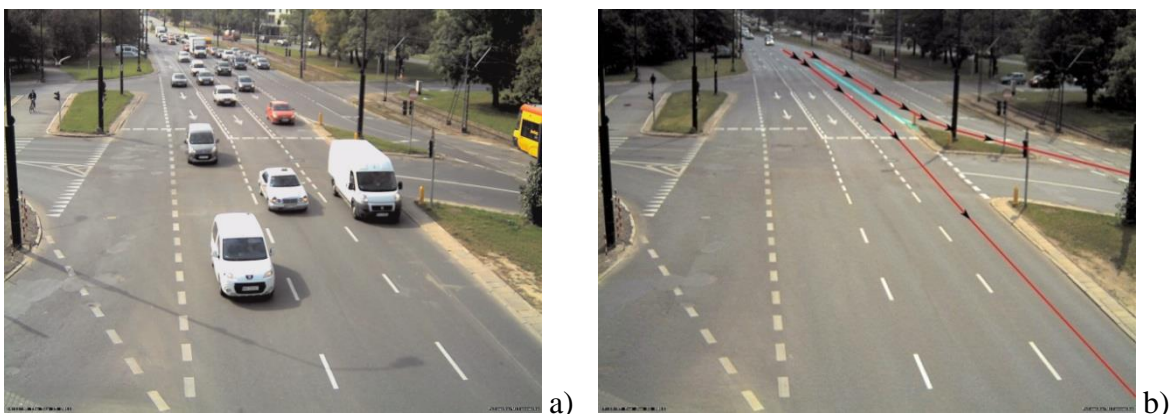
Ze względu na specyfikę algorytmu związaną z duplikowaniem ścieżek w procesie powiązywania ze sobą obiektów w sąsiednich ramkach strumienia wizyjnego, większość ścieżek kończących się w tym samym lub zbliżonym momencie czasu zawiera wspólne fragmenty. Dlatego ścieżki z tym samym punktem końcowym lub ścieżki z takim samym transjentem \mathbf{T}_p w określonym przedziale czasu są odrzucane w celu uniknięcia zduplikowanych alarmów dotyczących tego samego zdarzenia.



Rys. 6.14 Ilustracja działania algorytmu detekcji zmiany pasa (od lewej do prawej i od góry do dołu) na przykładzie ciemnej furgonetki. Kolorowe obszary przedstawiają łączny wynik detekcji ruchomych obiektów (GMM-Mod oraz przepływ optyczny), gdzie barwa oznacza kierunek ruchu, a jej nasycenie jest wprost proporcjonalne do modułu wektora prędkości. Pasy ruchu oznaczono czerwonymi i zielonymi liniami, a granica między dwoma pasami zainteresowania jest niebieska. Zakolorowane kwadraty symbolizują niezerowe elementy macierzy H_C , przy czym kolor oznacza stan poszczególnych ścieżek P : zielony – pierwszy punkt ścieżki, szary – kolejny punkt, biały – ścieżka została zakończona, czerwony – ścieżka zakończona z pozytywnym wynikiem detekcji; pogrubione linie białe i czerwone pokazują zakończone właśnie ścieżki w całości

6.2.3 Eksperymenty i wyniki detekcji zmiany pasa

Do eksperymentów z zakresu wykrywania nieprawidłowej zmiany pasa ruchu wykorzystano strumień wizyjny z kamery monitoringu należącej do Policji, zamontowanej w Warszawie, przy skrzyżowaniu ulicy Puławskiej z Wilanowską (rys. 6.15). Wykorzystana kamera to IQeye 702, która pracowała w rozdzielczości 1600x1200 i pozwalała w praktyce na rejestrację do 10 ramek obrazu na sekundę.



Rys. 6.15 Przykładowa ramka obrazu z kamery monitorującej ruch uliczny w Warszawie (a), oraz lokalizacja dwóch pasów ruchu (czerwony) i granicy między nimi (niebieski) gdzie zabroniona jest zmiana pasa (b)

Fragment kadru, w którym dochodzi do zabronionej zmiany pasa pokazano na rys. 6.11a. Przykładowe nagranie z kamery monitoringu, w którym występuje wykrywane zdarzenie, umieszczono również na płycie DVD dołączonej do rozprawy (plik „zmiana-pasa.avi”).

W tej lokalizacji pojazdy jadące prawym pasem (patrzac w kierunku ruchu) często mijają stojące z lewej strony pojazdy, a następnie wjeżdżają przed nie w celu skrętu w lewo na skrzyżowaniu (rys. 6.16). Ponieważ wykroczenie to zdarza się bardzo często, w godzinach szczytu nawet kilka razy na minutę, to głównym celem detektora nie jest wykrycie możliwie wszystkich przypadków łamania przepisów ruchu drogowego, ale to, aby wykryć przynajmniej ich istotną część, przy jednoczesnym zapewnieniu niewielkiego udziału fałszywych alarmów w otrzymywanych wynikach. W ten sposób, potencjalnie choć część kierowców może zostać pouczone lub ukarana mandatami, podczas gdy zwykle wykroczenie to uchodzi bezkarnie. Jednocześnie względnie niewielka liczba błędów pierwszego rodzaju sprawi, że większość zdarzeń wykrywanych przez detektor ma potwierdzenie w rzeczywistości, co znacznie ułatwia pracę operatorom systemu. W celu identyfikacji kierowcy odpowiedzialnego za wykroczenie można posłużyć się dodatkową kamerą z dużym zoomem optycznym, która pozwoliłaby odczytać tablice rejestracyjne pojazdów popełniających wykroczenie.



Rys. 6.16 Przycięte ramki obrazu z kamery monitoringu obrazujące przypadki nieprawidłowej zmiany pasa ruchu

W celu wykrycia zmiany pasa, konieczne było wcześniejsze ręczne oznaczenie w ramce obrazu położenia osi obu pasów ruchu oraz granicy między nimi na odcinku, gdzie obowiązuje zakaz (rys. 6.15b). Analizie poddano cztery godziny materiału wizyjnego gromadzonego w czasie porannego szczytu komunikacyjnego. Analiza obrazu i detekcja zdarzeń odbywała się w czasie rzeczywistym na serwerze zainstalowanym w Warszawie. Wyniki zapisywane były w pliku dziennika. Jednocześnie, strumień wizyjny był rejestrowany lokalnie dla celów walidacyjnych. Wszystkie automatycznie wykryte zdarzenia zostały zweryfikowane (tj. sprawdzono naocznie w nagraniach, czy zdarzenia te rzeczywiście miały miejsce). Następnie nagrania zostały przejrzane w celu znalezienia

przypadków zdarzeń nie wykrytych. Do oceny działania detektora użyto miar precyzji czułości R (ang. *Recall*), P (ang. *Precision*) i dokładności A (ang. *Accuracy*) zdefiniowanych wcześniej podanymi wzorami (6.2), (6.3) i (6.4).

Wyniki detekcji nieprawidłowej zmiany pasa pokazano w tab. 6.3. Opracowany algorytm jest w stanie wykryć ponad 40% przypadków nieprawidłowej zmiany pasa. Wartość ta może się wydawać niewielka na pierwszy rzut oka, jednak są to rezultaty satysfakcjonujące z praktycznego punktu widzenia, biorąc pod uwagę wysokie natężenie ruchu drogowego (prawie 2000 pojazdów przejeżdżających w ciągu godziny po dwóch tylko pasach znajdujących się w obszarze zainteresowania) i duże nagromadzenie pojazdów w jednym miejscu, a także zmienną prędkość ruchu – od postoju w zatorze drogowym, po szybkie manewry kierowców usiłujących wymusić pierwszeństwo. Osiągnięto przy tym precyzję detekcji zdarzeń P wynoszącą ponad 75%. Wszystkie przykładowe zdarzenia pokazane na rys. 6.16 zostały wykryte poprawnie. Błędy drugiego rodzaju spowodowane są przede wszystkim chwilowymi warunkami drogowymi; szczególnie często pojazd popełniający wykroczenie był przez większą część czasu zasłonięty przez poprzedzający go pojazd. Ilość błędów pierwszego rodzaju mogłaby być mniejsza; w tej chwili na każde cztery wykryte zdarzenia przypada jeden fałszywy alarm. Zmniejszenie liczby błędów pierwszego rodzaju byłoby możliwe poprzez wprowadzenie dodatkowych, bardziej rygorystycznych kryteriów podczas analizy obrazu lecz odbiłoby to się również na spadku czułości detektora. Wydaje się, że możliwość wykrycia niemal połowy zdarzeń nieprawidłowej zmiany pasa kosztem faktu, że co czwarte zdarzenie jest fałszywym alarmem wydaje się być akceptowalnym kompromisem. Istnieją ponadto duże możliwości dalszej poprawy skuteczności opracowanego rozwiązania poprzez wykorzystanie ujęć wielokamerowych i optymalizację ustawienia kamer w miejscu objętym testami.

Przykładowe filmy ilustrujące działanie algorytmu nieprawidłowej zmiany pasa (pliki od „zdarzenia-pas-01.avi” do „zdarzenia-pas-10.avi”) umieszczono na płycie DVD dołączonej do rozprawy.

Tab. 6.3 Wyniki detekcji nieprawidłowej zmiany pasa

Liczba wystąpień zdarzenia	TP	FN	FP	Czułość R	Precyzja P	Dokładność A
137	59	78	19	43,1%	75,6%	37,8%

7. Oprogramowanie do przetwarzania strumieni wizyjnych

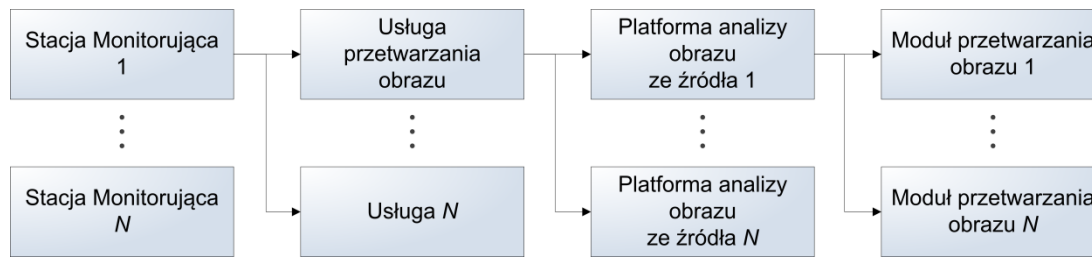
Niniejszy rozdział przedstawia platformę programistyczną, która została stworzona przez autora rozprawy w ramach rozwijania algorytmów przetwarzania obrazu. Platforma ta umożliwia analizę wielu strumieni wizyjnych jednocześnie w sposób elastyczny i skalowalny, a także w maksymalny sposób ułatwia twórcom modułów analizy obrazu ich tworzenie i testowanie. Została ona wykorzystana do implementacji i weryfikowania wszystkich opracowanych algorytmów przetwarzania obrazu. Platforma ta stanowi istotną część szerszego rozwiązania sprzętowo-programowego opracowanego ramach udziału Katedry Systemów Multimedialnych w projektach europejskich 7. Programu Ramowego „Indect” [270] i „Addpriv” [265], pozwalającego na tworzenie uniwersalnego, rozproszonego systemu monitoringu i automatycznej detekcji zdarzeń. Rozwiązanie to przedstawiono w załączniku 10.1. Prace innych wykonawców projektów wpisały się w rozwój omawianej w niniejszym rozdziale platformy programistycznej, przez co stała się ona uniwersalnym narzędziem do prowadzenia prac badawczych i eksperymentalnych z dziedziny monitoringu wizyjnego.

W kolejnych podrozdziałach scharakteryzowano platformę do tworzenia i uruchamiania modułów przetwarzania obrazu oraz przedstawiono zaimplementowane z jej wykorzystaniem moduły analizy obrazu, opracowane przez autora rozprawy samodzielnie lub z jego głównym udziałem. W ostatnim podrozdziale zbadano wydajność platformy pod kątem liczby strumieni wizyjnych, możliwych do jednoczesnego przetwarzania przez jedną jednostkę komputerową. Jest to szczególnie istotne w przypadku zastosowań przedstawionych w niniejszej rozprawie, w których strumienie wizyjne z kamer muszą być analizowane na bieżąco.

7.1 Platforma programistyczna

Platforma programistyczna pozwala na uruchamianie algorytmów przetwarzania obrazu w środowisku systemów operacyjnych Windows, Linux oraz MacOS X. Wchodzi ona w skład usługi przetwarzania obrazu, która działa na Stacji Monitorującej. Stacja Monitorująca jest funkcjonalną jednostką systemu monitoringu, odpowiedzialną m. in. za przetwarzanie obrazów z podłączonych do niej kamer i za wysyłanie wyników analizy do serwera systemu. System monitoringu, w tym Stację Monitorującą, przedstawiono w załączniku 10.1. Oprogramowanie Stacji Monitorującej ma charakter modułowy, co pokazano na rys. 7.1. Oprócz usługi przetwarzania obrazu, w ramach jednej Stacji

Monitorującej, mogą działać także inne usługi, odpowiedzialne np. za zdalne zarządzanie Stacją.



Rys. 7.1 Modułowa architektura oprogramowania Stacji Monitorującej

Usługa przetwarzania obrazu pozwala na równoczesne prowadzenie analizy dowolnej liczby strumieni wejściowych. Każdy strumień przetwarzany jest przez platformę analizy obrazu wg niezależnie konfigurowanego schematu przetwarzania (rys. 7.2). Schemat ten definiuje, porządkuje i ustala zależności pomiędzy poszczególnymi modułami przetwarzania obrazu. Każdy moduł implementuje pojedynczy algorytm analizy (np. śledzenie ruchomych obiektów). Moduły mogą być dowolnie włączane i wyłączane (biorąc pod uwagę zależności między nimi) w celu osiągnięcia oczekiwanych rezultatów przetwarzania strumienia.

Na podstawie zdefiniowanego schematu przetwarzania, platforma automatycznie maksymalizuje wydajność procesu analizy obrazu poprzez wielowątkowe i potokowe uruchamianie zadań obliczeniowych przez poszczególne moduły przetwarzania obrazu dla kolejnych ramek obrazu. Każdy moduł ma dostęp do oryginalnej ramki obrazu oraz do wyników przetwarzania wygenerowanych przez wszystkie inne moduły znajdujące się przed nim w schemacie przetwarzania. Dzięki temu moduły mogą bazować na wynikach uzyskanych przez inne moduły (np. śledzenie ruchomych obiektów bazuje na wynikach odejmowania tła).

Platforma programistyczna służąca do przetwarzania strumieni wizyjnych charakteryzuje się następującymi cechami:

- każdy moduł otrzymuje dane do przetworzenia (ramki obrazu) w kolejności ich odebrania ze źródła sygnału,
- określony moduł może w danej chwili przetwarzać dane dotyczące tylko jednej ramki obrazu, o ile nie ustawiono tego inaczej w konfiguracji modułu (zapewnienie przyczynowości przetwarzania koniecznej w większości

przypadków, gdy wynik analizy określonej ramki obrazu przez moduł jest zależny od wyników analizy ramek wcześniejszych),

- ta sama ramka obrazu może być równocześnie przetwarzana przez wiele modułów analizy obrazu.

Platforma udostępnia bazową klasę do tworzenia modułów przetwarzania obrazu. Dzięki jej użyciu, implementacja własnych modułów analizy sprowadza się tylko i wyłącznie do napisania algorytmu z wykorzystaniem dostarczonych funkcjonalności; wszelkie operacje związane z udostępnieniem danych wizyjnych do analizy oraz odebraniem z modułu wyników są realizowane automatycznie przez platformę.

W obecnej postaci platforma obsługuje następujące źródła obrazu:

- pliki filmowe znajdujące się lokalnie na dysku,
- filmy w postaci oddzielnych plików graficznych dla każdej ramki,
- kamery/serwery IP zgodne z protokołem VAPIX [274], produkowane przez firmę Axis,
- kamery IP firmy IQeye,
- dowolny serwer oferujący strumień RTSP,
- dowolna kamera USB rozpoznawana przez system operacyjny (tylko system Windows).

7.2 Moduły analizy obrazu

Kompletny schemat przetwarzania strumienia wizyjnego pokazano na rys. 7.2. Widoczne są w nim dwa rodzaje modułów. Moduły oznaczone kolorem zielonym i niebieskim stanowią implementację konkretnych algorytmów przetwarzania obrazu. Kolorem szarym zaznaczono moduły wewnętrzne platformy, które jednakże są traktowane na równi z modułami przetwarzania obrazu.

Autor rozprawy jest jedynym autorem większości modułów pokazanych na rys. 7.2. Wyjątkiem są moduły: detekcji ruchomych obiektów (`ObjectDetector`), klasyfikacji obiektów w zakresie ich podziału na osoby, pojazdy i inne obiekty (część modułu `ObjectClassifier`) oraz wykrywania zdarzeń podstawowych (`EventDetector`), przy tworzeniu których udział mieli również inni wykonawcy.

Wśród modułów implementujących algorytmy analizy obrazu znajdują się:

- `ObjectDetector (-PreBS, -BS, PostBS)` – detekcja ruchomych obiektów w ramce obrazu (szczegóły algorytmu przedstawiono w rozdziale

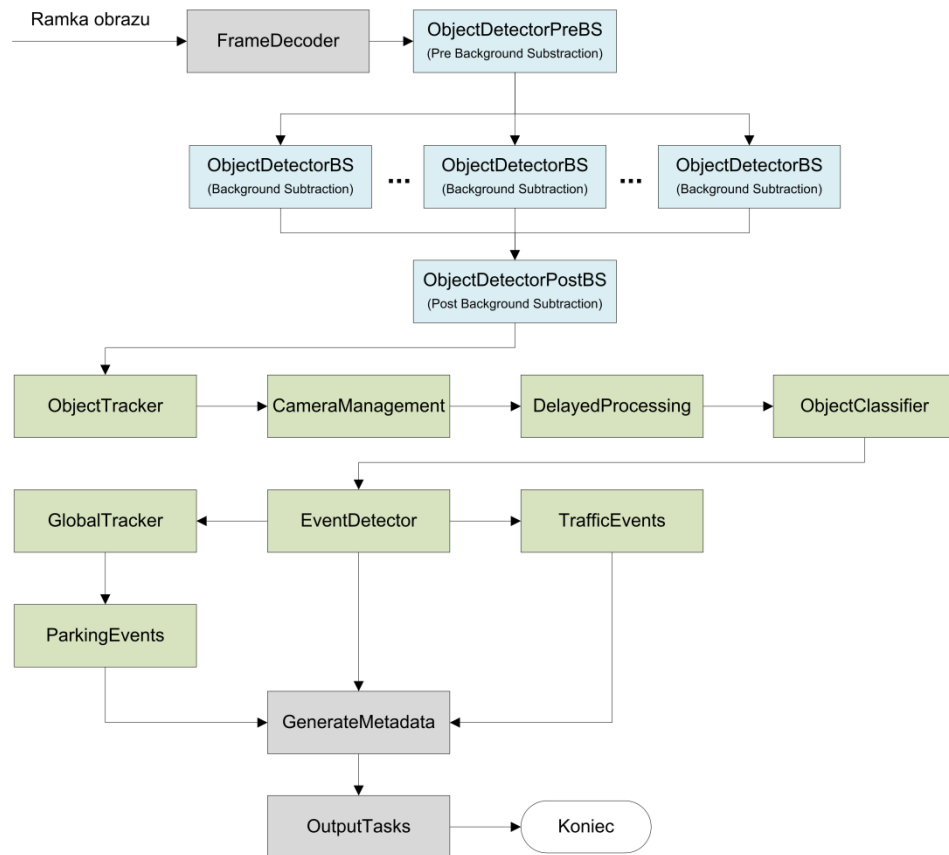
4.1). Ze względu na duże zapotrzebowanie na moc obliczeniową, moduł ten został podzielony na kilka podmodułów. `ObjectDetectorPreBS` przeprowadza wstępne skalowanie i filtrowanie obrazu, `ObjectDetectorBS` dokonuje właściwego odejmowania tła, natomiast `ObjectDetectorPostBS` realizuje filtrację końcową i detekcję cienia. W celu zwiększenia wydajności przetwarzania, jednocześnie uruchamianych jest kilka instancji `ObjectDetectorBS`, które działają wielowątkowo na określonych fragmentach ramki obrazu,

- `ObjectTracker` – śledzenie ruchomych obiektów poruszających się w polu widzenia kamery (rozdział 4.2),
- `CameraManagement` – przetwarzanie związane z kamerami, a przede wszystkim konwersja dwukierunkowa między współrzędnymi dwuwymiarowymi zdefiniowanymi wewnątrz ramki obrazu i współrzędnymi trójwymiarowymi świata rzeczywistego przy użyciu wyników kalibracji pola widzenia kamery (rozdział 6.1.1),
- `DelayedProcessing` – filtracja czasowa wyników detekcji i śledzenia obiektów (rozdział 4.2.2),
- `ObjectClassifier` – klasyfikacja typu obiektu (rozdział 3.3),
- `EventDetector` – detekcja zdarzeń podstawowych definiowanych w polu widzenia kamery, takich jak przekroczenie granicy, wejście w obszar, chwilowe zatrzymanie się lub rozpoczęcie poruszania; algorytmy zaimplementowane w tym module zostały przedstawione w rozdziałach 6.1 i 6.2 dotyczących detekcji zdarzeń wyższego poziomu, gdzie są wykorzystywane,
- `GlobalTracker` – reidentyfikacja i śledzenie obiektów przemieszczających się pomiędzy polami widzenia kamer (rozdział 4.4),
- `ParkingEvents` – detekcja zdarzeń na parkingu (rozdział 6.1),
- `TrafficEvents` – detekcja zdarzeń w ruchu drogowym (rozdział 6.2).

Moduły oznaczone na rys. 7.2 kolorem szarym mają charakter techniczny i pomocniczy:

- `FrameDecoder` – dekodowanie ramki obrazu, czyli konwersja danych natywnych otrzymanych ze źródła na mapę bitową, która może być analizowana algorytmami przetwarzania obrazu,

- `GenerateMetadata` – generowanie metadanych, umożliwiających wizualizację wyników działania algorytmów dla każdej ramki obrazu,
- `OutputTasks` – generowanie danych wyjściowych zawierających rezultaty działania algorytmów w celu wysłania ich do serwera.



Rys. 7.2 Kompletny schemat przetwarzania strumienia wizyjnego

7.3 Wydajność działania

Niniejszy podrozdział prezentuje analizę wydajności przetwarzania obrazu przez platformę. Przede wszystkim zbadano, jakie jest jej zapotrzebowanie na moc obliczeniową oraz na pamięć oraz określono jak zmienia się wydajność analizy w zależności od liczby dostępnych wątków oraz liczby równocześnie analizowanych strumieni wizyjnych. Ponadto przeprowadzono analizę wydajności działania poszczególnych modułów przetwarzania obrazu w celu określenia, czy któryś z nich nie stanowi „wąskiego gardła” całego procesu przetwarzania.

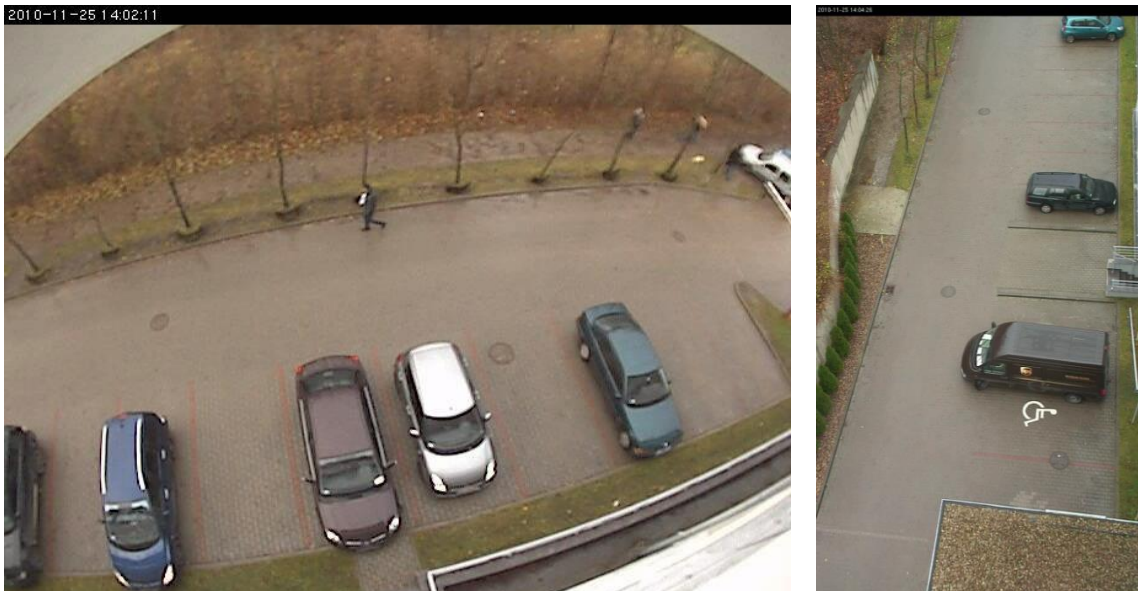
Do testów wydajności wykorzystano trzy rzeczywiste nagrania wykonane w Gdańskim Parku Naukowo-Technologicznym (GPNT). Celem analizy była detekcja

zdarzeń na parkingu. Oznacza to, że nie było możliwe równoczesne przetestowanie wydajności działania modułu detekcji zdarzeń w ruchu drogowym; detekcja obu tych typów zdarzeń nie jest możliwa w tym samym strumieniu wizyjnym, ze względu na ich odmienny charakter.

Użyte nagrania różnią się rozdzielczością (tab. 7.1). Ponieważ żadna kamera w GPNT nie działa z rozdzielczością HDTV 1080p, nagranie o tej rozdzielczości powstało poprzez przeskalowanie nagrania o niższej rozdzielczości. Ponadto, dla każdego kadru ustawiono obszar zainteresowania (ROI, ang. *Region of Interest*), w którym prowadzone było przetwarzanie obrazu; piksele spoza tego obszaru (np. trawniki) były wyłączone z analizy, co pozwala uzyskać wyższą wydajność. Na rys. 7.3 pokazano przykładowe ramki z wykorzystanych nagrań. Do analizy wydajności użyto komputer, którego konfigurację przedstawiono w tab. 7.2.

Tab. 7.1 Charakterystyka nagrań testowych do badania wydajności algorytmów

Numer nagrania	Rozdzielczość	Procent pikseli obrazu w ROI	Czas trwania [mm:ss]
N1	VGA (640x480)	82%	23:27
N2	HDTV 720p (1280x720)	78%	23:12
N3	HDTV 1080p (1920x1080)	78%	23:12



Rys. 7.3 Przykładowe ramki z nagrań testowych N1 (z lewej) oraz N2 i N3 (z prawej)

Tab. 7.2 Konfiguracja komputera użytego do testów wydajności

Cecha	Komputer testowy
Procesor	1x Intel Xeon W3530
Taktowanie procesora	2,80 GHz
Liczba rdzeni logicznych	8
Pamięć operacyjna	6 GB
Moduły pamięci	DDR3-1066
Max. przepustowość modułów pamięci	8,5 GB/s
System operacyjny	Windows 7 64 bit
Architektura oprogramowania	32 bit

7.3.1 Prędkość przetwarzania i wykorzystanie procesora

W pierwszym eksperymencie dokonano przetworzenia wszystkich trzech nagrań testowych tak szybko, jak to jest możliwe w celu określenia możliwej do osiągnięcia prędkości przetwarzania wyrażonej w liczbie ramek na sekundę. Eksperymenty wykonano przy różnej liczbie wątków przydzielonych do wykorzystania przez platformę. Liczba wątków określa ile modułów (zadań obliczeniowych) może maksymalnie działać w tej samej chwili oraz, jednocześnie, ile będzie wykorzystanych modułów `ObjectBackgroundBS`, realizujących funkcje detekcji obiektów ruchomych na częściach ramki obrazu. Wyniki pokazano w tab. 7.3.

Tab. 7.3 Wydajność obliczeniowa (w ramach na sekundę) w zależności od nagrania i liczby wykorzystanych wątków

Numer nagrania	Wydajność [fps – ramki na sekundę]					
	1 wątek	2 wątki	3 wątki	4 wątki	6 wątków	8 wątków
N1	20,9	39,3	54,7	61,6	73,5	79,8
N2	7,2	13,5	18,6	21,2	25,5	27,5
N3	3,2	5,9	8,1	9,5	11,3	12,2

Do optymalnego działania algorytmów analizy obrazu wymagane jest przetwarzanie strumienia wizyjnego z prędkością co najmniej 15 fps. Z powyższej tabeli widać, że warunek ten nie jest spełniony tylko dla nagrania o najwyższej rozdzielczości.

Aby osiągnąć założoną prędkość przetwarzania wprowadzono do algorytmu detekcji obiektów ruchomych tzw. współczynnik skalowania (rozdział 4.3.1), który powoduje proporcjonalne zmniejszenie rozdzielczości obrazu (np. dla współczynnika skalowania 0,5 zarówno wysokość jak i szerokość zmniejszana jest o połowę, a więc liczba pikseli zmniejsza się czterokrotnie) na potrzeby działania modułu `ObjectDetectorBS`, a jego wyniki są skalowane do oryginalnej wielkości obrazu wejściowego; wszystkie inne moduły działają na obrazie o oryginalnej rozdzielczości.

W tab. 7.4 pokazano osiągniętą wydajność przetwarzania obrazu w zależności od nagrania i liczby wątków dla współczynnika skalowania równego 0,5. W tym przypadku, nawet dla największej rozdzielczości, nie ma problemów z osiągnięciem wymaganej prędkości przetwarzania. Oznacza to, że pojedynczy komputer jest w stanie przetworzyć w czasie rzeczywistym, z prędkością co najmniej 15 fps, dowolny strumień wizyjny o rozdzielczości do HDTV 1080p.

Tab. 7.4 Wydajność obliczeniowa (w ramach na sekundę) w zależności od nagrania i liczby wykorzystanych wątków dla współczynnika skalowania równego 0,5

Numer nagrania	Wydajność [fps – ramki na sekundę]					
	1 wątek	2 wątki	3 wątki	4 wątki	6 wątków	8 wątków
N1	65,3	118,8	148,4	157,9	156,5	164,3
N2	23,1	42,3	54,8	61,2	61,8	60,6
N3	10,4	18,8	24,3	26,3	26,9	24,8

Analiza osiągniętej prędkości przetwarzania w funkcji liczby uruchomionych wątków wskazuje, że wydajność wyraźnie wzrasta przy zwiększaniu liczby wątków od 1 do 4; powyżej tej wartości nie widać istotnych różnic, szczególnie dla współczynnika skalowania 0,5. Przyczyną takiego stanu rzeczy jest zapewne ograniczenie fizycznej prędkości akwizycji sygnału wizyjnego, związane z odczytem nagrania testowego z dysku i jego dekodowaniem.

Wnioski te potwierdza analiza wykorzystania procesora przez proces analizy obrazu, pokazana w tab. 7.5. Wartości powyżej 12,5% w przypadku przetwarzania jednowątkowego wynikają z narzutu wnoszonego przez samą platformę, związanego przede wszystkim z odczytem plików z dysku, który jest tym większy, im większa jest liczba ramek przetwarzanych w jednostce czasu. Analizując tab. 7.5 można dostrzec, że zużycie procesora pozostaje niezależne od analizowanego nagrania (jego rozdzielczości) przy tej samej wartości współczynnika skalowania oraz liczby wątków. Ponadto zużycie procesora przestaje wzrastać przy zastosowaniu czterech i więcej wątków dla nagrań ze współczynnikiem skalowania 0,5. Potwierdza to tezę o przyczynie ograniczenia wydajności przetwarzania leżącej w procesie akwizycji sygnału wizyjnego, który nie zależy od wartości współczynnika skalowania; przy braku skalowania, konieczne jest przeprowadzenie odpowiednio większej ilości obliczeń, co znalazło odzwierciedlenie w ciągłym, blisko liniowym wzroście obciążenia procesora dla nagrań ze współczynnikiem skalowania równym 1 w funkcji liczby wątków.

Tab. 7.5 Średnie obciążenie procesora procesem przetwarzania obrazu (100% = całkowite obciążenie 8 rdzeni, 12,5% = całkowite obciążenie jednego rdzenia)

Numer nagrania	Wartość współczynnika skalowania	Średnie obciążenie procesora					
		1 wątek	2 wątki	3 wątki	4 wątki	6 wątków	8 wątków
N1	1	13,0%	26,4%	39,9%	52,9%	79,5%	94,7%
N2	1	13,0%	26,3%	39,6%	52,7%	78,4%	94,0%
N3	1	13,1%	26,5%	39,8%	53,0%	78,5%	93,9%
N1	0,5	15,4%	31,2%	46,1%	51,7%	57,2%	57,8%
N2	0,5	15,3%	30,7%	45,0%	57,0%	60,0%	60,8%
N3	0,5	15,7%	31,3%	46,7%	56,0%	57,4%	55,7%

7.3.2 Równoczesna analiza wielu strumieni wizyjnych

Widoczny w tab. 7.4 brak wzrostu wydajności przetwarzania w funkcji rosnącej liczby wątków oznacza wolne, możliwe do zagospodarowania moce obliczeniowe. Dlatego można przypuszczać, że możliwe będzie jednoczesne przetwarzanie kilku strumieni wizyjnych bez istotnego spadku wydajności. W tym celu równolegle, w ramach jednej usługi przetwarzania obrazu, uruchomiono wielokrotnie analizę tego samego nagrania testowego. Wyniki umieszczono w tab. 7.6. Wynika z nich, że przy braku skalowania obrazu możliwe jest równoczesne przetwarzanie nawet 5 strumieni wizyjnych o rozdzielczości VGA. Zastosowanie skalowania umożliwia przetwarzanie 15 strumieni VGA, 5 strumieni o rozdzielczości HDTV 720p lub dwóch o rozdzielczości 1080p przez pojedynczy komputer.

Tab. 7.6 Liczba strumieni wizyjnych możliwych do równoczesnego przetwarzania na jednym komputerze (8 wątków)

Numer nagrania	Wartość współczynnika skalowania	Wydajność [fps] dla określonej liczby strumieni					Maksymalna liczba strumieni wizyjnych dla wydajności 15 fps
		1	2	4	8	16	
N1	1	79,8	41,8	20,9	10,5	*	5
N2	1	27,5	14,2	9,5	*	*	1
N3	1	12,2	*	*	*	*	< 1
N1	0,5	164,3	113,2	59,8	29,1	14,6	15
N2	0,5	60,6	38,9	20,1	*	*	5
N3	0,5	24,8	16,8	*	*	*	2

* nie możliwe do zbadania ze względu na przekroczenie limitu 2GB wykorzystanej pamięci, nakładanego przez system operacyjny na procesy 32-bitowe

7.3.3 Przepustowość podsystemu pamięci i jej użycie

Istotną kwestią działania platformy analizy obrazu jest jej zapotrzebowanie na przepustowość systemu pamięci. W praktyce jest ono trudne do wyznaczenia, jednak można się pokusić o oszacowanie jego dolnej granicy. Wykorzystany zostanie do tego celu

moduł detekcji obiektów ruchomych, a w szczególności `ObjectDetectorBS`, którego zapotrzebowanie na zasoby pamięci jest, w porównaniu z innymi modułami pamięci, zdecydowanie największe. Moduł ten odpowiada m.in. za uaktualnianie modelu tła analizowanej sceny. Model ten zajmuje stosunkowo dużo pamięci (224 bajty na jeden piksel obrazu), a musi być w uaktualniany (tzn. w całości odczytywany i częściowo zmieniany) dla każdej ramki obrazu. Stosowne wyliczenia pokazano w tabeli tab. 7.7. Widać z nich, że dolne oszacowanie wymaganej przepustowości podsystemu pamięci, szczególnie dla rozdzielczości HDTV 1080p i braku skalowania (6,49 GB/s) zbliża się do maksymalnego, teoretycznego transferu modułów pamięci zainstalowanych w testowym komputerze. W związku z tym przepustowość pamięci może być wąskim gardłem dla dużych rozdzielczości, wysokiej liczby ramek na sekundę i wielu równocześnie przetwarzanych strumieni, biorąc pod uwagę zapotrzebowania na dostęp do pamięci pozostałych modułów analizy obrazu oraz samej platformy i systemu operacyjnego komputera.

Tab. 7.7 Szacunkowe zapotrzebowanie na przepustowość podsystemu pamięci

Numer nagrania	Wartość współczynnika skalowania	Wielkość modelu tła w module <code>ObjectDetectorBS</code>	Wymagana przepustowość dla 15 fps
N1	1	65,6 MB	984 MB/s
N2	1	196,9 MB	2954 MB/s
N3	1	443,0 MB	6645 MB/s
N1	0,5	16,4 MB	246 MB/s
N2	0,5	49,2 MB	738 MB/s
N3	0,5	110,7 MB	1661 MB/s

Ostatnim zbadanym elementem jest zapotrzebowanie na pamięć operacyjną, pokazane w tab. 7.8. Wynika z niego, że przetwarzanie strumienia o wysokiej rozdzielczości powoduje wykorzystanie pamięci w ilości zbliżającej się do 2GB limitu istniejącego dla aplikacji 32-bitowych. Oznacza to, że równoczesne przetwarzanie większej liczby strumieni wymaga kompilacji oprogramowania w architekturze 64-bitowej, co znalazło potwierdzenie w wynikach zaprezentowanych w tab. 7.6.

Tab. 7.8 Średnia zajętość pamięci operacyjnej przez oprogramowanie Stacji Monitorującej

Numer nagrania	Współczynnik skalowania równy 1 (brak skalowania)	Współczynnik skalowania równy 0,5
N1	193 MB	79 MB
N2	569 MB	187 MB
N3	1252 MB	390 MB

7.3.4 Analiza wydajności poszczególnych modułów przetwarzania obrazu

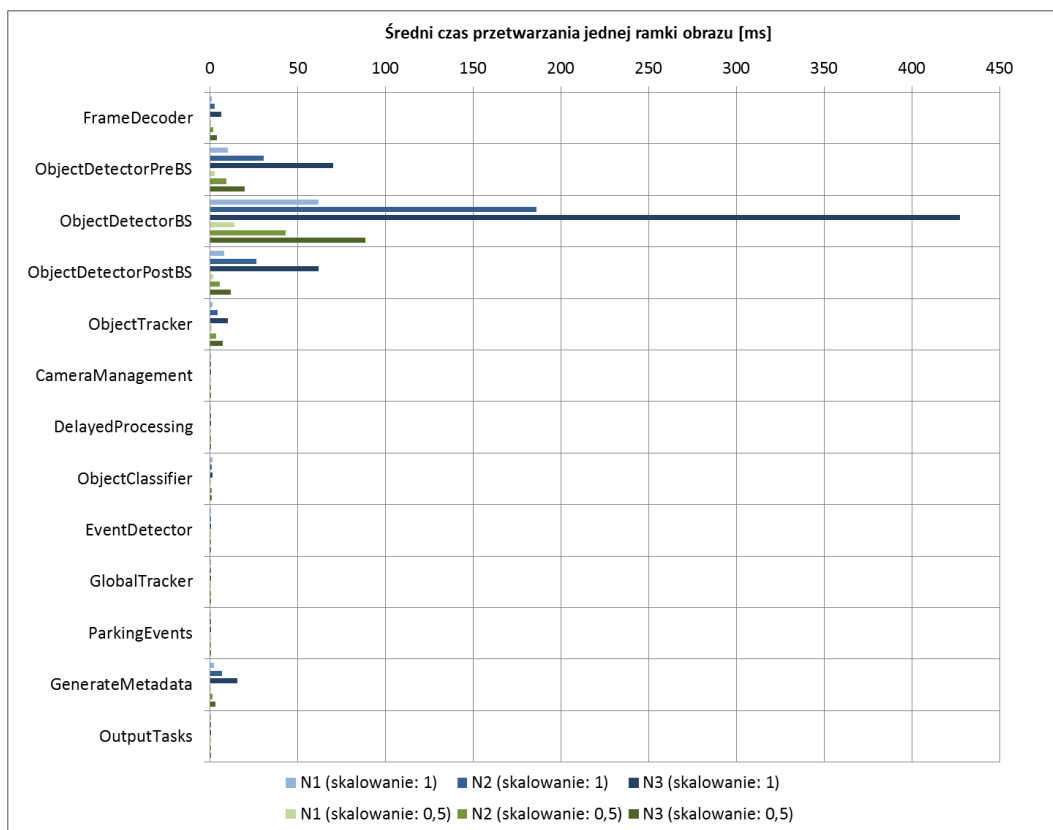
W tab. 7.9 przedstawiono szczegółowo wkład każdego modułu w średni czas przetwarzania jednej ramki obrazu (przy użyciu wszystkich 8 wątków). Rezultaty zobrazowano na rys. 7.4. Z tabeli wynika, że zdecydowanie najbardziej złożone obliczeniowo są moduły detekcji obiektów ruchomych (ObjectDetectorPreBS, ObjectDetectorBS i ObjectDetectorPostBS), które łącznie zajmują od 85% (dla współczynnika skalowania 0,5) do 94% (brak skalowania) czasu przetwarzania przez wszystkie moduły razem; pozostałe moduły zajmują łącznie tylko od kilku do kilkunastu procent czasu przetwarzania.

Tab. 7.9 Średni czas przetwarzania jednej ramki przez poszczególne moduły analizy obrazu (8 wątków)

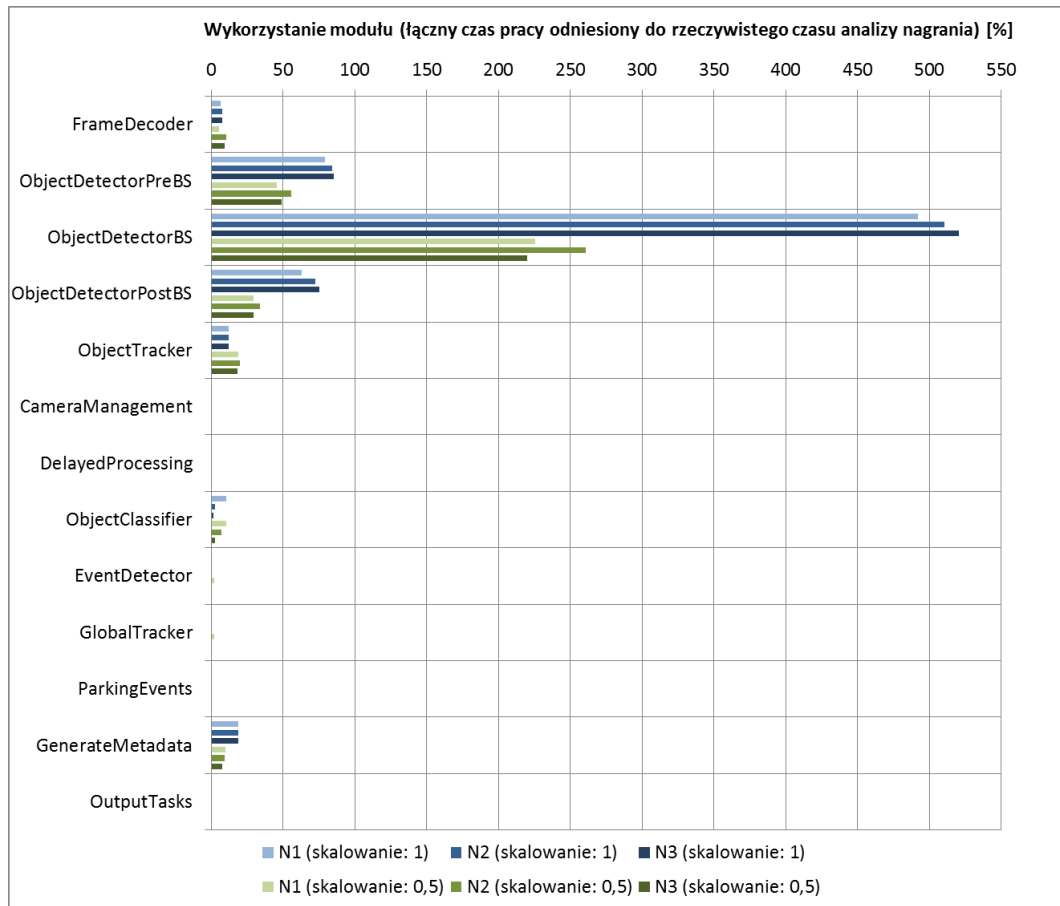
Nazwa modułu	Średni czas przetwarzania jednej ramki obrazu					
	Nagranie N1		Nagranie N2		Nagranie N3	
	[ms]	[%]	[ms]	[%]	[ms]	[%]
<i>Współczynnik skalowania obrazu równy 1</i>						
FrameDecoder	0,84	0,98	2,81	1,09	6,41	1,08
ObjectDetectorPreBS	9,97	11,62	30,62	11,84	70,04	11,82
ObjectDetectorBS	61,71	71,91	186,07	71,95	427,30	72,09
ObjectDetectorPostBS	7,89	9,19	26,44	10,22	61,61	10,39
ObjectTracker	1,50	1,74	4,37	1,69	10,15	1,71
CameraManagement	0,01	0,01	0,01	>0	0,01	>0
DelayedProcessing	0,03	0,03	0,04	0,02	0,03	>0
ObjectClassifier	1,29	1,51	1,04	0,40	1,39	0,23
EventDetector	0,09	0,10	0,21	0,08	0,15	0,03
GlobalTracker	0,09	0,10	0,07	0,03	0,10	0,02
ParkingEvents	0,02	0,02	0,02	0,01	0,02	>0
GenerateMetadata	2,35	2,74	6,89	2,66	15,50	2,62
OutputTasks	0,01	0,02	0,02	0,01	0,02	>0
Wszystkie moduły	85,8	100,00	258,6	100,00	592,73	100,00
<i>Współczynnik skalowania obrazu równy 0,5</i>						
FrameDecoder	0,35	1,62	1,70	2,58	3,70	2,73
ObjectDetectorPreBS	2,79	13,02	9,17	13,95	19,67	14,51
ObjectDetectorBS	13,77	64,32	43,05	65,47	88,64	65,38
ObjectDetectorPostBS	1,79	8,36	5,58	8,49	11,85	8,74
ObjectTracker	1,16	5,40	3,34	5,08	7,37	5,44
CameraManagement	0,01	0,04	0,01	0,02	0,01	0,01
DelayedProcessing	0,02	0,08	0,02	0,03	0,02	0,02
ObjectClassifier	0,64	2,97	1,16	1,77	1,01	0,75
EventDetector	0,13	0,59	0,10	0,15	0,11	0,08
GlobalTracker	0,13	0,60	0,05	0,08	0,06	0,04
ParkingEvents	0,01	0,07	0,01	0,02	0,01	0,01
GenerateMetadata	0,60	2,80	1,53	2,33	3,08	2,27
OutputTasks	0,01	0,06	0,01	0,02	0,01	0,01
Wszystkie moduły	21,41	100,00	65,73	100,00	135,54	100,00

W tab. 7.10 przedstawiono podsumowanie wykorzystania poszczególnych modułów w odniesieniu do rzeczywistego czasu analizy nagrania testowego. Wyniki zobrazowano na rys. 7.5. Należy przypomnieć, że wszystkie moduły oprócz `ObjectDetectorBS` działają jednowątkowo, więc ich maksymalne wykorzystanie wynosi 100%; dla modułu `ObjectDetectorBS` maksymalne teoretyczne wykorzystanie wynosi 800%.

Zawartość tab. 7.10 potwierdza wnioski wynikające z analizy tab. 7.9, tzn. zdecydowana większość czasu przetwarzania strumienia wizyjnego jest poświęcona zadaniom detekcji ruchomych obiektów. Widać poza tym, że wykorzystanie żadnego z modułów nie jest zbliżone do maksimum, co oznacza, że to nie implementacja konkretnego algorytmu przetwarzania obrazu stanowi ograniczenie wynikowej wydajności całego procesu analizy. Ponownie potwierdza to tezę, że jest nim odczyt i dekodowanie nagrania filmowego z dysku komputera.



Rys. 7.4 Średni czas przetwarzania jednej ramki obrazu przez poszczególne moduły analizy obrazu (8 wątków)



Rys. 7.5 Wykorzystanie poszczególnych modułów analizy obrazu (łączny czas przetwarzania obrazu przez moduł odniesiony do rzeczywistego czasu trwania analizy nagrania testowego)

Tab. 7.10 Wykorzystanie poszczególnych modułów analizy obrazu (łączy czas przetwarzania obrazu przez moduł odniesiony do rzeczywistego czasu trwania analizy nagrania testowego)

Nazwa modułu	Średni czas przetwarzania jednej ramki obrazu					
	Nagranie N1		Nagranie N2		Nagranie N3	
	[s]	[%]	[s]	[%]	[s]	[%]
<i>Współczynnik skalowania obrazu równy 1</i>						
FrameDecoder	17,71	6,69	58,65	7,71	133,85	7,81
ObjectDetectorPreBS	210,39	79,47	639,32	83,99	1462,3	85,29
ObjectDetectorBS	1302,13	491,82*	3884,88	510,38*	8920,49	520,29*
ObjectDetectorPostBS	166,39	62,85	552,08	72,53	1286,21	75,02
ObjectTracker	31,57	11,92	91,23	11,99	211,89	12,36
CameraManagement	0,24	0,09	0,25	0,03	0,27	0,02
DelayedProcessing	0,56	0,21	0,86	0,11	0,59	0,03
ObjectClassifier	27,31	10,32	21,63	2,84	28,92	1,69
EventDetector	1,8	0,68	4,4	0,58	3,18	0,19
GlobalTracker	1,81	0,69	1,5	0,2	2,02	0,12
ParkingEvents	0,4	0,15	0,33	0,04	0,34	0,02
GenerateMetadata	49,58	18,73	143,8	18,89	323,61	18,88
OutputTasks	0,29	0,11	0,33	0,04	0,36	0,02
Rzeczywisty czas trwania analizy	264,75	100,00	761,17	100,00	1714,50	100,00
<i>Współczynnik skalowania obrazu równy 0,5</i>						
FrameDecoder	7,3	5,68	35,48	10,29	77,2	9,17
ObjectDetectorPreBS	58,78	45,7	191,54	55,55	410,68	48,79
ObjectDetectorBS	290,32	225,7*	898,72	260,65*	1851,04	219,93*
ObjectDetectorPostBS	37,76	29,35	116,51	33,79	247,5	29,41
ObjectTracker	24,39	18,96	69,81	20,25	153,9	18,28
CameraManagement	0,2	0,16	0,22	0,06	0,2	0,02
DelayedProcessing	0,35	0,27	0,41	0,12	0,43	0,05
ObjectClassifier	13,41	10,43	24,28	7,04	21,16	2,51
EventDetector	2,65	2,06	2,11	0,61	2,23	0,26
GlobalTracker	2,73	2,12	1,14	0,33	1,23	0,15
ParkingEvents	0,31	0,24	0,27	0,08	0,27	0,03
GenerateMetadata	12,65	9,84	32,02	9,29	64,35	7,65
OutputTasks	0,25	0,2	0,25	0,07	0,26	0,03
Rzeczywisty czas trwania analizy	128,62	100,00	344,79	100,00	841,69	100,00

* moduł działa wielowątkowo

8. Wnioski i podsumowanie

W ramach rozprawy przedstawiono, opracowano, zaimplementowano i zbadano metody przetwarzania obrazu składające się na system detekcji zdarzeń w ruchu drogowym. W szczególności prace dotyczyły detekcji i śledzenia ruchomych obiektów, reidentyfikacji obiektów w polach widzenia różnych kamer oraz klasyfikacji typu obiektu i typu pojazdu. Elementy składowe zostały zintegrowane na potrzeby dwóch rozwiązań służących do detekcji zdarzeń na parkingu oraz do wykrywania niedozwolonej zmiany pasa ruchu przez pojazdy na ruchliwym skrzyżowaniu. Eksperymenty przeprowadzone w oparciu o nagrania testowe wykonane z użyciem rzeczywistych instalacji monitoringu wykazały, że poszczególne algorytmy indywidualnie, jak i kompletny system, charakteryzują się poprawnym działaniem i wysoką skutecznością wykrywania zdarzeń.

Do innowacyjnych rozwiązań przedstawionych w rozprawie, według orientacji jej autora, należą:

- rozwinięcie algorytmu detekcji obiektów ruchomych, pozwalające na jego poprawne działanie w warunkach zmiennego oświetlenia,
- algorytm rozwiązywania konfliktów śledzenia obiektów w obrazie,
- metoda reidentyfikacji tych samych obiektów obserwowanych przez różne kamery,
- udział w opracowaniu algorytmu klasyfikacji typu obiektu niezależnego od ustawienia obiektu względem kamery,
- badanie cech obrazu pojazdów i algorytmów uczących się oraz opracowanie rozwiązania klasyfikacji typu pojazdu niezależnego od orientacji kamery, nie wymagającego jej kalibracji,
- system detekcji zdarzeń na parkingu wdrożony praktycznie i działający nieprzerwanie od kilkunastu miesięcy przed zakończeniem redagowania rozprawy,
- algorytm wykrywania niepoprawnej zmiany pasa ruchu przez pojazdy w warunkach bardzo dużego natężenia ruchu i częstego, wzajemnego zasłaniania się pojazdów,
- udział w budowie uniwersalnej platformy sprzętowo-programistycznej do celu implementacji i uruchamiania algorytmów przetwarzania obrazu w czasie rzeczywistym.

W rozprawie przedstawiono badania związane z algorytmem klasyfikacji typu pojazdów na podstawie cech ich wyglądu, w których uwzględniono różne algorytmy uczące się. Eksperymenty dowiodły, że skuteczność rozpoznawania typu pojazdu jest wysoka i nie zależy istotnie od kierunku ruchu pojazdów oraz orientacji kamery względem jedni. Zaprezentowano także opracowane rozwiązanie zmierzające do identyfikacji tego samego pojazdu w strumieniach wizyjnych z różnych kamer, z wykorzystaniem lokalnych parametrów obrazu i sztucznych sieci neuronowych. Eksperymenty wykazały, że przy podobnym do siebie nawzajem ustawieniu kamer względem poruszających się pojazdów, opisywana metoda nadaje się do stosowania warunkach rzeczywistych. Stanowi to istotny element wyróżniający w stosunku do typowo stosowanych rozwiązań i dowodzi pierwszej tezy rozprawy, która brzmi:

1. Opracowane i przebadane eksperymentalnie cechy dystynktywne obrazu wizyjnego pojazdów umożliwiają skuteczną identyfikację i klasyfikację pojazdów dokonywaną z użyciem algorytmów uczących się.

W toku rozprawy opracowano i rozwinięto także algorytmy detekcji i śledzenia ruchomych obiektów w obrazie z kamery. Nacisk położono na dostosowanie istniejących rozwiązań do optymalnego działania w warunkach monitorowania ruchu drogowego. Opracowane rozwiązania zostały wykorzystane w autorskich algorytmach detekcji zdarzeń zweryfikowanych na parkingu i na ruchliwym skrzyżowaniu. Uzyskane eksperymentalnie wyniki działania świadczą o możliwości ich praktycznego stosowania. Badania wydajności opracowanej platformy przetwarzania obrazu wykazały, że możliwe jest przetwarzanie na jednym komputerze do kilkunastu różnych strumieni wizyjnych (w zależności od ich rozdzielczości). Dowodzi to drugiej tezy rozprawy:

2. Opracowane algorytmy detekcji i śledzenia ruchomych obiektów dostosowane do działania w warunkach rzeczywistych, pozwalają na szybkie i skuteczne z praktycznego punktu widzenia wykrywanie kluczowych zdarzeń związanych z parkowaniem pojazdów oraz istotnych zdarzeń zagrażających bezpieczeństwu ruchu drogowego.

Poprzez udowodnienie postawionych we wprowadzeniu tez osiągnięty został założony cel rozprawy.

Przyszłe planowane prace związane z tematyką niniejszej rozprawy będą dotyczyły rozwijania opracowanych modułów detekcji zdarzeń w ruchu drogowym w celu umożliwienia wykrywania większej liczby predefiniowanych sytuacji. W szczególności, badania będą dotyczyły takich zdarzeń jak zatrzymanie pojazdu na środku skrzyżowania,

stwarzające poważne zagrożenie dla innych uczestników ruchu drogowego oraz przekraczanie dozwolonej prędkości przez wiele pojazdów jednocześnie.

Opracowane algorytmy detekcji i śledzenia obiektów oraz wykrywania zdarzeń podstawowych posłużą również jako podstawa do opracowania nowego rodzaju algorytmów detekcji zdarzeń niebezpiecznych, typowo występujących na dworcach i lotniskach, takich jak porzucenie bagażu i poruszanie się w kierunku/miejscu niedozwolonym.

Osobny nurt planowanego rozwoju prac będzie dotyczył dostosowania opracowanych algorytmów do działania na procesorach sygnałowych DSP i na układach FPGA w celu dalszego zwiększania wydajności przetwarzania obrazu i jednocześnie zmniejszania zużycia energii.

9. Bibliografia

- [1] Arulampalam S., Maskell S., Gordon N., Clapp T.: “A Tutorial on Particle Filters for On-Line Non-Linear/Non-Gaussian Bayesian Tracking”, *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 174–189, 2002.
- [2] Ashbrook A. P., Thacker N. A., Rockett P. . I.: “Multiple shape recognition using pairwise geometric histogram based algorithms”, *Proc. International Conference on Image Processing and its Applications*, pp. 90–94, 1995.
- [3] Avidan S.: “Support vector tracking”, *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 184–191, 2001.
- [4] Bang-Jensen J., Gutin G., Yeo A.: “When the greedy algorithm fails”, *Discrete Optimization*, vol. 1, pp. 121–127, 2004.
- [5] Barron J. L., Fleet D. J., Beauchemin S. S.: “Performance of optical flow techniques”, *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.
- [6] Bar-Shalom Y., Fortmann T.: *Tracking and Data Association*. Academic Press, 1988.
- [7] Baugher M., McGrew D., Naslund M., Carrara E.: “The Secure Real-time Transport Protocol (SRTP), RFC 3711”. 2004.
- [8] Bay H., Tuytelaars T., Gool L. Van: “SURF: Speeded Up Robust Features”, *Proc. Computer Vision – ECCV*, vol. 3951, pp. 404–417, 2006.
- [9] Belaroussi R., Foucher P., Tarel J.-P., Soheilian B., Charbonnier P., Paparoditis N.: “Road Sign Detection in Images: A Case Study”, *Proc. International Conference on Pattern Recognition*, pp. 484–488, 2010.
- [10] Bertalmio M., Sapiro G., Andrandall G.: “Morphing active contours”, *IEEE Trans. Patt. Analy. Mach. Intell.*, vol. 22, no. 7, pp. 733–737, 2000.
- [11] Birchfield S.: “Elliptical head tracking using intensity gradients and color histograms”, *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 232–237, 1998.
- [12] Bird N. D., Masoud O., Papanikolopoulos N. P., Isaacs A.: “Detection of Loitering Individuals in Public Transportation Areas”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 2, pp. 167–177, Jun. 2005.
- [13] Black M., Anandan P.: “The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields”, *Comput. Vision Image Understand*, vol. 63, no. 1, pp. 75–104, 1996.
- [14] Black M., Jepson A.: “Eigentracking: Robust matching and tracking of articulated objects using a view-based representation”, *Int. J. Comput. Vision*, vol. 26, no. 1, pp. 63–84, 1998.
- [15] Bowyer K., Kranenburg C., Dougherty S.: “Edge detector evaluation using empirical roc curve”, *Comput. Vision Image Understand*, vol. 10, pp. 77–103, 2001.
- [16] Boykov Y., Huttenlocher D.: “Adaptive Bayesian Recognition in Tracking Rigid Objects”, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 697–704, 2000.
- [17] Breiman L.: “Random forests”, *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [18] Breiman L., Friedman J. H., Olshen R. A., Stone C. J.: *Classification and Regression Trees*. Wadsworth, 1984.
- [19] Bretzner L., Krantz M.: “Towards low-cost systems for measuring visual cues of driver fatigue and inattention in automotive applications”, *Proc. IEEE International Conference on Vehicular Electronics and Safety, 2005.*, pp. 161–164, 2005.

-
- [20] Buch N., Orwell J., Velastin S.: "Detection and classification of vehicles for urban traffic scenes", *Proc. 5th International Conference on Visual Information Engineering*, pp. 182–187, 2008.
- [21] Cai Q., Aggarwal J. K.: "Tracking human motion in structured environments using a distributed-camera system", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1241–1247, 1999.
- [22] Candamo J., Shreve M., Goldgof D., Sapper D. B., Kasturi R.: "Understanding Transit Scenes: A Survey on Human Behavior-Recognition Algorithms", *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 1, pp. 206–224, Mar. 2010.
- [23] Caselles V., Kimmel R., Sapiro G.: "Geodesic active contours", *Proc. IEEE International Conference on Computer Vision*, pp. 694–699, 1995.
- [24] Cédras C., Shah M.: "Motion-based recognition a survey", *Image and Vision Computing*, vol. 13, no. 2, pp. 129–155, Mar. 1995.
- [25] Cevikalp H., Kurt Z., Onarcan A. O.: "Return of the king: The Fourier transform based descriptor for visual object classification", *Proc. 21st Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, 2013.
- [26] Cham T., Rehg J.: "A Multiple Hypothesis Approach to Figure Tracking", *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 239–245, 1999.
- [27] Chang C. C., Lin C. J.: "LIBSVM: a library for support vector machines", *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.
- [28] Chen B.-H., Huang S.-C.: "An Advanced Moving Object Detection Algorithm for Automatic Traffic Monitoring in Real-World Limited Bandwidth Networks", *IEEE Transactions on Multimedia*, vol. PP, no. 99, 2014.
- [29] Chen G., Yu Z., Wen Q., Yu Y.: "Improved Gaussian Mixture Model for Moving Object Detection", *LNCS, Volume 7002, Artificial Intelligence and Computational Intelligence*, pp. 179–186, 2011.
- [30] Chen T.-H. (Chao-H., Lin Y.-F., Chen T.-Y.: "Intelligent Vehicle Counting Method Based on Blob Analysis in Traffic Surveillance", *Proc. International Conference on Innovative Computing, Informatio and Control (ICICIC)*, pp. 238–238, 2007.
- [31] Chen X., Zhang C.: "Incident Retrieval in Transportation Surveillance Videos - An Interactive Framework", *Proc. IEEE International Conference on Multimedia and Expo*, pp. 2186–2189, 2007.
- [32] Chen Y., Rui Y., Andhuang T.: "JPDAF based HMM for real-time contour tracking", *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 543–550, 2001.
- [33] Chen Z., Ellis T., Velastin S. A.: "Vehicle type categorization: A comparison of classification schemes", *Proc. 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 74–79, 2011.
- [34] Cheng H. Y., Hwang J. N.: "Multiple-Target Tracking for Crossroad Traffic Utilizing Modified Probabilistic Data Association", *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. I-921–I-924, 2007.
- [35] Chilgunde A., Kumar P., Ranganath S., Weimin H.: "Multi-Camera Target Tracking in Blind Regions of Cameras with Non-overlapping Fields of View", *Proc. British Machine Vision Conference*, 2004.
- [36] Choromanska A., Monteleoni C.: "Online Clustering with Experts", *Journal of Machine Learning Research: Workshop and Conference Proceedings, 15th*

- International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 22, pp. 227–235, 2012.
- [37] Collins R., Lipton A., Fujiyosji H., Kanade T.: “Algorithms for cooperative multisensor surveillance”, *Proceedings of IEEE*, vol. 89, no. 10, pp. 1456–1477, 2001.
- [38] Comaniciu D.: “Bayesian kernel tracking”, *Proc. Annual Conference of the German Society for Pattern Recognition*, pp. 438–445, 2002.
- [39] Comaniciu D., Meer P.: “Mean shift: a robust approach toward feature space analysis”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002.
- [40] Comaniciu D., Ramesh V., Meer P.: “Kernel-based object tracking”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, May 2003.
- [41] Cootes T. F., Edwards G. J., Taylor C. J.: “Active Appearance Models”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.
- [42] Cox I. J.: “A review of statistical data association techniques for motion correspondence”, *International Journal of Computer Vision*, vol. 10, no. 1, pp. 53–66, Feb. 1993.
- [43] Cox I. J., Hingorani S. L.: “An efficient implementation of Reid’s multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 2, pp. 138–150, 1996.
- [44] Cremers D., Schnorr C.: “Statistical shape knowledge in variational motion segmentation”, *Image and Vision Computing*, vol. 21, no. 1, pp. 77–86, 2003.
- [45] Cucchiara R., Piccardi M., Mello P.: “Image analysis and rule-based reasoning for a traffic monitoring system”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 2, pp. 119–130, Jun. 2000.
- [46] Czyżewski A. et al.: “Multimedialny system wspomagający identyfikację i zwalczanie przestępczości oraz terroryzmu”, *Praktyczne elementy zwalczania przestępczości zorganizowanej i terroryzmu: Nowoczesne technologie i praca operacyjna*, Paprzycki L., Rau Z. (Ed.). Wolter Kluwers Polska, pp. 211–227, 2009.
- [47] Czyżewski A., Dalka P.: “Examining Kalman Filters Applied to Tracking Objects in Motion”, *Proc. Ninth International Workshop on Image Analysis for Multimedia Interactive Services*, pp. 175–178, 2008.
- [48] Czyżewski A., Dalka P.: “Moving object detection and tracking for the purpose of multimodal surveillance system in urban areas”, *New Directions in Intelligent Interactive*, vol. 142, pp. 75–84, 2008.
- [49] Czyżewski A., Dalka P.: “Teleinformatyczna akwizycja i przetwarzanie danych dla potrzeb diagnostyki środowiska aglomeracji miejskich”, *Inteligentne wydobywanie informacji w celach diagnostycznych*, vol. 21, no. 5, Kowalczyk Z., Wiszniewski B. (Ed.). Pomorskie Wydawnictwo Naukowo-Techniczne, pp. 241–256, 2007.
- [50] Czyżewski A., Dalka P.: “Visual Traffic Noise Monitoring in Urban Areas”, *International Journal of Multimedia and Ubiquitous Engineering*, vol. 2, no. 3, pp. 91–102, 2007.
- [51] Czyżewski A., Kotus J., Szczodrak M.: “Online Urban Acoustic Noise Monitoring System”, *Noise Control Eng. J.*, vol. 60, no. 1, pp. 69–84, 2012.
- [52] D’Arminio P., Buendia-Iglesias R., Cichowski J., Dalka P., Ellwart D., Orwell J.: “Technologies for granting balance between security and privacy in video-

- surveillance”, Proc. *European Intelligence and Security Informatics Conference*, pp. 278–283, 2012.
- [53] Dalal N., Triggs B.: “Histograms of Oriented Gradients for Human Detection”, Proc. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 886–893, 2005.
- [54] Dalal N., Triggs B., Schmid C.: “Human detection using oriented histograms of flow and appearance”, Proc. *Eur. Conf. Comput. Vis.*, pp. 428–441, 2006.
- [55] Dalka P.: “Detection and segmentation of moving vehicles and trains using Gaussian mixtures, shadow detection and morphological processing”, *Machine Graphics & Vision International Journal*, vol. 15, no. 3/4, pp. 339–348, 2006.
- [56] Dalka P.: “Multi-camera Vehicle Tracking Using Local Image Features and Neural Networks”, Proc. *Multimedia Communications, Services and Security*, pp. 58–67, 2012.
- [57] Dalka P., Ciarkowski A., Szczuko P., Czyżewski A.: “Open standards-based communication system for distributed intelligent surveillance solution”, *Przeegląd Telekomunikacyjny*, no. 8–9, pp. 777–785, 2013.
- [58] Dalka P., Ciarkowski A., Szczuko P., Szwoch G., Czyżewski A.: “Surveillance Camera Tracking of Geo positioned Objects”, *New Directions in Intelligent Interactive Multimedia Systems and Services - 2*, vol. 226, Damiani E., Jeong J., Howlett R. J., Jain L. C. (Ed.). Berlin Heidelberg: Springer, pp. 21–30, 2009.
- [59] Dalka P., Czyżewski A.: “Human-Computer Interface Based on Visual Lip Movement and Gesture Recognition”, *International Journal of Computer Science and Applications*, vol. 7, no. 3, pp. 124–139, 2010.
- [60] Dalka P., Czyżewski A.: “Vehicle classification based on soft computing algorithms”, *Rough Sets and Current Trends in Computing*, pp. 70–79, 2010.
- [61] Dalka P., Ellwart D., Szwoch G.: “Camera orientation-independent parking events detection”, Proc. *12th Int. Workshop on Image Analysis for Mult. Interact. Services (WIAMIS)*, 2011.
- [62] Dalka P., Kostek B.: “Vowel recognition based on acoustic and visual features”, *Archives of Acoustics*, vol. 31, no. 3, pp. 1–14, 2006.
- [63] Dalka P., Szwoch G., Ciarkowski A.: “Distributed Framework for Visual Event Detection in Parking Lot Area”, *Multimedia Communications, Services and Security*, vol. 149, Dziech A., Czyżewski A. (Ed.). Berlin Heidelberg: Springer, pp. 37–45, 2011.
- [64] Dalka P., Szwoch G., Szczuko P., Czyżewski A.: “Video Content Analysis in the Urban Area Telemonitoring System”, *Multimedia Services in Intelligent Environments*, vol. 3, pp. 241–261, Dec. 2010.
- [65] Daugman J. G.: “Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression”, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 7, pp. 1169–1179, Jul. 1988.
- [66] Davis S. C., Diegel S. W.: *Transportation Energy Data Book: Edition 24-2004*. Oak Ridge National Laboratory, Oak Ridge, TN, USA, 2004.
- [67] Dempster P. A., Laird M. N., Rubin D. B.: “Maximum Likelihood from Incomplete Data via the EM Algorithm”, *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.
- [68] Dierks T., Allen C.: “The TLS Protocol Version 1.0, RFC 2246”. 1999.
- [69] Doermann D., Mihalcik D.: “Tools and techniques for video performance evaluation”, Proc. *15th International Conference on Pattern Recognition*, vol. 4, pp. 167–170, 2000.

- [70] Dong N. et al.: “Traffic Abnormality Detection through Directional Motion Behavior Map”, Proc. *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 80–84, 2010.
- [71] Dong Z., Jia Y.: “Vehicle type classification using distributions of structural and appearance-based features”, Proc. *IEEE International Conference on Image Processing*, pp. 4321–4324, 2013.
- [72] Doucet A., Godsill S., C. A.: “On Sequential Monte Carlo Sampling Methods for Bayesian Filtering”, *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.
- [73] Dougherty E. R., Lotufo R. A.: *Hands-on Morphological Image Processing*. SPIE Press, 2003.
- [74] Du W., Piater J.: “Multi-camera people tracking by collaborative particle filters and principal axis-based integration”, Proc. *8th Asian conference on Computer vision (ACCV)*, pp. 365–374, 2007.
- [75] Du Y., Feng Y.: “Vehicle detection from video sequence based on gabor filter”, Proc. *9th International Conference on Electronic Measurement & Instruments*, pp. 2–375–2–379, 2009.
- [76] Edwards G., Taylor C., Cootes T.: “Interpreting face images using active appearance models”, Proc. *International Conference on Face and Gesture Recognition*, pp. 300–305, 1998.
- [77] Elgammal A., Duraiswami R., Harwood D., Anddavis L.: “Background and foreground modeling using nonparametric kernel density estimation for visual surveillance”, *Proceedings of IEEE*, vol. 90, no. 7, pp. 1151–1163, 2002.
- [78] Ellwart D., Czyżewski A.: “Viewpoint independent shape-based object classification for video surveillance”, Proc. *12th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, 2011.
- [79] Farneback G.: “Two-frame motion estimation based on polynomial expansion”, Proc. *13th Scandinavian conference on Image analysis*, no. x, pp. 363–370, 2003.
- [80] Ferenbok J., Clement A.: “Hidden Changes: from CCTV to ‘Smart’ video surveillance”, *Eyes Everywhere: The Global Growth of Camera Surveillance*, pp. 235–253, 2011.
- [81] Fieguth P., Terzopoulos D.: “Color-based tracking of heads and other mobile objects at video frame rates”, Proc. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21–27, 1997.
- [82] Flusser J., Suk T.: “Pattern recognition by affine moment invariants”, *Pattern Recognition*, vol. 26, no. 1, pp. 167–174, Jan. 1993.
- [83] Freeman W., Adelson E.: “The design and use of steerable filters”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891–906, 1991.
- [84] Frejlichowski D.: “Application of Zernike Moments to the problem of General Shape Analysis”, *Control and Cybernetics*, vol. 40, no. 2, pp. 515–526, 2011.
- [85] Friedman J., Hastie T., Tibshirani R.: “Additive Logistic Regression: a Statistical View of Boosting”, *The Annals of Statistics*, vol. 28, no. 2, pp. 337–407, 2000.
- [86] Gao X., Boulton T., Coetzee F., Ramesh V.: “Error analysis of background adaption”, Proc. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 503–510, 2000.
- [87] Gavrilin D. M.: “The visual analysis of human movement: A survey”, *Comput. Vis. Image Underst.*, vol. 73, no. 1, pp. 82–98, 1999.
- [88] Van der Geer S.: *Empirical Processes in M-Estimation; Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, 2000, p. 298.

-
- [89] Gerrard G., Thompson R.: “How many CCTV cameras are there in the UK? Answer: 1.85 million”, *CCTV Image*, no. 42, 2011.
- [90] Gordon N., Salmond D., Smith A.: “A Novel Approach to Non-Linear and Non-Gaussian Bayesian State Estimation”, *Proceedings of IEEE*, vol. 140, pp. 107–113, 1993.
- [91] Graps A.: “An introduction to wavelets”, *IEEE Computational Science and Engineering*, vol. 2, no. 2, pp. 50–61, Jan. 1995.
- [92] Greenspan H., Belongie S., Goodman R., Perona P., Rakshit S., Anderson C.: “Overcomplete steerable pyramid filters and rotation invariance”, *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 222–228, 1994.
- [93] Grewe L., Kak A.: “Interactive learning of a multi-attribute hash table classifier for fast object recognition”, *Comput. Vision Image Understand.*, vol. 61, no. 3, pp. 387–416, 1995.
- [94] Grigorescu S. E., Petkov N., Kruizinga P.: “Comparison of texture features based on Gabor filters.”, *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, vol. 11, no. 10, pp. 1160–7, Jan. 2002.
- [95] Gupte S., Masoud O., Martin R. F. K., Papanikolopoulos N. P.: “Detection and classification of vehicles”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 37–47, Mar. 2002.
- [96] Gupte S., Masoud O., Papanikolopoulos P.: “Vision-based vehicle classification”, *Proc. IEEE Intelligent Transportation Systems*, no. Mi, pp. 46–51, 2000.
- [97] Hae-Kwang Kim, Jong-Deuk Kim, Dong-Gyu Sim, Dae-Il Oh: “A modified Zernike moment shape descriptor invariant to translation, rotation and scale for similarity-based image retrieval”, *Proc. IEEE International Conference on Multimedia and Expo*, vol. 1, pp. 307–310, 2000.
- [98] Haralick R. M., Shanmugam K., Dinstein I.: “Textural Features for Image Classification”, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, no. 6, pp. 610–621, Nov. 1973.
- [99] Harasse S., Bonnaud L., Desvignes M.: “Human model for people detection in dynamic scenes”, *Proc. Int. Conf. Pattern Recog.*, pp. 335–354, 2006.
- [100] Haritaoglu I., Harwod D., Davis L.: “W4: real-time surveillance of people and their activities”, *IEEE Trans. Patt. Analy. Mach. Intell.*, vol. 22, no. 8, pp. 809–830, 2000.
- [101] Harris C., Stephens M.: “A combined corner and edge detector”, *Proc. 4th Alvey Vision Conference*, pp. 147–151, 1988.
- [102] Hasegawa O., Kanade T.: “Type classification, color estimation, and specific target detection of moving targets on public streets”, *Machine Vision and Applications*, vol. 16, no. 2, pp. 116–121, Feb. 2005.
- [103] Hastie T., Tibshirani R., Friedman J.: *The Elements of Statistical Learning: Data Mining, Interference, and Prediction*, Second Edi. Springer Series in Statistics, 2009, p. 758.
- [104] Hausdorff F.: *Set Theory*. Chelsea, New York: , 1962.
- [105] Horn B., Schunk B.: “Determining optical flow”, *Artific. Intell.*, vol. 17, pp. 185–203, 1981.
- [106] Hota R. N., Venkoparao V., Rajagopal A.: “Shape based object classification for automated video surveillance with feature selection”, *Proc. IEEE Int. Conf. Inf. Technol.*, pp. 97–99, 2007.

- [107] Hou J.-S., Yang M.-T., Jhang R.-K.: “Traffic flow estimation and vehicle-type classification using vision-based spatial-temporal profile analysis”, *IET Computer Vision*, vol. 7, no. 5, pp. 394–404, Oct. 2013.
- [108] Hsieh J. W., Yu S., Chen Y. S., Hu W. F.: “Automatic Traffic Surveillance System for Vehicle Tracking and Classification”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 2, pp. 175–187, Jun. 2006.
- [109] Hsu C.-W., Chang C.-C., Lin C.-J.: “A practical guide to support vector classification”, Feb. 2003.
- [110] Hu M.-K.: “Visual pattern recognition by moment invariants”, *IEEE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, Feb. 1962.
- [111] Huang C. L., Liao W. C.: “A vision-based vehicle identification system”, Proc. *17th International Conference on Pattern Recognition*, vol. 4, pp. 364–367 Vol.4, 2004.
- [112] Huang T., Russell S.: “Object identification in a bayesian context”, Proc. *International Joint Conference on Artificial Intelligence*, pp. 1276–1283, 1997.
- [113] Huttenlocher D., Noh J., Rucklidge W.: “Tracking nonrigid objects in complex scenes”, Proc. *IEEE International Conference on Computer Vision (ICCV)*, pp. 93–101, 1993.
- [114] Iera A., Modafferi A., Musolino G., Vitetta A.: “An experimental station for real-time traffic monitoring on a urban road”, Proc. *IEEE International Conference on Intelligent Transportation Systems*, pp. 697–701, 2002.
- [115] Inigo R. M.: “Traffic Monitoring and Control Using Machine Vision: A Survey”, *IEEE Transactions on Industrial Electronics*, vol. IE-32, no. 3, pp. 177–185, Aug. 1985.
- [116] Irani M., Anandan P.: “Video indexing based on mosaic representations”, *IEEE Trans. Patt. Analy. Mach. Intell.*, vol. 20, no. 6, pp. 577–589, 1998.
- [117] Isard M., Blake A.: “Condensation - conditional density propagation for visual tracking”, *Int. J. Comput. Vision*, vol. 29, no. 1, pp. 5–28, 1998.
- [118] Isard M., MacCormick J.: “BraMBLe: a Bayesian multiple-blob tracker”, Proc. *8th IEEE International Conference on Computer Vision (ICCV)*, vol. 2, pp. 34–41, 2001.
- [119] Jain R., Nagel H.: “On the analysis of accumulative difference pictures from image sequences of real world scenes”, *IEEE Trans. Patt. Analy. Mach. Intell.*, vol. 1, no. 2, pp. 206–214, 1979.
- [120] Javed O., Shafique K., Rasheed Z., Shah M.: “Modeling inter-camera space-time and appearance relationships for tracking across non-overlapping views”, *Computer Vision and Image Understanding*, vol. 109, no. 2, pp. 146–162, Feb. 2008.
- [121] Jeong K., Jaynes C.: “Object matching in disjoint cameras using a color transfer approach”, *Machine Vision and Applications*, vol. 19, no. 5–6, pp. 443–455, May 2007.
- [122] Jepson A., Fleet D., Elmaraghi T.: “Robust online appearance models for visual tracking”, *IEEE Trans. Patt. Analy. Mach. Intell.*, vol. 25, no. 10, pp. 1296–1311, 2003.
- [123] Ji P., Jin L., Li X.: “Vision-based Vehicle Type Classification Using Partial Gabor Filter Bank”, Proc. *IEEE International Conference on Automation and Logistics*, no. 60275005, pp. 1037–1040, 2007.
- [124] Jones M. J., Snow D.: “Pedestrian detection using boosted features over many frames”, Proc. *Int. Conf. Pattern Recog.*, pp. 1–4, 2008.
- [125] Ju S.: “Human Motion Estimation and Recognition (Depth Oral Report)”, 1996.
- [126] Julier S., Uhlmann J.: “A New Extension of the Kalman Filter to Nonlinear Systems”, Proc. *SPIE*, vol. 3068, pp. 182–193, 1997.

- [127] Jung Y.-K., Lee K.-W., Ho Y.-S.: “Content-based event retrieval using semantic scene interpretation for automated traffic surveillance”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 2, no. 3, pp. 151–163, 2001.
- [128] Kaewtrakulpong P., Bowden R.: “An improved adaptive background mixture model for real-time tracking with shadow detection”, *Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems*, vol. 25, 2001.
- [129] Kałużny P.: *Telewizyjne systemy dozоровe*, Wydanie I. Warszawa: Wydawnictwa Komunikacji i Łączności, 2008.
- [130] Kamijo S., Matsushita Y., Ikeuchi K., Sakauchi M.: “Traffic monitoring and accident detection at intersections”, *Proc. IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems*, pp. 703–708, 1999.
- [131] Kanade T., Collins R., A. L., Burt P., Wixson L.: “Advances in cooperative multi-sensor video surveillance”, *Proc. Darpa IU Workshop*, pp. 3–24, 1998.
- [132] Kang J., Cohen I., Medioni G.: “Object reacquisition using geometric invariant appearance model”, *Proc. International Conference on Pattern Recognition (ICPR)*, pp. 759–762, 2004.
- [133] Kanistras K., Martins G., Rutherford M. J., Valavanis K. P.: “A survey of unmanned aerial vehicles (UAVs) for traffic monitoring”. pp. 221–234, 2013.
- [134] Kanungo T., Mount D., Netanyahu N., Piatko C. D., Silverman R., Wu A. Y.: “An efficient k-means clustering algorithm: analysis and implementation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, Jul. 2002.
- [135] Kass M., Witkin A., Terzopoulos D.: “Snakes: active contour models”, *International Journal of Computer Vision*, vol. 1, pp. 321–332, 1988.
- [136] Katsaggelos A. K.: “Abnormal event detection based on trajectory clustering by 2-depth greedy search”, *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2129–2132, 2008.
- [137] Kettner V., Brand M.: “Minimum-entropy models of scene activity”, *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 281–286, 1999.
- [138] Kettner V., Zabih R.: “Bayesian multi-camera surveillance”, *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 117–123, 1999.
- [139] Khan S., Shah M.: “Consistent labeling of tracked objects in multiple cameras with overlapping fields of view”, *IEEE Trans. Patt. Analy. Mach. Intell.*, vol. 25, no. 10, pp. 1355–1360, 2003.
- [140] Khare M., Nigam S., Srivastava R. K., Khare A.: “Contourlet transform based moving object segmentation”, *Proc. IEEE Conference on Information and Communication Technologies*, pp. 782–787, 2013.
- [141] Kim K., Chalidabhongse T. H., Harwood D., Davis L.: “Real-time foreground-background segmentation using codebook model”, *Real-time imaging*, vol. 11, no. 3, pp. 177–185, 2005.
- [142] Kitagawa G.: “Non-Gaussian State-Space Modeling of Nonstationary Time Series”, *J. Am. Statistical Assoc.*, vol. 82, pp. 1032–1063, 1987.
- [143] Klein L. A., Mills M. K., Gibson D. R.: *Traffic Detector Handbook*, Third Edit. U.S. Department of Transportation, Federal Highway Administration, No. FHWA-HRT-06-108, 2006.
- [144] Kopaczewski K., Szczodrak M., Czyzewski A., Krawczyk H.: “A method for counting people attending large public events”, *Multimedia Tools and Applications*, pp. 1–13, Aug. 2013.

- [145] Kopf S., Haenselmann T., Effelsberg W.: “Enhancing Curvature Scale Space Features for Robust Shape Classification”, Proc. *IEEE International Conference on Multimedia and Expo*, pp. 478–481, 2005.
- [146] Kotsiantis S. B.: “Supervised Machine Learning: A Review of Classification Techniques”, *Informatica*, vol. 31, no. 3, pp. 249–268, 2007.
- [147] Koutsia A., Semertzidis T., Dimitropoulos K., Grammalidis N., Georgouleas K.: “Intelligent traffic monitoring and surveillance with multiple cameras”, Proc. *International Workshop on Content-Based Multimedia Indexing*, pp. 125–132, 2008.
- [148] Kuhn H.: “The hungarian method for solving the assignment problem”, *Naval Research Logistics Quart.*, vol. 2, pp. 83–97, 1955.
- [149] Kumar P., Ranganath S., Weimin H., Sengupta K.: “Framework for Real-Time Behavior Interpretation From Traffic Video”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 1, pp. 43–53, Mar. 2005.
- [150] Kun A. J., Vamossy Z.: “Traffic monitoring with computer vision”, Proc. *International Symposium on Applied Machine Intelligence and Informatics*, pp. 131–134, 2009.
- [151] Kunfeng Wang, Zhenjiang Li, Qingming Yao, Wuling Huang, Fei-Yue Wang: “An automated vehicle counting system for traffic surveillance”, Proc. *IEEE International Conference on Vehicular Electronics and Safety*, pp. 1–6, 2007.
- [152] Lai A. H. S., Fung G. S. K., Yung N. H. C.: “Vehicle type classification from visual-based dimension estimation”, Proc. *IEEE Intelligent Transportation Systems*, pp. 201–206, 2001.
- [153] Lee L., Romano R., Stein G.: “Monitoring activities from multiple video streams: Establishing a common coordinate frame”, *IEEE Trans. Patt. Recogn. Mach. Intell.*, vol. 22, no. 8, pp. 758–768, 2000.
- [154] Lei M., Lefloch D., Gouton P., Madani K.: “A Video-Based Real-Time Vehicle Counting System Using Adaptive Background Method”, Proc. *IEEE International Conference on Signal Image Technology and Internet Based Systems*, pp. 523–528, 2008.
- [155] Li B., Chellappa R., Zheng Q., Der S.: “Model-based temporal object verification using video”, *IEEE Trans. Image Process.*, vol. 10, no. 6, pp. 897–908, 2001.
- [156] Li X., Fu H., Xu J.: “Automatic Vehicle Classification Based on Video with BP Neural Networks”, Proc. *4th International Conference on Wireless Communications, Networking and Mobile Computing*, no. 2, pp. 1–3, 2008.
- [157] Li Y.-B., Jiang T.-X., Qiao Z.-H., Qian H.-J.: “General methods and development actuality of gait recognition”, Proc. *IEEE Int. Conf. Wavelet Anal. Pattern Recog.*, pp. 1333–1340, 2007.
- [158] Lipton A. J., Fujiyoshi H., Patil R. S.: “Moving target classification and tracking from real-time video”, Proc. *4th IEEE Workshop on Applications of Computer Vision*, pp. 8–14, 1998.
- [159] Litzemberger M. et al.: “Vehicle Counting with an Embedded Traffic Data System using an Optical Transient Sensor”, Proc. *IEEE Intelligent Transportation Systems Conference*, pp. 36–40, 2007.
- [160] Lowe D. G.: “Distinctive Image Features from Scale-Invariant Keypoints”, *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
- [161] Lucas B. D., Kanade T.: “An iterative image registration technique with an application to stereo vision”, Proc. *International Joint Conference on Artificial Intelligence*, 1981.
- [162] Ludwig S., Beda J., Saint-Andre P., McQueen R., Egan S., Hildebrand J.: “Jingle, XEP-0166 (Draft)”. 2009.

- [163] Ma X., Grimson W. E. L.: "Edge-based rich representation for vehicle classification", Proc. *10th IEEE International Conference on Computer Vision (ICCV)*, vol. 2, pp. 1185–1192 Vol. 2, 2005.
- [164] Mackay D. J. C.: "Introduction to Monte Carlo methods", *Learning in Graphical Models*, Jordan M. I. (Ed.). Kluwer Academic Press, pp. 175–204, 1998.
- [165] Mallat S.: "A theory for multiresolution signal decomposition: The wavelet representation", *IEEE Trans. Patt. Analy. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, 1989.
- [166] Mansouri A.: "Region tracking via level set PDEs without motion computation", *IEEE Trans. Patt. Analy. Mach. Intell.*, vol. 24, no. 7, pp. 947–961, 2002.
- [167] Mariano V. Y. et al.: "Performance evaluation of object detection algorithms", Proc. *16th International Conference on Pattern Recognition*, vol. 3, pp. 965–969, 2002.
- [168] Marszałek Z., Sroka R., Stencel M.: "A new method of inductive sensor impedance measurement applied to the identification of vehicle parameters", *Metrology and Measurement Systems*, vol. 18, no. 1, pp. 69–76, 2011.
- [169] Martínez J. M.: "MPEG-7 Overview (version 10) ISO/IEC JTC1/SC29/WG11N6828". 2004.
- [170] McCall J. C., Trivedi M. M.: "Video-Based Lane Estimation and Tracking for Driver Assistance: Survey, System, and Evaluation", *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 20–37, Mar. 2006.
- [171] Mei Yu, Gangyi Jiang, Bokang Yu: "An integrative method for video based traffic parameter extraction in ITS", Proc. *IEEE Asia-Pacific Conference on Circuits and Systems. Electronic Communication Systems*, pp. 136–139, 2000.
- [172] Merwe R., Doucet A., Freitas N., Wan E.: "The Unscented Particle Filter", 2000.
- [173] Mikolajczyk K., Schmid C.: "Performance evaluation of local descriptors.", *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 10, pp. 1615–30, Oct. 2005.
- [174] Milan A., Roth S., Schindler K.: "Continuous energy minimization for multitarget tracking.", *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 1, pp. 58–72, Jan. 2014.
- [175] Monnet A., Mittal A., Paragios N., Ramesh V.: "Background modeling and subtraction of dynamic scenes.", Proc. *IEEE International Conference on Computer Vision (ICCV)*, pp. 1305–1312, 2003.
- [176] Moravec H.: "Visual mapping by a robot rover", Proc. *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 598–600, 1979.
- [177] Morris B., Trivedi M.: "Improved Vehicle Classification in Long Traffic Video by Cooperating Tracker and Classifier Modules", Proc. *IEEE International Conference on Video and Signal Based Surveillance*, 2006.
- [178] Morris B., Trivedi M.: "Robust classification and tracking of vehicles in traffic video streams", Proc. *IEEE Intelligent Transportation Systems Conference*, pp. 1078–1083, 2006.
- [179] Mughadam B., Pentland A.: "Probabilistic visual learning for object representation", *IEEE Trans. Patt. Analy. Mach. Intell.*, vol. 19, no. 7, pp. 696–710, 1997.
- [180] Myers J.: "Simple Authentication and Security Layer (SASL), RFC 2222". 1997.
- [181] Negri P., Clady X., Milgram M., Poulénard R.: "An Oriented-Contour Point Based Voting Algorithm for Vehicle Type Classification", Proc. *18th International Conference on Pattern Recognition (ICPR)*, pp. 574–577, 2006.
- [182] Oliver N., Rosario B., Pentland A.: "A bayesian computer vision system for modeling human interactions", *IEEE Trans. Patt. Analy. Mach. Intell.*, vol. 22, no. 8, pp. 831–843, 2000.

- [183] Otsuka K., Mukawa N.: “Multiview occlusion analysis for tracking densely populated objects based on 2-D visual angles”, Proc. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 90–97, 2004.
- [184] Pang C. C. C., Lam W. W. L., Yung N. H. C.: “A Method for Vehicle Count in the Presence of Multiple-Vehicle Occlusions in Traffic Images”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 3, pp. 441–459, Sep. 2007.
- [185] Papageorgiu C., Oren M., Poggio T.: “A general framework for object detection”, Proc. *IEEE International Conference on Computer Vision (ICCV)*, pp. 555–562, 1998.
- [186] Paragios N., Deriche R.: “Geodesic active contours and level sets for the detection and tracking of moving objects”, *IEEE Trans. Patt. Analy. Mach. Intell.*, vol. 22, no. 3, pp. 266–280, 2000.
- [187] Paragios N., Deriche R.: “Geodesic active regions and level set methods for supervised texture segmentation”, *Int. J. Comput. Vision*, vol. 46, no. 3, pp. 223–247, 2002.
- [188] Paschos G.: “Perceptually uniform color spaces for color texture analysis: an empirical evaluation”, *IEEE Trans. Image Process.*, vol. 10, pp. 932–937, 2001.
- [189] Petrovic V., Cootes T.: “Analysis of features for rigid structure vehicle type recognition”, Proc. *British Machine Vision Conference*, vol. 2, pp. 587–596, 2004.
- [190] Porikli F. M.: “A hidden markov model framework for traffic event detection using video features”, Proc. *International Conference on Image Processing*, vol. 5, pp. 2901–2904, 2004.
- [191] Qin Z.: “Method of vehicle classification based on video”, Proc. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 162–164, 2008.
- [192] Rasmussen C., Hager G. D.: “Probabilistic data association methods for tracking complex visual objects”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 560–576, Jun. 2001.
- [193] Reid D.: “An Algorithm for Tracking Multiple Targets”, *IEEE Trans. Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.
- [194] Reuter C.: *The V2 and the German, Russian and American Rocket Program*. New York: S. R. Research and Publishing, 2000.
- [195] Riedmiller M., Braun H.: “A direct adaptive method for faster backpropagation learning: the RPROP algorithm”, Proc. *IEEE International Conference on Neural Networks*, pp. 586–591, 1993.
- [196] Rittscher J., Kato J., Joga S., Blake A.: “A probabilistic background model for tracking”, Proc. *European Conference on Computer Vision (ECCV)*, pp. 336–350, 2000.
- [197] Ronfard R.: “Region based strategies for active contour models”, *Int. J. Comput. Vision*, vol. 13, no. 2, pp. 229–251, 1994.
- [198] Rowe S., Blake A.: “Statistical mosaics for tracking”, *Israel Verj. Cap. J.*, vol. 14, pp. 549–564, 1996.
- [199] Rowley H., S. B., Andkanade T.: “Neural network-based face detection”, *IEEE Trans. Patt. Analy. Mach. Intell.*, vol. 20, no. 1, pp. 23–38, 1998.
- [200] Roy A., Gale N., Hong L.: “Fusion of Doppler Radar and video information for automated traffic surveillance”. pp. 1989–1996, 2009.
- [201] Ryoo M. S., Riley M., Aggarwal J. K.: “Real-time detection of illegally parked vehicles using 1-D transformation”, Proc. *IEEE Conference on Advanced Video and Signal Based Surveillance*, pp. 254–259, 2007.
- [202] Saint-Andre P.: “Extensible Messaging and Presence Protocol (XMPP): Core, RFC 3920”. 2004.

- [203] Saint-Andre P.: “Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence, RFC 3921”. 2004.
- [204] Saito M., Kitaguchi K.: “Appearance Modeling for Object Pose Recognition using Canonical Correlation Analysis”, Proc. *SICE-ICASE International Joint Conference*, pp. 2818–2821, 2006.
- [205] Salari V., Sethi I.: “Feature point correspondence in the presence of occlusion”, *IEEE Trans. Patt. Analy. Mach. Intell.*, vol. 12, no. 1, pp. 87–91, 1990.
- [206] Sandhawalia H., Rodriguez-Serrano J. A., Poirier H., Csurka G.: “Vehicle type classification from laser scanner profiles: A benchmark of feature descriptors”, Proc. *16th International IEEE Conference on Intelligent Transportation Systems*, pp. 517–522, 2013.
- [207] Sato K., Aggarwal J.: “Temporal spatio-velocity transform and its application to tracking and interaction.”, *Comput. Vision Image Understand*, vol. 96, no. 2, pp. 100–128, 2004.
- [208] Satpathy A., Jiang X., Eng H.-L.: “Visual object detection by parts-based modeling using extended histogram of gradients”, Proc. *IEEE International Conference on Image Processing*, pp. 2738–2742, 2013.
- [209] Schulzrinne H., Casner S., Frederick R., Jacobson V.: “RTP: A Transport Protocol for Real-Time Applications, RFC 3550”. 2003.
- [210] Schweitzer H., Bell J. W., Wu F.: “Very fast template matching”, Proc. *European Conference on Computer Vision (ECCV)*, pp. 358–372, 2002.
- [211] Setchell C., Dagless E. L.: “Vision-based road-traffic monitoring sensor”, *IEE Proceedings - Vision, Image, and Signal Processing*, vol. 148, no. 1, p. 78, 2001.
- [212] Sethi I., Jain R.: “Finding trajectories of feature points in a monocular image sequence”, *IEEE Trans. Patt. Analy. Mach. Intell.*, vol. 9, no. 1, pp. 56–73, 1987.
- [213] Shafique K., Shah M.: “A non-iterative greedy algorithm for multi-frame point correspondence”, Proc. *IEEE International Conference on Computer Vision (ICCV)*, pp. 110–115, 2003.
- [214] Shaogang Gong, Tao Xiang: “Recognition of group activities using dynamic probabilistic networks”, Proc. *IEEE International Conference on Computer Vision*, pp. 742–749 vol.2, 2003.
- [215] Shi J., Malik J.: “Normalized cuts and image segmentation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.
- [216] Shi J., Tomasi C.: “Good features to track”, Proc. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600, 1994.
- [217] Stauffer C., Grimson W. E. L.: “Learning patterns of activity using real-time tracking”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000.
- [218] Stegmann M. B., Ersbřll B. K., Larsen R.: “FAME - A Flexible Appearance Modelling Environment”, *IEEE Transactions on Medical Imaging*, vol. 22, no. 10, pp. 1319–133, 2003.
- [219] Stenger B., Ramesh V., Paragios N., Coetzee F., Buhmann J.: “Topology free hidden markov models: Application to background modeling”, Proc. *IEEE International Conference on Computer Vision (ICCV)*, pp. 294–301, 2001.
- [220] Streit R. L., Luginbuhl T. E.: “Maximum likelihood method for probabilistic multi-hypothesis tracking”, *Proceedings of the International Society for Optical Engineering (SPIE)*, vol. 2235, pp. 394–405, 1994.
- [221] Sullivan G. D., Baker K. D., Worrall A. D., Attwood C. I., Remagnino P. M.: “Model-based vehicle detection and classification using orthographic

- approximations”, *Image and Vision Computing*, vol. 15, no. 8, pp. 649–654, Aug. 1997.
- [222] Sullivan G., Worrall A., Ferryman J.: “Visual object recognition using deformable models of vehicles”, Proc. *Workshop on Context-Based Vision*, vol. 19, pp. 75–86, 1995.
- [223] Sun Z., Bebis G., Miller R.: “On-road vehicle detection: a review.”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 694–711, May 2006.
- [224] Suo P., Wang Y.: “An improved adaptive background modeling algorithm based on Gaussian Mixture Model”, Proc. *9th International Conference on Signal Processing*, no. 60873163, pp. 1436–1439, 2008.
- [225] Szczodrak M., Czyżewski A.: “Video Analytics-Based Algorithm for Monitoring Egress from Buildings”, *Multimedia Communications, Services and Security*, Dziech A., Czyżewski A. (Ed.). Springer Berlin Heidelberg, pp. 224–232, 2013.
- [226] Szczodrak M., Czyżewski A., Kotus J.: “Investigation of the road noise source employing an automatic noise monitoring station”, *Archives of Acoustics*, vol. 33, no. 4, pp. 77–83, 2008.
- [227] Szczodrak M., Dalka P., Czyżewski A.: “Moving object tracking algorithm evaluation in autonomous surveillance system”, Proc. *IEEE International Conference on Multimedia Communications, Services and Security*, pp. 219–223, 2010.
- [228] Szeliski R., Coughlan J.: “Spline-based image registration”, *Int. J. Comput. Vision*, vol. 16, pp. 1–3, 1997.
- [229] Szwoch G., Dalka P.: “Automatic detection of abandoned luggage employing a dual camera system”, Proc. *IEEE International Conference on Multimedia Communications, Services and Security*, pp. 56–61, 2010.
- [230] Szwoch G., Dalka P.: “Identification of regions of interest in video for a traffic monitoring system”, Proc. *1st International Conference on Information Technology*, pp. 1–4, 2008.
- [231] Szwoch G., Dalka P.: “Layered background modeling for automatic detection of unattended objects in camera images”, Proc. *International Workshop on Image Analysis for Multimedia Interactive Services*, pp. 1–4, 2011.
- [232] Szwoch G., Dalka P., Ciarkowski A., Szczuko P., Czyżewski A.: “Visual object tracking system employing fixed and PTZ cameras”, *Intelligent Decision Technologies*, vol. 5, no. 2, pp. 177–188, 2011.
- [233] Szwoch G., Dalka P., Czyżewski A.: “Estimation of object size in the calibrated camera image”, *Elektronika: konstrukcje, technologie, zastosowania*, vol. 50, no. 3, pp. 10–14, 2009.
- [234] Szwoch G., Dalka P., Czyżewski A.: “Objects classification based on their physical sizes for detection of events in camera images”, Proc. *Signal Processing Algorithms, Architectures, Arrangements, and Applications (SPA)*, pp. 15–20, 2008.
- [235] Szwoch G., Dalka P., Czyżewski A.: “Resolving conflicts in object tracking for automatic detection of events in video”, *Elektronika: konstrukcje, technologie, zastosowania*, vol. 52, no. 1, pp. 52–54, 2011.
- [236] Szwoch G., Dalka P., Czyżewski A.: “Spatial Calibration of a Dual PTZ-Fixed Camera System for Tracking Moving Objects in Video”, *Journal of Imaging Science and Technology*, vol. 57, no. 2, pp. 1–10, Mar. 2013.
- [237] Talha M., Stolkin R.: “Particle Filter Tracking of Camouflaged Targets by Adaptive Fusion of Thermal and Visible Spectra Camera Data”, *IEEE Sensors Journal*, vol. 14, no. 1, pp. 159–166, Jan. 2014.

- [238] Tan T. N., Sullivan G. D., Baker K. D.: “Recognising Objects on the Ground-Plane”, Proc. *British Machine Vision Conference*, 1993.
- [239] Tao H., Sawhney H., Kumar R.: “Object tracking with bayesian estimation of dynamic layer representations”, *IEEE Trans. Patt. Analy. Mach. Intell.*, vol. 24, no. 1, pp. 75–89, 2002.
- [240] Terzopoulos D., Szeliski R.: “Tracking with Kalman snakes”, *Active Vision*, Blake A., Yuille A. (Ed.). MIT Press, 1992.
- [241] Tian B., Yao Q., Gu Y., Wang K., Li Y.: “Video processing techniques for traffic flow monitoring: A survey”, Proc. *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 1103–1108, 2011.
- [242] Toyama K., Krumm J., Brumitt B., Meyers B.: “Wallflower: Principles and practices of background maintenance”, Proc. *IEEE International Conference on Computer Vision (ICCV)*, pp. 255–261, 1999.
- [243] Tsai R.: “A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses”, *IEEE Journal on Robotics and Automation*, vol. 3, no. 4, pp. 323–344, Aug. 1987.
- [244] Tsuchiya M., Fujiyoshi H.: “Evaluating Feature Importance for Object Classification in Visual Surveillance”, Proc. *International Conference on Pattern Recognition*, vol. 2, pp. 978–981, 2006.
- [245] Veenman C., Reinders M., Backer E.: “Resolving motion correspondence for densely moving points”, *IEEE Trans. Patt. Analy. Mach. Intell.*, vol. 23, no. 1, pp. 54–72, 2001.
- [246] Veeraraghavan H., Schrater P., Papanikolopoulos N.: “Switching Kalman Filter-Based Approach for Tracking and Event Detection at Traffic Intersections”, Proc. *Mediterranean Conference on Control and Automation Intelligent Control*, pp. 1167–1172, 2005.
- [247] Viola P., Jones M.: “Rapid object detection using a boosted cascade of simple features”, Proc. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 00, no. C, pp. I–511–I–518, 2001.
- [248] Viola P., Jones M. J., Snow D.: “Detecting pedestrians using patterns of motion and appearance”, Proc. *IEEE International Conference on Computer Vision*, pp. 734–741, 2003.
- [249] Viterbi A. J.: “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm”, *IEEE Trans. Inform. Theory*, vol. 13, pp. 260–269, 1967.
- [250] Wang J., Dong L.: “Moving objects detection method based on a fast convergence Gaussian mixture model”, Proc. *3rd International Conference on Computer Research and Development*, no. 3, pp. 269–273, 2011.
- [251] Wang Q., Liu J.: “The Improved Particle Filter for Object Tracking”, Proc. *World Congress on Intelligent Control and Automation*, pp. 10275–10279, 2006.
- [252] Welch G., Bishop G.: “An introduction to the Kalman filter”, 2006.
- [253] Worrall A., Sullivan G., Baker K.: “A simple, intuitive camera calibration tool for natural images”, Proc. *British Conference on Machine Vision*, pp. 781–790, 1994.
- [254] Wren C., A. A., Pentland A.: “Pfinder: Real-time tracking of the human body”, *IEEE Trans. Patt. Analy. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, 1997.
- [255] Wu W., QiSen Z., Mingjun W.: “A method of vehicle classification using models and neural networks”, Proc. *53rd Vehicular Technology Conference*, vol. 4, pp. 3022–3026, 2001.
- [256] Wu Z., Leahy R.: “An optimal graph theoretic approach to data clustering: Theory and its applications to image segmentation”, *IEEE Trans. Patt. Analy. Mach. Intell.*, vol. 11, pp. 1101–1113, 1993.

- [257] Xu L.-Q., Landabaso J. L., Pardàs M.: “Shadow Removal with Blob-Based Morphological Reconstruction for Error Correction”, *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2, pp. 729–732, 2005.
- [258] Yilmaz A., Javed O., Shah M.: “Object tracking”, *ACM Computing Surveys*, vol. 38, no. 4, Dec. 2006.
- [259] Yilmaz A., Li X., Shah M.: “Contour based object tracking with occlusion handling in video acquired using mobile cameras”, *IEEE Trans. Patt. Analy. Mach. Intell.*, vol. 26, no. 11, pp. 1531–1536, 2004.
- [260] Yu H., Wang Y., Kuang F., Wan Q.: “Multi-moving targets detecting and tracking in a surveillance system”, *Proc. 5th World Congress on Intelligent Control and Automation*, vol. 6, pp. 5253–5257, 2004.
- [261] Zhang B.: “Reliable Classification of Vehicle Types Based on Cascade Classifier Ensembles”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 322–332, Mar. 2013.
- [262] Zhang T., Liu S., Xu C., Lu H.: “Mining Semantic Context Information for Intelligent Video Surveillance of Traffic Scenes”, *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 149–160, Feb. 2013.
- [263] Zhong J., Sclaroff S.: “Segmenting foreground objects from a dynamic textured background via a robust kalman filter”, *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 44–50, 2003.
- [264] Zhu S., Yuille A.: “Region competition: unifying snakes, region growing, and bayes/mdl for multi-band image segmentation”, *IEEE Trans. Patt. Analy. Mach. Intell.*, vol. 18, no. 9, pp. 884–900, 1996.
- [265] “Addpriv Project”. <http://www.addpriv.eu>. [widziany: 2013.07.28].
- [266] “Boost C++ Libraries”. <http://www.boost.org/>. [widziany: 2013.03.26].
- [267] “CGAL - Computational Geometry Algorithms Library”. <http://www.cgal.org/>. [widziany: 2013.03.26].
- [268] “FFmpeg - complete, cross-platform solution to record, convert and stream audio and video”. <http://www.ffmpeg.org/>. [widziany: 2013.03.26].
- [269] “Gdynia w oczach kamer”, 2013. http://www.gdynia.pl/dla/mieszkanow/bezpieczna/gdynia/7726_82731.html. [widziany: 2013.07.28].
- [270] “Indect Project”. <http://www.indect-project.eu>. [widziany: 2013.07.28].
- [271] “ITU-R Recommendation BT.601 : Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios”. <http://www.itu.int/rec/R-REC-BT.601/en>. [widziany: 2012.04.30].
- [272] “Monitoring w polskich miastach i w oczach społeczeństwa”, *Fundacja Panoptykon*, 2012. <http://panoptykon.org/biblioteka/monitoring-w-polskich-miastach-i-w-oczach-spolecenstwa>. [widziany: 2013.07.28].
- [273] “Ninth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2006)”, *Proc. Ninth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, no. Pets, 2006.
- [274] “Open Network Video Interface Forum Inc. (ONVIF): ONVIF™ Core Specification Version 2.2.1 (Dec 2012)”. <http://www.onvif.org/specs/core/ONVIF-Core-Specification-v221.pdf>. [widziany: 2013.03.26].
- [275] “OpenCV - Open Source Computer Vision”. <http://opencv.org/>. [widziany: 2013.03.26].
- [276] “OpenRTSP - libraries for standards-based RTP/RTCP/RTSP/SIP multimedia streaming”. <http://www.live555.com/>. [widziany: 2013.03.26].

-
- [277] “OpenSSL - Cryptography and SSL/TLS Toolkit”. <http://www.openssl.org/>.
[widziany: 2013.03.26].

10. Załączniki

10.1 Infrastruktura badawcza – rozproszony system monitoringu

Niniejszy rozdział przedstawia opis rozwiązania sprzętowo-programowego, które powstało w związku z potrzebami wynikającymi z prac badawczo rozwojowych, prowadzonych w ramach udziału Katedry Systemów Multimedialnych w projektach europejskich 7. Programu Ramowego INDECT [270] i ADDPRIV [265]. Udział autora rozprawy w tworzeniu tego rozwiązania dotyczył przede wszystkim stworzenia platformy programistycznej, umożliwiającej analizę wielu strumieni wizyjnych jednocześnie w sposób elastyczny i skalowalny (rozdziale 7). Celem tego załącznika jest przedstawianie kontekstu działania opracowanych algorytmów analizy obrazu i pokazanie kompleksowego rozwiązania, w ramach którego są one wykorzystywane i w którego tworzenie zaangażowany był autor rozprawy.

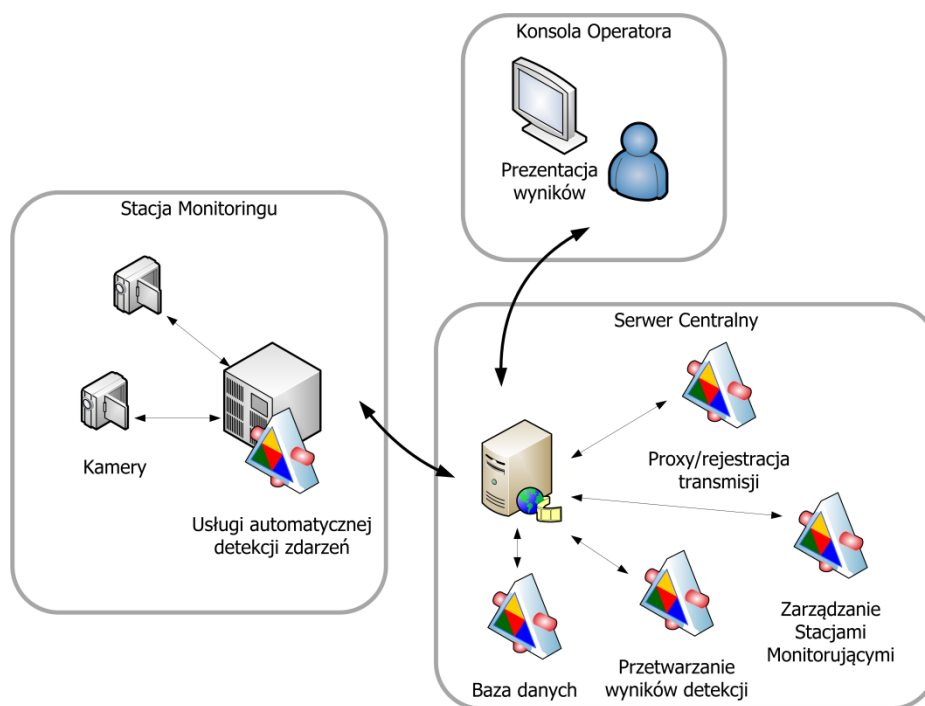
Stworzony w Katedrze Systemów Multimedialnych system charakteryzuje się następującymi cechami:

- zastosowanie uniwersalnych i otwartych standardów komunikacji, transmisji i przetwarzania danych wizyjnych,
- ujednoczona platforma programistyczna do tworzenia modułów analizy obrazu,
- równoległe przetwarzanie obrazu przez różne moduły,
- rozproszona architektura (rozdzielanie zadań systemu na wiele maszyn fizycznych),
- uniwersalna warstwa komunikacyjna, zapewniająca niezawodne połączenia między elementami systemu,
- modułowa budowa, ułatwiająca łatwe rozszerzanie funkcjonalności,
- możliwość pracy wieloplatformowej,
- bezpieczeństwo komunikacji.

10.1.1 Architektura systemu

Stworzony system ma architekturę rozproszoną, zarówno w sensie logicznym, w kontekście rozproszenia usług i funkcjonalności na wielu maszynach, jak i w sensie fizycznym, gdyż elementy systemu mogą być rozmieszczone w praktycznie dowolnych miejscach na świecie [57].

Z funkcjonalnego punktu widzenia w systemie monitoringu można wyróżnić trzy rodzaje elementów: Stacje Monitorujące, Serwer Centralny oraz Konsolę Operatora (rys. 10.1). W systemie może istnieć dowolna liczba Stacji i Konsol. Elementy systemu komunikują się ze sobą z wykorzystaniem dowolnego, dostępnego medium komunikacyjnego (LAN, Wi-Fi, GPRS, UMTS itd.).



Rys. 10.1 Funkcjonalne komponenty systemu monitoringu opracowanego w Katedrze Systemów Multimedialnych

10.1.2 Charakterystyka elementów składowych systemu

Stacja Monitorująca jest elementem dokonującym przetwarzania obrazu. Odbiera ona strumienie wizyjne z podłączonych do niej kamer, analizuje je, a wyniki przesyła do Serwera Centralnego. Realizuje tym samym koncepcję przetwarzania obrazu „na miejscu”, tzn. jak najbliżej źródła sygnału. Pozwala to na elastyczne zarządzanie mocą obliczeniową (jeden komputer przypada na kilka znajdujących się w pobliżu kamer, w przypadku braku mocy obliczeniowej można dostawić kolejny), znosi bariery geograficzne (stacje monitorujące mogą być rozmieszczone praktycznie w dowolnym miejscu na świecie) i uniezależnia działanie systemu od jakości medium transmisyjnego (brak konieczności przesyłania strumienia wizyjnego o wysokiej jakości na duże odległości na potrzeby analizy – transmitowane są tylko wyniki lub strumienie wizyjne na żądanie, które mogą charakteryzować się większym stopniem kompresji lub być transmitowane w trybie

offline). Strumienie wizyjne są archiwizowane na Stacjach Monitorujących; przy przekroczeniu przydzielonego miejsca na nośniku danych, najstarsze nagrania są usuwane automatycznie.

Zadaniem Serwera Centralnego jest zarządzanie całością systemu, odbieranie wyników ze Stacji Monitorujących, zachowywanie ich w bazie danych oraz rozsyłanie wiadomości o wykrytych zdarzeniach do odpowiednich Konsol Operatorów wg zadanych kryteriów. Przechowuje on również dane multimedialne (obrazy, nagrania wizyjne) związane z wykrytymi zdarzeniami w celu zabezpieczenia ich przed automatycznym usunięciem przez oprogramowanie Stacji Monitorujących.

Aplikacja Konsoli Operatora stanowi interfejs graficzny systemu, umożliwiający obserwację wyników działania systemu monitoringu (rys. 10.2). Pozwala na odbieranie strumieni wizyjnych z dowolnych kamer podłączonych do Stacji Monitorujących systemu (na żywo lub z archiwum znajdującego się na stacjach) i ich elastyczną aranżację na ekranie, przy użyciu od jednego do czterech monitorów.



Rys. 10.2 Aplikacja Konsoli Operatora powstała w Katedrze Systemów Multimedialnych na potrzeby wizualizacji wyników działania systemu monitoringu

Razem ze strumieniem wizyjnym odbierane są szczegółowe wyniki działania poszczególnych algorytmów w postaci metadanych, które są nakładane na ramki obrazu w celu ich wizualizacji. Użytkownik może wybrać rodzaj i sposób prezentacji danych. Może on dotyczyć jedynie informacji o wykrytych zdarzeniach wraz z zaznaczeniem obiektu, który je wywołał lub uwzględniać wyniki niskopoziomowej analizy obrazu, takie jak maski oznaczające ruchome obiekty, położenie wszystkich śledzonych obiektów w postaci prostokątów, obszary detekcji zdarzeń, rezultaty klasyfikacji obiektów itd. Dodatkowo Konsola prezentuje listę ostatnio wykrytych zdarzeń wraz z ich statystyką oraz spis wszystkich obiektów poruszających się aktualnie się po monitorowanym obszarze. W ten

sposób konsola operatora stanowi zarówno wygodne narzędzie do oceny skuteczności działania algorytmów jak i interfejs dla operatora systemu monitoringu.

10.1.3 Warstwa komunikacyjna

Konsekwencją przyjęcia rozproszonej architektury systemu jest możliwość znacznego, fizycznego oddalenia od siebie elementów systemu (Stacji Monitorujących, Serwera Centralnego, Konsol Operatorów), które uniemożliwia zastosowanie połączenia lokalnego (LAN) do komunikacji pomiędzy elementami systemu. Przyjęto wobec tego założenie, że komunikacja w systemie może się odbywać za pośrednictwem sieci publicznej, w tym również sieci Internet, poprzez zarówno przewodowe, jak i bezprzewodowe kanały dostępowe (sieci 3G, punkty dostępowe WiFi). W istotny sposób wpływa to na wymogi dotyczące bezpieczeństwa proponowanego systemu, zwłaszcza, że informacje przekazywane w systemie monitoringu częstokroć mają status danych wrażliwych. W konsekwencji wymogiem staje się, aby ustanowiony kanał komunikacyjny był szyfrowany, zaś dostęp do elementów systemu niemożliwy bez właściwej autoryzacji. Ponadto należy uwzględnić fakt, iż wspomniane kanały dostępu do Internetu wykorzystują zwykle urządzenia pośredniczące w transmisji, takie jak translatory adresu (NAT, ang. *Network Address Translator*) lub zapory sieciowe (ang. *Firewall*), istotną kwestią staje się więc możliwość dostępu do elementów systemu znajdujących się w takiej konfiguracji.

Analiza postawionych powyżej wymagań pozwoliła na zastosowanie do realizacji systemu komunikacji protokołu XMPP (ang. *Extensible Messaging and Presence Protocol*), oryginalnie zaprojektowanego i powszechnie stosowanego do budowy systemów typu Instant Messaging [57][202][203]. Systemy oparte o protokół XMPP cechują się architekturą typu klient-serwer, zaś stroną inicjującą połączenie jest zawsze klient, więc jeżeli serwer posiada publiczny i dostępny z zewnątrz adres IP, to rozwiązywany jest problem ewentualnych urządzeń pośredniczących, np. translatorów adresów NAT, które utrudniłyby inicjację przez serwer takiego połączenia.

Każdy agent XMPP połączony jest z serwerem XMPP. Agentami takimi są elementy systemu (Stacje Monitorujące, Serwer Centralny, Konsola Operatora), które posiadają unikatowy identyfikator, umożliwiający jednoznaczne adresowanie przekazywanych komunikatów, w sposób analogiczny do wiadomości e-mail (nazwa_agenta@adres.serwera.xmpp). Należy zwrócić uwagę, że w kontekście omawianego systemu monitoringu protokół XMPP dostarcza warstwy transportowej, a zarazem stanowi platformę służącą do uruchamiania usług realizujących właściwą

funkcjonalność związaną z monitoringiem (Stacje Monitorujące, Serwer Centralny, Konsola Operatora).

W warstwie transportowej wykorzystywany jest mechanizm TLS 1.0 (ang. *Transport Layer Security*), możliwe jest więc zastosowanie szyfrowania i kontroli integralności transmisji [68]. Ponadto, w system wbudowana jest autoryzacja klientów oparta o standardowy mechanizm SASL (ang. *Simple Authentication and Security Layer*) [180].

10.1.4 Transmisja danych multimedialnych

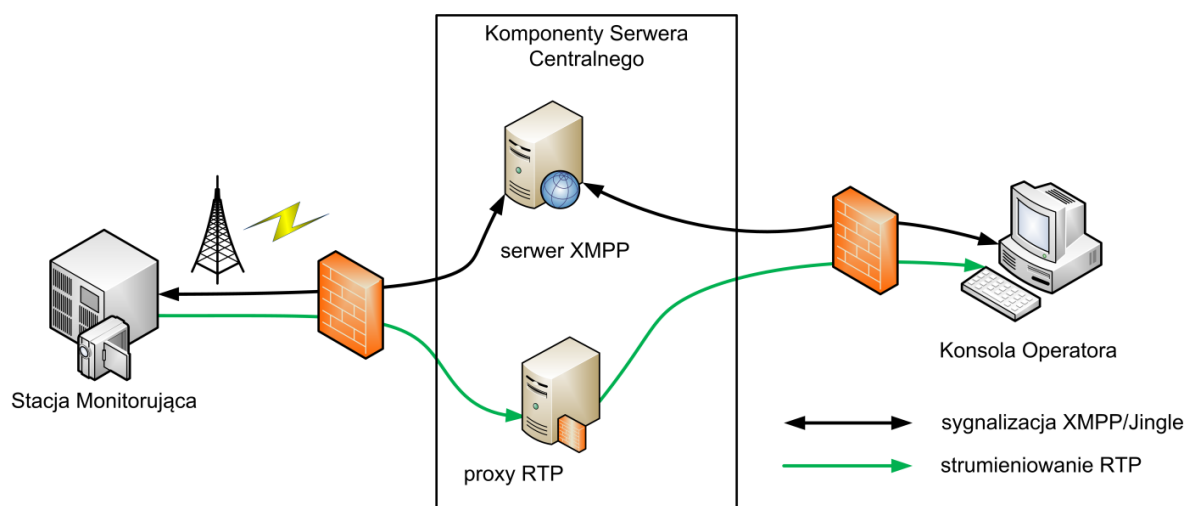
Istotnym rozszerzeniem protokołu XMPP znajdującym zastosowanie w omawianym systemie jest podsystem Jingle [57][162]. XMPP, jako protokół bazujący na TCP/TLS i wymianie komunikatów XML jest rozwiązaniem suboptymalnym w kontekście transmisji czasu rzeczywistego danych multimedialnych, gdyż jego stosowanie wprowadza znaczący narzut związany z kodowaniem danych binarnych do postaci akceptowalnej oraz przyczynia się do dużych opóźnień transmisji. Rozszerzenie Jingle umożliwia natomiast nawiązywanie sesji komunikacji czasu rzeczywistego zewnętrznych (ang. *out-of-band*) w stosunku do sesji XMPP. Wymiana komunikatów związanych z realizacją transmisji odbywa się w całości za pośrednictwem istniejącej sesji XMPP, korzysta więc z zapewnianego przez XMPP adresowania, autoryzacji i bezpieczeństwa kanału transmisyjnego.

Rozszerzenie Jingle definiuje protokół kontrolny (sygnalizacyjny) służący do inicjowania i nadzoru nad właściwą sesją komunikacji, która w omawianym systemie wykorzystuje protokół RTP (ang. *Real-Time Transport Protocol*) do transmisji treści multimedialnych. Protokół RTP jest standardowym narzędziem służącym do realizacji transmisji czasu rzeczywistego [209]. Ze względu na kwestie bezpieczeństwa systemu zastosowano rozszerzenie SRTP, umożliwiające szyfrowanie oraz kontrolę integralności transmisji [7]. Wymiana kluczy SRTP odbywa się w ramach nawiązanej uprzednio, zabezpieczonej sesji XMPP za pośrednictwem rozszerzeń protokołu Jingle.

Transmisja obrazu wykorzystuje kompresję stratną, której rodzaj i parametry jakościowe uzależnione są od możliwości terminala docelowego i uzgadniane są na etapie negocjacji transmisji, czyli wstępnej wymiany komunikatów XMPP przez klienta transmisji (Konsolę Operatora) i serwer (Stację Monitorującą).

Ponieważ transmisja RTP odbywa się zewnętrznie w stosunku do sesji XMPP, powraca problem związany z obecnością urządzeń typu NAT i zapór. Problem ten jest tym istotniejszy, że w omawianym zastosowaniu strona znajdująca się za tego typu przeszkodą

nie zawsze będzie stroną inicjującą transmisję, zaś jej odbiornik końcowy również może znajdować za urządzeniem pośredniczącym. Aby umożliwić transmisję danych multimedialnych zastosowano rozwiązanie oparte na serwerze pośredniczącym (proxy), który jest dostępny za pomocą zewnętrznego adresu IP, przez co może być wykorzystany jako adresat transmisji RTP i przekazywać ją następnie do właściwego odbiorcy. Schemat komunikacji przy transmisji strumieni multimedialnych zilustrowano na rys. 10.3.



Rys. 10.3 Schemat transmisji strumieni multimedialnych w systemie monitoringu opracowanym w Katedrze Systemów Multimedialnych

10.1.5 Implementacja

Omówiona wyżej funkcjonalność została zaimplementowana w systemie w postaci zbioru tzw. usług XMPP (ang. *XMPP Services*). Każda z usług ma postać „wtyczki” programowej uruchamianej w aplikacji ujednoliconego środowiska wykonawczego usług XMPP, dostarczającej funkcjonalności agenta XMPP uruchamianym usługom. Zarówno wspomniana aplikacja, jak i każda z usług, zaimplementowane zostały w języku C++ w oparciu o szereg opracowanych na potrzeby systemu bibliotek programistycznych oferujących warstwę abstrakcji wobec systemu operacyjnego oraz implementujących w sposób niezależny od używanego środowiska poszczególne komponenty stosu protokołów i inne udogodnienia. Pozwala to na uruchamianie opracowanych usług XMPP w środowisku systemów operacyjnych Windows, Linux oraz MacOS X.

Ponadto w implementacji wykorzystano szereg zewnętrznych bibliotek otwartego oprogramowania, z których do najważniejszych należą:

- Boost 1.55 – kolekcja uniwersalnych bibliotek programistycznych poszerzających możliwości języka C++ [266],

- OpenCV 2.4.8 – zestaw funkcji wykorzystywanych podczas przetwarzania obrazu [275],
- CGAL 4.3 – zbiór procedur do obliczeń geometrycznych [267],
- FFmpeg 0.9 – pakiet narzędzi do nagrywania, konwertowania i przesyłania strumieniowej treści audiowizualnych [268],
- OpenRTSP – zestaw bibliotek do strumieniowej transmisji multimediiów zgodnie ze standardami RTP/RTCP/RTSP/SIP [276],
- OpenSSL 1.0.0e – implementacja protokołów SSL i TLS oraz algorytmów kryptograficznych ogólnego przeznaczenia [277].

Jedną z zaimplementowanych usług jest usługa przetwarzania obrazu, która jest kluczowa z punktu widzenia tematyki doktoratu. Inne usługi działające na Stacji Monitorującej to np. usługa umożliwiająca automatyczne uaktualnienie oprogramowania Stacji oraz usługa dostępu do wiersza poleceń systemu, umożliwiająca zarządzanie Stacją za pomocą standardowego, dowolnego klienta sieci XMPP. Dowlone inne usługi zwiększające funkcjonalność Stacji mogą być łatwo dodawane w miarę potrzeb.

10.1.6 Budowa Stacji Monitorującej

Stacja Monitorująca jest głównym elementem opracowanego systemu z punktu widzenia algorytmów przetwarzania obrazu i detekcji zdarzeń. Chociaż oprogramowanie Stacji Monitorującej nie ma żadnych wymagań co do sprzętu komputerowego, na którym jest uruchomione, to jednak stworzono jedną wzorcową konstrukcję, które umożliwia montaż takiej stacji i jej całoroczne działanie w warunkach zewnętrznych.

Wybór platformy sprzętowej dla Stacji Monitorującej podyktowany był jednoczesnym spełnieniem wymogów gabarytowych (możliwe najmniejszych wymiarów), maksymalizacją mocy obliczeniowej, liczbą możliwych do obsłużenia urządzeń peryferyjnych i łatwością rozwoju platformy sprzętowej. W związku z tym wybrano płytę główną o rozmiarach mniejszych od standardowego ATX, tj. model microATX, o rozmiarach 270mm x 270mm.

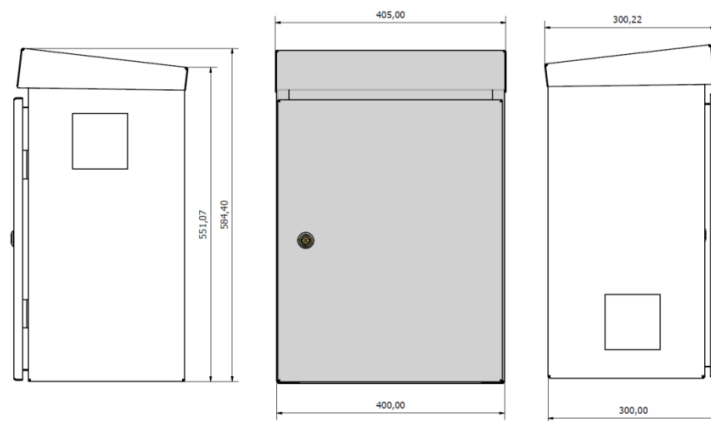
Platforma sprzętowa stanowiąca Stację Monitorującą może być eksploatowana w warunkach:

- wewnętrznych, laboratoryjnych, z wykorzystaniem obudowy fabrycznej (rys. 10.4)
- zewnętrznych, po demontażu elementów plastikowych obudowy i umieszczeniu w dedykowanej stalowej obudowie, wykonanej w Katedrze

Systemów Multimedialnych i odpornej na czynniki atmosferyczne (rys. 10.5 i rys. 10.6).



Rys. 10.4 Komputer Stacji Monitorującej w oryginalnej obudowie



Rys. 10.5 Projekt zrealizowanej obudowy zewnętrznej Stacji Monitorującej



Rys. 10.6 Stacja Monitorująca zamontowana w obudowie zewnętrznej (z lewej) oraz przykładowa instalacja stacji wraz z kamerą na fasadzie budynku (z prawej)

10.2 Płyta DVD

Dołączona do rozprawy płyta DVD zawiera materiały (głównie filmy) ilustrujące działanie poszczególnych algorytmów przetwarzania obrazu, które były przedmiotem badania i opracowania:

- detekcja i śledzenie ruchomych obiektów:
 - detekcja-ludzie.avi
 - gmm-mod-1.avi
 - gmm-mod-2.avi
 - I-1.avi
 - I-2.avi
 - O-Ch.avi
 - O-S.avi
 - S1-T1-C3-detekcja.avi
 - S1-T1-C3-konflikty.avi
 - sledzenie-deszcz.avi
 - sledzenie-ludzie.avi
 - sledzenie-parking.avi
 - sledzenie-pojazdy-1.avi
 - sledzenie-pojazdy-2.avi
 - sledzenie-pojazdy-3.avi
- klasyfikacja obiektów:
 - klasyfikacja.avi
- detekcja zdarzeń na parkingu:
 - parkowanie-1.avi
 - parkowanie-2.avi
 - parkowanie-3.avi
 - zdarzenia-parking-1.avi
 - 151 ramek obrazu z nałożonymi wynikami analizy w momencie wykrycia zdarzenia parkowania pojazdu
- detekcja nieprawidłowej zmiany pasa ruchu:
 - zmiana-pasa.avi
 - zdarzenia-pas-01.avi
 - zdarzenia-pas-02.avi

- zdarzenia-pas-03.avi
- zdarzenia-pas-04.avi
- zdarzenia-pas-05.avi
- zdarzenia-pas-06.avi
- zdarzenia-pas-07.avi
- zdarzenia-pas-08.avi
- zdarzenia-pas-09.avi
- zdarzenia-pas-10.avi

10.3 Spis rysunków

RYS. 1.1	CENTRUM MONITORINGU WIZYJNEGO W GDYNI [269].....	2
RYS. 3.1	ELEMENTY SKŁADOWE WIZYJNEJ ANALIZY OBRAZU	12
RYS. 3.2	WYNIKI DETEKЦИИ PUNKTÓW CHARAKTERYSTYCZNYCH ZA POMOCĄ DETEKTORÓW A) HARRISA, B) KLT I C) SIFT [258]	15
RYS. 3.3	WYNIKI SEGMENTACJI OBRAZU (A) ALGORYTMAMI MEAN-SHIFT (B) I CIĘCIA NORMALIZOWANEGO (C) [258]	17
RYS. 3.4	PRZYKŁAD DZIAŁANIA ALGORYTMU ODEJMOWANIA TŁA Z WYKORZYSTANIEM SUMY WAŻONYCH ROZKŁADÓW GAUSSOWSKICH A) ORYGINALNA RAMKA OBRAZU, B) WARTOŚCI ŚREDNIE ROZKŁADU O NAJWIĘKSZEJ WADZE, C) WARTOŚCI ŚREDNIE NASTĘPNEGO W KOLEJNOŚCI ROZKŁADU, NIE ZALICZANEGO W TYM MOMENCIE DO MODELU TŁA, D) WYKRYTE PIKSELE NALEŻĄCE DO OBIEKTU RUCHOMEGO	20
RYS. 3.5	PODZIAŁ METOD ŚLEDZENIA OBIEKTÓW RUCHOMYCH.....	24
RYS. 3.6	ILUSTRACJA DZIAŁANIA ŚLEDZENIA CIEMNEGO POJAZDU ALGORYTMEM MEAN-SHIFT: A) ESTYMOWANA POZYCJA OBIEKTU W RAMCE POPRZEDNIEJ, B),C), D), E), F) WYNIKI LOKALIZACJI OBIEKTU W BIEŻĄCEJ RAMCE PO PRZEPROWADZENIU ITERACJI NUMER 1, 2, 3, 4 I 6.....	34
RYS. 3.7	PRZYKŁADOWE WEKTORY RUCHU OBIEKTÓW RÓŻNEGO TYPU, OD LEWEJ: POJAZD, OSOBA, GRUPA OSÓB ORAZ ROWERZYSTA, OBLICZONE W PUNKTACH CHARAKTERYSTYCZNYCH ZNALEZIONYCH ZA POMOCĄ DETEKTORA KLT [244]. PUNKT ZACZEPIENIA KAŻDEGO WEKTORA ZILUSTROWANO CIEMNĄ KROPKĄ, A KOLOR ODPOWIADA LOKALIZACJI WEKTORÓW W JEDNYM Z SZEŚCIU PODOBSZARÓW, NA KTÓRE W PIONIE DZIELONY JEST OBRAZ W CELU OKREŚLENIA STOPNIA ZMIENNOŚCI WEKTORÓW W ZALEŻNOŚCI OD LOKALIZACJI W OBRAZIE	45
RYS. 3.8	W ZALEŻNOŚCI OD ORIENTACJI KAMERY WZGLĘDEM POJAZDÓW, WYMIARY PROSTOKĄTA OPISANEGO NA MASCE POJAZDU MOGĄ OZNACZAĆ A) WYSOKOŚĆ I DŁUGOŚĆ POJAZDU [177] LUB B) SZEROKOŚĆ ORAZ KOMBINACJĘ WYSOKOŚCI I DŁUGOŚCI POJAZDU [191]	47
RYS. 3.9	RÓŻNICE W „LINIOWOŚCI” GÓRNEJ KRAWĘDZI CIĘŻARÓWKI (Z LEWEJ) I AUTOBUSU (Z PRAWEJ) [108]	48
RYS. 3.10	ZASTOSOWANIE DESKRYPTORA SIFT DO KLASYFIKACJI TYPU POJAZDU [18], A) OBRAZ POJAZDU I JEGO KRAWĘDZIE WYZNACZONE W OPARCIU O DETEKTOR CANNY’EGO, B) HISTOGRAMY ORIENTACJI KRAWĘDZI DLA OTOCZENIA DWÓCH RÓŻNYCH PUNKTÓW CHARAKTERYSTYCZNYCH (OZNACZONYCH NA CZERWONO I POŁOŻONYCH W ŚRODKACH NIEBIESKICH KWADRATÓW)	49
RYS. 3.11	PRZYKŁADOWE MODELE 3D POJAZDÓW: A) NAJPROSTSZY [152], B) BARDZIEJ ZŁOŻONE [222]	50
RYS. 3.12	PRZYKŁADOWE ZDJĘCIA WYKORZYSTYWANE W SYSTEMIE ROZPOZNAJĄCYM 77 MODELI SAMOCHODÓW [189]	52
RYS. 3.13	PRZYKŁADOWE ZDJĘCIA POJAZDÓW UŻYTE W SYSTEMIE, KTÓRY OSIĄGNĄŁ WYSOKĄ SKUTECZNOŚĆ KLASYFIKACJI [123]	53
RYS. 3.14	PODZIAŁ METOD DETEKЦИИ ZDARZEŃ W RUCHU DROGOWYM	54

RYS. 3.15	ZLICZANIE POJAZDÓW ZA POMOCĄ WIRTUALNEJ BRAMKI [154]	55
RYS. 3.16	POŁOŻENIE STATYCZNYCH OBIEKTÓW TŁA ISTOTNYCH Z PUNKTU WIDZENIA REGUŁ DETEKCJI ZDARZEŃ [149]	56
RYS. 3.17	PRZYKŁADOWY KADR OBRAZU (Z LEWEJ) I WYGENEROWANA AUTOMATYCZNIE MAPA ZACHOWAŃ ZAWIERAJĄCA SKUMULOWANE W CZASIE PARAMETRY RUCHU OBIEKTÓW (Z PRAWEJ) [70]	58
RYS. 4.1	WYBRANE RAMKI Z NAGRANIA PRZETWORZONEGO ALGORYTMEM GMM (LEWA KOLUMNA) ORAZ ALGORYTMEM Z WPROWADZONĄ PRZEZ AUTORA ROZPRAWY MODYFIKACJĄ UMOŻLIWIAJĄCĄ REAGOWANIE NA NAGLE ZMIANY OŚWIETLENIA (PRAWA KOLUMNA); RAMKI POCHODZĄ (ODPOWIEDNIO OD GÓRY DO DOŁU) Z 6., 9., 15. ORAZ 27. SEKUNDY NAGRANIA TESTOWEGO. CZERWONY KOLOR OZNACZA REGIONY Z WYKRYTYM CIENIEM, A ZIELONY – OBSZARY WYKRYTE JAKO ZAWIERAJĄCE PIKSELE NALEŻĄCE DO RUCHOMYCH OBIEKTÓW	65
RYS. 4.2	PRZYKŁADOWE WYNIKI DETEKCJI I ELIMINACJI CIENIA OBIEKTÓW RUCHOMYCH (SZARE OBSZARY W PRAWEJ CZĘŚCI GÓRNEGO RYSUNKU ORAZ CZERWONE OBSZARY NA DOLNYM RYSUNKU).....	67
RYS. 4.3	KOLEJNE ETAPY ALGORYTMU DETEKCJI OBIEKTÓW RUCHOMYCH: A) ORYGINALNA RAMKA OBRAZU Z NAGRANIA, B) MODEL TŁA DLA TEJ RAMKI (DLA KAŻDEGO PIKSELA POKAZANO WARTOŚĆ ŚREDNIĄ ROZKŁADU CHARAKTERYZUJĄCEGO SIĘ NAJWYŻSZĄ WAGĄ), C) BEZPOŚREDNI WYNIK ODEJMOWANIA TŁA, BEZ PRZETWARZANIA MORFOLOGICZNEGO I USUWANIA CIENIA, D) OSTATECZNA MASKA BINARNA, E) WYNIK DETEKCJI RUCHOMYCH OBIEKTÓW: NA ZIELONO ZAZNACZONO OBIEKT, NA CZERWONO – JEGO CIEŃ, F) OBRAZ RUCHOMEGO OBIEKTU WYEKSTRAHOWANY Z RAMKI ŹRÓDŁOWEJ	69
RYS. 4.4	ALGORYTM USTALANIA POWIĄZAŃ MIĘDZY TRACKERAMI A REGIONAMI BĘDĄCYMI WYNIKAMI ODEJMOWANIA TŁA W BIEŻĄCEJ RAMCE (Z LEWEJ) ORAZ TYPY RELACJI UWZGLĘDNIANE W ALGORYTMIE POWIĄZYWANIA (Z PRAWEJ)	74
RYS. 4.5	PRZYKŁADOWE WYNIK ŚLEDZENIA OBIEKTÓW W SYTUACJI KONFLIKTOWEJ; W LEWEJ KOLUMNIE UŻYTO FILTRÓW KALMANA Z 6-CIO ELEMENTOWYM WEKTOREM STANU, W PRAWEJ – Z 8-MIO ELEMENTOWYM; W KOLEJNYCH WIERSZACH ZNAJDUJĄ SIĘ RAMKI NR 1051, 1068, 1085 I 110 Z NAGRANIA S1-T1-C3 Z ZESTAWU NAGRAŃ REFERENCYJNYCH PETS2006 [273]	81
RYS. 4.6	PRZYKŁADOWE RAMKI OBRAZU ORAZ CHARAKTERYSTYKA NAGRAŃ TESTOWYCH UŻYTYCH DO WALIDACJI ALGORYTMÓW DETEKCJI I ŚLEDZENIA RUCHOMYCH OBIEKTÓW	83
RYS. 4.7	WARTOŚCI ŚREDNIE I ODCHYLENIA STANDARDOWE MIAR SŁUŻĄCYCH DO OCENY ALGORYTMÓW DETEKCJI OBIEKTÓW RUCHOMYCH, WYZNACZONYCH DLA KAŻDEGO KAŻDEJ NAGRANIA TESTOWEGO I KAŻDEJ RAMKI WZORCOWEJ	87
RYS. 4.8	PRZYKŁADOWA RAMKA OBRAZU Z NAGRANIA O-S, ANALIZOWANA ALGORYTMEM GMM (U GÓRY) I GMM-MOD (U DOŁU). ZIELONE OBSZARY OZNACZAJĄ WYKRYTE PIKSELE NALEŻĄCE DO RUCHOMYCH OBIEKTÓW (TP I FP), CZERWONE – WYKRYTY CIEŃ. DODATKOWO KOLOROWYMI PROSTOKĄTAMI OZNACZONO WSZYSTKIE WYKRYTE OBIEKTY SPEŁNIAJĄCE KRYTERIUM WIELKOŚCI	88
RYS. 4.9	MIARY OCENY ALGORYTMÓW DETEKCJI RUCHOMYCH OBIEKTÓW UŚREDNIONE DLA WSZYSTKICH NAGRAŃ TESTOWYCH	89
RYS. 4.10	WARTOŚCI MIAR SŁUŻĄCYCH DO OCENY ALGORYTMU ŚLEDZENIA OBIEKTÓW RUCHOMYCH, WYZNACZONE DLA KAŻDEGO NAGRANIA TESTOWEGO	94
RYS. 4.11	ŚREDNIE WARTOŚCI MIAR OCENY ALGORYTMU ŚLEDZENIA RUCHOMYCH OBIEKTÓW	95
RYS. 4.12	SCHEMAT ALGORYTMU WIELOKAMEROWEGO ŚLEDZENIA OBIEKTÓW	96
RYS. 4.13	PRZYKŁAD POJAZDU (ZIELONA MASKA I CZERWONY PROSTOKĄT) POJAWIAJĄCEGO SIĘ W POLU WIDZENIA W JEDNYM Z OBSZARÓW WEJŚCIOWYCH (POMARAŃCZOWE WIELOKĄTY)	98
RYS. 4.14	ILUSTRACJA DWÓCH RODZAJÓW PARAMETRÓW (RÓŻNICA SUM WARTOŚCI PIKSELI W OBSZARACH JASNYCH I CIEMNYCH) WYKORZYSTYWANYCH DO WYZNACZENIA DESKRYPTORA SURF	100
RYS. 4.15	ZALEŻNOŚCI MIĘDZY OBIEKTAMI O, ICH OBRAZAMI I ORAZ WEKTORAMI PARAMETRÓW LOKALNYCH V WYSTĘPUJĄCE W OPRACOWANYM PRZEZ AUTORA ALGORYTMIE REIDENTYFIKACJI OBIEKTÓW .	102
RYS. 4.16	WARTOŚCI FUNKCJI WAGI w_{IK} W ZALEŻNOŚCI OD WARTOŚCI NA OBU WYJŚCIACH SIECI NEURONOWEJ	104
RYS. 4.17	ILUSTRACJA PROCESU REIDENTYFIKACJI OBIEKTU NA PODSTAWIE WARTOŚCI SKUMULOWANEJ ODPOWIEDZI SIECI NEURONOWEJ	104

RYS. 4.18	PRZYKŁADOWE RAMKI OBRAZU Z KAMER (WRAZ Z ICH NUMERAMI) UŻYTYCH W EKSPERYMENTACH ZWIĄZANYCH Z WIELOKAMEROWYM ŚLEDZENIEM POJAZDÓW; OBSZARY WEJŚCIA/WYJŚCIA W POLU WIDZENIA KAŻDEJ KAMERY ZAZNACZONO ZA POMOCĄ POMARAŃCZOWYCH WIELOKĄTÓW.....	105
RYS. 4.19	PRZYKŁADOWE OBRAZY POJAZDU WJEŹDŻAJĄCEGO (GÓRNY RZĄD) I WYJEŹDŻAJĄCEGO (DOLNY RZĄD) Z MONITOROWANEGO TERENU ZAREJESTROWANE PRZEZ WSZYSTKIE KAMERY UŻYTE W EKSPERYMENTACH; NUMER KAMERY POKAZANO POD OBRAZAMI	106
RYS. 4.20	WYNIKI REIDENTYFIKACJI POJAZDÓW DLA KAŻDEJ PARY KAMER I DWÓCH RÓŻNYCH DESKRYPTORÓW OBRAZU	110
RYS. 5.1	PRZYKŁADOWE OBRAZY POJAZDÓW ROZPOZNAWANYCH TYPÓW	111
RYS. 5.2	OKNO APLIKACJI DO GENEROWANIA DWUWYMIAROWYCH SYLWETEK WYBRANYCH TRÓJWYMIAROWYCH MODELI OBIEKTÓW DLA DOWOLNYCH KĄTÓW WIDZENIA	113
RYS. 5.3	PRZYKŁADOWE SYLWETKI POJAZDÓW (Z LEWEJ) I OSÓB (Z PRAWEJ) WYGENEROWANE NA PODSTAWIE MODELI TRÓJWYMIAROWYCH DLA RÓŻNYCH KĄTÓW WIDZENIA	113
RYS. 5.4	ILUSTRACJA PROCESU PARAMETRYZACJI SYLWETKI OBIEKTU: A) PRZYKŁADOWY KSZTAŁT DO PARAMETRYZACJI Z OZNACZONYM ŚRODKIEM CIĘŻKOŚCI, B) PRZEDZIAŁY RADIALNE R_{BIN} , C) PRZEDZIAŁY KĄTOWE α_{BIN} , D) PODOBSZARY (R_{BIN} , α_{BIN}) DO WYZNACZANIA WARTOŚCI DESKRYPTORA, E) WARTOŚCI $R(R_{BIN}, \alpha_{BIN})$ STANOWIĄCE WEKTOR DEFINIUJĄCY KSZTAŁT OBIEKTU	115
RYS. 5.5	SCHEMAT PROCESU KLASYFIKACJI	115
RYS. 5.6	PRZYKŁADOWE RAMKI OBRAZU Z EKSPERYMENTALNYCH NAGRAŃ DLA KĄTÓW WIDZENIA 20° (GÓRNY WIERSZ) I 60° (WIERSZ DOLNY)	117
RYS. 5.7	POŁOŻENIE AUTOMATYCZNIE ZNALEZIONYCH PUNKTÓW CHARAKTERYSTYCZNYCH (CZERWONE ELIPSY) ORAZ ŚRODKI OŚMIU KLASTRÓW WYZNACZONE ALGORYTMEM CENTROIDÓW (ŻÓLTE OKRĘGI) DLA DWÓCH PRZYKŁADOWYCH OBRAZÓW POJAZDÓW O RÓŻNEJ ROZDZIELCZOŚCI.....	121
RYS. 5.8	ŚRODEK CIĘŻKOŚCI MASKI POJAZDU (ŻÓŁTY PUNKT) ORAZ POŁOŻENIE (I ROZMIAR) CZTERECH PUNKTÓW CHARAKTERYSTYCZNYCH (CZERWONE OKRĘGI)	122
RYS. 5.9	OBLICZANIE PARAMETRÓW OBRAZU POJAZDU Z WYKORZYSTANIEM FILTRU GABORA: A) PRZYKŁADOWY OBRAZ POJAZDU, B) JEGO OBRAZ GRADIENTU I_G ORAZ WYNIKI FILTRACJI BANKIEM FILTRÓW GABORA Z PARAMETREM C) $\lambda = 2,5$ I D) $\lambda = 4$; W GÓRNEJ CZĘŚCI C) I D) ILUSTRACJA FILTRU GABORA DLA KIERUNKÓW Θ RÓWNYCH 0° , 45° , 90° I 135° , NIŻEJ: PRZEFILTROWANY OBRAZ GRADIENTU POJAZDU I_G	124
RYS. 5.10	SIGMOIDALNA BIPOLARNA FUNKCJA AKTYWACJI NEURONÓW SIECI (TANGENS HIPERBOLICZNY)..	127
RYS. 5.11	PRZYKŁADOWE RAMKI OBRAZU Z POSZCZEGÓLNYCH NAGRAŃ	131
RYS. 5.12	WYNIKI KLASYFIKACJI TYPU POJAZDU, Z AGREGACJĄ WYNIKÓW PRZY WYKORZYSTANIU KLASYFIKATORA BOOST; SŁUPKI BŁĘDU ILUSTRUJĄ WARTOŚCI ODCHYLENIA STANDARDOWEGO WYNIKÓW UZYSKANYCH W PIĘCIU SERIACH EKSPERYMENTALNYCH RÓŻNIĄCYCH SIĘ DOBOREM WEKTORÓW DO ZBIORU TRENINGOWEGO I TESTOWEGO.....	136
RYS. 5.13	WYNIKI KLASYFIKACJI TYPU POJAZDU W PARACH OSOBOWE-CIĘŻAROWE, OSOBOWE-FURGONETKI I FURGONETKI-CIĘŻAROWE, Z AGREGACJĄ WYNIKÓW I Z WYKORZYSTANIEM KLASYFIKATORA BOOST	138
RYS. 6.1	MIEJSCA PARKINGOWE (ZIELONE WIELOKĄTY) OZNACZONE W POLU WIDZENIA DWÓCH PRZYKŁADOWYCH KAMER	141
RYS. 6.2	SCHEMAT ALGORYTMU WYKRYWANIA ZDARZEŃ PARKOWANIA POJAZDU	141
RYS. 6.3	POŁOŻENIE PUNKTÓW KALIBRACYJNYCH DLA JEDNEJ Z KALIBROWANYCH KAMER.....	144
RYS. 6.4	DOLNE (KOLOR CIEMNOZIELONY) I GÓRNE (KOLOR JASNOZIELONY) PODSTAWY GRANIASTOSŁUPÓW MODELUJĄCYCH POŁOŻENIE MIEJSC PARKINGOWYCH W POLU WIDZENIA DWÓCH KAMER.....	144
RYS. 6.5	FRAGMENT RAMKI OBRAZU, ZWIĄZANEJ Z WYKRYWANIEM WEJŚCIA I WYJŚCIA OSÓB Z BUDYNKU; KOLEM POMARAŃCZOWYM ZAZNACZONO OBSZARY PODSTAWOWY I DODATKOWY WYKORZYSTYWANE PRZEZ ALGORYTM, A PROSTOKĄTAMI W KOLORACH JASNO- I CIEMNONIEBIESKIM ALGORYTM ZAZNACZYŁ DWIE OSOBY WYCHODZĄCE Z BUDYNKU	146
RYS. 6.6	SCHEMAT ROZMIESZCZENIA KAMER STACJONARNYCH (WRAZ Z ICH NUMERAMI) WYKORZYSTYWANYCH DO DETEKЦИИ ZDARZEŃ NA PARKINGU; NUMERACJA KAMER NIE JEST CIĄGLA	

	ZE WZGLĘDU NA OBECNOŚĆ W SYSTEMIE INNYCH KAMER (W TYM OBROTOWYCH, NIE UŻYWANYCH W EKSPERYMENTACH OMAWIANYCH W ROZPRAWIE).....	147
Rys. 6.7	PRZYKŁADOWE RAMKI OBRAZU ZE WSZYSTKICH KAMER STACJONARNYCH UŻYTYCH W EKSPERYMENTACH Z ZAKRESU WYKRYWANIA ZDARZEŃ NA PARKINGU WRAZ Z NANIESIONYM POŁOŻENIEM MIEJSC PARKINGOWYCH (ZIELONE WIELOKĄTY); W POLU WIDZENIA KAMERY NR 9 NIE MA ZDEFINIOWANYCH ŻADNYCH MIEJSC PARKINGOWYCH	147
Rys. 6.8	ŁĄCZNA CZUŁOŚĆ, PRECYZJA I DOKŁADNOŚĆ DETEKЦИИ ZDARZEŃ PARKOWANIA I ZATRZYMANIA SIĘ POZA MIEJSCEM PARKINGOWYM W ZALEŻNOŚCI OD KAMERY	150
Rys. 6.9	HISTOGRAM WYSTĘPOWANIA ZDARZEŃ NA PARKINGU W TYGODNIOWYCH PRZEDZIAŁACH CZASU	151
Rys. 6.10	HISTOGRAM WYSTĘPOWANIA ZDARZEŃ NA PARKINGU W GODZINNYCH PRZEDZIAŁACH CZASU	152
Rys. 6.11	DETEKCJA RUCHOMYCH OBIEKTÓW UZUPEŁNIONA O KIERUNEK RUCHU: A) OBSZAR OBRAZU I Z GRANICĄ MIĘDZY PASAMI RUCHU ZAZNACZONĄ NA CZERWONO, B) WYNIKI DETEKЦИИ OBIEKTÓW ALGORYTMEM GMM-MOD M , C) WYNIK DETEKЦИИ OBIEKTÓW ZA POMOCĄ PRZEPLYWU OPTYCZNEGO O , D) ILOCZYN LOGICZNY OBRAZÓW B) I C); BARWA PIKSELI NA RYSUNKACH C) I D) OZNACZA KIERUNEK RUCHU, PODCZAS GDY JASNOŚĆ PIKSELI JEST WPROST PROPORCJONALNA DO MODUŁU WEKTORA PRĘDKOŚCI	154
Rys. 6.12	İLUSTRACJA MACIERZY WYKORZYSTYWANYCH DO ANALIZY RUCHU OBIEKTÓW W OBRAZIE I ; IM MNIEJSZA WARTOŚĆ ELEMENTU MACIERZY, TYM CIEMNIEJSZY KOLOR NA RYSUNKU: A) UDZIAŁ PIKSELI SPEŁNIAJĄCYCH WARUNEK DOTYCZĄCY KIERUNKU RUCHU H , B) LOKALNE MAKSYMUM H_M , C) ŚRODKÓW DOLNYCH KRAWĘDZI POŁĄCZONYCH OBSZARÓW H_C , D) ŁĄCZNY WYNIK DETEKЦИИ RUCHOMYCH OBIEKTÓW Z NAŁOŻONĄ ZAWARTOŚCIĄ MACIERZY H_C WRAZ Z ZAZNACZONYMI ŚRODKAMI PASÓW RUCHU (ZIELONE I CZERWONE LINIE) I GRANICĄ MIĘDZY DWOMA PASAMI ZAINTERESOWANIA (NIEBIESKA LINIA)	157
Rys. 6.13	İLUSTRACJA PIĘCIU OBSZARÓW, NA JAKIE DZIELONY JEST OBRAZ DLA PRZYKŁADOWEGO KIERUNKU RUCHU POJAZDÓW ZAZNACZONEGO NIEBIESKĄ STRZAŁKĄ I ZABRONIONEGO KIERUNKU ZMIANY PASA (CZERWONA STRZAŁKA); OD LEWEGO DOLNEGO ROGU RYSUNKU DO PRAWEGO GÓRNEGO: OBSZAR NA ZEWNĄTRZ PASA ŹRÓDŁOWEGO O_s , PAS ŹRÓDŁOWY L_s , GRANICA MIĘDZY PASAMI B , PAS DOCELOWY L_D ORAZ OBSZAR NA ZEWNĄTRZ PASA DOCELOWEGO O_D	158
Rys. 6.14	İLUSTRACJA DZIAŁANIA ALGORYTMU DETEKЦИИ ZMIANY PASA (OD LEWEJ DO PRAWEJ I OD GÓRY DO DOŁU) NA PRZYKŁADZIE CIEMNEJ FURGONETKI. KOLOROWE OBSZARY PRZEDSTAWIAJĄ ŁĄCZNY WYNIK DETEKЦИИ RUCHOMYCH OBIEKTÓW (GMM-MOD ORAZ PRZEPLYW OPTYCZNY), GDZIE BARWA OZNACZA KIERUNEK RUCHU, A JEJ NASYCENIE JEST WPROST PROPORCJONALNE DO MODUŁU WEKTORA PRĘDKOŚCI. PASY RUCHU OZNACZONO CZERWONYMI I ZIELONYMI LINIAMI, A GRANICA MIĘDZY DWOMA PASAMI ZAINTERESOWANIA JEST NIEBIESKA. ZAKOLOROWANE KWADRATY SYMBOLIZUJĄ NIEZEROWE ELEMENTY MACIERZY H_C , PRZY CZYM KOLOR OZNACZA STAN POSZCZEGÓLNYCH ŚCIEŻEK P : ZIELONY – PIERWSZY PUNKT ŚCIEŻKI, SZARY – KOLEJNY PUNKT, BIAŁY – ŚCIEŻKA ZOSTAŁA ZAKOŃCZONA, CZERWONY – ŚCIEŻKA ZAKOŃCZONA Z POZYTYWNYM WYNIKIEM DETEKЦИИ; POGRUBIONE LINIE BIAŁE I CZERWONE POKAZUJĄ ZAKOŃCZONE WŁAŚNIE ŚCIEŻKI W CAŁOŚCI.....	160
Rys. 6.15	PRZYKŁADOWA RAMKA OBRAZU Z KAMERY MONITORUJĄCEJ RUCH ULICZNY W WARSZAWIE (A), ORAZ LOKALIZACJA DWÓCH PASÓW RUCHU (CZERWONY) I GRANICY MIĘDZY NIMI (NIEBIESKI) GDZIE ZABRONIONA JEST ZMIANA PASA (B).....	160
Rys. 6.16	PRZYCIĘTE RAMKI OBRAZU Z KAMERY MONITORINGU OBRAZUJĄCE PRZYPADKI NIEPRAWIDŁOWEJ ZMIANY PASA RUCHU.....	161
Rys. 7.1	MODUŁOWA ARCHITEKTURA OPROGRAMOWANIA STACJI MONITORUJĄCEJ.....	164
Rys. 7.2	KOMPLETNY SCHEMAT PRZETWARZANIA STRUMIENIA WIZYJNEGO.....	167
Rys. 7.3	PRZYKŁADOWE RAMKI Z NAGRAŃ TESTOWYCH N1 (Z LEWEJ) ORAZ N2 I N3 (Z PRAWEJ).....	168
Rys. 7.4	ŚREDNI CZAS PRZETWARZANIA JEDNEJ RAMKI OBRAZU PRZEZ POSZCZEGÓLNE MODUŁY ANALIZY OBRAZU (8 WĄTKÓW)	174
Rys. 7.5	WYKORZYSTANIE POSZCZEGÓLNYCH MODUŁÓW ANALIZY OBRAZU (ŁĄCZNY CZAS PRZETWARZANIA OBRAZU PRZEZ MODUŁ ODNIESIONY DO RZECZYWISTEGO CZASU TRWANIA ANALIZY NAGRANIA TESTOWEGO)	175

RYS. 10.1	FUNKcjONALNE KOMPONENTY SYSTEMU MONITORINGU OPRACOWANEGO W KATEDRZE SYSTEMÓW MULTIMEDIAŁNYCH	197
RYS. 10.2	APLIKACJA KONSOLI OPERATORA POWSTAŁA W KATEDRZE SYSTEMÓW MULTIMEDIAŁNYCH NA POTRZEBY WIZUALIZACJI WYNIKÓW DZIAŁANIA SYSTEMU MONITORINGU	198
RYS. 10.3	SCHEMAT TRANSMISJI STRUMIENI MULTIMEDIAŁNYCH W SYSTEMIE MONITORINGU OPRACOWANYM W KATEDRZE SYSTEMÓW MULTIMEDIAŁNYCH	201
RYS. 10.4	KOMPUTER STACJI MONITORUJĄCEJ W ORYGINALNEJ OBUDOWIE	203
RYS. 10.5	PROJEKT ZREALIZOWANEJ OBUDOWY ZEWNĘTRZNEJ STACJI MONITORUJĄCEJ	203
RYS. 10.6	STACJA MONITORUJĄCA ZAMONTOWANA W OBUDOWIE ZEWNĘTRZNEJ (Z LEWEJ) ORAZ PRZYKŁADOWA INSTALACJA STACJI WRAZ Z KAMERĄ NA FASADZIE BUDYNKU (Z PRAWYJ)	203

10.4 Spis tabel

TAB. 1.1	PUBLIKACJE AUTORA W TEMATYCE ROZPRAWY	6
TAB. 2.1	PORÓWNANIE TECHNOLOGII MONITOROWANIA RUCHU DROGOWEGO [143]	11
TAB. 3.1	PORÓWNANIE METOD DETEKcJI OBIEKTÓW W OBRAZIE.....	22
TAB. 3.2	PORÓWNANIE METOD ŚLEDZENIA OBIEKTÓW RUCHOMYCH.....	41
TAB. 4.1	WARTOŚCI PARAMETRÓW ALGORYTMÓW DETEKcJI OBIEKTÓW RUCHOMYCH TŁA UŻYTE DO ANALIZY NAGRAŃ TESTOWYCH	84
TAB. 4.2	WARTOŚCI MIAR OCENY ALGORYTMU DETEKcJI OBIEKTÓW DLA PRZYKŁADOWYCH RAMEK OBRAZU; KOLOREM ZIELONYM ZAZNACZONO WYKRYTE, RUCHOME OBIEKTY.....	86
TAB. 4.3	WARTOŚCI PARAMETRÓW ALGORYTMU ŚLEDZENIA OBIEKTÓW RUCHOMYCH UŻYTE DO ANALIZY NAGRAŃ TESTOWYCH	90
TAB. 4.4	WARTOŚCI MIAR OCENY ALGORYTMU ŚLEDZENIA OBIEKTÓW DLA PRZYKŁADOWYCH RAMEK OBRAZU	93
TAB. 4.5	ŚREDNIE LICZBY POJAZDÓW (P) I OBRAZÓW NA JEDEN POJAZD (O) W ZBIORACH UCZĄCYM I TESTOWYM PRZYPADAJĄCE NA JEDNĄ SIEĆ NEURONOWĄ.....	107
TAB. 4.6	ZBIORCZE WYNIKI REIDENTYFIKACJI POJAZDÓW DLA KAŻDEJ KAMERY UŻYTEJ JAKO ŹRÓDŁOWA LUB DOCELOWA (WYKORZYSTANO DESKRYPTORY SURF-64) Z WYRÓŻNIONYMI WYNIKAMI Z LICZBĄ POPRAWNYCH KLASYFIKACJI PONAD 50%	108
TAB. 4.7	WYNIKI REIDENTYFIKACJI POJAZDÓW DLA KAŻDEJ PARY KAMER I DWÓCH RÓŻNYCH DESKRYPTORÓW OBRAZU Z WYRÓŻNIONYMI WYNIKAMI DLA KAMER NR 2, 3 I 4.	109
TAB. 4.8	STATYSTYKA WARTOŚCI SKUMULOWANEJ ODPOWIEDZI SIECI NEURONOWEJ R DLA POJAZDÓW POZYTYWNYCH (POPRAWNYCH) I NEGATYWNYCH (NIEPOPRAWNYCH) ZNAJDUJĄCYCH SIĘ W ZBIORZE TESTOWYM	110
TAB. 5.1	LICZEBNOŚCI OBIEKTÓW POSZCZEGÓLNYCH KLAS W DWÓCH TYPACH NAGRAŃ UŻYTYCH W EKSPERYMENTACH	117
TAB. 5.2	MACIERZ BŁĘDU KLASYFIKACJI TYPU OBIEKTU DLA KĄTA WIDZENIA KAMERY 20°	118
TAB. 5.3	MACIERZ BŁĘDU KLASYFIKACJI TYPU OBIEKTU DLA KĄTA WIDZENIA KAMERY 60°	118
TAB. 5.4	CHARAKTERYSTYKA NAGRAŃ UŻYTYCH W EKSPERYMENTACH.....	131
TAB. 5.5	SUMARYCZNE WYNIKI KLASYFIKACJI TYPU POJAZDÓW NA PODSTAWIE NAGRANIA A Z WYKORZYSTANIEM RÓŻNYCH KLASYFIKATORÓW I PARAMETRÓW BEZ AGREGACJI WYNIKÓW (KAŻDY OBRAZ POJAZDU KLASYFIKOWANY NIEZALEŻNIE)	132
TAB. 5.6	SUMARYCZNE WYNIKI BINARNEJ KLASYFIKACJI TYPU POJAZDÓW NA PODSTAWIE NAGRANIA A Z WYKORZYSTANIEM DWÓCH KLASYFIKATORÓW I RÓŻNYCH PARAMETRÓW, BEZ AGREGACJI WYNIKÓW (KAŻDY OBRAZ POJAZDU KLASYFIKOWANY NIEZALEŻNIE)	133
TAB. 5.7	SZCZEGÓŁOWE WYNIKI KLASYFIKACJI POJAZDÓW POMIĘDZY TRZY TYPY, BEZ AGREGACJI WYNIKÓW, Z WYKORZYSTANIEM DWÓCH RODZAJÓW KLASYFIKATORÓW I RÓŻNYCH NAGRAŃ	135
TAB. 5.8	SZCZEGÓŁOWE WYNIKI KLASYFIKACJI POJAZDÓW POMIĘDZY TRZY TYPY, Z AGREGACJĄ WYNIKÓW, Z WYKORZYSTANIEM DWÓCH RODZAJÓW KLASYFIKATORÓW I RÓŻNYCH NAGRAŃ	136
TAB. 5.9	MACIERZ BŁĘDU DLA KLASYFIKATORA BOOST I NAGRANIA A	137

TAB. 6.1	MODELE KAMER UŻYTYCH W EKSPERYMENTACH DOTYCZĄCYCH DETEKCJI ZDARZEŃ NA PARKINGU WRAZ Z PARAMETRAMI AKWIZYCJI OBRAZU	148
TAB. 6.2	MIARY OCENY SKUTECZNOŚCI DETEKCJI ZDARZEŃ NA PARKINGU.....	149
TAB. 6.3	WYNIKI DETEKCJI NIEPRAWIDŁOWEJ ZMIANY PASA	162
TAB. 7.1	CHARAKTERYSTYKA NAGRAŃ TESTOWYCH DO BADANIA WYDAJNOŚCI ALGORYTMÓW	168
TAB. 7.2	KONFIGURACJA KOMPUTERA UŻYTEGO DO TESTÓW WYDAJNOŚCI	169
TAB. 7.3	WYDAJNOŚĆ OBLICZENIOWA (W RAMKACH NA SEKUNDĘ) W ZALEŻNOŚCI OD NAGRANIA I LICZBY WYKORZYSTANYCH WĄTKÓW	169
TAB. 7.4	WYDAJNOŚĆ OBLICZENIOWA (W RAMKACH NA SEKUNDĘ) W ZALEŻNOŚCI OD NAGRANIA I LICZBY WYKORZYSTANYCH WĄTKÓW DLA WSPÓŁCZYNNIKA SKALOWANIA RÓWNEGO 0,5.....	170
TAB. 7.5	ŚREDNIE OBCIĄŻENIE PROCESORA PROCESEM PRZETWARZANIA OBRAZU (100% = CAŁKOWITE OBCIĄŻENIE 8 RDZENI, 12,5% = CAŁKOWITE OBCIĄŻENIE JEDNEGO RDZENIA)	171
TAB. 7.6	LICZBA STRUMIENI WIZYJNYCH MOŻLIWYCH DO RÓWNOCZESNEGO PRZETWARZANIA NA JEDNYM KOMPUTERZE (8 WĄTKÓW).....	171
TAB. 7.7	SZACUNKOWE ZAPOTRZEBOWANIE NA PRZEPUSTOWOŚĆ PODSYSTEMU PAMIĘCI	172
TAB. 7.8	ŚREDNIA ZAJĘTOŚĆ PAMIĘCI OPERACYJNEJ PRZEZ OPROGRAMOWANIE STACJI MONITORUJĄCEJ ..	172
TAB. 7.9	ŚREDNI CZAS PRZETWARZANIA JEDNEJ RAMKI PRZEZ POSZCZEGÓLNE MODUŁY ANALIZY OBRAZU (8 WĄTKÓW).....	173
TAB. 7.10	WYKORZYSTANIE POSZCZEGÓLNYCH MODUŁÓW ANALIZY OBRAZU (ŁĄCZNY CZAS PRZETWARZANIA OBRAZU PRZEZ MODUŁ ODNIESIONY DO RZECZYWISTEGO CZASU TRWANIA ANALIZY NAGRANIA TESTOWEGO)	176