

Tomasz Kryjak*

Analiza i testowanie algorytmów generacji tła na potrzeby systemu monitoringu przestrzeni publicznej**

1. Wprowadzenie

W ostatnich latach można zaobserwować dynamiczny rozwój systemów monitoringu. Wynika on z jednej strony z coraz większej potrzeby zapewnienia bezpieczeństwa publicznego, a z drugiej ma związek z rozwojem technologii kamer i transmisji obrazu. Pierwsze systemy wykorzystywały kamery niskiej rozdzielczości, najczęściej pracujące w odcieniach szarości, a transmisja obrazu odbywała się po kablu koncentrycznym. Miało to szereg wad, do których należy zaliczyć niską jakość obrazu oraz konieczność prowadzenia osobnego kabla do każdej z kamer w systemie. Współcześnie obserwuje się rozwój kamer cyfrowych, które umożliwiają transmisję obrazu poprzez sieć Ethernet. Zatem grupa kamer może być połączona z centralą jednym kablem sieciowym. Upraszcza to architekturę systemu monitoringu i ułatwia jego rozbudowę. Dodatkowo coraz częściej wykorzystuje się kamery pracujące w kolorze oraz kamery dzień/noc z wbudowanym oświetlaczem podczerwieni.

Podstawowym problemem wynikającym ze wzrostu liczby kamer w systemie jest potrzeba analizy wielu strumieni wideo. Klasyczny schemat pracy centrali monitoringu zakłada analizę obrazu wyświetlanego na wielu monitorach przez operatora. Podejście to ma jedną niezaprzeczalną zaletę. Jak dotąd nie udało się skonstruować sztucznego systemu wizyjnego, który choć częściowo byłby w stanie dokonać analizy sceny z szybkością, dokładnością i trafnością porównywalną do człowieka. Jednakże ma też szereg wad. Człowiek nie jest w stanie cały czas skupić się na analizie kilku, kilkunastu strumieni wideo, szczególnie, że przez większość czasu na obrazie nie ma sytuacji nietypowych. Ponadto praca taka jest bardzo nużąca. Z wyżej wymienionych powodów monitoring w wielu przypadkach

* AGH Akademia Górniczo-Hutnicza, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki, Katedra Automatyki, Laboratorium Biocybernetyki, al. A. Mickiewicza 30, 30-059 Kraków

** Praca wykonana w ramach projektu „SIMPOZ” z Ministerstwa Nauki i Szkolnictwa Wyższego nr 0128/R/t00/2010/12

sprowadza się do analizy sekwencji wideo już po zdarzeniu (np. kradzieży). Zatem jego skuteczność jest mniejsza niż gdyby możliwa była analiza na bieżąco.

Rozwiązaniem przedstawionych problemów jest automatyzacja monitoringu. Obecny etap rozwoju algorytmów wizyjnych, szczególnie tych zaliczanych do grupy rozumienia obrazów, nie pozwala na całkowitą automatyzację. Tym niemniej odpowiedni system może znacznie ułatwić pracę operatorowi, głównie przez wskazywanie sytuacji nietypowych, które powinien przeanalizować. Do takich zaliczyć można: pozostawienie bagażu (potencjalny ładunek wybuchowy), nietypowe zachowanie (upadek człowieka, nagłe zgromadzenie, bójka), naruszenie strefy zabronionej (np. przejście przez barierki w muzeum), zniknięcie nadzorowanego przedmiotu itp.

Analiza wymienionych powyżej sytuacji wskazuje, że podstawą działania automatycznego systemu wspomagania monitoringu wizyjnego powinien być algorytm, który operuje na dwóch obrazach: aktualnej ramce oraz modelu tła. Metoda ta, znana w literaturze jako generacja tła lub odejmowanie tła (*background generation, background subtraction*), pozwala na wykrywanie obiektów, które pojawiły się na scenie, oraz tych, które ze sceny zostały usunięte. Pośrednio metoda wykrywa również obiekty ruchome. Temat generacji tła jest obecny w badaniach naukowych od ponad 20 lat. Ostatnio zaproponowano szereg metod, których obszerne zestawienie zawiera praca [5]. W konkluzji autorzy pracy stwierdzili, że nie istnieje jeden dobry algorytm generacji tła, zatem konieczne jest dopasowywanie algorytmu dla konkretnego rozwiązania.

W artykule opisano i przetestowano szereg algorytmów generacji tła, które mogły by być podstawą konstrukcji automatycznego systemu monitoringu wizyjnego. W rozdziale 2 zdefiniowano cele pracy, w rozdziale 3 omówiono ideę generacji tła. Rozdział 4 dotyczy wyboru przestrzeni barw, która ma być wykorzystana w systemie. W rozdziale 5 dokonano analizy wybranych algorytmów generacji tła, a w rozdziale 6 przedstawiono dyskusję na temat selektywnej aktualizacji tła. Artykuł kończy podsumowanie i wskazanie dalszych kierunków prac.

2. Cel pracy

Celem niniejszej pracy jest uzyskanie odpowiedzi na szereg pytań, które są kluczowe przy wyborze algorytmu generacji tła. Analiza uzyskanych rezultatów powinna wskazać, który z opisanych w literaturze algorytmów najlepiej spełnia wymagania stawiane przez system automatycznego monitoringu wizyjnego.

Problemem, który trzeba rozstrzygnąć na wstępie, jest dokonanie wyboru, czy algorytm ma działać w odcieniach szarości, czy w kolorze, i jeżeli w kolorze, to w jakiej przestrzeni barw (np. RGB, znormalizowane RGB, HSV, YCrCb, CIE Lab itp.). Analiza obrazu kolorowego oznacza w przybliżeniu trzykrotnie większą złożoność obliczeniową i pamięciową algorytmu. Dodatkowo, praca w innej przestrzeni barw niż dostępna na wejściu systemu (zwykle RGB), wprowadza narzut w postaci operacji konwersji. Potencjalnymi zaletami

pracy w kolorze są lepsze rezultaty segmentacji (obiektów, które mają zbliżoną jasność do tła, ale różnią się kolorem) oraz możliwość redukcji wpływu cieni [2].

Kolejną, bardzo ważną, kwestią jest wybór algorytmu generacji tła. Dostępnych metod i ich modyfikacji jest bardzo dużo [5]. Przy wyborze trzeba wziąć pod uwagę następujące kryteria: złożoność pamięciową, złożoność obliczeniową, dokładność w określaniu konturów obiektów, stabilność wykrycia obiektów w czasie, zdolność do wykrywania niewielkich zmian (wolno poruszających się obiektów) oraz niewrażliwość na zmiany warunków (głównie oświetlenia). Dodatkowo w literaturze [3] wskazanych jest szereg problemów, które powinien rozwiązać idealny algorytm generacji tła. Należą do nich:

- odporność na zaszumienie obrazu (np. niska jakość kamery, słaba czułość przy braku wystarczającego oświetlenia itp.),
- drżenie kamery (np. kamera zamocowana na słupie sygnalizacji drogowej przy silnym wietrze),
- automatyczna zmiana nastaw kamery (balans bieli, wzmocnienie),
- pora dnia (płynne, powolne zmiany oświetlenia),
- nagła zmiana oświetlenia (włączenie oświetlenia, słoneczno/pochmurny dzień),
- obecność obiektów ruchomych w sekwencji inicjalizacyjnej,
- tło multimodalne – w tle obecny jest ruch, który nie powinien być wykrywany: płynąca woda, ruszające się liście,
- kamuflaż – obiekt ruchomy jest bardzo zbliżony do tła,
- obecność cieni.

Dodatkowo wyróżnia się cztery złożone problemy, których rozwiązanie na poziomie algorytmu generacji tła jest bardzo trudne lub niemożliwe:

- poruszone obiekty w tle (np. przesunięte krzesło),
- obiekty wstawione w tło (np. osoba usiadła na krześle w poczekalni i została po pewnym czasie uznana za element tła),
- obiekty, które początkowo były nieruchome, zaczynają się ruszać (np. ruszający samochód na parkingu).

W pracy większy nacisk położono na problemy z pierwszej grupy, ponieważ wskazane, złożone zagadnienia trudno rozwiązać bez semantycznej analizy sceny, która wymaga wykonania segmentacji, indeksacji oraz klasyfikacji każdego ze znalezionych obiektów i wykorzystania kontekstu czasowego, w postaci śledzenia obiektów.

3. Algorytmy generacji tła

Idea generacji tła wywodzi się z metody stałego tła referencyjnego. W pierwszym kroku tworzony jest model tła, który w tym przypadku stanowi pusta scena (bez obiektów ruchomych). Następnie od każdej ramki z sekwencji wideo odejmowany jest model tła,

a wynik poddany binaryzacji. Rezultatem jest maska obiektów, których nie było wcześniej na scenie. Podstawowa wada tego podejścia to brak adaptacyjności. Już niewielka zmiana oświetlenia sceny spowoduje pojawianie się wielu „fałszywych” obiektów. Jednakże w ściśle kontrolowanych warunkach (przykładowo studia telewizyjne) metoda ta jest stosowana.

Rozwinięciem opisaney idei jest dopuszczenie modyfikacji modelu tła. Algorytmy generacji tła można podzielić na nierekurencyjne lub rekurencyjne. W pierwszym przypadku model tła wyznaczany jest na podstawie bufora N ostatnich ramek z sekwencji (np. średnia, mediana, itp.). W drugim model tła jest uaktualniany tylko na podstawie aktualnej ramki. W pracy analizowano algorytmy należące do obu grup.

3.1. Algorytmy nierekurencyjne

Podstawą działania algorytmów nierekurencyjnych są operacje na buforze N ostatnich ramek z sekwencji wideo. Pierwszą cechą takiego podejścia jest znaczne zapotrzebowanie na pamięć. Przykładowo dla sekwencji o rozdzielczości 640×480 , w kolorze RGB (24 bity) zapamiętanie $N = 30$ ramek wymaga ok. 26 MB pamięci RAM. W przypadku współczesnych komputerów PC, w których rozmiar pamięci operacyjnej wynosi co najmniej 1GB, wielkość ta nie stanowi istotnego ograniczenia. Problemem może być natomiast sposób wykorzystywania bufora. Dla najprostszego podejścia – średniej z bufora, możliwe jest wykorzystanie następującego schematu obliczeń:

$$S_{t+1} = S_t + \frac{I_{t+1}}{N} - \frac{B_{t-N}}{N} \quad (1)$$

gdzie:

- S_{t+1} – nowa wartość średniej,
- S_t – poprzednia wartość średniej,
- I_{t+1} – wartość z aktualnej ramki,
- B_{t-N} – wartość z bufora ramek,
- N – rozmiar bufora.

Zaletą wykorzystania wzoru (1) jest ograniczenie liczby potrzebnych operacji odczytu/zapisu z pamięci RAM. Jeżeli bufor ramek zorganizuje się jako bufor cykliczny, to przy wyznaczaniu nowej wartości średniej konieczne jest wykonanie tylko jednego odczytu i jednego zapisu.

W przypadku innych wariantów metody, np. średniej ważonej (ramki wcześniejsze mają mniejszą wagę niż późniejsze), nie da się uprościć obliczeń i konieczny jest dostęp do każdego elementu bufora. Przykładowo dla parametrów rozdzielczość 640×480 i $N = 30$ liczba odczytów wynosi 9 216 000 w każdej iteracji algorytmu (zakłada się, że w jednym odczycie uzyskuje się informacje o trzech składowych barwnych).

Wyliczenie mediany z bufora ramek można przyspieszyć, wykorzystując histogramy [7] lub odpowiednie struktury danych [6]. Jednakże, każde z rozwiązań, będzie miało

złożoność obliczeniową większą od metody ze średnią. Do innych technik nierekurencyjnych zaliczyć też można algorytm W4, filtr Wienera i modelowanie nieparametryczne (KDE – *Kernel Density Estimation*), których opis można znaleźć w pracy [5].

W literaturze wskazuje się następujące zalety algorytmów nierekurencyjnych: wysoką adaptacyjność i brak wpływu na model tła historii sprzed N ramek. Wada to przede wszystkim złożoność pamięciowa, szczególnie, gdy konieczne jest śledzenie obiektów, które poruszają się wolno, gdyż wtedy konieczne jest wykorzystanie dużego bufora na ramki.

3.2. Algorytmy rekurencyjne

Większość algorytmów generacji tła opisywanych w literaturze zalicza się do grupy rekurencyjnych. Zasada działania zakłada aktualizację modelu tła tylko na podstawie aktualnej ramki, bez konieczności zapamiętywania bufora ramek. Wybrane i poddane analizie algorytmy należące do tej grupy to metoda sigma-delta [10], pojedynczy rozkład Gaussa [19], wielokrotny rozkład Gaussa (MOG) [18] oraz klasteryzacja [4]. Warto zaznaczyć, że dwa pierwsze algorytmy przechowują tylko jeden wariant tła, a dwa ostatnie należą do metod wielowariantowych. Oznacza to, że dla każdej lokalizacji na obrazie generowany jest więcej niż jeden wariant tła (zwykle od 3–5). Pozwala to na eliminację wykrywania drobnych ruchów elementów tła (przykładowo liści, płynącej wody) oraz umożliwia stworzenie mechanizmu inicjalizacji tła w obecności obiektów ruchomych na scenie.

Dokładny opis algorytmów znaleźć można w podanej literaturze. W tym miejscu podano jedynie równania będące podstawą adaptacji wymienionych metod. Dla pojedynczego i wielokrotnego rozkładu Gaussa oraz klasteryzacji opisany on jest poprzez równanie:

$$\mu_t = \alpha I_t + (1 - \alpha) \cdot \mu_{t-1} \quad (2)$$

gdzie:

I_t – aktualna ramka, μ_{t-1} ,

μ_t – poprzednia i aktualna wartość średniej (model),

α – empirycznie dobierany współczynnik z przedziału (0;1) – zwykle kompromis pomiędzy stabilnością a szybkim uzupełnianiem tła.

Dla metody sigma-delta uaktualnianie modelu tła opisuje równanie:

$$M_t = M_{t-1} + \text{sgn}(I_t - M_{t-1}) \quad (3)$$

gdzie:

I_t – ramka obrazu,

M_t – model tła (aproksymacja mediany),

sgn – funkcja signum.

W literaturze podkreśla się następujące zalety metod rekurencyjnych: brak konieczności tworzenia dużego bufora na ramki, szybsze i prostsze obliczenia (dla metod z jednym wariantem tła). Ponadto metody da się prosto rozwinąć w schemat wielu wariantów tła.

Podstawowa wada to duża podatność na błędy w tle, które jak już powstaną to są długo eliminowane (w zależności od wartości parametru α z równania (2)).

4. Wybór przestrzeni barw

Przy realizacji systemu wizyjnego istotna jest kwestia wyboru przestrzeni barw spośród szeregu dostępnych: odcienie szarości, RGB, HSV, YCbCr, CIE Lab, CIE Luv. Podstawą jest przestrzeń RGB, gdyż zwykle w takiej postaci dostępny jest sygnał na wejściu systemu wizyjnego (kamera kolorowa, sekwencja wideo). Wybór pracy w innej przestrzeni wiąże się z koniecznością wykonania konwersji, która, mimo iż jest operacją punktową i dość prostą, może mieć wpływ na wydajność systemu. W tabeli 1 zaprezentowano wyniki pomiarów czasu konwersji przeprowadzonych dla biblioteki przetwarzania obrazów OpenCV w wersji 2.1. Testy wykonano na komputerze PC wyposażonym w procesor Intel i7 920 taktowny zegarem 2,66 GHz, 8 GB RAM'u i 64-bitowy system operacyjny Windows7. Programy kompilowano w Visual Studio 2008. W tabeli podano uśredniony czas wykonania konwersji dla 10000 prób dla obrazu w rozdzielczości 720×576. Uwzględniono zarówno obrazy zapisane w formacie *uint8* (8-bitowa liczba całkowita bez znaku) oraz *float* (32-bitowa liczba zmiennoprzecinkowa).

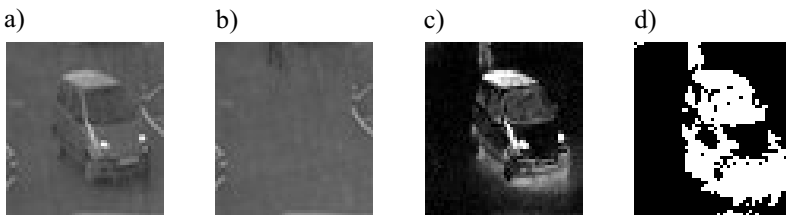
Tabela 1
Zestawienie czasów konwersji pomiędzy wybranymi przestrzeniami barw

Konwersja	Uśredniony czas wykonania [ms]	
	Format <i>uint8</i>	Format <i>float</i>
RGB -> GRAY	1,2088	0,7756
RGB -> YCrCb	2,4263	2,3259
RGB -> HSV	6,5015	7,4008
RGB -> CIE Lab	5,1977	37,2393

Analiza danych z tabeli 1 pokazuje, że operacja konwersji pomiędzy przestrzeniami barw może stanowić istotny, z punktu widzenia czasu przetwarzania, element algorytmu wizyjnego. W przypadku strumienia wizyjnego z obrazem próbkowanym 25 razy na sekundę, wymaganie pracy w czasie rzeczywistym oznacza, że maksymalny czas przetwarzania jednej ramki nie może przekroczyć 40 ms. Zatem dla formatu *uint8* sama konwersja stanowi od 3% do 16% dopuszczalnego czasu obliczeń. Wyniki przeprowadzonego eksperymentu pokazują, że przy wyborze przestrzeni barw, oprócz pożądanych cech, takich jak: zgodność z percepcją barw przez człowieka i przydatnością w redukcji cieni [2], należy również uwzględnić czas obliczeń.

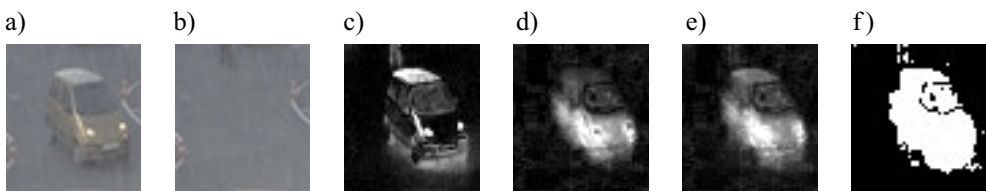
Kolejnym etapem badań była próba wskazania sytuacji, w których wykorzystanie informacji o kolorze pozwala poprawić wyniki segmentacji obiektów na scenie. W tym celu przeanalizowano szereg sekwencji wideo. Przyjęto następujące założenia. Odległość pomiędzy aktualną ramką a tłem liczona jest jako Euklidesowa oraz próg binaryzacji dobierany jest ręcznie tak, aby uzyskać jak najlepszą maskę obiektu. W trakcie analizy sekwencji testowych stwierdzono, że wskazanie sytuacji, w której tylko informacja o kolorze umożliwia poprawną segmentację, jest zadaniem trudnym. W większości przypadków daje się ręcznie dobrać taki próg binaryzacji w odcieniach szarości, aby wynik był poprawny, tym niemniej przykładowo dla sceny przedstawionej na rysunku 1 jest to niemożliwe.

Na scenie (rys. 1a) widoczny jest samochód. Analiza rysunku 1c, na którym pokazano moduł z różnicy pomiędzy sceną w odcieniach szarości a tłem w odcieniach szarości, wskazuje, że karoseria ma prawie identyczną luminancję jak asfalt. Przeprowadzenie binaryzacji tego obrazu nie pozwala na poprawne wykrycie całego samochodu (rys. 1d). Ustawienie niskiego progu powoduje wykrywanie cieni pod samochodem, ale obszar maski pozostaje niewykryty.



Rys. 1. Segmentacja w odcieniach szarości: a) scena; b) tło; c) obraz różnicowy; d) rezultat binaryzacji obrazu różnicowego

Na rysunku 2a i b przedstawiono scenę i tło w kolorze. Żółta karoseria samochodu wyraźnie odróżnia się od szarego tła. Zatem wykorzystanie informacji o kolorze powinno pozwolić na poprawną segmentację. W celu potwierdzenia tezy przeprowadzono szereg eksperymentów z różnymi przestrzeniami barw: RGB, znormalizowane RGB (Y_{xy} [8]) YCbCr, Lst [8], CIE Lab.



Rys. 2. Segmentacja w przestrzeni Y_{xy} : a) scena; b) tło; c) obraz różnicowy dY ; d) obraz różnicowy dxY ; e) średnia ważona obu obrazów różnicowych; f) rezultat binaryzacji

W książce [8] autor stwierdza, że przestrzeń barw, w której przeprowadzana jest segmentacja na podstawie koloru, powinna składać się ze składowej luminancji oraz składowych

chrominacji (niezależnych od składowej luminancji). Wyniki wykonanych eksperymentów potwierdzają tę tezę. W przestrzeni RGB nie udało się dokonać poprawnej segmentacji samochodu z rysunku 2a, pomimo wypróbowania szeregu miar odległości: Euklidesowej, kąto-odległościowej [13], Canberra [14]. Natomiast w przestrzeniach Yxy, YCbCr, CIE Lab i Lst uzyskano znacznie lepsze rezultaty. Przykładowo dla przestrzeni Yxy na rysunku 2c zaprezentowano odległość Euklidesową pomiędzy składowymi Y sceny i tła (dY), a na rysunku 2d odległość pomiędzy składowymi xy sceny i tła (dxy). Średnią ważoną odległości dY i dxy (wagę składowej dY ustalono na 1, a wagę składowej dxy na 2) pokazano na rysunku 2e, a rezultat binaryzacji z progami 0,15 na rysunku 2f.

Podsumowując wyniki przeprowadzonych badań wstępnych, należy stwierdzić, że wykorzystanie informacji o kolorze umożliwia segmentację obiektów w pewnej grupie szczególnych przypadków (do takiej należy zaprezentowany przykład z samochodem) oraz poprawia rezultaty segmentacji w pozostałych przypadkach. Jedną z niewątpliwych zalet podejścia jest mniejsza wrażliwość algorytmu na wybór końcowego progu binaryzacji.

Wskazanie przestrzeni barw, która najlepiej sprawdza się w systemie automatycznego monitoringu wizyjnego, wymaga przeprowadzenia dalszych badań. W przedstawionych eksperymentach nie analizowano wpływu przestrzeni na redukcję cieni, która powinna być jedną z cech dobrej segmentacji. Przeprowadzone pomiary czasu konwersji pokazują, że z punktu widzenia wydajności systemu najlepiej byłoby zrezygnować z konieczności konwersji i prowadzić obliczenia w takiej przestrzeni barw, w jakiej obraz dostarcza kamera lub *frame-grabber*. Dlatego też w dalszych badaniach zdecydowano się wykorzystać obraz w RGB.

5. Testowanie algorytmów

Najczęściej spotykaną w literaturze metodą testowania algorytmów generacji tła jest analiza wyników segmentacji obiektów ruchomych [5]. Porównuje się maskę zwracaną przez algorytm z rezultatem segmentacji ręcznej (wzorcem, *ground truth*). Na tej podstawie wprowadza się podział na piksele poprawnie sklasyfikowane jako obiekty (TP – *true positives*), piksele poprawnie sklasyfikowane jako tło (TN – *true negatives*), piksele będące tłem, a sklasyfikowane jako obiekty (FP – *false positives*) oraz piksele będące obiektami, a sklasyfikowane jako tło (FN – *false negatives*). Następnie wyznaczane są współczynniki będące podstawą do porównywania (przykłady i szersze omówienie zagadnienia w [5]).

Opisane podejście ma jedną niezaprzeczną zaletę: jest to metoda, która w obiektywny sposób pozwala porównać dwa lub więcej algorytmów generacji tła. Jednak wykorzystanie metodologii ma też szereg wad. Po pierwsze podejście niejako wymusza dostosowywanie parametrów algorytmu do konkretnej sekwencji. Aspekt ten jest zazwyczaj pomijany w opisach i testach, a ma bardzo duże znaczenie praktyczne. Przykładowo, z punktu widzenia aplikacji, bardziej pożądany może okazać się algorytm, który daje gorsze wyniki (w sensie opisanym powyżej), ale ma mniej parametrów i łatwiej go dostosować do różnych

scen (wewnątrz, na zewnątrz, różne warunki pogodowe i oświetleniowe). Po drugie, na ostateczny rezultat segmentacji obiektów ma jeszcze wpływ szereg czynników: szeroko rozumiane przetwarzanie końcowe (*post-processing*) np. filtracja medianowa, filtracja morfologiczna, a także indeksacja i inne operacje (np. wykorzystanie przepływu optycznego). Zagadnienie to omówione zostało w pracy [12].

Uwzględniając wszystkie powyższe czynniki, zdecydowano się na poprzestaniu na jakościowej ocenie algorytmów (cechy z rozdz. 2), uzupełnionej o przedstawienie obserwacji poczynionych w trakcie testów. Wykorzystano szereg sekwencji, z których część należy uznać za wymagające, z uwagi na dynamicznie zmieniające się oświetlenie, obecność ruchu w tle (ruszające się na wietrze krzewy) i występujące cienie.

Do testów wybrano następujące algorytmy: średnia, mediana i maksimum z bufora N ramek (metody nierekurencyjne) oraz sigma-delta (SD), pojedynczy rozkład Gaussa (SG), klasteryzacja (CL) i wielokrotny rozkład Gaussa (MOG) (metody rekurencyjne). Implementację metod, z wyłączeniem MOG, wykonano samodzielnie, w języku C++ z wykorzystaniem biblioteki Open CV [11]. Dla metody MOG wykorzystano dostępną w bibliotece OpenCV implementację. Wszystkie algorytmy działały w przestrzeni barw RGB. Większość zaprezentowanych wyników wymaga analizy maski obiektów ruchomych. Dla metod z jednym wariantem tła maska taka wyznaczana była zgodnie z równaniem:

$$\text{maska} = (|R_r - R_t| + |G_r - G_t| + |B_r - B_t|) > \text{próg} \quad (4)$$

gdzie:

- RGB_r – piksel z aktualnej ramki,
- RGB_t – piksel z modelu tła,
- próg – empirycznie ustalona wartość progów binaryzacji.

Dla metod CL i MOG wykorzystano wbudowany w algorytm mechanizm, który za piksele należące do obiektów ruchomych uznaje takie, dla których nie uda się dopasować żadnego wariantu tła.

5.1. Uwagi ogólne

Analizując cechy algorytmu generacji tła, takie jak odporność na szum, odporność na ruch kamery oraz odporność na automatyczną zmianę nastaw kamery, należy stwierdzić, że zagadnienia te powinny być rozwiązane poza algorytmem generacji tła. Szum można próbować zredukować filtracją wstępną, regulacją parametrów kamery lub ustawieniem progów lub progów wykorzystywanych przy segmentacji. Wykrywanie i kompensacja ruchu lub drżenia kamery, to złożone zagadnienie, które raczej wymaga osobnego algorytmu (programowa stabilizacja obrazu). Powinien on współpracować z modułem tła (np. w przypadku, gdy zmiana położenia jest trwała wymusić szybką aktualizację sceny).

Automatyka kamery (wzmocnienie, balans bieli itp.) powoduje w wielu przypadkach błędne działanie algorytmów generacji tła, co zostało stwierdzone podczas prowadzonych

eksperymentów (część sekwencji wideo nagrywana była kamerą telewizyjną). Przykładowo automatyczna korekcja wzmocnienia powoduje nagłe rozjaśnienie lub zaciemnienie sceny, niezwiązane z faktyczną zmianą oświetlenia, a z pojawieniem się w kadrze dużego obiektu (np. samochodu). W praktycznej aplikacji powinny być wykorzystane kamery, w których da się albo automatyką sterować (współpraca z modułem generacji tła) albo ją wyłączyć.

5.2. Inicjalizacja tła

W warunkach normalnej eksploatacji systemu monitoringu przestrzeni publicznej trzeba założyć, że inicjalizacja tła odbywać się będzie w warunkach obecności na scenie obiektów ruchomych. Zatem dobry algorytm generacji tła powinien być w stanie sobie z taką sytuacją poradzić. Większość metod (przegląd w pracy [5]) zakłada, że tło w danej lokalizacji pojawia się przez więcej niż 50% czasu trwania sekwencji uczącej. Wtedy wyznaczenie np. mediany z sekwencji zapewnia poprawne działania. W sytuacji, gdy to założenie nie jest spełnione, konieczne jest albo dokonanie bardziej zaawansowanej analizy zmian wartości pikseli w czasie, albo wykorzystanie kontekstu przestrzennego [1].

W praktycznej aplikacji interesujące są dwa aspekty związane z inicjalizacją: czy sam algorytm generacji tła zapewnia stworzenie poprawnego tła po uruchomieniu, co oznacza, że osobny algorytm inicjalizacyjny nie jest potrzebny, oraz jak długo to trwa. Metoda średniej z bufora nie zawiera mechanizmu inicjalizacji, również sposobem działania nie wspiera poprawnej inicjalizacji. Każda wartość piksela w buforze ma wpływ na aktualny model tła. Zatem, w zależności od parametrów (rozmiaru bufora N oraz analizy co K -tej ramki), uzyskuje się różne przybliżenia tła, jednak nigdy tło nie jest „w pełni” poprawne (oczywiście trzeba wykluczyć sytuację, w której przez $N \times K$ klatek tło się nie zmienia – wtedy algorytm zadziała poprawnie). W przypadku mediany z bufora N ramek uzyskiwane tło jest lepsze, szczególnie dla dużych rozmiarów parametru N . W skład modelu tła wchodzi tylko piksele występujące na sekwencji (brak wartości pośrednich), a mediana „premiuje” wartości pikseli, które występują często. Niestety wykorzystanie dużego bufora ma poważną wadę w postaci zbyt wolnej adaptacji do nagłych zmian oświetlenia. Trzecia analizowana metoda, maksimum z bufora N ramek, ma właściwości podobne do metody medianowej.

Podsumowując analizę fazy inicjalizacji, w przypadku metod nierekurencyjnych należy stwierdzić, że w pewnych warunkach metody mediana i maksimum pozwalają na poprawną inicjalizację tła, jednak konieczny w tym przypadku duży rozmiar bufora powoduje, że algorytm wolno adaptuje się do nagłych zmian oświetlenia.

Dla metod rekurencyjnych z pojedynczym wariantem tła: sigma-delta i pojedynczy rozkład Gaussa można zaobserwować zachowanie podobne jak przy średniej z bufora. W przypadku stałej obecności ruchu na scenie, żadna z metod nie jest w stanie wygenerować w pełni poprawnego modelu tła, zatem nie rozwiązuje problemu inicjalizacji.

Algorytmy wielowariantowe (CL i MOG) mają „wbudowany” mechanizm inicjalizacji tła. Polega on na przypisywaniu wagi każdemu z wariantów tła dla danej lokalizacji na

obrazie (piksele modelowane są niezależnie). Na podstawie wagi można określić, który z wariantów występuje wystarczająco długo (często), aby zostać uznany za tło. W obu metodach decyduje o tym próg. Przeprowadzone eksperymenty potwierdzają, że mechanizm inicjalizacji tła, zawarty w metodach wielowariantowych, działa poprawie. Wariant tła, któremu przypisana jest największa waga (a zatem można uznać go za najbardziej reprezentatywny), po dość krótkim czasie, rzędu kilku sekund, staje się wolny od obiektów ruchomych. Oczywiście, czas ten zależy od dwóch parametrów obecnych w obu algorytmach: progu, który decyduje o tym, czy dany piksel zostaje uznany za należący do konkretnego wariantu, oraz współczynnika wtapienia w tło (α – równanie (2)).

Podsumowując badania na temat inicjalizacji tła, należy stwierdzić, że najlepsze wyniki uzyskano dla algorytmów wielowariantowych. Wynika to z własności algorytmu, którego integralną częścią jest prowadzona w trakcie obliczeń jawna analiza częstości występowania pikseli o określonych wartościach jasności (koloru). Poprawne wyniki dają również metody mediana i maksimum z bufora N ramek. Jednak w tym przypadku bardzo istotny jest rozmiar bufora. Pozostałe analizowane metody nie rozwiązują problemu inicjalizacji, zatem w praktycznym zastosowaniu należałoby rozważyć dodanie osobnego algorytmu, który będzie odpowiadał za stworzenie pierwszego, poprawnego modelu tła.

5.3. Zmiany oświetlenia

Podczas pracy algorytmu generacji tła występują dwa rodzaje zmian oświetlenia: płynne i gwałtowne, dynamiczne. Przykładem pierwszych z nich jest stopniowa zmiana natężenia światła związana z porą dnia. Tego typu zakłócenia nie powodują błędnego działania metod, ponieważ w każdej z nich istnieje mechanizm kompensacji. W algorytmach nierekurencyjnych jest to ograniczenie analizowanej historii do N ostatnich ramek, a w rekurencyjnych aktualizacja zgodnie z równaniem (2) lub (3).

Dynamiczne zmiany oświetlenia stanowią duże wyzwanie dla twórców algorytmów generacji tła i mogą stanowić przyczynę poważnych błędów w segmentacji. Takie zmiany występują przykładowo podczas częściowego zachmurzenia w wietrzny dzień lub w sytuacji, gdy załączone lub wyłączone zostanie światło w pomieszczeniu. Zatem każdy system monitoringu, poza takimi, które działają w bardzo specyficznych, kontrolowanych warunkach, powinien taką sytuację uwzględniać.

Metody z jednym wariantem tła, nierekurencyjne i rekurencyjne, nie mają mechanizmu kompensacji szybkich zmian oświetlenia. Eliminacja tego typu zakłócenia musi odbywać się poprzez zwykły mechanizm adaptacji (zaprojektowany dla powolnych zmian). W trakcie tej adaptacji wyniki segmentacji są niepoprawne – najczęściej większość pikseli na obrazie oznaczona jest jako ruchome. Na czas adaptacji wpływają parametry metody: rozmiar bufora N lub współczynnik α . Z punktu widzenia odporności na omawianą sytuację korzystne są: mały bufor lub duże α , czyli szybka adaptacja. Niestety w trakcie normalnego działania takie parametry powodują zbyt szybkie wtapienie się obiektów w tło. Podczas implementacji algorytmu warto zatem rozważyć dodanie mechanizmów modyfikacji

współczynnika odpowiadającego za aktualizację. Przykładowo w chwili zmiany oświetlenia (wykrytej jako nagły wzrost liczby obiektów po segmentacji) można tło stworzyć od nowa lub szybko aktualizować, a następnie wrócić do poprzednich ustawień. Podejście to jednak powoduje, że do tła przenikają obecne na scenie obiekty ruchome.

W sytuacji nagłej zmiany oświetlenia metody wielowariantowe działają w odmienny sposób. Zamiast dostosowywania aktualnego wariantu, tworzony jest nowy, dla inaczej oświetlonej sceny. W przypadku gdy zmiana wystąpiła jednokrotnie (przykładowo włączenie światła), rezultaty segmentacji i czas adaptacji są podobne jak dla metod z jednym wariantem, natomiast dla wielokrotnych (ew. powtarzających się) zmian, warianty tła powinny być przełączane, co zapewnia szybką adaptację. Aby sprawdzić poprawność tej tezy, przeprowadzono eksperyment na dwóch sekwencjach: prostszej, nagranej wewnątrz budynku, w trakcie nagrania światło było gaszone, a następnie załączane, oraz trudniejszej, w warunkach zewnętrznych przy trwających dynamicznych zmianach oświetlenia słonecznego.

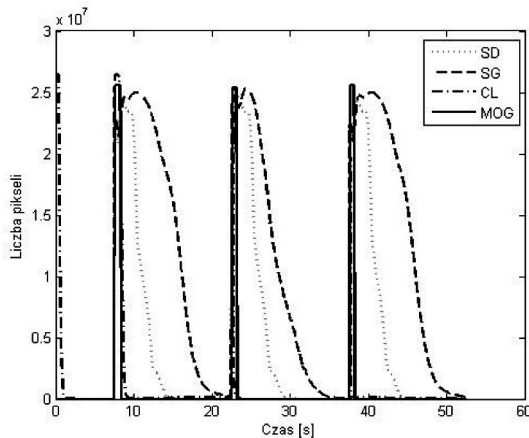
Do testów wybrano algorytmy: sigma-delta (SD), pojedynczy rozkład Gaussa (SG), klasteryzację (CL) i wielokrotny rozkład Gaussa (MOG). Metody nierekurencyjne pominięto, ponieważ nawet dla małego rozmiaru bufora $N = 25$ adaptacja następowała znacznie wolniej niż dla pozostałych metod. Na potrzeby eksperymentu należało przyjąć pewne parametry algorytmów (każdy z nich ma kilka), w taki sposób, aby maski obiektów ruchomych były podobne. Przy czym starano się zbytnio nie dostrajać parametrów do danej sekwencji. Przykładowo, dla prostego testu z włączeniem i wyłączeniem światła, gdy na scenie nie ma obiektów ruchomych, najbardziej korzystne są ustawienia zapewniające szybką (w zasadzie natychmiastową) adaptację. Dla większości typowych sekwencji takie zachowanie algorytmu uznawane jest za niekorzystne. Dlatego też celem eksperymentu było głównie pokazanie różnicy w działaniu pomiędzy metodami z jednym wariantem, a wieloma wariantami tła, przy pewnych arbitralnie wybranych parametrach (tab. 2).

Tabela 2
Parametry algorytmów wykorzystane w eksperymencie

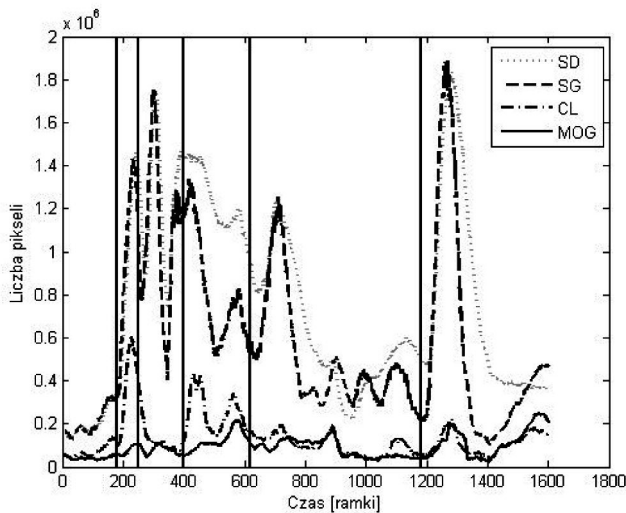
Algorytm	Parametry:
SG	$\alpha = 0,02$, Th = 35 (próg binaryzacji)
SD	Th = 35 (próg binaryzacji)
CL	$\alpha = 0,02$, $K = 3$ (liczba wariantów tła) Th = 35 (próg dopasowania do wariantu)
MOG	$\alpha = 0,02$, $K = 3$ (liczba modeli wariantów) Th (próg binaryzacji) – wyliczany automatycznie w algorytmie

W eksperymencie zliczano liczbę pikseli wykrytych jako punkty obiektów i analizowano przebieg tej wielkości w czasie. Na rysunku 3 zaprezentowano przebieg przy trzech zmianach oświetlenia.

Analiza rysunku 3 pozwala zauważyć, że metody SD i SG (jednowariantowe), przy każdej zmianie oświetlenia, muszą na nowo stworzyć model tła. Proces ten trwa od 7 do 10 sekund. Metody wielowariantowe (CL, MOG) dużo szybciej tworzą (pierwsza zmiana oświetlenia), a następnie przełączają się (kolejne zmiany) pomiędzy wariantami tła. Adaptacja do nowych warunków zajmuje mniej niż 1 sekundę.



Rys. 3. Analiza działania algorytmów w przypadku nagłej zmiany oświetlenia



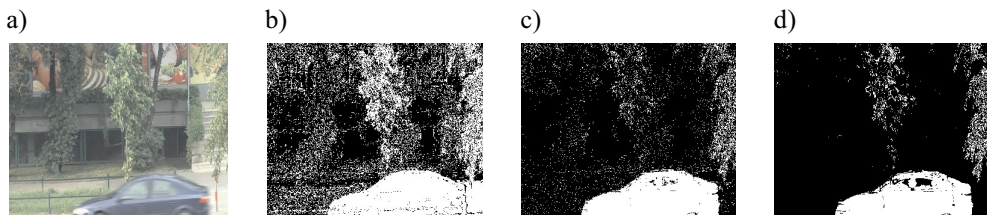
Rys. 4. Analiza działania algorytmów dla sekwencji z dynamicznymi zmianami oświetlenia. Pionowymi kreskami oznaczono chwile zmian oświetlenia

Druga z sekwencji testowych została nagrana w warunkach dynamicznie zmieniającego się oświetlenia słonecznego. Parametry algorytmów dobrano tak, aby w rezultacie uzyskać porównywalne maski obiektów ruchomych. Na rysunku 4 przedstawiono wynik

eksperymentu. Należy szczególną uwagę zwrócić na działanie metod po zmianach oświetlenia (oznaczonych pionowymi kreskami). Algorytmy SD oraz SG, których działanie oparte jest na pojedynczym wariancie tła, muszą dostosowywać się na nowo po każdej zmianie oświetlenia. Natomiast w przypadku CL i MOG można zaobserwować mechanizm przełączania pomiędzy wariantami, który skutkuje szybszą adaptacją. We wszystkich metodach przełączanie nie jest natychmiastowe, co wynika po pierwsze z płynnych zmian oświetlenia, a po drugie z różniącą się w kolejnych fragmentach sekwencji średniej jasności sceny.

5.4. Tło multimodalne

W literaturze mianem tła multimodalnego określa się obecność drobnego ruchu w tle: kołyszące się drzewa lub krzewy, płynąca woda itp. Zdolność do poprawnej pracy w takich warunkach jest zwykle wymieniana jako podstawowa zaleta algorytmów wielowariantowych [18, 4]. W celu zweryfikowania tej tezy wybrano sekwencję, na której widoczne są dość intensywnie ruszające się (kołyszące się) na wietrze krzew liściasty i drzewo (rys. 5a). Następnie przeanalizowano rezultaty segmentacji obiektów ruchomych generowane przez algorytmy.



Rys. 5. Wynik testu metod dla tła multimodalnego: a) scena; b) metody z jednym wariantem tła; c) metoda CL; d) metoda MOG

Dla testowanych metod z jednym wariantem tła wynik segmentacji był podobny (rys. 5b). Należy stwierdzić, że algorytmy z tej grupy zawodzą w przypadku tego typu tła, generując wiele, de facto, fałszywych obiektów, których eliminacja na etapie prostego przetwarzania końcowego jest raczej niemożliwa. Dla metody CL (rys. 5c) i MOG (rys. 5d) wyniki są znacznie lepsze, jednak ruch krzewu po prawej stronie, wzdłuż pionowej krawędzi ramki obrazu nadal jest wykrywany. Dla tej konkretnej sekwencji (którą należy uznać za bardzo trudną), nie udało się dobrać takich parametrów algorytmu, przy których drobny ruch byłby całkowicie wyeliminowany. Jednakże wykorzystanie np. erozji morfologicznej pozwala w większości wyeliminować zakłócenia.

Dodatkowo, podczas testów zauważono przewagę metody MOG nad CL. Ma to związek z mechanizmem automatycznego progowania, który jest wbudowany w algorytm. Próg wyznaczany jest na podstawie wariancji, co oznacza, że dla pikseli, których wartości często się zmieniają (np. liście krzewu), ma on dużą wartość i w ten sposób eliminowana jest część ruchu. Podsumowując ten etap badań, należy stwierdzić, że przypadek tła multimodalnego

pokazuje największą przewagę metod z wieloma wariantami tła nad metodami z jednym wariantem.

5.5. Uwagi dotyczące generacji tła

Podczas przeprowadzonych testów dokonano pewnych obserwacji, które wydają się istotne z punktu widzenia wyboru, implementacji i testowania algorytmu generacji tła. Po pierwsze ważny jest dobór sekwencji wideo. W niniejszej pracy zdecydowano się wykorzystywać „trudniejsze” sekwencje, zarejestrowane w warunkach zewnętrznych i zmiennego oświetlenia. Dodatkowo stwierdzono, że istotny jest kierunek ruchu obiektów (osób, samochodów) względem ustawienia kamery. Jeżeli jest on w przybliżeniu prostopadły do osi optycznej kamery, to zjawisko wtapienia się obiektów w tło występuje na dużo mniejszą skalę niż w przypadku ruchu wzdłuż osi (piksele obiektu mają niezmienną wartość w trakcie analizy większej liczby ramek). Zatem lepsze, z punktu widzenia oceny algorytmu, są sceny wskazane jako drugie, szczególnie, że tego typu kadr dominuje w rzeczywistych systemach monitoringu wizyjnego.

Dla metod wielowariantowych zaobserwowano, że obecność na scenie osób podobnie ubranych (np. na czarno) powoduje powstawanie jednego wariantu tła odpowiadającego dominującemu kolorowi. Nie wpływa to wprawdzie negatywnie na „główny” wariant tła, ale sprawia, że segmentacja jest niepoprawna, gdyż w metodach CL i MOG za obiekty ruchome uznawane są tylko piksele, które nie pasują do żadnego z wariantów. Eliminacja tego zjawiska, bez wykorzystania innego rodzaju informacji: (odejmowanie ramek, przepływ optyczny) lub sprzężenia zwrotnego od modułów analizy (semantyka sceny), wydaje się niemożliwa. Zatem o ile metody z jednym wariantem tła mają tendencję do generowania maski obiektów ruchomych większej niż rzeczywiste obiekty, głównie z uwagi na „śląd” pozostawiany w tle, o tyle metody wielowariantowe generują niepełne maski.

5.6. Ocena końcowa algorytmów

W tabeli 3 zestawiono oceny poszczególnych algorytmów w skali 0–5.

Tabela 3
Ocena testowanych algorytmów

Zagadnienie	Nierekurencyjne			Rekurencyjne			
	średnia	mediana	maksimum	SD	SG	CL	MOG
Inicjalizacja	2	3	3	2	2	4	4
Płynne zmiany oświetlenia	5	5	5	5	5	5	5
Dynamiczne zmiany oświetlenia	1	1	1	1	1	3	3
Tło multimodalne	0	0	0	0	0	4	5

Algorytmy wykorzystujące więcej niż jeden wariant tła wydają się mieć przewagę nad metodami z jednym wariantem. Zapewniają mechanizm inicjalizacji, w znaczącym stopniu eliminują problem ruchu w tle oraz lepiej sprawdzają się w zmiennych warunkach oświetlenia. Oczywiście wiąże się to z w przybliżeniu K -krotnie (gdzie K oznacza liczbę wariantów tła) większą złożonością obliczeniową i pamięciową w porównaniu z metodami jednowariantowymi. Podczas testów nie stwierdzono dużej przewagi metod nierekurencyjnych nad rekurencyjnymi (jedynie lepsze rezultaty osiągają w przypadku inicjalizacji), zatem ich przydatność w systemach monitoringu jest raczej niewielka. Najlepszą metodą wydaje się MOG. Algorytm ten jest na pewno najpopularniejszy i powstało na jego temat najwięcej prac (150) i ulepszeń [3]. Jednakże jest też najbardziej złożony obliczeniowo (a większość modyfikacji i ulepszeń tą złożoność jeszcze zwiększa) i jego działanie zależy od kilku parametrów, które trzeba uważanie dobrać. Metoda CL, która de facto jest uproszczeniem MOG, generuje wyniki nieznacznie gorsze, przy mniejszej złożoności obliczeniowej.

6. Selektywna aktualizacja tła

W dotychczasowych rozważaniach zakładano, że aktualizacja tła będzie dokonywana dla całej sceny. Podejście takie powoduje, że do modelu tła przenikają elementy obiektów ruchomych. Zjawisko to występuje w różnym stopniu w zależności od metody generacji tła i jej parametrów (np. α z równania (2)). Ponieważ zjawisko zbyt szybkiego przenikania obiektów do tła może być uznane za niekorzystne, można mu przeciwdziałać, stosując selektywną aktualizację tła tylko dla pikseli, dla których istnieje wysokie prawdopodobieństwo, że są tłem. Problemem jest wybór kryterium aktualizacji. W literaturze przedmiotu nie występuje jedno, powszechnie przyjęte i skuteczne rozwiązanie.

Najprostsze podejście polega na wykorzystaniu do aktualizacji tła maski obiektów ruchomych powstałej z odejmowania aktualnej ramki obrazu od modelu tła. Niestety jest ono bardzo podatne na błędy w modelu tła, które dodatkowo, jeżeli raz powstaną, nie są eliminowane. Przykładem jest zjawisko określane w literaturze jako *ghost* (duch), czyli sytuacja w której stacjonarny obiekt (człowiek, samochód), będący elementem tła, w pewnej chwili zaczął się ruszać. W miejscu gdzie uprzednio był obiekt, powstaje błąd w tle (*ghost*), który zostanie wykryty jako element ruchomy. Zatem tło w tym miejscu nie będzie aktualizowane i błędna segmentacja będzie propagowana w czasie. Warto dodać, że źródłem błędów mogą też być zmiany oświetlenia, ruch w tle itp. Aby zapobiegać tego typu sytuacjom, zaproponowano rozmaite heurystyki, np. zliczanie ile razy piksel został sklasyfikowany jako obiekt ruchomy i jeżeli wartość ta przekracza ustalony próg, poddanie go aktualizacji, analizy czy piksel nie jest naprzemiennie sklasyfikowany jako tło i obiekt itp.

Wydaje się jednak, że podejmowanie decyzji na temat aktualizacji tylko na podstawie informacji z generacji tła jest podejściem, które zawsze będzie podatne na błędy. Zatem celowe wydaje się wykorzystanie innego rodzaju informacji. W idealnym rozwiązaniu jej

źródłem powinien być moduł semantycznej analizy sceny. Wtedy każdy wydzielony obiekt na scenie zostałby sklasyfikowany (nazwany), a aktualizacja tła byłaby przeprowadzona tylko dla pikseli przypisanych do tła. Niestety, realizacja opisanego schematu na obecnym etapie rozwoju sztucznych systemów wizyjnych jest bardzo trudna i wymagająca obliczeniowo. Innym źródłem dodatkowej informacji może być wykrywanie ruchu na dwóch lub kilku sąsiednich ramkach. Wtedy aktualizacja tła następowałaby tylko dla pikseli, które pozostają stacjonarne w danym przedziale czasu. Podejście to nie rozwiązuje wielu problemów, np. zatrzymania się osoby, ale może zapobiegać powstawaniu „smug” w tle za poruszającymi się obiektami oraz poprawić wyniki segmentacji (integracja rezultatów uzyskanych z modułu generacji tła oraz detekcji ruchu).

W przeprowadzonych badaniach wstępnych wykorzystano dwa algorytmy detekcji ruchu: proste różnicowanie trzech sąsiednich ramek (implementacja własna w C++) oraz przepływ optyczny (*optical flow*) – implementacja algorytmu Lucasa–Kanade’a [9] w bibliotece OpenCV. Przykładowe wyniki zaprezentowano na rysunku 6. Pokazano maski binarne obiektów ruchomych, progi dobrano ręcznie, dla przepływu optycznego progowano jego amplitudę. Obliczanie różnicy sąsiednich ramek działa w sposób zbliżony do wykrywania krawędzi (rys 6b) i informacja ta nie wydaje się specjalnie przydatna, przy konstruowaniu reguły aktualizacji tła. Poprawność tezy wykazały przeprowadzone eksperymenty. Obiekty ruchome nadal praktycznie w całości przenikały do tła.

Przepływ optyczny (rys. 6c) pozwala na względnie poprawną segmentację poruszających się obiektów. Duży szum, który generuje metoda, można częściowo wyeliminować odpowiednim doбором progu binaryzacji. Przeprowadzone badania wstępne pokazały, że wykorzystanie maski ruchu, uzyskanej metodą przepływu optycznego, w znacznym stopniu zapobiega przenikaniu obiektów ruchomych do tła i tym samym poprawia rezultaty segmentacji, nawet z wykorzystaniem prostej metody SG. W innym eksperymencie zauważono, że integracja rezultatów odejmowania tła i amplitudy przepływu optycznego pozwala poprawić wynik segmentacji obiektów. Dla metody wielowariantowej CL, dodanie informacji o ruchu również powoduje polepszenie wyników działania, pozwalając zniwelować przenikanie obiektów do modelu tła, co sprawia, że po segmentacji maski obiektów są bardziej kompletne.



Rys. 6. Rezultat działania metod wykrywających ruch: a) oryginalna ramka; b) odejmowanie trzech ramek; c) przepływ optyczny Lucasa–Kanade’a

Opisane w rozdziale eksperymenty stanowią tylko wstęp do badań nad integracją informacji z odejmowania aktualnego obrazu i modelu tła oraz przepływu optycznego. Uzyskane wyniki pokazują jednak duży potencjał podejścia i warto wykonać szerszą analizę tej metody. Wykorzystanie przepływu optycznego ma jedną wadę. Jego wyliczanie jest dość złożone i czasochłonne. Przykładowo dla ramek o rozdzielczości 360×288 wykonanie trwa ok. 15 ms, co stanowi 37% procent czasu dostępnego na przetwarzanie jednej ramki w systemie działającym w czasie rzeczywistym.

7. Podsumowanie

W artykule poruszono szereg zagadnień związanych z segmentacją obiektów ruchomych za pomocą generacji i odejmowania tła. W pierwszym etapie badań pokazano, że wykorzystanie koloru pozwala poprawić wyniki segmentacji, a w niektórych sytuacjach jest wręcz niezbędne do poprawnego wykrycia obiektów. Jednakże analiza obrazu kolorowego zwiększa złożoność obliczeniową algorytmu i użyte zasoby. Przebadano kilka modeli barw, a także zmierzono, ile czasu zajmuje konwersja obrazu do wybranych przestrzeni. Wyniki eksperymentów wskazują, że przy segmentacji ważne jest, aby model barw miał oddzielne składowe dla jasności i koloru. W eksperymentach wykorzystano miarę odległości pomiędzy aktualną ramką a modelem tła, skonstruowaną na bazie przestrzeni Y_{xy} (znormalizowane RGB), uzyskując dobre wyniki.

W kolejnym etapie prac dokonano przeglądu najpopularniejszych algorytmów generacji tła. Przeprowadzono szereg testów, analizując zdolność metod do poprawnej inicjalizacji tła, w przypadku obecności obiektów ruchomych na scenie oraz działanie metod w warunkach dynamicznie zmieniającego się oświetlenia i drobnego ruchu w tle (tło multimodalne). Uzyskane wyniki (zebrane w tabeli 3) pokazują pewną przewagę metod wielowariantowych, a szczególnie metody wielokrotnego rozkładu Gaussa. Ostatni etap prac to wstępna analiza przydatności informacji o ruchu na scenie uzyskanej innymi metodami: odejmowaniem sąsiednich ramek oraz z wykorzystaniem przepływu optycznego. Otrzymane wyniki pokazują, że druga z metod ma duży potencjał i pozwala znacznie poprawić działanie, nawet w przypadku prostych algorytmów jak pojedynczy rozkład Gaussa.

Opisane prace wymagają kontynuacji. Przede wszystkim należy spróbować stworzyć schemat redukcji wpływu cieni na wyniki segmentacji. W pierwszym podejściu w przestrzeni RGB. Kolejnym problemem, który wymaga rozwiązania, jest opracowanie metod automatycznego generowania progów używanych w algorytmach. Należy też przeprowadzić osobne badania, które pozwolą odpowiedzieć na pytanie: czy lepsze wyniki można uzyskać, integrując proste metody, typu pojedynczy rozkład Gaussa lub sigma-delta, z przepływem optycznym, czy lepiej wykorzystać algorytm wielowariantowy typu MOG. Niewy-

kluczone jednak, że jednoznaczna odpowiedź nie istnieje, a algorytm należy dobierać w zależności od obserwowanej sceny (przykładowo prosty, jednowariantowy wewnątrz, a wielowariantowy na zewnątrz). Osobnym tematem badań, już częściowo prowadzonym, jest sprzętowa akceleracja algorytmów generacji tła, przykładowo przy wykorzystaniu układów FPGA. Uzyskane wyniki pokazują, że możliwa jest implementacja i uruchomienie algorytmu wielowariantowego w zasobach układu rekonfigurowalnego. Podejście takie, pozwala wykorzystać moc obliczeniową procesora na realizację dalszych etapów analizy obrazu, co w świetle zaprezentowanych wyników wydaje się bardzo uzasadnione.

Literatura

- [1] Baltieri D., Vezzani R., Cucchiara R., *Fast Background Initialization with Recursive Hadamard Transform*. Proc. of IEEE Seventh International Conference on Advanced Video and Signal Based Surveillance, 2010, 165–171.
- [2] Benedek C., Sziranyi T., *Study on color space selection for detecting cast shadows in video surveillance*. Articles. Int. J. Imaging Syst. Technol., 17, 3, 2007, 190–201.
- [3] Bouwmans T., El Baf F., Vachon B., *Background modeling using mixture of Gaussians for foreground detection. A survey*. Recent Patents Comput. Sci., 1, No. 3, 2008, 219–237.
- [4] Butler D., Sridharan S., Bove V.M. (Jr.), *Real-time Adaptive Background Segmentation*. Acoustics, Speech, and Signal Processing. Proc. (ICASSP '03). IEEE Int. Conf. on April 2003, 349–52.
- [5] Elhabian S.Y., El-Sayed K.M., Ahmed S.H., *Moving object detection in spatial domain using background removal techniques, state-of-art*. Recent Patents on Computer Science, 1, 2008, 32–54.
- [6] Hardle W., Steiger W., *Optimal Median Smoothing*. 1994.
- [7] Huang T.S., Yang G.J., Tang G.Y., *A fast two-dimensional median filtering algorithm*. IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 27(1), 1979.
- [8] Lee H.-C., *Introduction to Color Imaging Science*. Cambridge University Press, 2005.
- [9] Lucas B.D., Kanade T., *An iterative image registration technique with an application to stereovision*. Proc. of the 7th International Joint Conference on Artificial Intelligence (IJCAI'81), April 1981, 674–679.
- [10] McFarlane N.J.B., Schofield C.P., *Segmentation and tracking of piglets in images*. Machine Vision and Applications, vol. 8, No. 3, 1995, 187–193.
- [11] OpenCV, <http://sourceforge.net/projects/opencvlibrary/>.
- [12] Parks D.H., Fels S.S., *Evaluation of Background Subtraction Algorithms with Post-Processing*. Proc. of the 2008 IEEE Fifth international Conference on Advanced Video and Signal Based Surveillance, AVSS. IEEE Computer Society, Washington, 2008, 192–199.
- [13] Plataniotis K.N., Sri V., Androustos D., Venetsanopoulos A.N., *An Adaptive Nearest Neighbor Multichannel Filter*. IEEE Transactions Circuits and Systems for Video Technology, 1996, 699–703.
- [14] Plataniotis K.N., Venetsanopoulos A.N., *Adaptive filters for multichannel signal processing*. IEEE Symposium Advances in Digital Filtering and Signal Processing, 1998, 67–71.
- [15] Prati A., Mikic I., Trivedi M.M., Cucchiara R., *Detecting moving shadows: algorithms and evaluation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, No.7, July 2003, 918–923.

-
- [16] Qin R., Liao S., Lei Z., Li S.Z., *Moving Cast Shadow Removal Based on Local Descriptors*. 20th International Conference on Pattern Recognition (ICPR), 23–26, Aug. 2010, 1377–1380.
 - [17] Sanin A., Sanderson C., Lovell B.C., *Improved Shadow Removal for Robust Person Tracking in Surveillance Scenarios*. 20th International Conference on Pattern Recognition (ICPR), 23–26 Aug. 2010, 141–144.
 - [18] Stauffer C., Grimson W.E.L., *Adaptive background mixture models for real-time tracking*. Proc. IEEE CVPR 1999, 1, vol. 2, 246–252.
 - [19] Wren C.R., Azarbayejani A., Darrell T., Pentland A.P., *Pfinder: real-time tracking of the human body*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, No. 7, 1997, 780–785.