

Tomasz KRYJAK, Marek GORGONAGH AKADEMIA GÓRNICZO-HUTNICZA,
Al. Mickiewicza 30, 30-059 Kraków**Implementacja sprzętowa odejmowania tła metodą ViBe w układzie FPGA****Dr inż. Tomasz KRYJAK**

Autor jest absolwentem kierunku Automatyka i Robotyka na Akademii Górniczo-Hutniczej im. Stanisława Staszica w Krakowie (2007). Pracuje w Laboratorium Biocybernetyki Katedry Automatyki i Inżynierii Biomedycznej AGH na stanowisku adiunkta. Interesuje się przetwarzaniem i analizą obrazów, ze szczególnym uwzględnieniem zaawansowanych systemów monitoringu wizyjnego oraz sprzętową akceleracją algorytmów wizyjnych z wykorzystaniem układów FPGA. Autor ponad 20 publikacji.

e-mail: kryjak@agh.edu.pl

**Dr hab. inż. Marek GORGON**

Autor jest absolwentem Akademii Górniczo-Hutniczej w Krakowie, gdzie uzyskał kolejno dyplom magistra inżyniera elektroniki (1988) oraz stopnie doktora (1995) i doktora habilitowanego (2007), oba w dyscyplinie Automatyka i Robotyka. Pracuje w Katedrze Automatyki AGH i zajmuje się metodami i systemami do przetwarzania i analizy obrazu. Od 1992 roku jego głównym obszarem zainteresowania są systemy obrazowe oparte na układach FPGA. Jest członkiem IEEE, autorem 70 publikacji.

e-mail: mago@agh.edu.pl

**Streszczenie**

W artykule zaprezentowano implementację sprzętową nowatorskiego algorytmu odejmowania tła ViBe (ang. *Visual Background Extractor*) w układzie rekonfigurowalnym FPGA. Metoda ta opiera się na odmiennej od dotychczas opisywanych i realizowanych koncepcji modelowania tła. W pracy dokonano oceny możliwości przeniesienia algorytmu na platformę sprzętową, pokazano dwie modyfikacje, które pozwoliły poprawić działanie metody oraz omówiono zrealizowany system sprzętowy. Według wiedzy autorów jest to pierwszy opis implementacji tego algorytmu w układzie FPGA.

Słowa kluczowe: układy rekonfigurowalne FPGA, przetwarzanie i analiza obrazów, generacja tła, odejmowanie tła, segmentacja obiektów ruchomych, algorytm ViBe.

Hardware implementation of the ViBe background subtraction method in FPGA**Abstract**

This paper presents a hardware implementation in the FPGA reconfigurable device of ViBe - a novel background subtraction algorithm. The method is based on a different, from those previously described and implemented, background modelling concept. It partly uses random numbers, which allowed us to significantly reduce the buffer size in relation to the standard methods like mean or median form a buffer. A detailed description of ViBe can be found in papers [6, 7, 8]. In this paper the role of background generation algorithms in image processing and analysis systems, with particular emphasis on hardware implementations is discussed (Section 1). The ViBe algorithm is described in Section 2. Then an analysis of the possibility of implementing ViBe in FPGA is presented (Section 3). Section 4 describes two proposed modifications: the use of the CIE Lab colour space and the enhanced flashing pixels detection method. Their desirability has been confirmed quantitatively using the "ChangeDetection" database [9]. A detailed description of the designed ViBe hardware module and image processing system is presented in Section 5. The scheme of the ViBe module is shown in Figure 5 and the whole system in Figure 4. Table 3 summarizes the hardware resource utilization. The proposed solution enables the detection of objects using the method ViBe and enables real-time processing of a colour 640 x 480 video stream at 60 frames per second. The obtained results confirm the high usefulness of FPGA in the implementation of advanced image processing and analysis algorithms.

Keywords: FPGA, reconfigurable devices, image processing and analysis, background generation, background subtraction, moving object detection, ViBe algorithm.

1. Wprowadzenie

Detekcja obiektów ruchomych (obiektów pierwszoplanowych) jest jednym z najważniejszych zagadnień w przetwarzaniu i analizie sekwencji wideo. Ma zastosowanie w zaawansowanych, automatycznych systemach monitoringu wizyjnego oraz systemach nadzoru ruchu drogowego, których bardzo dynamiczny rozwój można obserwować w ostatnich latach.

Metody te można podzielić na trzy kategorie: oparte o wyznaczanie różnic pomiędzy kolejnymi ramkami, oparte o tzw. modelowanie tła oraz wykorzystujące przepływ optyczny. W tej pracy przedstawiono sprzętową implementację metody należącej do drugiej, z wymienionych kategorii. Obszerne omówienie różnych podejść do detekcji obiektów można odnaleźć w pracy [1].

Koncepcja odejmowania tła zakłada wykrywanie obiektów na podstawie obrazu różnicowego pomiędzy aktualnym widokiem sceny, a tłem, gdzie przez tło rozumiana jest pusta scena tj. bez obiektów będących przedmiotem zainteresowania (na przykład pojawiających się ludzi, samochodów). Warto zwrócić uwagę, że tło może zawierać elementy ruchome: płynącą wodę, ruszające się liście i krzewy, które nie powinny być wykrywane. Reprezentacja tła powinna podlegać modyfikacjom, aby dostosowywać się do pewnych naturalnie występujących zmian np. oświetlenia, przemieszczenia pewnych przedmiotów, co określa się mianem generacji lub modelowania tła.

W literaturze można odnaleźć wiele opisów implementacji algorytmów generacji tła w układach FPGA. Obszerne omówienie zagadnienia przedstawiono w pracy [2]. Do ważniejszych artykułów, w których w ostatnim czasie zaprezentowano sprzętową realizację algorytmów, można zaliczyć:

- MOG [3] - przetwarzanie strumienia HD w odcieniach szarości (1920 x 1080 @ 20 fps),
- Horparserta [4] - przetwarzanie strumienia 1024 x 1024 @ 32,8 fps, częściowe wykorzystanie języka wysokiego poziomu Impulse-C,
- Codebook [5] - przetwarzanie strumienia 768 x 576 @ 60 fps,
- Clustering [2] - przetwarzanie kolorowego strumienia HD.

Sprzętowa realizacja algorytmów generacji tła może zostać wykorzystana w akceleratorach sprzętowych (np. w frame-grabberze z układem FPGA, który realizuje pewne operacje wstępnej przetwarzania i analizy obrazu) lub w kamerach inteligentnych (ang. *smart cameras*), gdzie całe przetwarzanie, analiza i rozpoznawanie obrazu realizowane jest w kamerze, a transmitowane są jedynie rezultaty.

2. Algorytm ViBe

Algorytm segmentacji obiektów pierwszoplanowych ViBe (ang. *Visual Background Extractor*) został zaproponowany przez O. Barnicha i M. Van Droogmbroecka i opisany w pracach [6, 7, 8]. Zawiera on kilka nowatorskich elementów (rozwiązanie jest opatentowane) oraz pozwala na uzyskanie bardzo dobrych wyników segmentacji, co jest potwierdzone wysokim miejscem w rankingu algorytmów detekcji obiektów [9].

Model tła w algorytmie ViBe zbudowany jest z zbioru zaobserwowanych wartości piksela. Jest to istotna różnica w stosunku do najbardziej rozpowszechnionych metod, gdzie model tła oparty jest o rozkład prawdopodobieństwa. Autorzy algorytmu ViBe wybór ten uzasadniają trudnością z właściwym doбором rozkładu prawdopodobieństwa oraz jego odpowiednim uaktualnianiem.

Niech $v(x)$ oznacza wartości piksela w danej przestrzeni barw w punkcie x na obrazie, a v_i i -tą próbkę z modelu tła. Wtedy model dla każdego piksela x określany jest jako zbiór N próbek:

$$M(x) = \{v_1, v_2, \dots, v_N\} \quad (1)$$

W celu dokonania klasyfikacji piksela $v(x)$ definiowana jest sfera $S_r(v(x))$ o promieniu R ze środkiem w punkcie $v(x)$. Analizowany piksel jest uznawany za tło, jeżeli co najmniej $\#_{\min}$ próbek z modelu $M(x)$ leży wewnątrz sfery. Odległość określona jest jako euklidesowa, a sama procedura wymaga, w najgorszym przypadku, wykonania N obliczeń odległości oraz N porównań.

Autorzy metody zaproponowali mechanizm inicjalizacji modelu tła na podstawie jednej ramki. Za takim podejściem przemawia szybka możliwość realizowania segmentacji po uruchomieniu systemu i prosta reinicjalizacja, przydatna w przypadku wystąpienia np. nagłej zmiany oświetlenia sceny.

W tego typu podejściu, nie jest dostępny kontekst czasowy (historia piksela), dlatego poczyniono założenie, że sąsiednie piksele powinny mieć podobne wartości. Procedura inicjalizacyjna polega na wypełnianiu bufora $M(x)$ wybranymi losowo próbkami z otoczenia rozpatrywanego piksela (rozmiar otoczenia 3×3). Wadą takiego rozwiązania, jest podatność na występowanie artefaktów w postaci "duchów", czyli zbiorów pikseli sklasyfikowanych jako należące do pierwszego planu, ale faktycznie nie związanych z żadnym obiektem. Eliminację takich zakłóceń zapewnia omówiony poniżej mechanizm aktualizacji modelu tła.

W algorytmie ViBe zastosowano konserwatywne podejście do zagadnienia aktualizacji - model tła podlega modyfikacji tylko w przypadku zaklasyfikowania piksela jako elementu tła. Z jednej strony zapobiega to przenikaniu elementów obiektów ruchomych do modelu tła, ale jednocześnie może prowadzić do wystąpienia nieusuwalnych błędów segmentacji (np. "puste" miejsce parkingowe po samochodzie, który odjechał zostanie zakwalifikowane jako obiekt).

W odróżnieniu od popularnych algorytmów generacji tła, które wykorzystują bufor pikseli (średnia z bufora, mediana z bufora), w których w trakcie aktualizacji najstarsza próbka zostaje zastąpiona nową wartością (wykorzystanie FIFO), w metodzie ViBe kontekst czasowy nie jest uwzględniany. Próbka, która ma zostać uaktualniona, wybierana jest w sposób losowy. W połączeniu z konserwatywnym podejściem powoduje to, że "czas życia" próbki w modelu tła opisany jest zależnością eksponencjalną. W celu dalszego rozszerzenia przedziału czasowego, który opisuje model tła, aktualizacja przeprowadzana jest z ustalonym prawdopodobieństwem (np. 1/16).

W celu przeciwdziałania negatywnym skutkom założonego podejścia konserwatywnego, w algorytmie ViBe zaproponowano mechanizm, który zapewnia uaktualnianie modeli sąsiadujących z aktualnie rozpatrywanym. W przypadku, gdy bieżący piksel $v(x)$ zostanie uznany za należący do tła, uruchamiane są dwie procedury aktualizacji: bieżącego i sąsiedniego modelu tła. Na wstępie, w sposób losowy określane jest, czy dana aktualizacja ma mieć miejsce (prawdopodobieństwo zaproponowane przez Autorów - 1/16). Następnie, w pierwszym przypadku losowana jest próbka do podmiany (1 z N), a w drugim, najpierw określany jest model (1 z 8, przy założeniu otoczenia o rozmiarze 3×3), a następnie próbka do zamiany (1 z N). Wybrane próbki zastępowane są wartością $v(x)$.

Warto podkreślić, że metoda ViBe wymaga ustawienia bardzo niewielu parametrów. Autorzy zaproponowali następujące ich wartości: $N = 20$ (liczba próbek w modelu), $R = 20$ (promień sfery, wartość dla analizy w odcieniach szarości), $\#_{\min} = 2$ (minimalna liczba próbek, które muszą leżeć wewnątrz sfery) oraz prawdopodobieństwo aktualizacji (1/16).

3. Analiza możliwości implementacji sprzętowej algorytmu ViBe

Jednym z podstawowych problemów przy sprzętowej implementacji algorytmów generacji tła jest zapewnienie szybkiego

dostępu do zewnętrznej pamięci RAM, w której przechowywany jest model tła [2]. W przypadku algorytmu ViBe konieczne jest zapewnienie następującego transferu:

$$T = N \times B \times PC \times 2 \quad (2)$$

gdzie: N - rozmiar modelu (liczba próbek), B - liczba bitów przypadająca na jedną próbkę (dla obrazów w odcieniach szarości $B=8$, dla RGB $B=24$, dla CIE Lab $B=23$), PC - zegar piksela (dla rozdzielczości 640×480 - VGA - wynosi on 25 MHz). Zastosowanie mnożnika dwa, wynika z konieczności dokonania zapisu i odczytu. Podstawiając odpowiednie wartości otrzymuje się wartość parametru $T \cong 690$ MB/s dla odcieni szarości i $T \cong 2070$ MB/s dla RGB. W przypadku pracy z rozdzielczością HD (tj. 1920×1080 , zegar piksela 148,5 MHz) wartości te wynoszą odpowiednio: $T \cong 4898$ MB/s i $T \cong 12293$ MB/s.

Współczesne karty z układem FPGA wyposażone są zwykle w zewnętrzną pamięć DDR3 RAM. W niniejszej pracy brano pod uwagę dwie platformy: ML605 (układ Virtex 6) oraz VC707 (układ Virtex 7), obie firmy Xilinx. Pierwsza z nich wyposażona jest w pamięć o maksymalnej, teoretycznej przepustowości 6400 MB/s, a druga 12800 MB/s. Przy czym, w przypadku pamięci dynamicznej, niemożliwe jest uzyskanie wartości maksymalnych, z uwagi na konieczność realizacji odświeżania oraz dostępu do poszczególnych banków i kolumn.

Analiza zaprezentowanych liczb pozwala sformułować następujące wnioski. Algorytm dla rozdzielczości VGA można zaimplementować na obu platformach sprzętowych. W przypadku rozdzielczości HD możliwe jest jedynie zrealizowanie wersji działającej w odcieniach szarości na nowszej platformie VC707.

Metoda ViBe może zostać dość łatwo zrealizowana w sprzęcie. Obliczanie odległości pomiędzy pikselem a próbkami w modelu jest możliwe do implementacji w sposób równoległy. Pozostałe operacje, w tym generacja liczb pseudolosowych, również są wykonalne. Dość złożonym elementem jest jedynie mechanizm propagacji pikseli do sąsiednich modeli tła, który wymaga wygenerowania bardzo dużego kontekstu oraz dużej liczby linii opóźniających.

4. Zaproponowane modyfikacje algorytmu ViBe

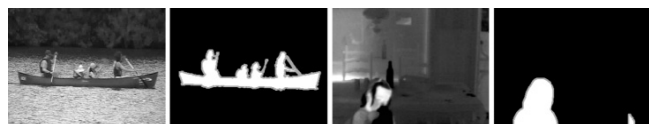
W pierwszym etapie prowadzonych badań szczegółowo przeanalizowano artykuł [7], w którym Autorzy algorytmu ViBe proponują szereg zmian poprawiających działanie metody. Niestety implementacja większości z nich w zasobach rekonfigurowalnych okazała się trudna lub niemożliwa do zrealizowania w systemie przetwarzającym dane w sposób potokowy. Bardziej szczegółowo rozpatrzono jednak dwa pomysły: zmianę przestrzeni barw oraz mechanizm pozwalający na redukcję fałszywych detekcji w obszarach o niewielkim ruchu (np. płynąca woda).

W celu przebadania wskazanych ulepszeń skonstruowano model algorytmu w języku C++ z wykorzystaniem biblioteki OpenCV [10]. Jako zbiór testowy użyta została baza filmów, wykonana na potrzeby warsztatów *IEEE Workshop on Change Detection* [9]. Zawiera ona 6 kategorii sekwencji wideo: podstawowa, dynamiczne tła (np. płynąca rzeka), drgania kamery, obiekty pozostawione lub zatrzymujące się na scenie, cienie oraz obrazy termowizyjne. W każdej z nich znalazło się od 4 do 6 filmów. Można stwierdzić, że na sekwencjach występuje duża część sytuacji, które są problematyczne dla algorytmów generacji tła. Jednakże podstawową zaletą bazy, i tym co wyróżnia ją od innych zbiorów (np. Wallflower [11]), jest duża liczba ramek referencyjnych z ręcznie zaznaczonymi obszarami: tła, cienia, nieznacznego ruchu i rozmazania oraz ruchu obiektów pierwszoplanowych. Pozwala to na wiarygodną ocenę działania algorytmów w różnych sytuacjach. Przykładowe obrazy zamieszczono na rysunku 1.

Metodologia eksperymentów była następująca. Porównywano maskę obiektów zwracaną przez algorytm z maską referencyjną. Ponieważ w algorytmie ViBe nie wbudowanego mechanizmu

wykrywania cienia, uwzględniano tylko klasyfikację na tło lub obiekt. Zliczano:

- TP (*true positive*) - piksel należący do obiektu z pierwszego planu wykrywany jako piksel należący do obiektu z pierwszego planu,
- TN (*true negative*) - piksel należący do tła wykrywany jako piksel należący do tła,
- FP (*false positive*) - piksel należący do tła wykrywany jako piksel należący do obiektu z pierwszego planu,
- FN (*false negative*) - piksel należący do obiektu z pierwszego planu wykrywany jako piksel należący do tła.



Rys. 1. Przykłady obrazów użytych do badań. Lewa kolumna obraz oryginalny, prawa kolumna referencyjna maska obiektów [9]

Fig. 1. Example test images. Left column – original images, right column – reference mask [9]

Następnie na podstawie wyliczonych parametrów wyznaczono dwie miary: procent niepoprawnych klasyfikacji:

$$PNK = \frac{FN + FP}{TP + FN + FP + TN} \times 100\% \quad (3)$$

oraz precyzję:

$$P = TP / (TP + FP) \quad (4)$$

W pierwszym eksperymencie przebadano trzy reprezentacje piksela: odcienie jasności, RGB oraz CIE Lab. W pierwszych dwóch przypadkach obliczano odległość modułową (metryka L1) oraz dodatkowo euklidesową (metryka L2) dla RGB, a w trzecim określonej następującą zależnością:

$$d_{CIELab} = \alpha |L_I - L_B| + \beta (|a_I - b_B| + |a_I - b_B|) \quad (5)$$

gdzie: L_I, a_I, b_I - aktualny piksel w przestrzeni barw CIE Lab, L_B, a_B, b_B - próbka z modelu tła w przestrzeni barw CIE Lab, α, β - wagi (w eksperymentach ustalone na $\alpha = 1, \beta = 1,5$). Na analizę przestrzeni CIE Lab zdecydowano się, gdyż we wcześniejszych pracach [2] jej użycie pozwoliło uzyskać dobre wyniki segmentacji obiektów. Wyniki eksperymentów zestawiono w tabeli 1.

Tabel 1. Analiza działania algorytmu ViBe w zależności od użytej przestrzeni barw
Table 1. Performance of the ViBe algorithm depending on the used colour space

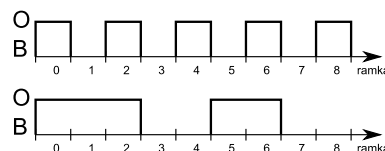
Przestrzeń barw	Odległość	PNK [%]	P
Odcienie szarości	L1	3,78	0,67
RGB	L1	2,71	0,62
RGB	L2	2,28	0,69
CIE Lab	wzór (5)	2,18	0,71

Przedstawione wartości są średnimi wskaźnikami dla całego zbioru testowego. Jedynym parametrem algorytmu, który był modyfikowany jest próg R . Starano się dobrać go tak, aby uzyskać jak najlepsze wartości PNK i P. Analiza wyników pokazuje nieznaczną przewagę wykorzystania przestrzeni CIE Lab nad RGB (norma L2). Dodatkowo, sprzętowa implementacja równania (5) jest znacznie prostsza w realizacji, niż obliczanie odległości euklidesowej (konieczność wykonywania podnoszenia do kwadratu i pierwiastkowania). W docelowej implementacji, zdecydowano się zatem na wykorzystanie przestrzeni CIE Lab, co stanowi modyfikację w stosunku do oryginalnej propozycji z pracy [7].

W publikacji [8] zaproponowane zostało rozszerzenie metody ViBe o moduł detekcji pikseli, które naprzemiennie klasyfikowane są jako obiekt i tło. Występują one najczęściej w przypadku, gdy w tle obecny jest niewielki ruch (płynąca woda, ruszająca się

trawa lub liście). Autorzy zaproponowali następujący mechanizm kompensacji tych zakłóceń. Wykrywane są piksele należące do wewnętrznej granicy obszaru tła, które w poprzedniej iteracji zostały zaklasyfikowane inaczej niż w obecnej. W takim przypadku zmienna pomocnicza "wskaźnik migania" inkrementowana jest o 15, a w przeciwnym dekrementowana o 1. W przypadku, gdy wskaźnik przekroczy wartość progu (ustalonego na 30), wtedy piksel usuwany jest z maski obiektów¹.

W niniejszej pracy zaproponowano rozszerzenie analizy pikseli polegające na zastosowaniu dwóch liczników: kolejnych klasyfikacji piksela jako tła i jako obiektu. Dzięki temu możliwe stało się nie tylko wykrycie pikseli, które zmieniają się co jedną iterację (ramkę obrazu), ale również co kilka (rysunek 2, wykres dolny).



Rys. 2. Dwa możliwe warianty migającego piksela. O - klasyfikacja jako obiekt, B - klasyfikacja jako tło

Fig. 2. Two kinds of blinking pixel. O – classification as foreground object, B – classification as background

Zaproponowane podejście pozwoliło uzyskać nieznacznie lepsze wyniki, przykładowo dla sekwencji "Canoe" (płynąca woda - górna część rysunku 1) dla oryginalnego podejścia uzyskano $PNK = 2,19$ i $P = 0,63$, a dla zaproponowanego $PNK = 1,97$ i $P = 0,68$. Warto podkreślić, że modyfikacja w bardzo nieznacznym stopniu komplikuje algorytm, a szczególnie jego implementację sprzętową.

Do przetwarzania końcowego wybrano filtr medianowy o rozmiarze 7×7 . Warto podkreślić, że jego użycie znacząco poprawia wyniki uzyskiwane przez algorytm. Przykład zaprezentowano w tabeli 2.

Tabel 2. Wpływ końcowej filtracji medianowej na wyniki działania algorytmu. Dane średnie dla całej bazy

Table 2. The impact of the post processing median filtering on the algorithm performance. Mean results for the whole database

Przetwarzanie końcowe	PNK [%]	P
brak	2,18	0,71
mediana 7×7	1,76	0,88

5. Implementacja sprzętowa

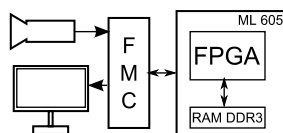
Schematycznie proponowany system zaprezentowano na rysunku 3. Składa się on ze źródła sygnału HDMI (kamera lub karta graficzna), monitora HDMI, modułu FMC (ang. FPGA Mezzanine Card) Avnet FMC DVI IO z wejściem i wyjściem HDMI oraz karty ML605 z układem FPGA Virtex 6 (XC6VLX240T) firmy Xilinx. Karta ta wyposażona jest w zewnętrzną pamięć RAM DDR3.

Funkcjonalność logiki opisana została w językach VHDL i Verilog. Schemat blokowy przedstawiono na rysunku 4. Moduł RGB TO CIELab odpowiada za realizację konwersji przestrzeni barwnych [2]. Generację liczb pseudolosowych (RNG) zrealizowano wykorzystując koncepcję opisaną w pracy [12]. Warto podkreślić, że Autorzy artykułu udostępnili gotowy kod (VHDL), który można bardzo łatwo zintegrować z realizowanym projektem. Do obsługi zewnętrznej pamięci RAM wykorzystano kontroler zbliżony do opisanego w [2].

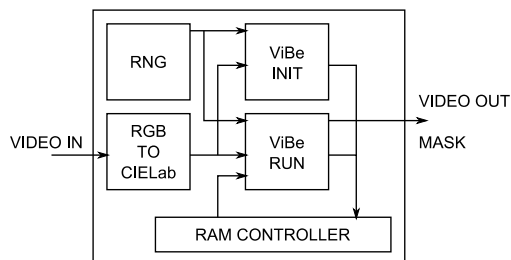
Moduł ViBe INIT odpowiada za inicjalizację modelu tła. Składa się z linii opóźniającej odpowiadającej za generację kontekstu o rozmiarze (3×3) oraz N multiplexerów ($N=20$), odpowiadających za wybór z kontekstu odpowiedniej próbki, która następnie jest zapisywana w modelu tła. Multiplexery sterowane są na

¹ w pracy [8] wprowadzono podział na dwie maski (obiektów i do uaktualnienia), a wykrycie migającego piksela skutkowało zmianą w drugiej z nich. Z racji na znaczną komplikację realizacji operacji wypełniania dziur w systemie potokowym, w przedstawianych badaniach wątek ten został pominięty.

podstawie wektora z modułu RNG, w ten sposób uzyskano losowo inicjalizowane modelu.



Rys. 3. Schemat proponowanego systemu sprzętowego do detekcji obiektów
Fig. 3. Scheme of the proposed foreground object detection system



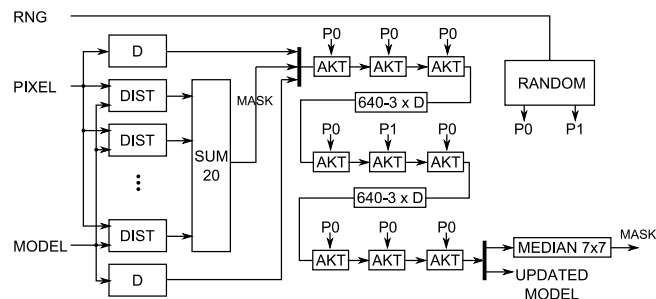
Rys. 4. Schemat blokowy modułów zrealizowanych w układzie FPGA
Fig. 4. Block diagram of the modules implemented in the FPGA device

Zasadniczym modułem jest ViBe RUN, którego bardziej szczegółowy schemat zaprezentowano na rysunku 5. Wejście do modułu stanowią RNG (wektor z generatora liczb pseudolosowych), PIXEL (bieżący piksel w przestrzeni CIE Lab), MODEL (model tła odczytany z zewnętrznej pamięci RAM). W pierwszej fazie obliczane są odległości pomiędzy bieżącym pikselem, a próbkami z modelu i porównywane z wartością R (DIST - realizacja równania (5)). Następnie sprawdzane jest czy liczba odległości mniejszych od R przekracza wartości $\#_{min}$. W kolejnym etapie generowany jest kontekst 3×3 składający się z sygnałów PIXEL, MODEL i MASK (maska obiektów). Warto zwrócić uwagę na znaczną zasobochłonność rozwiązania - wymaga ono użycia 28 modułów pamięci blokowej (Block RAM). Moduł AKT pełni zarówno funkcję pojedynczego opóźnienia, jak i zawiera w sobie logikę, która pozwala na realizację aktualizacji. Proces zamiany jednej z próbek modelu z bieżącym pikselem sterowany jest zmienną P0 lub P1 (piksel centralny) i uzależniony od czynnika losowego (por. rozdział 2) - schematycznie zostało to przedstawione w postaci modułu RANDOM. Ostatnim etapem przetwarzania jest filtracja medianowa z maską 7×7 (MEDIAN 7×7). Uaktualniany model jest zapisywany do pamięci RAM, a maska wyświetlana. Dla czytelności pominięto logikę detekcji migających pikseli.

Przedstawiony system zintegrowano i zsyntezowano dla układu Virtex 6 z wykorzystaniem narzędzia Xilinx ISE 14.4 Design Suite. Maksymalna częstotliwość pracy (raportowana po fazie place & route) wynosiła 140 MHz, co jest wartością wystarczającą do analizy kolorowego strumienia wideo o rozdzielczości 640×480 i częstotliwości 60 ramek na sekundę, w czasie rzeczywistym. Zużycie zasobów FPGA zestawiono w tabeli 3. Zgodność modułu sprzętowego z modelem programowym potwierdzono wykonując symulację w narzędziu ISim.

6. Podsumowanie

W artykule opisano implementację algorytmu generacji tła ViBe w układzie FPGA. Zaproponowano dwie modyfikacje: wykorzystanie przestrzeni barw CIE Lab oraz ulepszoną detekcję migających pikseli, które poprawiły skuteczność metody. Rezultaty pokazują, że dysponując odpowiednią platformą sprzętową, z szybką pamięcią zewnętrzną, można, w sposób w pełni potokowy, zrealizować nawet dość złożony algorytm analizy obrazu. Zaproponowany system umożliwia płynne przetwarzanie kolorowego strumienia wideo o rozdzielczości 640×480 i częstotliwości 60 klatek na sekundę. Wykonany moduł może zostać wykorzystany w zaawansowanych, automatycznych systemach monitoringu wizyjnego.



Rys. 5. Schemat implementacji modułu ViBe RUN
Fig. 5. Scheme of the ViBe RUN module

Tab. 3. Wykorzystanie zasobów FPGA
Tab. 3. FPGA resource utilisation

Zasób	Wykorzystane	Dostępne	Procent
FF	13572	301440	3%
LUT6	9278	150720	6%
DSP 48	13	768	1%
BRAM_18	172	832	20%

Przedstawione w artykule prace były wspierane przez AGH w Krakowie grant nr 11.11.120.612.

7. Literatura

- [1] Elhajian S., El-Sayed K., Ahmed S.: Moving Object Detection in Spatial Domain using Background Removal Techniques — State-of-Art. Recent Patents on Computer Science, 1:32-54, 2008.
- [2] Kryjak T., Komorkiewicz M., Gorgoń M.: Real-time background generation and foreground object segmentation for high definition colour video stream in FPGA device. Journal of Real-Time Image Processing, pages 1–17, 2012.
- [3] Genovese M., Napoli E.: FPGA-based architecture for real time segmentation and denoising of HD video. Journal of Real-Time Image Processing, pages 1–13, 2011.
- [4] Rodriguez-Gomez R., Fernandez-Sanchez E.J., Diaz J., Ros E.: FPGA implementation for real-time background subtraction based on Horprasert model. Sensors, 12(1):585–611, 2012.
- [5] Rodriguez-Gomez R., Fernandez-Sanchez E.J., Diaz J., Ros E.: Codebook hardware implementation on FPGA for background subtraction. Journal of Real-Time Image Processing, pages 1–15, April 2012.
- [6] Barnich O., Van Droogenbroeck M.: ViBe: a powerful random technique to estimate the background in video sequences. In International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2009), pages 945-948, April 2009.
- [7] Barnich O., Van Droogenbroeck M.: ViBe: A universal background subtraction algorithm for video sequences. In IEEE Transactions on Image Processing, 20(6):1709-1724, June 2011.
- [8] Van Droogenbroeck M., Paquot O.: Background Subtraction: Experiments and Improvements for ViBe. In Change Detection Workshop (CDW), Providence, Rhode Island, June 2012.
- [9] Goyette N., Jodoin P.M., Porikli F., Konrad J., Ishwar P., changedetection.net: A new change detection benchmark dataset, in Proc. IEEE Workshop on Change Detection (CDW'12) at CVPR'12, Providence, RI, 16-21 Jun., 2012 (www.changedetection.net).
- [10] OpenCV - strona www: <http://opencv.org/> (ostatni dostęp 10.01.2013r.)
- [11] Toyama K., Krumm J., Brumitt B., Meyers B.: Wallflower: principles and practice of background maintenance. In Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, volume 1, pages 255–261, 1999.
- [12] Thomas D.B., Luk W.: FPGA-Optimised Uniform Random Number Generators Using LUTs and Shift Registers, Field Programmable Logic and Applications (FPL), 2010 International Conference on pages 77-82, 2010.

otrzymano / received: 16.08.2013

przyjęto do druku / accepted: 01.10.2013

artykuł recenzowany / revised paper