

Tomasz Kryjak

AGH AKADEMIA GÓRNICZO-HUTNICZA, AL. MICKIEWICZA 30, 30-059 KRAKÓW

Marek Gorgoń

AGH 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, prof. nadzw. AGH**

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, komitetów naukowych wielu konferencji międzynarodowych i krajowych, 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 FPGA reconfigurable device of ViBe (ang. *Visual Background Extractor*) - a novel background subtraction algorithm. The method is based on a different, from those previously described and implemented, background modelling concept. In this paper the possibility of implementing ViBe in hardware is discussed, two modifications to the algorithm, which increase the performance, are proposed and the designed system is presented. This is the first, known to the authors, FPGA implementation of the ViBe algorithm.

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 [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). 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: MOG [3], Codebook [4] Clustering [2]. Obszerne omówienie zagadnienia przedstawiono w pracy [2].

2. Algorytm ViBe

Algorytm segmentacji obiektów pierwszoplanowych ViBe (ang. *Visual Background Extractor*) został zaproponowany przez O. Barnicha i M. Van Droogbroeckę i opisany w pracach [5, 6]. 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 [7].

Model tła w algorytmie ViBe zbudowany jest z zbioru zaobserwowanych wartości piksela. 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 $Sr(v(x))$ 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 Euklidesowa).

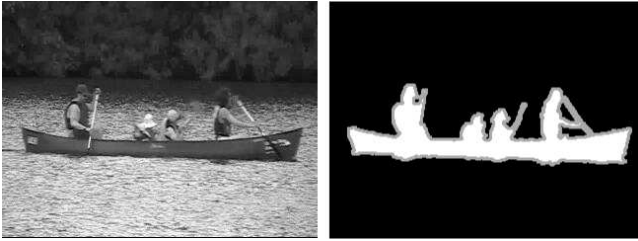
Autorzy metody zaproponowali mechanizm inicjalizacji modelu tła na podstawie jednej ramki oraz metodę aktualizacji z czynnikiem losowym. Dokładny opis zawarty jest w pracy [5]. Warto podkreślić, że metoda ViBe wymaga ustawienia bardzo niewielu parametrów. Autorzy zaproponowali następujące ich wartości: $N = 20$ (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. Zaproponowane modyfikacje metody ViBe

Na podstawie wyników zaprezentowanych w pracy [6] zdecydowano się przeanalizować następujące modyfikacje algorytmu ViBe: 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. Jako zbiór testowy użyta została baza filmów, wykonana na potrzeby warsztatów *IEEE Workshop on Change Detection* [7]. Przykładowe obraz zamieszczono na rysunku 1.

Metodologia eksperymentów była następująca. Porównywano maskę obiektów zwracaną przez algorytm z maską referencyjną. Zliczano klasyfikacje TP (*true positive* – prawdziwie dodatna), TN (*true negative* – prawdziwie ujemna), FP (*false positive* – fałszywie dodatnia), FN (*false negative* – fałszywie ujemna).



Rys. 1. Przykład obrazu użytego do badań. Lewa kolumna obraz oryginalny, prawa kolumna referencyjna maska obiektów [7]

Fig. 1. Example test image. Left column – original image, right column - reference mask [7]

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

$$PNK = (FN + FP) / (TP + FN + FP + TN) \times 100\% \quad (2)$$

oraz precyzję:

$$P = TP / (TP + FP) \quad (3)$$

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 określoną następującą zależnością:

$$d_{CIE Lab} = \alpha |L_I - L_B| + \beta (|a_I - b_B| + |a_I - b_B|) \quad (4)$$

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 $\alpha = 1, \beta = 1,5$). Wyniki eksperymentów zestawiono w tabeli 1.

Tab. 1. Analiza działania algorytmu ViBe w zależności od użytej przestrzeni barw
Tab. 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

Na podstawie uzyskanych wyników, w docelowej implementacji, zdecydowano się na wykorzystanie przestrzeni CIE Lab. Zaproponowano również rozszerzoną, w stosunku do pracy [6], metodę detekcji pikseli, które naprzemiennie klasyfikowane są jako obiekt i tło. Pozwoliło to uzyskać nieznacznie lepsze wyniki, w porównaniu do propozycji oryginalnej ($PNK = 1,97$ i $P = 0,68$ wobec $PNK = 2,19$ i $P = 0,63$). Do przetwarzania końcowego wybrano filtr medianowy o rozmiarze 7×7 . Warto podkreślić, że jego użycie znacząco poprawia wyniki uzyskiwane przez algorytm.

4. Implementacja sprzętowa

Zaproponowany system składa się ze źródła sygnału HDMI, 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 oraz zewnętrzną pamięcią RAM DDR3 firmy Xilinx.

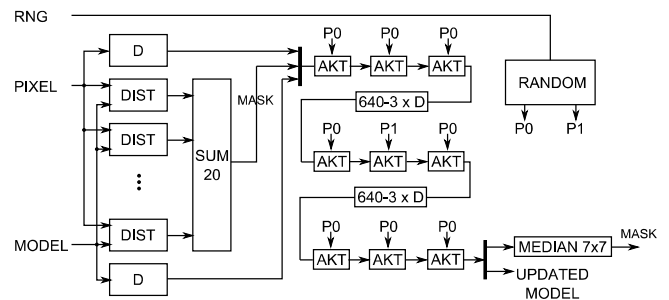
Funkcjonalność logiki opisana została w językach VHDL i Verilog. Schemat modułu realizującego segmentację i aktualizację modelu tła zaprezentowano na rysunku 2. Wejście do niego stanowią RNG (wektor z generatora liczb pseudolosowych), PIXEL (bieżący piksel), MODEL (model tła). W pierwszej fazie obliczane są odległości pomiędzy bieżącym pikselem, a próbkami z modelu i porównywane z wartością R (DIST - realizacja równania (4)). Następnie sprawdzane jest czy liczba odległości mniejszych od R przekracza wartości $\#_{min}$. W kolejnym etapie generowany jest kontekst 3×3 . Moduł AKT pełni zarówno funkcję pojedynczego opóźnienia, jak i zawiera w sobie logikę, która pozwala na realizację aktualizacji (sterowaną sygnałami P0 i P1 zależnymi od czynnika losowego - RANDOM). Generacja liczb

pseudolosowych została zrealizowana z wykorzystaniem koncepcji opisanej w pracy [8]), a do obsługi zewnętrznej pamięci RAM wykorzystano kontroler zbliżony do opisanego w [2]. Ostatnim etapem przetwarzania jest filtracja medianowa z maską 7×7 (MEDIAN 7×7). Uaktualniony model jest zapisywany do pamięci RAM, a maska wyświetlana.

Przedstawiony system zintegrowano i zsyntezowano z wykorzystaniem narzędzia Xilinx ISE 14.4 Design Suite. Maksymalna częstotliwość pracy wynosiła 140 MHz, co jest wartością wystarczającą do analizy kolorowego strumienia wideo o rozdzielczości 640×480 w czasie rzeczywistym. Zgodność modułu sprzętowego z modelem programowym potwierdzono symulacyjnie (ISim).

5. 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 60 klatkach na sekundę.



Rys. 2. Schemat implementacji modułu ViBe RUN.

Fig. 2. Scheme of the ViBe RUN module

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

6. Literatura

- [1] Elhabian 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.: Codebook hardware implementation on FPGA for background subtraction. Journal of Real-Time Image Processing, pages 1–15, April 2012.
- [5] 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.
- [6] Van Droogenbroeck M., Paquot O.: Background Subtraction: Experiments and Improvements for ViBe. In Change Detection Workshop (CDW), Providence, Rhode Island, June 2012.
- [7] 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).
- [8] 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