

**Wiesław PAMUŁA**

POLITECHNIKA ŚLĄSKA  
ul. Akademicka 2 A, 44-100 Gliwice

## Metoda implementacji trójwymiarowej dyskretnej transformaty falkowej strumienia wideo w układach FPGA

Dr inż. Wiesław PAMUŁA

Dr nauk technicznych (1998). Adiunkt w Katedrze Inżynierii Ruchu na Wydziale Transportu Politechniki Śląskiej. Prowadzone prace badawcze dotyczą zastosowania przetwarzania obrazów do określania parametrów ruchu drogowego. Bierze udział w opracowaniu i wdrożeniu wideo detektorów pojazdów dla sterowników sygnalizacji świetlnej.



e-mail: wieslaw.pamula@polsl.pl

### Streszczenie

Współczynniki dyskretnej transformaty falkowej reprezentują własności przestrzenne otoczenia punktu, dla którego są wyliczane i wyliczane są na podstawie wartości z tego otoczenia. Rozmiar otoczenia zależy od poziomu dekompozycji i długości filtrów. Korzystając z tej własności zaproponowano metodę implementacji polegającą na potokowym przetwarzaniu wektorów wartości z otoczenia. Potok realizuje algorytm transformaty na zadanym poziomie dekompozycji. Brak jest iteracyjnego wyliczania kolejnych poziomów dekompozycji i następuje znaczna redukcja liczby wymian danych z pamięcią. Przedstawioną jest przykład z wykorzystaniem falki  $S(1, 1)$ , użyteczny do wykorzystania w rozwiązaniach detektorów ruchomych obiektów np. w ruchu drogowym.

**Słowa kluczowe:** trójwymiarowa dyskretna transformata falkowa, FPGA, detekcja ruchomych obiektów.

### Implementation of 3D DWT of video stream in FPGA

#### Abstract

The paper presents a method for implementation of 3D discrete wavelet transform in FPGA. The method is based on direct calculation of coefficients at the desired level of decomposition. The previous methods [5, 6, 7] use complex architectures with multilevel processing, mapping the lifting scheme or the convolution procedure. The direct calculation of coefficients is done using the set of neighbourhood data. This set is derivative of the level of decomposition and the number of vanishing moments of the used wavelet filters - Eq. (3). An example of implementing the  $S(1, 1)$  transform at the third level of decomposition is presented. The  $S$  transform coefficients are the weighted sums of  $4 \times 4 \times 4$  volumes of data (Eqs. (4), (5)). Fig. 2 shows the block diagram of the implementation. Data from the stream is stored in a buffer memory of the neighbourhood vectors. The addressing scheme, which is carried out by the addressing module, assures appropriate ordering of data in the vectors, in the memory. Further refinement consists in summing in place the consecutive values and thus replacing the neighbourhood data with sums of data that are used for calculating coefficients. This reduces significantly the vector size and streamlines calculations. The results of logic utilisation (Table 2) of different FPGA components for the implementation are presented. The designed 3D DWT component is incorporated in a moving object detecting device processing video from a road traffic camera. The method may be used for developing specialised hardware for compressing 3D data streams in a way compatible with the JPEG2000 standard.

**Keywords:** 3D DWT (Discrete Wavelet Transform), FPGA, detection of moving objects.

### 1. Wstęp

Postęp w technologiach obrazowania prowadzi do generowania strumieni wideo o bardzo dużych rozmiarach. Ciąg obrazów z kamery, na przykład w rozdzielczości HD, stanowi strumień ponad 150MB/sek danych. Sekwencje obrazów z urządzeń diagnostyki obrazowej mogą zawierać nawet po kilkaset MB. Prze-

tworzenie strumieni wymaga zachowania najwyższej wierności aby zminimalizować ryzyko popełnienia błędów w ocenie treści zawartości. Przy tak dużych rozmiarach strumieni przetwarzanie może być bardzo uciążliwe.

Dyskretna transformata falkowa dominuje jako narzędzie przygotowania reprezentacji strumienia. W oparciu o nią znane standardy kompresji MPEG4, JPEG2000 zapewniają duże stopnie kompresji i możliwość elastycznego dopasowania jakości odtworzenia do potrzeb analizy treści.

Bardzo efektywnym rozwiązaniem jest implementacja przetwarzania z użyciem układów logicznych zapewniających równoległe obliczenia oraz pracę z bardzo dużymi prędkościami. Sprzętowe rozwiązania układów wyliczania transformat falkowych skupiają się na realizacji algorytmów ww. standardów. Są to przede wszystkim jedno i dwuwymiarowe wersje obliczające transformaty z filtrem falkowym CDF 5/3 (Cohen Daubechies Faveau) dla kompresji bezstratnej i filtrem CDF 9/7 dla stratnej. Można wyróżnić rozwiązania sprzętowe oparte na bezpośredniej implementacji algorytmu splatania, wykorzystujące schemat liftingu oraz wykonane z użyciem układów MAC (multiply-accumulate) [1, 2, 3].

Trójwymiarowa wersja transformaty falkowej spotykana jest w kompresji obrazów przestrzennych. Wczesne opracowania sprzętowych implementacji opierają się na rozszerzeniu dwuwymiarowych wersji przez powielenie układów wyliczania i modyfikacje organizacji dostępu do pamięci [5].

Zwiększenie zasobów układów logicznych umożliwiło zaproponowanie realizacji schematu lifting i potokowego wyliczania współczynników [6]. Najnowsze rozwiązania korzystają z dynamicznej rekonfiguracji aby zoptymalizować stopień wykorzystania zasobów układów FPGA [7]. Idea implementacji pozostaje rozwinięciem wcześniejszych pomysłów. Przykłady implementacji realizowane są z użyciem zaawansowanych układów FPGA np. z rodziny Virtex firmy Xilinx lub Stratix firmy Altera.

Implementacje realizują obliczenia dla kolejnych wymiarów strumienia. Wyniki pośrednie przechowywane są w pamięci i wielokrotnie wymieniane przez układy przetwarzające. Dyskusja cech rozwiązań prezentowanych w literaturze pomija zagadnienia optymalizacji dostępu do pamięci. Czas dostępu do pamięci, szczególnie dynamicznych, które stosowane są do przechowywania dużej liczby danych, co jest charakterystyczne dla obrazów z kamery, stanowi istotny czynnik spowolnienia procesu wyliczania. Propozowane zastosowanie bezpośredniego wyliczania transformaty z użyciem wektora danych otoczenia znacznie zmniejsza liczbę dostępu do pamięci. Dodatkowo wykorzystanie długiego słowa danych umożliwia dalszą redukcję liczby operacji pamięciowych.

Odmiernym od kompresji zastosowaniem trójwymiarowej falki jest segmentacja strumienia wideo dla wydzielenia poruszających się obiektów [4]. Współczynniki transformaty służą do opisu obiektów, ruchome obiekty wyróżnione są przez wartości współczynników określających zmiany wartości w czasie – trzecim wymiarze transformaty.

### 2. Dyskretna transformata falkowa

Dyskretna transformata falkowa realizowana jest z użyciem filtrów falkowych. Wyliczana jest odpowiedź z użyciem splotu lub operacje filtrowania dekomponowane są na ciąg działań w schemacie lifting.

Dyskretny sygnał wejściowy  $x(n)$  poddawany jest filtracji z użyciem filtru dolnoprzepustowego  $\hat{h}$  i górnoprzepustowego  $\hat{g}$ . Odpowiedzi filtrów poddane są decymacji, odrzucana jest co druga wartość i uzyskuje się:

$$y_L(n) = \sum_{i=0}^{k-1} \tilde{h}(i)x(2n-i),$$

$$y_H(n) = \sum_{i=0}^{l-1} \tilde{g}(i)x(2n-i),$$
(1)

gdzie:  $k, l$  – długości filtrów.

W [8] wykazano, gdy para  $\tilde{h}$  i  $\tilde{g}$  jest komplementarną parą filtrów macierz polifazową tej pary  $\tilde{P}(z)$  można z użyciem algorytmu Euklidesa rozłożyć na czynniki reprezentujące działania schematu lifting:

$$\tilde{P}(z) = \begin{bmatrix} \tilde{h}_e(z) & \tilde{h}_o(z) \\ \tilde{g}_e(z) & \tilde{g}_o(z) \end{bmatrix},$$

$$\tilde{P}(z) = \prod_{i=1}^m \begin{bmatrix} 1 & \tilde{u}(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1/K \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix},$$
(2)

gdzie:  $m$  – liczba kroków schematu lifting.

Operacje  $\tilde{u}(z)$  określane są mianem uaktualnienia a  $\tilde{p}(z)$  predykcją. Tabela 1. przedstawia wybrane filtry falkowe i ich wyrażenia określające predykcję dla wyznaczenia  $d$  (szczegóły) oraz uaktualnienie dla wyznaczenia  $s$  (średnie) w wersji jednowymiarowej [9, 10].

Tab. 1. Funkcje predykcji i uaktualniania całkowitoliczbowych filtrów falkowych  
Tab. 1. Prediction and update functions of integer wavelet functions

S(1,1)	$d_{1,i} = s_{0,2i+1} - s_{0,2i}$ $s_{1,i} = s_{0,2i} + \lfloor d_{1,i}/2 \rfloor$
CDF 5/3 (2,2)	$d_{1,i} = s_{0,2i+1} - \lfloor (s_{0,2i} + s_{0,2i+2})/2 + 1/2 \rfloor$ $s_{1,i} = s_{0,2i} + \lfloor (d_{1,i-1} + d_{1,i})/4 + 1/2 \rfloor$
(4,4)	$d_{1,i} = s_{0,2i+1} - \lfloor 9(s_{0,2i} + s_{0,2i+2})/16 - (s_{0,2i-2} + s_{0,2i+4})/16 + 1/2 \rfloor$ $s_{1,i} = s_{0,2i} + \lfloor 9(d_{1,i-1} + d_{1,i})/32 - (d_{1,i-2} + d_{1,i+1})/32 + 1/2 \rfloor$
9/7-F (4,4)	$d_{1,i} = s_{0,2i+1} + \lfloor -203(s_{0,2i} + s_{0,2i+2})/128 + 1/2 \rfloor$ $s_{1,i} = s_{0,2i} + \lfloor -217(d_{1,i-1} + d_{1,i})/4096 + 1/2 \rfloor$ $d_{1,i} = d_{1,i} + \lfloor 113(s_{1,2i} + s_{1,2i+2})/128 + 1/2 \rfloor$ $s_{1,i} = s_{s,i} + \lfloor 1817(d_{1,i-1} + d_{1,i})/4096 + 1/2 \rfloor$

Wykonanie operacji w schemacie lifting zawiera mniej działań w porównaniu do splatania i pozwala na zmniejszenie wymagań pamięciowych. Schemat jest używany też do konstrukcji transformacji całkowitoliczbowych poprzez zaokrąglenie wyników filtracji w każdym kroku algorytmu przed korektą wyznaczonych wartości próbek odpowiednio nieparzystych i parzystych.

### 3. Implementacja

Reprezentacja strumienia wideo opiera się na współczynnikach transformaty co najwyżej kilku poziomów dekompozycji. Analiza wyrażen przedstawiających funkcje predykcji i uaktualniania filtrów falkowych wskazuje, że wyliczane są z użyciem pikseli z niewielkiego sąsiedztwa. Wraz z poziomem  $n$  dekompozycji rozszerza się jego zasięg. Rozmiar sąsiedztwa zależy od liczby zer  $k$  transmitancji ( $z$ ) filtru falkowego:

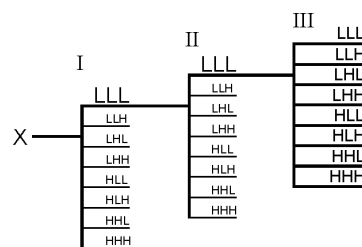
$$\text{gdy } k = 1: \quad P = 2^n 2^n 2^n,$$

$$\text{gdy } k > 1: \quad P = (2^n + (2k-1)(2^n-1))^3.$$
(3)

Dla transformacji S (1, 1) na trzecim poziomie dekompozycji otoczenie obejmuje kwadrat 8x8 pikseli w ośmiu kolejnych klatkach strumienia wideo w sumie 512 wartości. W przypadku CDF 5/3 otoczenie zwiększa się do 29x29x29 obejmując 24389 danych.

Niewielka liczba danych, w porównaniu do rozmiaru strumienia, wymaganych aby wyliczyć współczynnik na zadanym poziomie dekompozycji wskazuje na możliwość wykonania obliczeń z wykorzystaniem wewnętrznych zasobów pamięciowych typowych układów FPGA (low cost).

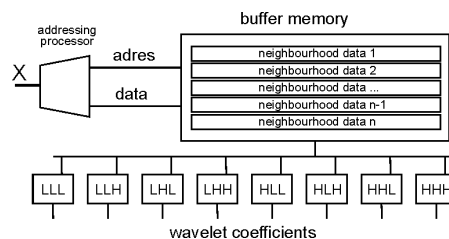
Dyskretna transformata falkowa może być wykonana kolejno dla wymiarów strumienia. Zgodnie ze schematem Mallata rys.1 do kolejnych poziomów dekompozycji wykorzystywane są dane z uaktualnień (filtracji dolnoprzepustowej), pozwala to pominąć wykonanie pełnej dekompozycji dla wcześniejszych poziomów redukując znaczną liczbę operacji obliczeniowych.



Rys. 1. Schemat Mallata wyliczenia 3-wymiarowej dyskretnej transformaty falkowej na trzecim poziomie dekompozycji

Fig. 1. Mallat's scheme for calculation of the 3D DWT on the third level of decomposition

Trzywymiarowa transformacja falkowa dla danego poziomu dekompozycji wyznacza 8 wartości współczynników odpowiadających kombinacjom odpowiedzi filtrów dolno- i górno- przepustowym. Wyliczanie współczynników proponuje się zrealizować z użyciem autonomicznych modułów przetwarzających wartości z otoczenia pikseli o rozmiarach zgodnie z (3). Moduły działają na wektorach wartości otoczenia, które przygotowywane są przez układ konwersji strumienia wideo na wektory otoczeń. Konwersja polega na pobraniu ze strumienia danych i uporządkowaniu ich do postaci wektorów. Realizowane jest to przez wyznaczenie adresów w pamięci buforującej odpowiadających kolejnym elementom wektorów z użyciem procesora adresowego. Rys. 2 przedstawia schemat blokowy rozwiązania.



Rys. 2. Schemat blokowy rozwiązania wyliczenia trójwymiarowej dyskretnej transformaty falkowej

Fig. 2. Block scheme of the solution for calculation of the 3D DWT

Operacje schematu lifting dla najczęściej używanych filtrów falkowych obejmują sumowania i mnożenia całkowitoliczbowe (tabela 1) stąd moduły wyliczania zawierają będą sumatory i układy mnożące (dedicated multipliers).

Blok konwersji strumienia współpracujący z pamięcią buforującą wektory otoczeń może wspomagać wyliczanie sum wartości pikseli przez sumowanie w miejscu wymaganych danych. W przypadku szeregowo podawanych danych np. z kamery skompletowanie pierwszego wektora wymaga pobrania danych z kolejnych wierszy i klatek obrazu.

#### Transformacja S (1, 1)

Sąsiedztwa pikseli dla danego poziomu dekompozycji, dla transformacji S, nie nakładają się. Współczynniki można zatem obliczyć dla z góry podzielonej przestrzeni danych bez konieczności wielokrotnego ponownego pobierania tych samych danych. Na

trzecim poziomie dekompozycji dla wektora  $X=(x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7)$  wartości współczynników wyrażone są:

$$\begin{aligned} d_3 &= \frac{x_4 + x_5 + x_6 + x_7}{4} - \frac{x_0 + x_1 + x_2 + x_3}{4}, \\ s_3 &= \frac{x_4 + x_5 + x_6 + x_7}{8} + \frac{x_0 + x_1 + x_2 + x_3}{8}. \end{aligned} \quad (3)$$

założono brak zaokrąglania wyników. Charakterystycznym wyrażeniem jest suma czterech kolejnych wartości pikseli. Dokonując transformacji względem kolejnych wymiarów  $i, j, k$  powtarzane będzie sumowanie dla każdej współrzędnej pikseli. Trójwymiarowa transformata sześcianu pikseli o rozmiarze  $8 \times 8 \times 8$ , trzeci wymiar może oznaczać następstwo w czasie lub kolejne warstwy analizy przestrzennej, operować będzie na sumach:

$$\begin{aligned} S_0 &= \sum_{i=0}^3 \sum_{j=0}^3 \sum_{k=0}^3 x_{ijk}, & S_1 &= \sum_{i=0}^3 \sum_{j=0}^3 \sum_{k=0}^3 x_{ijk+4}, \\ S_2 &= \sum_{i=0}^3 \sum_{j=0}^3 \sum_{k=0}^3 x_{ij+4k}, & S_3 &= \sum_{i=0}^3 \sum_{j=0}^3 \sum_{k=0}^3 x_{ij+4k+4}, \\ S_4 &= \sum_{i=0}^3 \sum_{j=0}^3 \sum_{k=0}^3 x_{i+4jk}, & S_5 &= \sum_{i=0}^3 \sum_{j=0}^3 \sum_{k=0}^3 x_{i+4jk+4}, \\ S_6 &= \sum_{i=0}^3 \sum_{j=0}^3 \sum_{k=0}^3 x_{i+4j+4k}, & S_7 &= \sum_{i=0}^3 \sum_{j=0}^3 \sum_{k=0}^3 x_{i+4j+4k+4}. \end{aligned} \quad (4)$$

Dla strumienia wideo indeksy  $i, j$  określają położenie piksela w klatce, a indeks  $k$  określa kolejną klatkę strumienia wideo. Współczynniki odpowiadające złożeniu operacji  $d$  (szczegóły lub odpowiedzi filtru górnoprzepustowego) oraz  $s$  (średnie lub odpowiedzi filtru dolnoprzepustowego) na trzecim poziomie dekompozycji wynoszą:

$$\begin{aligned} sss_3 &= \frac{1}{512} (S_0 + S_1 + S_2 + S_3 + S_4 + S_5 + S_6 + S_7), \\ ssd_3 &= \frac{1}{256} (S_0 - S_1 + S_2 - S_3 + S_4 - S_5 + S_6 - S_7), \\ sds_3 &= \frac{1}{256} (S_0 + S_1 - S_2 - S_3 + S_4 + S_5 - S_6 - S_7), \\ sdd_3 &= \frac{1}{128} (S_0 - S_1 - S_2 + S_3 + S_4 - S_5 - S_6 + S_7), \\ dss_3 &= \frac{1}{256} (S_0 + S_1 + S_2 + S_3 - S_4 - S_5 - S_6 - S_7), \\ dsd_3 &= \frac{1}{128} (S_0 - S_1 + S_2 - S_3 - S_4 + S_5 - S_6 + S_7), \\ dds_3 &= \frac{1}{128} (S_0 + S_1 - S_2 - S_3 - S_4 - S_5 + S_6 + S_7), \\ ddd_3 &= \frac{1}{64} (S_0 - S_1 - S_2 + S_3 - S_4 + S_5 + S_6 - S_7). \end{aligned} \quad (5)$$

Rozwiązanie układu z rys. 2 rozszerzone jest o podzespół sumowania w miejscu wartości składających się na sumy  $S_i$ . Wektor danych z otoczenia zawiera sumy cząstkowe  $S$ . Gdy pobierana ze strumienia dana należy do  $i$ -tej ósmej części sześcianu otoczenia dodawana jest do  $S_i$  w wektorze. Po pobraniu wszystkich danych z otoczenia wyznaczany jest kolejny adres wektora w pamięci buforowej przez procesor adresów. Pamiętanie sum w miejsce samych danych znacznie skraca rozmiar wektora.

Po skompletowaniu w pamięci otoczeń z ośmiu kolejnych klatek obrazu następuje wyliczenie współczynników falkowych z użyciem sukcesywnie pobieranych z pamięci wektorów otoczeń.

Układ obliczania trójwymiarowej dyskretnej transformaty falkowej zrealizowano jako podzespół detektora obiektów w strumieniu wideo. Zmiany wartości współczynników transformaty, związanych z współrzędną czasową, są bardzo mało czułe na złożone wahania oświetlenia obserwowanej sceny, analizując ich zmienność uzyskuje się wskaźniki ruchu obiektów [4].

Podzespół opracowano w VHDL w środowisku ISE firmy Xilinx w układzie starszej generacji Spartan3E. Tabela 2 przedstawia złożoność implementacji z użyciem układów FPGA. Realizacja z użyciem układów firmy Altera (po modyfikacji źródeł VHDL polegającej na podmianie elementów bibliotecznych) obejmuje porównywalne zasoby logiczne.

Tab. 2. Zasoby wykorzystane w implementacji prototypu transformacji S (1, 1)  
Tab. 2. Resources used by a prototype of the S (1, 1) transform implementation

zasób	Xilinx	
	Spartan 3E	Virtex 6
przerzutniki	2288	1298
LUT	3833 (4wej)	3386 (6wej)
pamięć wewnętrzna	11 bloków	10 bloków

### Transformacja CDF 5/3

Implementacja transformaty falkowej CDF 5/3 wymaga uważnej konstrukcji układu adresowania danych w buforze. CDF 5/3 na trzecim poziomie dekompozycji operuje na otoczeniach o rozmiarze  $29 \times 29 \times 29$ . Sumowanie w miejscu pozwala skrócić długość wektora do rozmiaru  $8 \times 12b$  i zastosować podobne jak dla S (1, 1) rozwiązanie kaskadowego sumowania w modułach wyliczania współczynników.

## 4. Wnioski

Uzyskane wyniki wskazują na możliwość realizacji trójwymiarowej transformaty falkowej w typowym układzie FPGA. Zaproponowana metoda może znaleźć zastosowanie w sprzętowych układach kompresji strumienia wideo oraz jako podzespół w urządzeniach przetwarzania obrazów np. do wykrywania poruszających się obiektów.

Prototypowy podzespół wideo detektora wykonuje obliczenia w czasie rzeczywistym pracując na strumieniu wideo w standardzie ITU-R BT 656.

## 5. Literatura

- [1] Acharya T., Chakrabarti C.: A survey on lifting-based discrete wavelet transform architectures, *J. VLSI Signal Process.*, vol. 42, s. 321–339, 2006.
- [2] Shi G., Liu W., Zhang L., Li F.: An efficient folded architecture for lifting-based discrete wavelet transform, *IEEE Trans. on Circuits and Systems II: Express Briefs*, vol. 56, s. 290–294, 2009.
- [3] Zhang C., Wang C., Ahmad M. O.: A pipeline VLSI architecture for high speed computation of the 1-D discrete wavelet transform, *IEEE Trans. on Circuits and Systems I*, vol. 57, s. 2729–2740, 2010.
- [4] Pamula W.: Wavelet-based data reduction for detection of moving objects, *Machine Graphics and Vision*, Institute of Computer Science Polish Academy of Sciences, vol. 20, s. 67–78, 2011.
- [5] Weeks M., Bayoumi M. A.: Three-dimensional discrete wavelet transform architectures, *IEEE Trans. on Signal Processing*, vol. 50, s. 2050–2063, 2002.
- [6] Salem M., Winkler F.: Concept for hardware implementation for the 3D wavelet transform, *Proceedings of the 4th Int. Conference on Intelligent Computing and Information Systems*, Cairo, s. 624–627, 2009.
- [7] Ahmad A., Krill B., Amira A., Rabah H.: Efficient architectures for 3D HWT using dynamic partial reconfiguration. *Journal of Systems Architecture* vol. 56, s. 305–316, 2010.
- [8] Daubechies I., Sweldens W.: Factoring wavelet transforms into lifting steps, *J. Fourier Anal. Appl.*, vol. 4, s. 247–269, 1998.
- [9] Calderbank A. R., Daubechies I., Sweldens W., Yeo Boon-Lock: Wavelet transforms that map integers to integers, *Applied and Computational Harmonic Analysis* vol. 5, s. 332–369, 1998.
- [10] Zheng Jing-jing, Fang Jin-yun, Han Cheng-de: The selection of reversible integer-to-integer wavelet transforms for DEM multi-scale representation and progressive compression, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. vol. 37 part B4, Beijing s. 709–714, 2008.