

Krzysztof SIEMEK

Konrad WOJCIECHOWSKI

DYSKRETNE PRZEKSZTAŁCENIA SUMACYJNE  
DLA FUNKCJI WALSHA

Streszczenie. Przedstawiono sposób zapisu dyskretnych skalarnych funkcji Walsh'a w uporządkowaniu naturalnym, diadycznym oraz sekwencyjnym. Zaprezentowano algorytmy prostego i odwrotnego szybkiego przekształcania Walsh'a. Pokazano sposób tworzenia wektorowych dyskretnych funkcji Walsh'a oraz algorytmy obliczania szybkiego przekształcania Walsh'a na podstawie takich układów.

1. Wstęp

Wraz z szerokim zastosowaniem techniki komputerowej w przetwarzaniu obrazów rośnie znaczenie dyskretnych przekształceń sumacyjnych będących odpowiednikami przekształceń całkowych dla funkcji ciągłych. Jednym z często wykorzystywanych w tym celu przekształceń jest transformacja Walsh'a. Przekształcenia te mogą być wykorzystywane do analizy widmowej obrazów. Szczególnie interesująca jest filtracja polegająca na odrzuceniu części widma obrazu i prowadzeniu dalszego przetwarzania dla problemu o zmniejszonym wymiarze. Ma to duże znaczenie w trakcie analizy obrazów w czasie rzeczywistym.

W przypadku transformacji Walsh'a sposób filtracji, tzn. obojęcia widma oraz jej efekty zależą silnie od uporządkowania układu funkcji Walsh'a, w oparciu o które dokonujemy przekształceń.

W stosunku do istniejącej literatury [1] - [5] praca nie wnosi nowych koncepcji w zakresie szybkiej transformacji Walsh'a. Jej celem jest jedynie stworzenie uporządkowanych podstaw dla numerycznej realizacji tej transformacji i wyjaśnienie związku pomiędzy uporządkowaniem układu funkcji Walsh'a a efektami filtracji polegającej na wycięciu fragmentu "widma".

## 2. Układy dyskretnych funkcji ortogonalnych

Rozważmy dyskretną funkcję zespoloną  $f(n)$  określoną na zbiorze  $\{n : n = 0, \dots, N-1\}$  i załóżmy, że jest ona sumowalna z kwadratem, tzn. że istnieją sumy:

$$\sum_{n=0}^{N-1} f(n) \quad \text{oraz} \quad \sum_{n=0}^{N-1} |f(n)|^2 \quad (2.1)$$

Każdą taką funkcję można przedstawić w postaci:

$$f(n) = \sum_{k=0}^{N-1} a_k \varphi(k, n), \quad n = 0, \dots, N-1 \quad (2.2)$$

gdzie zbiór dyskretnych funkcji  $\{\varphi(k, n)\}$  jest układem funkcji ortogonalnych, tzn. spełniających zależność:

$$\sum_{n=0}^{N-1} \varphi(k, n) \varphi(j, n) = \delta_{kj} c_k \quad \begin{array}{l} k = 0, \dots, N-1 \\ j = 0, \dots, N-1, \end{array} \quad (2.3)$$

gdzie  $\delta_{kj}$  - delta Kroneckera.

Szereg (2.1) nazywany rozwinięciem funkcji  $f(n)$  w układzie funkcji ortogonalnych  $\varphi(k, n)$ .

Współczynniki szeregu  $a_k$  można wyznaczyć ze wzoru:

$$a_k = \sum_{n=0}^{N-1} f(n) \varphi(k, n) \quad (2.4)$$

Są one wybrane w ten sposób, by minimalizować błąd aproksymacji w sensie najmniejszych kwadratów, a więc:

$$\min_{a_k} \sum_{n=0}^{N-1} \left[ f(n) - \sum_{k=0}^{N-1} a_k \varphi(k, n) \right]^2 \rightarrow a_k$$

Istnieje bardzo wiele różnych układów ortogonalnych. Przedstawimy niektóre z nich.



2.1. Funkcje Haara

Ortogonalny układ funkcji dyskretnych  $\varphi(k,n) = \text{har}(k,n)$ , nazywany funkcjami Haara [2], ma postać:

$$\text{har}(0,n) = 1$$

$$\text{har}(k,n) = \begin{cases} \sqrt{2^i} & n = \frac{j}{2^i} N, \dots, \frac{j+1}{2^i} N-1 \\ \sqrt{2^i} & n = \frac{j+1}{2^i} N, \dots, \frac{j+1}{2^i} N-1, \\ 0 & \end{cases} \quad (2.5)$$

gdzie:

$$N = 2^p$$

$$n = 0, \dots, N-1$$

$$k = 2^i + j$$

$$i = 1, \dots, p$$

$$j = 0, 1, \dots, 2^i - 1$$

Przykładowo, wartości funkcji Haara dla  $N = 8$  przedstawiono w poniższej macierzy o wymiarach  $N \times N$ , w której  $k$ -ty wiersz zawiera wartości  $k$ -tej funkcji  $\text{har}(k,n)$ .

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 2 & 2 & -2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & -2 & -2 \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix} = \begin{bmatrix} \text{har}(0,n) \\ \text{har}(1,n) \\ \text{har}(2,n) \\ \text{har}(3,n) \\ \text{har}(4,n) \\ \text{har}(5,n) \\ \text{har}(6,n) \\ \text{har}(7,n) \end{bmatrix}$$

2.2. Funkcje Walsh

Dyskretne funkcje Walsh przybierają wartości  $+1$  lub  $-1$  w punktach  $n = 0, 1, \dots, N-1$ , gdzie  $N = 2^p$ .

Wyróżnia się trzy podstawowe uporządkowania tych funkcji [2], [4] naturalne wg Hadamarda, diadyczne wg Paleya i sekwencyjne wg Walsh. W definicji uporządkowania naturalnego korzysta się z opisanych w 1893 r. macierzy Hadamarda, podających wartości dyskretnych funkcji Walsh do rzędu  $2^p-1$ . Uporządkowanie diadyczne zostało zaproponowane w 1932 przez Paleya. Podana przez niego definicja funkcji Walsh bazuje na iloczynach funkcji Rademachera i binarnym rozwinięciu kolejnych wskaźników funkcji.

Uporządkowanie sekwencyjne dane przez Walsha w 1932 roku charakteryzuje się tym, że rząd (numer) funkcji Walsha jest równy liczbie zmian znaku tej funkcji.

### 2.2.1. Uporządkowanie naturalne

Podstawą generacji układu funkcji Walsha w uporządkowaniu naturalnym są macierze Hadamarda. Są to macierze kwadratowe o elementach równych +1 lub -1. Kolumny (wiersze) tych macierzy są ortogonalne. Ogólnie dla N-tej macierzy Hadamarda mamy:

$$\begin{aligned} H_N H_N^T &= N \cdot I, \\ H_N &= H_N^T, \\ H_N^{-1} &= \frac{1}{N} H_N \end{aligned} \quad (2.6)$$

Struktura macierzy Hadamarda jest następująca:

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$H_{2N} = \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix}$$

Zastępując, dla uproszczenia zapisu elementy +1 i -1 odpowiednio przez "+" i "-", możemy napisać:

$$H_2 = \begin{bmatrix} + & + \\ + & - \end{bmatrix}$$

$$H_4 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} = \begin{bmatrix} + & + & + & + \\ + & - & + & - \\ + & + & - & - \\ + & - & - & + \end{bmatrix}$$



$$H_8 = \begin{bmatrix} H_4 & H_4 \\ H_4 & -H_4 \end{bmatrix} = \begin{bmatrix} + & + & + & + & + & + & + & + \\ + & - & + & - & + & - & + & - \\ + & + & - & - & + & + & - & - \\ + & - & - & + & + & - & - & + \\ + & + & + & + & - & - & - & - \\ + & - & + & - & - & + & - & + \\ + & + & - & - & - & - & + & + \\ + & - & - & + & - & + & + & - \end{bmatrix}$$

Oznaczając  $k$ -tą funkcję Walsha w uporządkowaniu naturalnym przez  $\text{had}(k,n)$ , możemy powiedzieć, że  $k$ -ty wiersz macierzy Hadamarda odpowiada  $k$ -tej funkcji Walsha, a więc dla  $N = 8$  mamy:

$$H_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} = \begin{bmatrix} \text{had } 0,n \\ \text{had } 1,n \\ \text{had } 2,n \\ \text{had } 3,n \\ \text{had } 4,n \\ \text{had } 5,n \\ \text{had } 6,n \\ \text{had } 7,n \end{bmatrix} \quad (2.7)$$

Formalnie proces obliczania wartości poszczególnych funkcji  $\text{had}(k,n)$  można zapisać [4]:

$$\begin{aligned} \text{had}(0,n) &= 1 \\ \text{had}\left(\frac{N}{2}, n\right) &= \begin{cases} 1 & n = 0, 1, \dots, \frac{N}{2} - 1 \\ -1 & n = \frac{N}{2}, \frac{N}{2} + 1, \dots, N-1 \end{cases} \end{aligned} \quad (2.8)$$

$$\text{had}(k,n) = \text{had}(2k, 2n) \quad \text{had}(0,n), \quad k = 1, 2, \dots, \frac{N}{2} - 1$$

$$\text{had}(k,n) = \text{had}\left(2k - \frac{N}{2}, 2n\right) \text{had}\left(\frac{N}{2}, n\right), \quad k = \frac{N}{2} + 1, \dots, N-1$$

### 2.2.2. Uporządkowanie diadyczne

Funkcje Rademachera, będące podstawą konstrukcji funkcji Walsha w uporządkowaniu diadycznym, mogą być uzyskane z funkcji sinusoidalnej wg wzoru [2]:

$$R(k,n) = \text{sign} \left[ \sin\left(2^k \pi \frac{n}{N}\right) \right] \quad (2.9)$$

Funkcje te przyjmują wartości +1 lub -1 i mają tę samą liczbę zmian znaku jak sinusoida o odpowiedniej częstotliwości. Oznaczając przez  $\text{pal}(k,n)$  układ funkcji Walsha w uporządkowaniu diadycznym wzór definiujący ten układ ma postać:

$$\text{pal}(u,n) = \sum_{r=0}^{p-1} k_r R(r,n) \quad (2.10)$$

gdzie  $k_r$  są współczynnikami rozwinięcia binarnego liczby  $k$ :

$$k = \sum_{r=0}^{p-1} k_r 2^r$$

Przykładowo, aby znaleźć funkcję dla  $k = 13 = 1101_2$ , musimy obliczyć iloczyn trzech funkcji Rademachera:

$$\text{pal}(13,n) = R(4,n) R(3,n) R(1,n)$$

Sposób generacji funkcji  $\text{pal}(k,n)$  można również opisać wzorami analogicznymi do funkcji  $\text{had}(k,n)$  w rozdz. 2.2.1 [4]:

$$\begin{aligned} \text{pal}(0,n) &= 1 \quad n = 0, 1, \dots, N-1 \\ \text{pal}(1,n) &= \begin{cases} 1 & n = 0, 1, \dots, \frac{N}{2} - 1 \\ -1 & n = \frac{N}{2}, \frac{N}{2} + 1, \dots, N-1 \end{cases} \end{aligned} \quad (2.11)$$

$$\text{pal}(k,n) = \text{pal}\left(\frac{k}{2}, 2n\right) \text{pal}(0,n)$$

$$\text{pal}(k+1,n) = \text{pal}\left(\frac{k}{2}, 2n\right) \text{pal}(1,n), \quad n = 2, 4, \dots, N-2$$

### 2.2.3. Uporządkowanie sekwencyjne

Układ funkcji Walsha w uporządkowaniu sekwencyjnym można zdefiniować następująco:

$$\begin{aligned} \text{wal}(k,n) &= \text{wal}(k_{p-1}, k_{p-2}, \dots, k_0; n_{p-1}, n_{p-2}, \dots, n_0) = \\ &= \sum_{r=0}^{p-1} (11) k_{p-1-r} (n_r + n_{r+1}), \end{aligned} \quad (2.12)$$



gdzie:

$$k = \sum_{r=0}^{p-1} k_r 2^r ; \quad n = \sum_{r=0}^{p-1} n_r 2^r$$

Istnieje prosta zależność pomiędzy układami w uporządkowaniu diadycznym i sekwencyjnym. Związek ten można wyrazić następująco:

$$\text{wal}(k, n) = \text{pal}(g(k), n),$$

gdzie  $g(k)$  oznacza liczbę  $k$  zapisaną w cyklicznym kodzie Graya, czyli dla  $k = (k_{p-1}, \dots, k_0)_2$  mamy:

$$g_1(k) = k_i \oplus k_{i+1},$$

gdzie  $\oplus$  - dodawanie modulo-2. Np. dla  $k = 6_{10} = 0110$  mamy:

$$\begin{array}{r} 0110 \\ \oplus 0110 \\ \hline 0101_2 = 5 \end{array} \quad g(6) = 5,$$

a więc

$$\text{wal}(6, n) = \text{pal}(5, n)$$

Proces tworzenia układu w uporządkowaniu sekwencyjnym można zapisać również w postaci podobnej jak w przypadku omówionych już uporządkowań. Odpowiednie zależności mają postać [4]:

$$\begin{aligned} \text{wal}(0, n) &= 1 & n &= 0, 1, \dots, N-1 \\ \text{wal}(1, n) &= \begin{cases} 1 & n = 0, 1, \dots, \frac{N}{2} - 1 \\ 1 & n = \frac{N}{2}, \dots, N-1 \end{cases} & (2.13) \\ \text{wal}(m, n) &= \text{wal}\left(\frac{m}{2}, 2n\right) \text{wal}\left(1 + (-1)^{\frac{m}{2} + 1} / 2, n\right) \\ \text{wal}(m+1, n) &= \text{wal}\left(\frac{m}{2}, 2n\right) \text{wal}\left(1 + (-1)^{\frac{m}{2}} / 2, n\right), \quad m = 2, 4, \dots, N-2 \end{aligned}$$

### 3. Dyskretne przekształcenia sumacyjne

Na początku poprzedniego rozdziału podano wzory (2.2) - (2.4) określające rozkład funkcji dyskretnej  $f(n)$  określonej na zbiorze  $\{n : n = 0, 1, \dots, N-1\}$ , o własnościach (2.1), względem układu funkcji ortogonalnych. Rozwinięcie to można traktować jako przyporządkowanie funkcji  $f(n)$  pewnej funkcji dyskretnej  $F(k)$ , której wartości są równe współczynnikom rozwinięcia  $f(n)$  względem danego układu. Można to zapisać w następujący sposób:

$$F(k) = \sum_{n=0}^{N-1} f(n) g(k,n), \quad k = 0, \dots, N-1 \quad (3.1)$$

Przyporządkowanie to nazywa się często dyskretnym przekształceniem sumacyjnym. Przekształceniem odwrotnym nazywamy przyporządkowanie transformacie  $F(k)$  oryginału  $f(n)$  w następujący sposób:

$$f(n) = \sum_{k=0}^{N-1} F(k) h(k,n) \quad n = 0, \dots, N-1 \quad (3.2)$$

Funkcje  $g(k,n)$  oraz  $h(k,n)$  nazywamy jądrami przekształceń. Nazwa przekształcenia zależy od wybranego jądra.

W dalszej części rozważań zajmiemy się dyskretnym przekształceniem Walsha.

#### 3.1. Dyskretne przekształcenie Walsha

Dyskretnym przekształceniem Walsha nazywamy przekształcenie (3.1), w którym układ funkcji  $g(k,n)$  jest ortonormalnym układem funkcji Walsha. W zależności od uporządkowania dyskretnej funkcji Walsha wzór ten przybiera postać:

$$F_H(k) = \frac{1}{N} \sum_{n=0}^{N-1} f(n) h_{ad}(k,n) \quad (3.3a)$$

$$F_P(k) = \frac{1}{N} \sum_{n=0}^{N-1} f(n) p_{al}(k,n) \quad (3.3b)$$

$$F_w(k) = \frac{1}{N} \sum_{n=0}^{N-1} f(n) w_{al}(k,n) \quad k = 0, 1, \dots, N-1 \quad (3.3c)$$

Uporządkowanie otrzymanego w ten sposób widma jest zgodne z uporządkowaniem jądra.



W przypadku przekształcenia Walsh'a jądro przekształcenia odwrotnego  $h(k,n)$  pokrywa się z jądrem przekształcenia prostego z dokładnością do współczynnika  $\frac{1}{N}$ . Mamy więc:

$$f(n) = \sum_{k=0}^{N-1} F_H(k) \text{ had}(k,n) \quad (3.4a)$$

$$f(n) = \sum_{k=0}^{N-1} F_P(n) \text{ pal}(k,n) \quad (3.4b)$$

$$f(n) = \sum_{k=0}^{N-1} F_W(k) \text{ wal}(k,n) \quad (3.4c)$$

### 3.2. Szybkie dyskretne przekształcenie Walsh'a

Dużym ograniczeniem w stosowaniu transformacji Walsh'a (3.3) jest jej złożoność obliczeniowa. Pomijając skalowanie przez  $N$  wzór (2.3c), określający transformację Walsh'a w uporządkowaniu sekwencyjnym, można napisać w postaci:

$$F_W(k) = \sum_{n=0}^{N-1} f(n) \text{ wal}(k,n), \quad k = 0, \dots, N-1 \quad (3.5)$$

Zależność ta jest układem  $N$  równań odpowiadających  $N$  różnym wartościom zmiennej  $k$ . Łatwo można zauważyć, że w celu uzyskania wartości  $F_W(k)$  dla pojedynczej wartości  $k$  należy wykonać  $N$  operacji dodawania lub odejmowania. Uzyskanie więc transformaty Walsh'a funkcja  $f(n)$  dla  $k = 0, \dots, N-1$  wymaga  $N^2$ -tych operacji.

Z drugiej strony widoczna jest nieefektywność takiego sposobu liczenia. Dla różnych wartości  $k$  wykonywane są te same działania, np. sumowanie  $f(0) + f(1)$  wykonywane jest  $\frac{N}{2}$  razy. Umiejętna dekompozycja obliczeń może prowadzić do znacznego zmniejszenia liczby potrzebnych działań. Stosowane w praktyce algorytmy pozwalają na uzyskanie transformaty  $F_W(k)$  po wykonaniu  $N \cdot \log_2 N$  operacji. Posiadają one wspólną nazwę FWT - Szybka Transformacja Walsh'a.

Poniżej zostaną przedstawione trzy szybkie algorytmy pozwalające wyliczyć przekształcenie Walsh'a.

#### 3.2.1. Algorytm 1

Przyjmując zależność określającą funkcje Walsh'a  $\text{wal}(k,n)$  zgodnie ze wzorem (2.12) otrzymujemy (3.5) w postaci:

$$F_W(k) = \sum_{r=0}^{p-1} \sum_{n=1}^{N-1} (-1)^k p^{-1-r} \binom{n+n}{r} f(n) \quad (3.6)$$

Podzielmy proces obliczeń na  $p$ -kroków. Wyniki  $r$ -tego kroku będą wykorzystywane jako dane wejściowe kroku  $(r+1)$ -ego. Algorytm postępowania będzie identyczny w każdym kroku. Do pierwszego kroku  $r=0$  mamy:

$$A_1(k_{p-1}, n) = \sum_{n=0}^{N-1} (-1)^{k_{p-1}(n_0+n_1)} f(n) \quad (3.7)$$

Rozpatrując (2.7) dla sąsiednich wartości  $f(n)$  oraz  $f(n+1)$  możemy napisać:

$$A(k_{p-1}, n_{p-1} \dots n_0) = \sum_{n_0=0}^1 -1^{k_{p-1}(n_0+n_1)} f(k_{p-1}, k_{p-2} \dots n_1, n_0) \quad (3.8)$$

Wtedy wartość wyrażenia:

$$(-1)^{k_{p-1}(n_0+n_1)}, \quad n_0 = 0, 1; \quad k_{p-1} = 0, 1$$

decyduje czy wartości  $f(n)$ ,  $f(n+1)$  zostaną zsumowane, czy odjęte dla danej wartości  $A_0$ .

Ten proces jest powtarzany dla kolejnych par  $f(n)$ ,  $f(n+1)$ . Ogólnie w  $r$ -tym kroku mamy:

$$A_r(k_{p-1}, k_{p-2} \dots k_{p-r}, n_{p-1}, n_{p-2}, \dots n_r) = \sum_{n_{r-1}=0}^1 (-1)^{k_{p-r}(n_r+n_{r-1})} \cdot A_{r-1}(k_{p-1}, \dots, k_{p-r+1}, n_{p-1} \dots n_{r-1}) \quad (3.9)$$

W wyniku takiego postępowania otrzymujemy wartość  $A_r(k_{p-1}, \dots, k_0)$ . Po podzieleniu jej przez  $N$  otrzymujemy szukaną wartość transformaty funkcji  $f(n)$ :

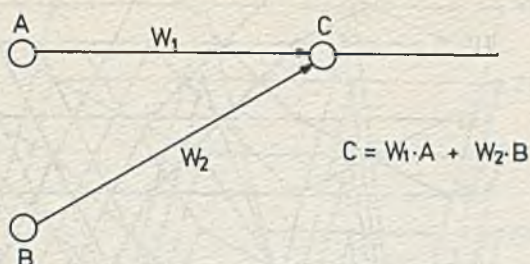
$$F_N(k_{p-1}, \dots, k_0) = \frac{1}{N} A_p(k_{p-1}, \dots, k_0)$$

Otrzymane w ten sposób widmo jest zapisane w porządku sekwencyjnym.

Powyższy algorytm można przedstawić graficznie w postaci grafu przepływu sygnałów. Podstawowy element takiego grafu ma postać przedstawioną na rys. 1.

Schemat algorytmu dla  $N = 16$  jest pokazany na rys. 2. Linie ciągłe oznaczają  $W_1 = 1$ , a linie przerywane  $W_1 = -1$ .





Rys. 1. Podstawowy fragment grafu szybkiej transformacji Walsha dla uporządkowania sekwencyjnego

Fig. 1. Basic fragment of the graph for fast Walsh transform for sequential ordering

### 3.2.2. Algorytm 2

Rozumowanie prowadzące do stworzenia algorytmu zobrazowanego graficznie na rys. 3 jest podobne jak w rozdziale 3.2.1. Inny jest sposób doboru punktów. Tak więc w kroku pierwszym nie operujemy na sąsiednich wartościach  $f(n)$ ,  $f(n+1)$ , lecz na  $f(n)$  oraz  $f(n + \frac{N}{2})$ , w kroku drugim na wartościach pośrednich  $A_0, A_{0+\frac{N}{2}}$  itd.

Uzyskane widmo funkcji  $f(n)$  jest zapisane również w porządku sekwencyjnym.

### 3.2.3. Algorytm 3

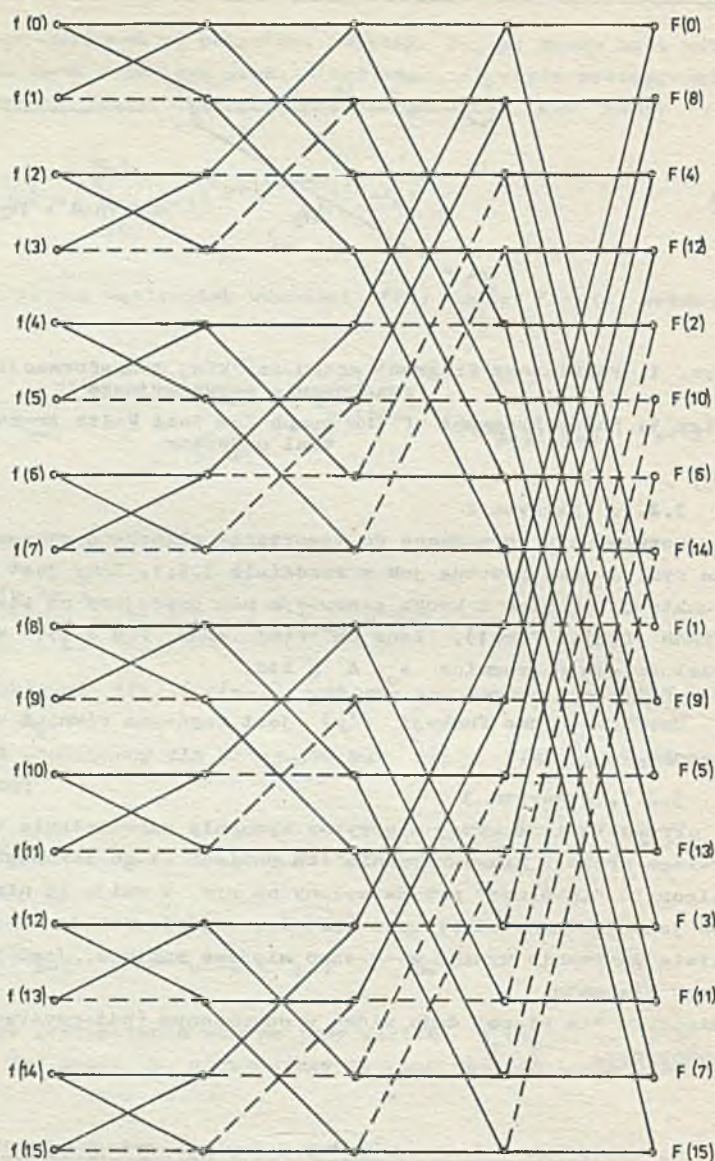
Przedstawione wyżej algorytmy wymagały zapamiętania wyników pośrednich  $r$ -tego kroku i niewymazywania ich pamięci aż do zakończenia kroku  $r+1$ . Algorytm "in-place" przedstawiony na rys. 4 omija tę niedogodność. Wymaga on jednokrotnego odczytania wartości wejściowych do danego kroku i umożliwia zapisanie wyniku w to samo miejsce pamięci. Jest to istotna zaleta tego elementu.

Algorytm "in place" daje widmo w odwróconym (bit-reversed) porządku sekwencyjnym.

## 4. Wektorowe układy dyskretnych funkcji ortogonalnych

Prowadzone dotychczas rozważania dotyczyły funkcji dyskretnych o argumentie skalarnym. W prosty sposób można je rozszerzyć na przypadek dyskretnych funkcji o argumentie wektorowym.

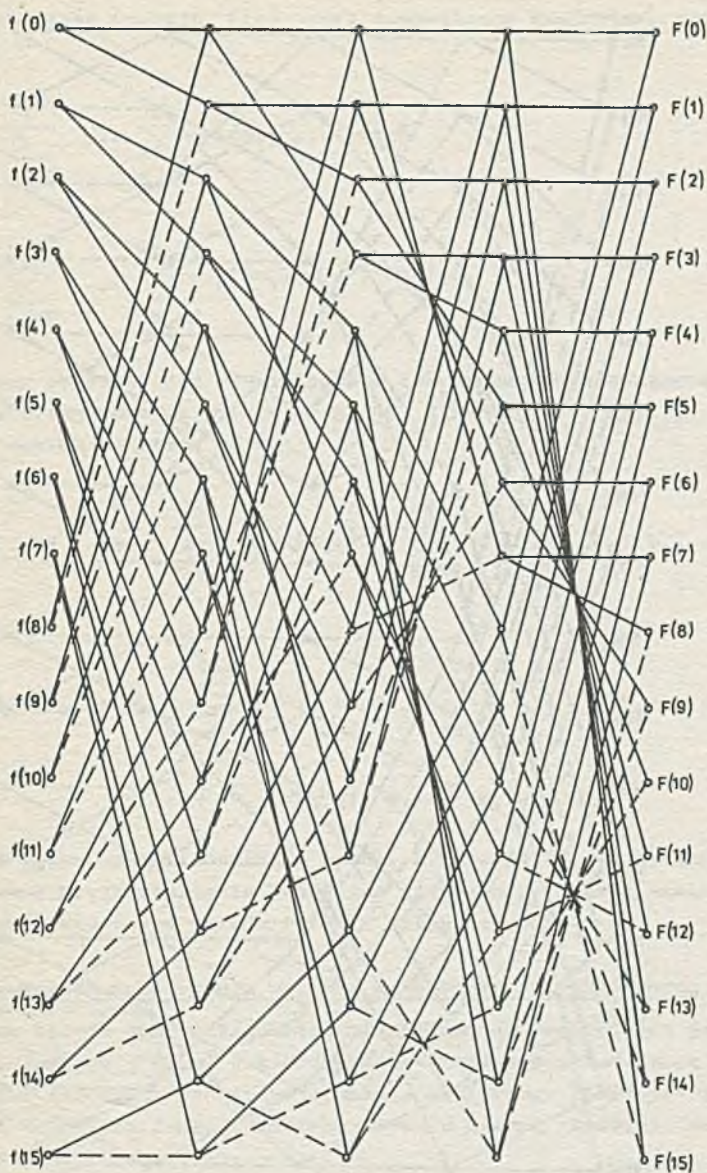
Funkcję dyskretną  $f(n)$ , gdzie  $n^T = [n_1, \dots, n_s]$ , określoną na zbiorze  $\{n : n = 0, 1, \dots, N, \quad N^T = [N_1, \dots, N_s]\}$  nazywamy funkcją o argumentie wektorowym.

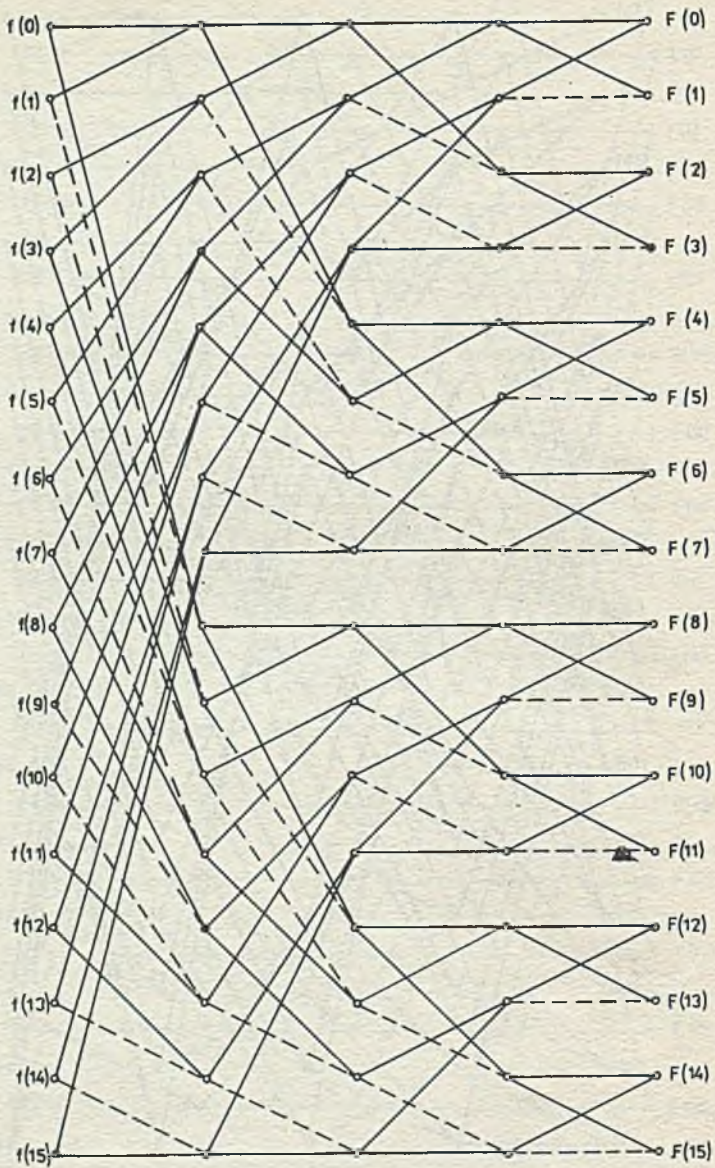


Rys. 2. Graf szybkiej transformacji Walsha dla uporządkowania sekwencyjnego i  $N = 16$

Fig. 2. The graph of the fast Walsh transform for the sequential ordering and  $N = 16$



Rys. 3. Graf szybkiej transformacji Walsha dla  $N=16$ Fig. 3. The graph of fast Walsh transform for  $N = 16$



Rys. 4. Graf szybkiej transformacji Walsha realizujący algorytm "in-place"

Fig. 4. The graph of the fast Walsh transform realizing "in place" algorithm



Jeżeli taka funkcja  $f(n)$  jest sumowalna z kwadratem, tzn. istnieją sumy:

$$\sum_{n_1=0}^{N_1-1} \dots \sum_{n_s=0}^{N_s-1} f(n_1, \dots, n_s)$$

(4.1)

oraz

$$\sum_{n_1=0}^{N_1-1} \dots \sum_{n_s=0}^{N_s-1} |f(n_1, \dots, n_s)|^2,$$

to możemy określić jej rozwinięcie w szereg względem wektorowego układu funkcji ortogonalnych  $\varphi(k, n)$ , gdzie  $k^T = [k_1, \dots, k_s]$ , określonych w tych samych punktach w  $f(n)$ :

$$f(n_1, \dots, n_s) = \sum_{n=0}^{N-1} a_n \varphi(k, n) = \sum_{k_1=0}^{N_1-1} \dots \sum_{k_s=0}^{N_s-1} a_k \varphi(k_1, \dots, k_s; n_1, \dots, n_s)$$

(4.2)

gdzie:

$$a_k = A_{k_1} \dots k_s = \sum_{n=0}^{N-1} f(n) (k, n) = \sum_{n_1=0}^{N_1-1} \sum_{n=0}^{N-1} f(n_1, \dots, n_s) \cdot \varphi(k_1, \dots, k_s; n_1, \dots, n_s)$$

(4.3)

Zgodnie z tym co powiedziano w rozdz. 2 para (4.3), (4.2) określa proste i odwrotne przekształcenie sumacyjne dyskretnej funkcji wielu zmiennych względem wektorowego układu funkcji ortogonalnych.

#### 4.1. Metoda konstrukcji układów ortogonalnych wektorowych

Niech będzie dany układ ortogonalny funkcji dyskretnych jednoargumentowych. Funkcje  $\varphi(k, n)$  wektorowego ortogonalnego układu mogą być przyjęte w postaci iloczynów  $s$  funkcji układu skalarnego [5]. Dyskretnymi argumentami kolejnych funkcji występujących w każdym iloczynie są kolejno  $n_1, \dots, n_s$ . Mamy więc:

$$\begin{aligned} \varphi(k, n_1, \dots, n_s) &= \varphi(k_1, \dots, k_s; n_1, \dots, n_s) = \\ &= \varphi(k_1, n_1) \dots \varphi(k_s, n_s) \end{aligned}$$

(4.4)

Ułatwieniem w konstruowaniu układów ortogonalnych wektorowych dla funkcji dyskretnych jest fakt, że liczba funkcji układu skalarnego jest skończona.

W charakterze przykładu ilustrującego powyższe rozważania przedstawimy konstrukcję ortogonalnego układu wektorowego dla  $s = 2$ . Mamy:

$$\varphi(k, n) = \varphi(k_1, k_2; n_1, n_2) = \varphi(k_1, n_1) \varphi(k_2, n_2) \quad (4.5)$$

$$k_1, n_1 = 0, 1, \dots, N_1 - 1$$

$$k_2, n_2 = 0, 1, \dots, N_2 - 1$$

Uwzględniając (3.5) we wzorze (2.3) dostajemy:

$$\begin{aligned} r(n_1, n_2) &= \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} a_{k_1, k_2} \varphi(k_1, n_1) \cdot \varphi(k_2, n_2) = \\ &= a_{00} \varphi(0, n_1) \cdot \varphi(0, n_2) + a_{01} \varphi(0, n_1) \cdot \varphi(1, n_2) + \\ &+ \dots + a_{N_1 N_2} \varphi(N_1, n_1) \cdot \varphi(N_2, n_2), \end{aligned} \quad (4.6)$$

gdzie:

$$\begin{aligned} a_{k_1, k_2} &= \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} r(n_1, n_2) \varphi(k_1, n_1) \varphi(k_2, n_2) = \\ &= r(0, 0) \varphi(k_1, 0) \varphi(k_2, 0) + r(0, 1) \varphi(k_1, 0) \varphi(k_2, 1) + \\ &+ \dots + r(N_1, N_2) \varphi(k_1, N_1) \varphi(k_2, N_2) \end{aligned} \quad (4.7)$$

#### 4.2. Dwuwymiarowe przekształcenie Walsha

W oparciu o to co przedstawiliśmy dotychczas możemy podać wzory określające proste i odwrotne dwuwymiarowe przekształcenie Walsha. W zależności od uporządkowania dyskretnych funkcji Walsha mamy:

- dla przekształcenia prostego:

$$F_H(k, m) = \frac{1}{N^2} \sum_{n=0}^{N-1} \sum_{i=0}^{N-1} f(n, i) \text{had}(k, n) \text{had}(m, i)$$

$$F_P(k, m) = \frac{1}{N^2} \sum_{n=0}^{N-1} \sum_{i=0}^{N-1} f(n, i) \text{pal}(k, n) \text{pal}(m, i) \quad (4.8)$$



$$F_w(k,m) = \frac{1}{N^2} \sum_{n=0}^{N-1} \sum_{i=0}^{N-1} f(n,i) \text{ wal}(k,n) \text{ wal}(m,i)$$

$$N_1 = N_2 = N = 2^p$$

- dla przekształcenia odwrotnego:

$$f(n,i) = \sum_{n=0}^{N-1} \sum_{i=0}^{N-1} F_H(k,m) \text{ had}(k,n) \text{ had}(m,i)$$

$$f(n,i) = \sum_{k \neq 0}^{N-1} \sum_{i=0}^{N-1} F_p(k,m) \text{ pal}(k,n) \text{ pal}(m,i) \quad (4.9)$$

$$f(n,i) = \sum_{k=0}^{N-1} \sum_{i=0}^{N-1} F_w(k,m) \text{ wal}(k,n) \text{ wal}(m,i)$$

Ze względu na sposób konstruacji dwuwymiarowego układu funkcji Walsha obliczenia widma funkcji  $f(n,i)$  można podzielić na dwa etapy. W tym celu przekształciliśmy wzór (3.8) do postaci:

$$F_w(k,m) = \frac{1}{N} \sum_{i=0}^{N-1} \left\{ \frac{1}{N} \sum_{n=0}^{N-1} f(n,i) \text{ wal}(k,n) \right\} \text{ wal}(m,i) =$$

$$= \frac{1}{N} \sum_{i=0}^{N-1} F_w^*(k,i) \text{ wal}(m,i)$$

W kroku pierwszym obliczamy skalarną transformatę Walsha wzdłuż kolejnych kolumn funkcji  $f(n,i)$ , otrzymując widmo częściowe  $F_w^*(k,i)$ . W kroku drugim liczymy podobnie wzdłuż wierszy widma  $F_w^*(k,i)$ , otrzymując poszukiwaną funkcję  $F_w(k,m)$ .

W opisanym powyżej procesie można stosować szybkie algorytmy przedstawione w 2.2. Tak więc algorytm obliczeń dwuwymiarowego dyskretnego przekształcenia Walsha wymaga dwukrotnego zastosowania algorytmu dla przykładu skalarnego.

## 5. Zakończenie

Przedstawione w pracy uporządkowanie dyskretnych funkcji Walsha oraz odpowiadające im algorytmy szybkich przekształceń są wykorzystywane w

w programach numerycznych przetwarzających dyskretne obrazy dwuwymiarowe opisane funkcją obrazu  $f(n,i)$ . Wybór konkretnego uporządkowania jest złożony i nie podlega ogólnym zależnościom. Mają na niego wpływ zarówno cele, jakim służy przetwarzanie, jak i rodzaj analizowanego obrazu.

## LITERATURA

- [1] Beauchamp K.G.: Walsh Functions and their applications. Academic Press 1975.
- [2] Harmuth H.F.: Sequency Theory. Academic Press 1977.
- [3] Sing-Tze Bow: Pattern Recognition. Marcel Dekker, Inc. 1984.
- [4] Drygałło A., Pietraszek St.: Uporządkowanie naturalne, diadyzjne i sekwencyjne w algorytmach szybkiego przekształcenia Walsh'a. Archiwum Automatyki i Telemekhaniki, t.XXVI, 1981.
- [5] Wojciechowski K.: Rozpoznanie obrazów. Skrypt Pol. Śl., Gliwice 1977.

Recenzent: Doc. dr hab. inż. Mariusz Nieniewski

Wpłynęło do Redakcji 3.11.1987 r.

ДИСКРЕТНОЕ СУММАЦИОННОЕ ПРЕОБРАЗОВАНИЕ  
ДЛЯ ФУНКЦИИ УОЛША

## Р е з ю м е

Представлен способ записи дискретных скалярных функций Уолша в естественном диодическом и последовательном упорядочениях. Предложен алгоритм прямого и обратного быстрого преобразования Уолша. Показан способ создания преобразования Уолша на базе таких систем.

## DISCRETE SUMMATION TRANSFOR FOR THE WALSH FUNCTIONS

## S u m m a r y

The way of description of discrete scalar Walsh functions in natural, dyadic and sequential ordering is presented. Simple and inverse fast Walsh Transform algorithm is presented. The way of discrete vector Walsh function generation is presented and algorithm of fast Walsh transformation using these systems is proposed.