

Article

Matrix Factorization and Some Fast Discrete Transforms

Iliya Bouyukliev ^{1,*} , Mariya Dzhumalieva-Stoeva ²  and Paskal Piperkov ² 
¹ Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria

² Faculty of Mathematics and Informatics, St. Cyril and St. Methodius University of Veliko Tarnovo, 5003 Veliko Tarnovo, Bulgaria; m.dzhumalieva@ts.uni-vt.bg (M.D.-S.); p.piperkov@ts.uni-vt.bg (P.P.)

* Correspondence: iliyab@math.bas.bg

Abstract: In this paper, three discrete transforms related to vector spaces over finite fields are studied. For our purposes, and according to the properties of the finite fields, the most suitable transforms are as follows: for binary fields, this is the Walsh–Hadamard transform; for odd prime fields, the Vilenkin–Chrestenson transform; and for composite fields, the trace transform. A factorization of the transform matrices using Kronecker power is given so that the considered discrete transforms are reduced to the fast discrete transforms. Examples and applications are also presented of the considered transforms in coding theory for calculating the weight distribution of a linear code.

Keywords: fast discrete transform; Walsh–Hadamard; Vilenkin–Chrestenson; trace; linear code; weight distribution

MSC: 65T50; 15B33; 94B03

1. Introduction

The Fourier-type discrete transforms have various applications [1], including signal processing, communications, and cryptography. As their special application, we indicate the implementation for digital devices related to spectral logic [2]. The discrete transformations are also important in studying the cryptographic properties of Boolean functions [3–5], and they are used in the study of self-dual bent functions (see [6]). A description of fast Fourier-type discrete transforms can be found in the popular book by Joux [3] (Chapter 9).

Fast discrete transforms are widely used in coding theory, especially for computing some important parameters of linear codes over finite fields. The problems for computing the weight distribution, minimum distance, and covering radius of a linear code are NP-complete [7]. Algorithms that solve these problems are usually based on an exhaustive search [8–10]. Connections between discrete Fourier-type transforms, on the one hand, and weight distribution and covering radius, on the other, were established by Karpovsky [11,12], but he mainly considered binary codes. The algorithms based on fast discrete transforms are well suited for linear codes with small rates (large length and small dimension). The present study is a continuation of the recent work of the authors on the transforms' applications in coding theory [13,14].

Fast discrete transforms, in most cases, can be represented as a matrix by vector multiplication, where the matrix is recursively defined by a Kronecker product. The transition from discrete to fast discrete transform is based on the fact that the transform matrix defined recursively by a Kronecker product can be factorized into a product of sparse matrices. We prove that the sparse matrices in this product commute, which is important for some implementations [15]. As a result, this manages to reduce the complexity of computing the discrete transform from $O(n^2)$ to $O(n \log n)$, where n is the size of the transform matrix.



Citation: Bouyukliev, I.; Dzhumalieva-Stoeva, M.; Piperkov, P. Matrix Factorization and Some Fast Discrete Transforms. *Axioms* **2024**, *13*, 495. <https://doi.org/10.3390/axioms13080495>

Academic Editor: Hyun Soo Chung

Received: 4 July 2024

Revised: 18 July 2024

Accepted: 21 July 2024

Published: 23 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

The fast discrete transforms are implemented with butterfly diagrams and algorithms (see, for example, [16]). These types of algorithms have very efficient natural implementations with the SIMD model of parallelization, especially with the CUDA platform. The speedup of the parallel implementation of Walsh–Hadamard transform in GPU can be seen in [15].

In addition to the basic properties of the considered discrete transforms, we dwell in more detail on the following cases concerning linear codes over a finite field with q elements:

- (1) If $q = 2$, we consider the Walsh–Hadamard (Walsh) transform. This transform is a square-wave analog to the Fourier transform, based on Walsh functions [17,18]. The Walsh transform matrices are related to the Hadamard matrices of the Sylvester type. The Hadamard matrix H_n , $n \in \mathbb{N}$, is the n -th Kronecker power of the matrix $H_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. This allows us to represent H_n as a product of rare matrices [19], which leads to a fast algorithm. As the computations of the transform are addition and subtraction only, it can be implemented by a butterfly algorithm [3,16]. Therefore, this transform is best suited for vector spaces over the binary field.
- (2) If q is an odd prime, we use the Vilenkin–Chrestenson transform [20,21]. This transform is defined over the ring of integers modulo q $\mathbb{Z}_q = \{0, 1, 2, \dots, q-1\}$. As it is well known, $\mathbb{Z}_q$ is a field only when q is a prime. We consider linear codes over a field, so this transform is suitable for our research for prime q . The Walsh–Hadamard transform is a special case of the Vilenkin–Chrestenson transform, namely for $q = 2$. Information about the Vilenkin–Chrestenson transform, as well as other discrete Fourier-type transforms, is given in [2,22,23].
- (3) If $q = p^s$ for a prime p and positive integer $s > 1$, we use the trace transform for linear codes over the field $\mathbb{F}_q$.

The aim of this paper is to present methods for factorization of transform matrices when they are represented as a Kronecker product and their application for faster computation of the weight distribution of linear codes over prime and composite finite fields. We also give specific examples for better understanding. This consideration is essential for the modification of algorithms for computing the weight spectrum and covering radius of a linear code in some cases [13,14].

Section 2 presents the information we need about finite fields and linear codes. In Section 3, we present some useful properties of the Kronecker product. The factorization of the k -th Kronecker power of a square matrix is also given in this part. Sections 4–6 are devoted to the Walsh–Hadamard, Vilenkin–Chrestenson, and trace transform, respectively.

2. Finite Fields and Linear Codes

Let $\mathbb{F}_q$ be a finite field, or Galois field, with q elements and the prime p be its characteristic. Detailed information about finite field properties is given in [24]. Any finite extension $\mathbf{F}$ over the field $\mathbf{K}$ is a linear space over $\mathbf{K}$ and a basis of the field $\mathbf{F}$ is any basis of this linear space. In particular, the field $\mathbb{F}_{p^m}$, as a finite m -powered extension of its prime subfield $\mathbb{F}_p$, is isomorphic to the linear space $\mathbb{F}_p^m$. If $\beta_1, \beta_2, \dots, \beta_m \in \mathbb{F}_{p^m}$ is a basis of $\mathbb{F}_{p^m}$ over the prime subfield $\mathbb{F}_p$ and $\alpha \in \mathbb{F}_{p^m}$, then $\alpha = \lambda_1\beta_1 + \lambda_2\beta_2 + \dots + \lambda_m\beta_m$, where $\lambda_1, \lambda_2, \dots, \lambda_m \in \mathbb{F}_p$.

Definition 1. Let $K = \mathbb{F}_q$, $F = \mathbb{F}_{q^m}$ and $\alpha \in F$. Trace $Tr_{F/K}(\alpha)$ of the element α over K is

$$Tr_{F/K}(\alpha) = \alpha + \alpha^q + \alpha^{q^2} + \dots + \alpha^{q^{m-1}}.$$

If K is a prime subfield of F , then $Tr_{F/K}(\alpha)$ is called absolute trace of the element α and it is denoted by $Tr_F(\alpha)$.

If the field is given in advance, the absolute trace is denoted only by $Tr(\alpha)$ and called just trace.

Definition 2. Two bases $\beta_1, \dots, \beta_m \in F$ and $\beta'_1, \dots, \beta'_m \in F$ of the field F over the subfield K are called dual, if

$$\text{Tr}_{F/K}(\beta_i \beta'_j) = \begin{cases} 0, & \text{if } i \neq j, \\ 1, & \text{if } i = j. \end{cases}$$

for any $1 \leq i, j \leq m$. A basis that is dual of itself is called a self-dual basis.

In other words, the trace $\text{Tr}(\beta_i \beta_j)$ is equal to the Kronecker δ of i and j for all $i, j \in \{1, 2, \dots, m\}$. Each basis has exactly one dual basis. The following theorem gives a sufficient and necessary condition for the existence of a self-dual basis.

Theorem 1 ([25] (Theorem 5.1.18), [26]). *There exists a self-dual basis of $F = \mathbb{F}_{q^m}$ over $K = \mathbb{F}_q$ if and only if either q is even or both q and m are odd.*

For the purposes of the transforms, the elements of $\mathbb{F}_q$ are ordered, i.e., $\mathbb{F}_q = \{\alpha_0 = 0, \alpha_1, \dots, \alpha_{q-1}\}$. In most cases, when q is composite, we use the lexicographic ordering of the elements, depending on the chosen basis. If $\beta_1, \dots, \beta_m$ is a basis of $\mathbb{F}_q$, $q = p^m$, over $\mathbb{F}_p$ for the prime p , then any elements $\alpha \in \mathbb{F}_q$ can be written as $\alpha = \lambda_1(\alpha)\beta_1 + \dots + \lambda_m(\alpha)\beta_m$, where $\lambda_i(\alpha) \in \mathbb{F}_p$, $i = 1, \dots, m$. There is one-to-one correspondence σ_β between $\mathbb{F}_{p^m}$ and $\mathbb{F}_p^m$ defined by this basis. For $\alpha \in \mathbb{F}_{p^m}$, $\sigma_\beta(\alpha) = (\lambda_1(\alpha), \dots, \lambda_m(\alpha))$. To order the elements of $\mathbb{F}_q$, we use the lexicographic ordering of the vectors in $\mathbb{F}_p^m$, namely

$$(0, \dots, 0, 0) < (0, \dots, 0, 1) < (0, \dots, 0, 2) < \dots < (p-1, \dots, p-1).$$

The position of the zero element is constant, whereas all other elements could change their positions if the basis is changed.

Let $\mathbb{F}_q^n$ be the n -dimensional vector space over the finite field $\mathbb{F}_q = \{0, \alpha_1, \alpha_2, \dots, \alpha_{q-1}\}$, where q is a power of a prime number. Any k -dimensional linear subspace $C \subset \mathbb{F}_q^n$ is called linear code, and it is said to be $[n, k]_q$ code with length n and dimension k . A vector $v \in C$ is called a codeword of C . The (Hamming) weight $wt(v)$ of a vector $v \in \mathbb{F}_q^n$ is the number of the nonzero coordinates of v . The weight distribution of the code C is the finite sequence $(A_0, A_1, \dots, A_n) \in \mathbb{Z}^{n+1}$, where A_u is the number of the codewords with weight u , $u = 0, 1, \dots, n$, and the weight enumerator of the code C is the polynomial $W_C(y) = 1 + A_1 y^1 + \dots + A_n y^n$.

A linear code can be represented by its generator matrix, i.e., a $k \times n$ matrix whose rows form a basis of C as a vector space over $\mathbb{F}_q$. We use another representation of the $[n, k]$ linear code C that depends on the chosen generator matrix G . We define a characteristic function $f_G : \mathbb{F}_q^k \rightarrow \mathbb{Z}$ of the code C with respect to its generator matrix G in the following way.

Definition 3. *The value of the characteristic function $f_G(x)$ of the linear $[n, k]_q$ code C with respect to the generator matrix G is the number of the columns in G , which are proportional (with non zero coefficients) of x , for $x \in \mathbb{F}_q^k$. If $f_G(0) = 0$ we say that the code has full length.*

Obviously, if x_1 and x_2 are proportional vectors in $\mathbb{F}_q^k$, then $f_G(x_1) = f_G(x_2)$. Since a linear code has more than one generator matrix (in the general case), it has more than one characteristic function.

3. Factorization of Kronecker Power

Kronecker product of the matrices $A = (a_{ij})_{s_1 \times t_1}$ and $B = (b_{ij})_{s_2 \times t_2}$ is the $(s_1 s_2 \times t_1 t_2)$ matrix

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \dots & a_{1t_1}B \\ a_{21}B & a_{22}B & \dots & a_{2t_1}B \\ \dots & \dots & \dots & \dots \\ a_{s_11}B & a_{s_12}B & \dots & a_{s_1t_1}B \end{pmatrix}.$$

We give some elementary properties of the Kronecker product:

$$A \otimes (B \otimes C) = (A \otimes B) \otimes C, \quad (1)$$

$$(A + B) \otimes (C + D) = A \otimes C + A \otimes D + B \otimes C + B \otimes D, \quad (2)$$

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD). \quad (3)$$

Kronecker's product is not commutative.

The k -th Kronecker power of a square matrix M is defined as follows:

$$\otimes^2 M = M \otimes M, \quad \otimes^{k+1} M = M \otimes (\otimes^k M), k > 1.$$

If M is a square matrix of size t , then $\otimes^k M$ is a square matrix of size t^k .

Good [19] showed that the Kronecker power could be reduced to the usual row-by-column multiplication of some sparse matrices. This fact stands also for the Kronecker power of a square matrix.

Theorem 2. Let M be a square matrix of size t and k be a positive integer. Then

$$\otimes^k M = B_1 \cdot B_2 \cdots B_k, \quad (4)$$

where $B_l = I_{t^{l-1}} \otimes M \otimes I_{t^{k-l}}, 1 \leq l \leq k$, and I_s is the identity matrix of size s .

Proof. The proof of the theorem is made by induction. The statement is obvious for $k = 1$:

$$\otimes^1 M = B_1 = I_{t^{1-1}} \otimes M \otimes I_{t^{k-1}} = I_1 \otimes M \otimes I_1 = M.$$

Let (4) be true for all Kronecker powers smaller than k . Specifically,

$$\otimes^{k-1} M = B'_1 \cdot B'_2 \cdots B'_{k-1},$$

where $B'_l = I_{t^{l-1}} \otimes M \otimes I_{t^{k-l-1}}, l = 1, 2, \dots, k-1$. Because of (1), (3) and the inductive hypothesis, we have that

$$\begin{aligned} \otimes^k M &= M \otimes (\otimes^{k-1} M) = (M \cdot I_t) \otimes (I_{t^{k-1}} \cdot (\otimes^{k-1} M)) \\ &= (M \otimes I_{t^{k-1}}) \cdot (I_t \otimes (\otimes^{k-1} M)) \\ &= B_1 \cdot (I_t^{k-1} \otimes (B'_1 \cdot B'_2 \cdots B'_{k-1})) \\ &= B_1 \cdot (I_t \otimes B'_1) \cdot (I_t \otimes B'_2) \cdots (I_t \otimes B'_{k-1}) \\ &= B_1 \cdot (I_t \otimes M \otimes I_{t^{k-2}}) \cdot (I_t \otimes I_t \otimes M \otimes I_{t^{k-3}}) \cdots (I_t \otimes I_{t^{k-2}} \otimes M) \\ &= B_1 \cdot B_2 \cdot B_3 \cdots B_k. \end{aligned}$$

So the theorem holds for all $k \in \mathbb{N}$. $\square$

Another important result is that the order of the factors in (4) does not matter.

Theorem 3. The factors in (4) commute.

Proof. We apply the properties of the Kronecker product.

$$\begin{aligned}
 B_i \cdot B_j &= (I_{t^{i-1}} \otimes M \otimes I_{t^{n-i}}) \cdot (I_{t^{j-1}} \otimes M \otimes I_{t^{k-j}}) \\
 &= I_{t^{i-1}} \otimes ((M \otimes I_{t^{k-i}}) \cdot (I_{t^{j-i}} \otimes M \otimes I_{t^{k-j}})) \\
 &= I_{t^{i-1}} \otimes ((M \otimes I_{t^{j-i}}) \cdot (I_{t^{j-i}} \otimes M)) \otimes I_{t^{k-j}} \\
 &= I_{t^{i-1}} \otimes ((M \otimes I_{t^{j-i}}) \cdot (I_t \otimes I_{t^{j-i-1}} \otimes M)) \otimes I_{t^{k-j}} \\
 &= I_{t^{i-1}} \otimes (M \cdot I_t) \otimes (I_{t^{j-i}} \cdot (I_{t^{j-i-1}} \otimes M)) \otimes I_{t^{k-j}} \\
 &= I_{t^{i-1}} \otimes M \otimes I_{t^{j-i-1}} \otimes M \otimes I_{t^{k-j}} \\
 &= I_{t^{i-1}} \otimes (I_t \cdot M) \otimes ((I_{t^{j-i-1}} \otimes M) \cdot I_{t^{j-i}}) \otimes I_{t^{k-j}} \\
 &= I_{t^{i-1}} \otimes ((I_t \otimes I_{t^{j-i-1}} \otimes M) \cdot (M \otimes I_{t^{j-i}})) \otimes I_{t^{k-j}} \\
 &= I_{t^{i-1}} \otimes ((I_{t^{j-i}} \otimes M) \cdot (M \otimes I_{t^{j-i}})) \otimes I_{t^{k-j}} = B_j \cdot B_i.
 \end{aligned}$$

□

If we take a square $t \times t$ matrix M without zero elements, then the number of nonzero elements in each row and column in the matrices B_l is t .

$$I_u \otimes M \otimes I_s = \begin{pmatrix} M & & & \\ & M & & \\ & & \ddots & \\ & & & M \end{pmatrix} \otimes I_s = \begin{pmatrix} M \otimes I_s & & & \\ & M \otimes I_s & & \\ & & \ddots & \\ & & & M \otimes I_s \end{pmatrix},$$

where

$$M \otimes I_s = \begin{pmatrix} m_{11}I_s & m_{12}I_s & \dots & m_{1t}I_s \\ m_{21}I_s & m_{22}I_s & \dots & m_{2t}I_s \\ \dots & & \ddots & \\ m_{t1}I_s & m_{t2}I_s & \dots & m_{tt}I_s \end{pmatrix}$$

If $m_{ij} \neq 0$, then each block $m_{ij}I_s$ has exactly one nonzero element in each row and column. Hence, $M \otimes I_s$ has the same number of nonzero elements by rows and columns as the matrix M itself. Obviously, this holds for $I_u \otimes M \otimes I_s$, too.

Let M be a square $t \times t$ matrix without zero elements and v be a vector of length t^k for some integer k . To compute the product of the k -th Kronecker power $\otimes^k M$ with the vector v , one needs $t^k \cdot t^k$ multiplications and $t^k(t^k - 1)$ additions. Applying Theorem 2, $\otimes^k M$ is factorized into k square matrices of size t^k , such that each one has the same number t of nonzero elements by rows and columns as M . We have that

$$\otimes^k M \cdot v = B_1(B_2(B_3 \dots (B_{k-1}(B_k v)) \dots))$$

So, calculating $\otimes^k M \cdot v$ consists of k multiplications $B_l \cdot v'$, where v' is the vector, obtained in the previous step. Computing $B_l \cdot v'$ consists of $t^k \cdot t$ multiplications and $t^k(t - 1)$ additions. In the general case, the whole calculation consists of at most $k(t \cdot t^k + (t - 1)t^k) = kt^k(2t - 1)$ operations. Hence, $\otimes^k M \cdot v$ is computed much faster. This is the core of the presented fast discrete transforms.

4. Fast Walsh Transform

Let n be positive integer and $\mathbb{F}_2 = \{0, 1\}$ be the Galois field of two elements. There is one-to-one correspondence between the non negative integers $u < 2^n$ and the binary vectors of length n , defined by $u \mapsto \bar{u} = (u_0, u_1, \dots, u_{n-1}) \in \mathbb{F}_2^n$, where $u = u_0 2^{n-1} + u_1 2^{n-2} + \dots + u_{n-1}$ is the binary representation of u .

Let $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$ be a Boolean function of n variables. Truth Table TT_f of the function f is a 2^n -dimensional column vector whose $(i + 1)$ -th coordinate is $f(\bar{i})$, $i = 0, 1, \dots, 2^n - 1$.

Walsh transform of the function f is defined as follows:

$$\hat{f}(\omega) = \sum_{x \in \mathbb{F}_2^k} f(x)(-1)^{\langle x, \omega \rangle}, \quad \omega \in \mathbb{F}_2^k, \quad (5)$$

where $\langle x, y \rangle = x_1y_1 + x_2y_2 + \dots + x_ky_k$ and it means the Euclidean inner product of the vectors x and y . The column vector of the function $\hat{f}$ is denoted also by $TT_{\hat{f}}$.

We use a special type of Hadamard matrices, defined recurrently as follows:

$$H_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad H_k = \begin{pmatrix} H_{k-1} & H_{k-1} \\ H_{k-1} & -H_{k-1} \end{pmatrix}, \quad k > 1. \quad (6)$$

Theorem 4. *If f is a Boolean function of n variables then $TT_{\hat{f}} = H_n \cdot TT_f$.*

Proof. We prove the theorem by induction. For $k = 1$, let $f : \mathbb{F}_2 \rightarrow \mathbb{F}_2$ be a Boolean function of one variable. Then $TT_f = \begin{pmatrix} f(0) \\ f(1) \end{pmatrix}$ and

$$W_f(0) = f(0) \cdot (-1)^{0 \cdot 0} + f(1) \cdot (-1)^{1 \cdot 0} = f(0) + f(1),$$

$$W_f(1) = f(0) \cdot (-1)^{0 \cdot 1} + f(1) \cdot (-1)^{1 \cdot 1} = f(0) - f(1).$$

Obviously, $W_f = H_1 \cdot TT_f$.

Let the theorem holds for $n = k$, $k \geq 1$, and $f : \mathbb{F}_2^{k+1} \rightarrow \mathbb{F}_2$ be a Boolean function of $k + 1$ variables with truth table TT_f . We use the Boolean functions of k variables $f_0(x_1, \dots, x_k) = f(0, x_1, \dots, x_k)$ and $f_1(x_1, \dots, x_k) = f(1, x_1, \dots, x_k)$. According to the lexicographic ordering, the set of all vectors in $\mathbb{F}_2^{k+1}$ can be split into two subsets, namely $V_0 = \{(0, x_1, \dots, x_k), \forall (x_1, \dots, x_k) \in \mathbb{F}_2^k\}$ and $V_1 = \{(1, x_1, \dots, x_k), \forall (x_1, \dots, x_k) \in \mathbb{F}_2^k\}$. Then $TT_f = (TT_{f_0}, TT_{f_1})$. According to the induction hypothesis $W_{f_0} = H_k \cdot TT_{f_0}$ and $W_{f_1} = H_k \cdot TT_{f_1}$. Then

$$\begin{aligned} W_f(\omega) &= \sum_{x \in \mathbb{F}_2^{k+1}} f(x)(-1)^{\langle x, \omega \rangle} \\ &= \sum_{(x_1, \dots, x_k) \in \mathbb{F}_2^k} f(0, x_1, \dots, x_k)(-1)^{\langle (0, x_1, \dots, x_k), (\omega_0, \omega_1, \dots, \omega_k) \rangle} \\ &\quad + \sum_{(x_1, \dots, x_k) \in \mathbb{F}_2^k} f(1, x_1, \dots, x_k)(-1)^{\langle (1, x_1, \dots, x_k), (\omega_0, \omega_1, \dots, \omega_k) \rangle} \\ &= \sum_{(x_1, \dots, x_k) \in \mathbb{F}_2^k} f_0(x_1, \dots, x_k)(-1)^{\langle (x_1, \dots, x_k), (\omega_1, \dots, \omega_k) \rangle} \\ &\quad + \sum_{(x_1, \dots, x_k) \in \mathbb{F}_2^k} f_1(x_1, \dots, x_k)(-1)^{\omega_0 + \langle (x_1, \dots, x_k), (\omega_1, \dots, \omega_k) \rangle} \\ &= \sum_{(x_1, \dots, x_k) \in \mathbb{F}_2^k} f_0(x_1, \dots, x_k)(-1)^{\langle (x_1, \dots, x_k), (\omega_1, \dots, \omega_k) \rangle} \\ &\quad + (-1)^{\omega_0} \sum_{(x_1, \dots, x_k) \in \mathbb{F}_2^k} f_1(x_1, \dots, x_k)(-1)^{\langle (x_1, \dots, x_k), (\omega_1, \dots, \omega_k) \rangle}. \end{aligned}$$

It follows that

$$W_f(\omega) = W_{f_0}(\omega_1, \dots, \omega_k) + W_{f_1}(\omega_1, \dots, \omega_k) \quad \text{for } \omega_0 = 0,$$

$$W_f(\omega) = W_{f_0}(\omega_1, \dots, \omega_k) - W_{f_1}(\omega_1, \dots, \omega_k) \quad \text{for } \omega_0 = 1.$$

These equations and the induction hypothesis show that

$$\begin{aligned} W_f &= \begin{pmatrix} W_{f_0} + W_{f_1} \\ W_{f_0} - W_{f_1} \end{pmatrix} = \begin{pmatrix} H_k \cdot TT_{f_0} + H_k \cdot TT_{f_1} \\ H_k \cdot TT_{f_0} - H_k \cdot TT_{f_1} \end{pmatrix} \\ &= \begin{pmatrix} H_k & H_k \\ H_k & -H_k \end{pmatrix} \cdot \begin{pmatrix} TT_{f_0} \\ TT_{f_1} \end{pmatrix}. \end{aligned}$$

Hence, the induction step is proven. $\square$

Thus, the computation of the Walsh transform is reduced to a matrix by vector multiplication. The elements of the matrix H_n are only 1's and -1 's. Hence, the multiplication of a row of the matrix by the considered vector means only addition and subtraction.

There is a fast algorithm to calculate $H_n \cdot v$. First, the vector v is partitioned into vectors of two coordinates. For each small vector

$$H_1 \cdot \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} v_1 + v_2 \\ v_1 - v_2 \end{pmatrix}.$$

Using these results, H_2 is multiplied by 4-length parts of v . The k -th step consists of multiplying the matrix H_k by a part v' of v , where v' has length 2^k . Let $v^{(1)}$ and $v^{(2)}$ be 2^{k-1} parts of v' from the previous step of the algorithm and also $H_{k-1} \cdot v^{(1)}$ and $H_{k-1} \cdot v^{(2)}$ are computed. Then

$$H_k \cdot v' = \begin{pmatrix} H_{k-1} & H_{k-1} \\ H_{k-1} & -H_{k-1} \end{pmatrix} \cdot \begin{pmatrix} v^{(1)} \\ v^{(2)} \end{pmatrix} = \begin{pmatrix} H_{k-1} \cdot v^{(1)} + H_{k-1} \cdot v^{(2)} \\ H_{k-1} \cdot v^{(1)} - H_{k-1} \cdot v^{(2)} \end{pmatrix}.$$

Therefore, we may consequently calculate $H_n \cdot v$.

The presented algorithm consists of n steps, each one performing only additions and subtractions. Therefore, $n \cdot 2^n$ arithmetic operations are made, instead of $2^n \cdot 2^n$ operations of the ordinary matrix by vector multiplication. An implementation of the Walsh–Hadamard fast transform in CUDA, as well as analysis of the complexity of the algorithm and comparison to other existing algorithms, are given in [15].

Example 1. Let $M = H_1$ and $\otimes^k H_1 = H_k$. For $k = 3$, it holds that $H_3 = (H_1 \otimes I_4) \cdot (I_2 \otimes H_1 \otimes I_2) \cdot (I_4 \otimes H_1)$.

$$H_3 = \begin{pmatrix} I_4 & I_4 \\ I_4 & -I_4 \end{pmatrix} \cdot \begin{pmatrix} I_2 & I_2 \\ I_2 & -I_2 \\ I_2 & I_2 \\ I_2 & -I_2 \end{pmatrix} \cdot \begin{pmatrix} H_1 & & & \\ & H_1 & & \\ & & H_1 & \\ & & & H_1 \end{pmatrix}$$

Each row in the above multipliers consists of only two nonzero elements, namely 1 and -1 . So, the row by vector multiplication costs one arithmetic operation (addition or subtraction). Let TT_f be the Truth Table of a function $f : \mathbb{F}_2^3 \rightarrow \mathbb{F}_2$. The process of computing $H_3 \cdot TT_f$ is given below in Figure 1. In the given diagram, the continuous line means addition, and the dotted line means subtraction.

In some cases, especially by parallelization of the algorithm and using shared memory, it is more convenient to rearrange the matrices. Because of Theorem 3, there is another way to compute $H_3 \cdot TT_f$.

$$H_3 = (I_4 \otimes H_1) \cdot (H_1 \otimes I_4) \cdot (I_2 \otimes H_1 \otimes I_2).$$

This leads to another fast algorithm for computing $V_3.b$ (b is a vector of suitable length), whose diagram is given in Figure 2.

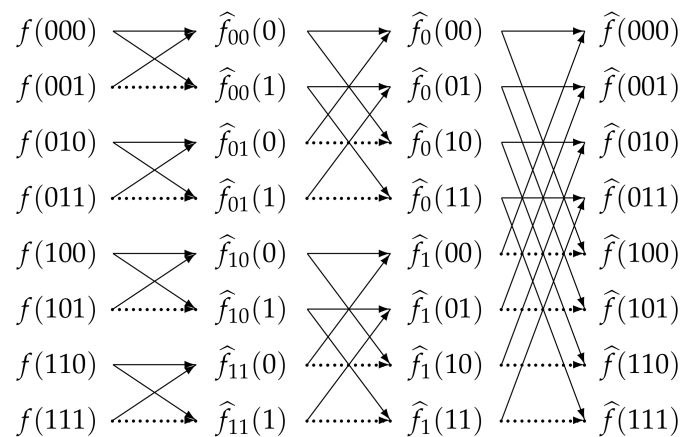


Figure 1. Fast Walsh transform diagram.

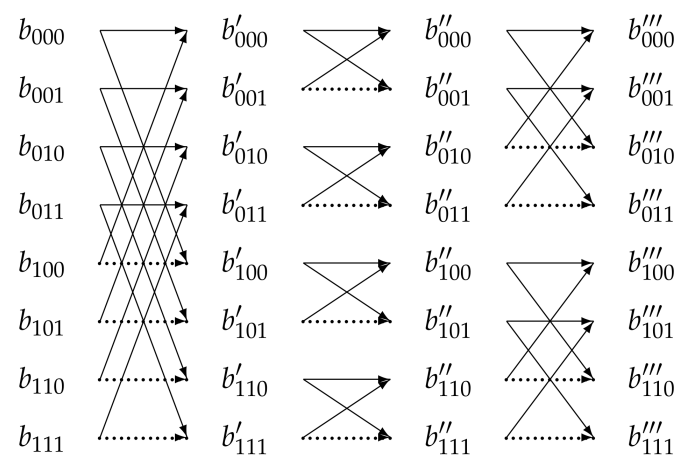


Figure 2. Fast Walsh transform diagram with rearranged steps.

As a factorization of the Kronecker power is made, the Vilenkin–Chrestenson and the trace transforms also have similar fast algorithms.

5. Discrete Transform of Vilenkin–Chrestenson

Let ζ be a primitive complex q -th root of unity. The matrices of Vilenkin–Chrestenson of size k are recurrently defined as follows:

$$V_1 = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \zeta & \zeta^2 & \dots & \zeta^{q-1} \\ 1 & \zeta^2 & \zeta^4 & \dots & \zeta^{2(q-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \zeta^{q-1} & \zeta^{2(q-1)} & \dots & \zeta^{(q-1)^2} \end{pmatrix}, \quad V_{k+1} = V_1 \otimes V_k, \quad k \in \mathbb{Z}, \quad k \geq 1, \quad (7)$$

where $\otimes$ means the Kronecker product. The elements of the matrix V_k are $v_\omega(x) = \zeta^{\langle \omega, x \rangle}$, where $\omega = (\omega_1, \omega_2, \dots, \omega_k) \in \mathbb{Z}_q^k$ are the row labels and $x = (x_1, x_2, \dots, x_k) \in \mathbb{Z}_q^k$ are the column labels, both in lexicographical order, $\mathbb{Z}_q = \{0, 1, 2, \dots, q-1\}$ and $\langle \omega, x \rangle = \sum_{i=1}^k \omega_i \cdot x_i \in \mathbb{Z}_q$.

Some useful properties directly following the definition are given below:

$$v_\omega(x) = v_x(\omega), \quad v_\omega(x)v_\omega(x') = v_\omega(x+x'), \quad v_\omega(0) = 1. \quad (8)$$

The first property shows that the matrices V_k are symmetric.

Definition 4. Let $f : \mathbb{Z}_q^k \rightarrow \mathbb{C}$ be a function. Transform of Vilenkin–Chrestenson of f is the function $\hat{f}_{VC} : \mathbb{Z}_q^k \rightarrow \mathbb{C}$, defined as follows:

$$\hat{f}_{VC}(\omega) = \sum_{x \in \mathbb{Z}_q^k} f(x) v_\omega(x), \quad \omega \in \mathbb{Z}_q^k. \quad (9)$$

Let TT_f be the vector whose coordinates are all values of the function f when the elements of $\mathbb{Z}_q^k$ are lexicographically ordered. It is called a truth table, and it is similar to the truth table of a Boolean function but with complex numbers. The value vectors of f and $\hat{f}_{VC}$ are related as follows:

$$TT_{\hat{f}_{VC}} = V_k \cdot TT_f.$$

Hence, the Vilenkin–Chrestenson transform can be reduced to a matrix by vector multiplication (see, for example, [16]).

One very useful application of the presented transform is to calculate the weight distribution of a linear code over a prime field. Let q be a prime and G be a generator matrix of the linear $[n, k]_q$ code C . We take a code C of full length, so $f_G(0) = 0$.

The relation between the Vilenkin–Chrestenson transform $\hat{f}_{VC}$ and the weight of the codeword $\omega G \in C$ is given by the following theorem.

Theorem 5. Let G be a generator matrix of the linear $[n, k]_q$ code C with full length. Then, for the weight of the codeword $\omega G \in C$, it holds that

$$wt(\omega G) = \frac{(q-1)n - \hat{f}_{VC}(\omega)}{q}, \quad \omega \in \mathbb{F}_q^k, \quad (10)$$

where $\hat{f}_{VC}$ is the Vilenkin–Chrestenson transform of the characteristic function f_G of the code C .

Proof. It is known that all codewords of C are linear combinations of the rows of its generator matrix, so any codeword is generated by multiplication of G by a vector $\omega \in \mathbb{F}_q^k$. Let us denote the columns of the matrix G by $g_1, \dots, g_n$. It follows that $\omega G = (\langle \omega, g_1 \rangle, \dots, \langle \omega, g_n \rangle)$, where $\langle \omega, g_i \rangle$ is the inner product of ω and the vector g_i . Since $wt(\omega G)$ is the number of the nonzero coordinates of the vector, so $n - wt(\omega G)$ is the number of the zeros in ωG . On the other hand, the summands in (9) are $\neq 0$ only if there exists a proportional to x column in G . Furthermore, if $f(x) \neq 0$ for some $x \neq 0$, then $f(x) = f(\alpha x)$ for each $\alpha \in \mathbb{F}_q \setminus \{0\}$. Hence, for each column x of the generator matrix G it holds that

$$\sum_{\alpha \in \mathbb{F}_q \setminus \{0\}} v_\omega(\alpha x) = \sum_{\alpha \in \mathbb{F}_q \setminus \{0\}} \zeta^{\langle \omega, \alpha x \rangle} = \begin{cases} q-1, & \text{if } \langle \omega, x \rangle = 0, \\ -1, & \text{if } \langle \omega, x \rangle \neq 0. \end{cases} \quad (11)$$

As the code has full length, then $f(0) = 0$. The summands in (9) can be rearranged such that exactly one sum of type (11) corresponds to each column of G . Then

$$\hat{f}_{VC}(\omega) = (q-1)(n - wt(\omega G)) - wt(\omega G).$$

This proves that in this case $\hat{f}_{VC}$ is an integer-valued function. Calculating $wt(\omega G)$, we obtain (10). $\square$

Example 2. Let C be $[7, 3]_3$ code and its generator matrix is

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 2 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The characteristic function $f_G(x) : \mathbb{F}_3^3 \rightarrow \mathbb{Z}$ of C with respect to G has the following Truth Table:

$$TT_{f_G} = (0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1)$$

We have that $TT_{\hat{f}_{VC}} = V_3 \cdot TT_{f_G}$, so

$$TT_{\hat{f}_{VC}} = (14, 5, 5, 2, -4, -1, 2, -1, -4, -1, -4, 2, -1, -1, -1, 2, -4, -1, -1, 2, -4, 2, -1, -4, -1, -1, -1)$$

Applying Theorem 5, we obtain that the nonzero weights in the code are 3, 4, 5, 6, and the weight enumerator is $W_C(y) = 1 + 2y^3 + 6y^4 + 12y^5 + 6y^6$. The structure of the matrix V_3 is given below.

$$\begin{aligned} V_3 &= (I_{3^{1-1}} \otimes V_1 \otimes I_{3^{3-1}}) \cdot (I_{3^{2-1}} \otimes V_1 \otimes I_{3^{3-2}}) \cdot (I_{3^{3-1}} \otimes V_1 \otimes I_{3^{3-3}}) \\ &= (V_1 \otimes I_9) \cdot (I_3 \otimes V_1 \otimes I_3) \cdot (I_9 \otimes V_1), \end{aligned}$$

$$\text{where } V_1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & \zeta & \zeta^2 \\ 1 & \zeta^2 & \zeta \end{pmatrix}.$$

$$V_3 = \begin{pmatrix} I_9 & I_9 & I_9 \\ I_9 & \zeta I_9 & \zeta^2 I_9 \\ I_9 & \zeta^2 I_9 & \zeta I_9 \end{pmatrix} \cdot \begin{pmatrix} I_3 & I_3 & I_3 \\ I_3 & \zeta I_3 & \zeta^2 I_3 \\ I_3 & \zeta^2 I_3 & \zeta I_3 \end{pmatrix} \cdot \begin{pmatrix} I_3 & I_3 & I_3 \\ I_3 & \zeta I_3 & \zeta^2 I_3 \\ I_3 & \zeta^2 I_3 & \zeta I_3 \end{pmatrix} \cdot \begin{pmatrix} I_3 & I_3 & I_3 \\ I_3 & \zeta I_3 & \zeta^2 I_3 \\ I_3 & \zeta^2 I_3 & \zeta I_3 \end{pmatrix} \cdot \begin{pmatrix} V_1 & & & \\ & V_1 & & \\ & & \ddots & \\ & & & V_1 \end{pmatrix}$$

Each of the sparse matrices above has exactly three nonzero elements in its rows and columns.

In the general case, a sparse matrix for the Vilenkin–Chrestenson transform has exactly q nonzero elements since V_1 is a $q \times q$ matrix. That is why the number of the necessary operations is reduced to $2kq^{k+1}$, approximately, while the same result would be obtained with $2q^{2k}$ operations without the V_k factorization. The same conclusion could be drawn for the trace transform as well.

6. Trace Transform

Let $\mathbb{F}_q$ be a composite finite field with q elements, where $q = p^m$ for the prime p and the integer $s > 1$. For the trace transform, the absolute trace of the inner product is used instead of the inner product itself, as in the Vilenkin–Chrestenson transform [27]. Let ζ be a primitive complex p -th root of unity, and

$$\tau_\omega(x) = \zeta^{\text{Tr}(\langle \omega, x \rangle)} \quad (12)$$

for any $\omega, x \in \mathbb{F}_q^k$. We define the trace transform similarly to the Vilenkin–Chrestenson transform.

Definition 5. Let $\mathbb{F}_q$ be a finite field with q elements, $q = p^m$ for the prime number p , and ζ be a primitive complex p -th root of unity. Trace transform of the function $f : \mathbb{F}_q^k \rightarrow \mathbb{C}$ is the function $\hat{f}_T : \mathbb{F}_q^k \rightarrow \mathbb{C}$, defined as follows:

$$\hat{f}_T(\omega) = \sum_{x \in \mathbb{F}_q^k} f(x) \tau_\omega(x) = \sum_{x \in \mathbb{F}_q^k} f(x) \zeta^{\text{Tr}(\langle \omega, x \rangle)}, \quad \omega \in \mathbb{F}_q^k. \quad (13)$$

The transform matrix here is the $q^k \times q^k$ matrix $T_k = (\tau_\omega(x))$, where the vectors (in our case denoted by ω and x) in $\mathbb{F}_q^k$ are lexicographically ordered. Then the Truth Tables of f and $\hat{f}_T$ are connected by the equality

$$TT_{\hat{f}_T} = T_k \cdot TT_f.$$

Because of the symmetry and linearity of the inner product, the following properties hold:

$$\tau_\omega(x) = \tau_x(\omega), \quad \tau_\omega(x)\tau_\omega(x') = \tau_\omega(x + x'). \quad (14)$$

Lemma 1. For $x \in \mathbb{F}_q^k$ it holds that

$$\sum_{\omega \in \mathbb{F}_q^k} \tau_\omega(x) = \begin{cases} q^k, & \text{for } x = 0, \\ 0, & \text{for } x \neq 0. \end{cases}$$

Proof. The lemma is a modification of [28] (Chapter 5, Lemma 9), and its proof is by induction on k . In the base step, $k = 1$ it holds

$$\sum_{\omega \in \mathbb{F}_q} \tau_\omega(x) = \sum_{\omega \in \mathbb{F}_q} \zeta^{\text{Tr}(x\omega)} = \begin{cases} q, & \text{for } x = 0, \\ 0, & \text{for } x \neq 0. \end{cases}$$

For $x \neq 0$, when ω takes all possible values of $\mathbb{F}_q$, then $x\omega$ takes all the values of $\mathbb{F}_q$. Moreover, for any element $\alpha \in \mathbb{F}_q$, there are exactly p^{m-1} elements $\alpha' \in \mathbb{F}_q$ with the same trace value $\text{Tr}(\alpha') = \text{Tr}(\alpha)$. Thus

$$\sum_{\omega \in \mathbb{F}_q} \zeta^{\text{Tr}(x\omega)} = \sum_{\omega \in \mathbb{F}_q} \zeta^{\text{Tr}(\omega)} = p^{m-1} \sum_{\omega \in \mathbb{F}_p} \zeta^\omega = p^{m-1} \frac{1 - \zeta^p}{1 - \zeta} = 0.$$

Let the equality hold for some integer k . Consider the vectors in $\mathbb{F}_q^{k+1}$ in the following form: $x = (x_0, x')$, $x_0 \in \mathbb{F}_q$, $x' \in \mathbb{F}_q^k$. Because of the trace linearity

$$\begin{aligned} \sum_{\omega \in \mathbb{F}_q^{k+1}} \tau_\omega(x) &= \sum_{\omega \in \mathbb{F}_q^{k+1}} \zeta^{\text{Tr}(\langle \omega, x \rangle)} = \sum_{j \in \mathbb{F}_q} \sum_{\omega' \in \mathbb{F}_q^k} \zeta^{\text{Tr}(jx_0 + \langle \omega', x' \rangle)} \\ &= \sum_{j \in \mathbb{F}_q} \sum_{\omega' \in \mathbb{F}_q^k} \zeta^{\text{Tr}(jx_0) + \text{Tr}(\langle \omega', x' \rangle)} = \sum_{j \in \mathbb{F}_q} \sum_{\omega' \in \mathbb{F}_q^k} \zeta^{\text{Tr}(jx_0)} \zeta^{\text{Tr}(\langle \omega', x' \rangle)} \\ &= \left(\sum_{j \in \mathbb{F}_q} \zeta^{\text{Tr}(jx_0)} \right) \left(\sum_{\omega' \in \mathbb{F}_q^k} \zeta^{\text{Tr}(\langle \omega', x' \rangle)} \right) \\ &= \left(\sum_{j \in \mathbb{F}_q} \tau_j(x_0) \right) \left(\sum_{\omega' \in \mathbb{F}_q^k} \tau_{\omega'}(x') \right) \end{aligned} \quad (15)$$

According to the induction hypothesis and its base step, the sum is $\neq 0$ exactly when $x_0 = 0$ and $x' = 0$. In this case, the sum is $q \cdot q^k = q^{k+1}$.

Hence, the equality holds for each $k \in \mathbb{N}$. $\square$

To reduce (15), it is also taken that

$$\zeta^{\text{Tr}(\langle \omega, x \rangle)} = \zeta^{\text{Tr}(jx_0)} \zeta^{\text{Tr}(\langle \omega', x' \rangle)},$$

where $\omega = (j, \omega')$, $x = (x_0, x') \in \mathbb{F}_q^{k+1}$, $j, x_0 \in \mathbb{F}_q$, $\omega', x' \in \mathbb{F}_q^k$. This means that the matrices T_k are generated as a Kronecker power, i.e., $T_{k+1} = T_1 \otimes T_k$ and $T_k = \otimes^k T_1$ for $k \in \mathbb{N}$.

The trace transform also has applications in coding theory, but it is used for linear codes over composite finite fields. In this case, a theorem similar to Theorem 5 is proved.

Theorem 6. Let G be a generator matrix of the linear $[n, k]_q$ code C with full length. Then, for the weights of all codewords of C , it holds that

$$wt(\omega G) = \frac{(q-1)n - \hat{f}_T(\omega)}{q}, \quad \omega \in \mathbb{F}_q^k, \quad (16)$$

where $\hat{f}_T$ is the trace transform of the characteristic function f_G of the code C .

Proof. The proof is similar to the proof of Theorem 5. Each coordinate in ωG is equal to the inner product of ω and the corresponding column of G . Since $wt(\omega G)$ is the number of the nonzero coordinates of ωG , so $n - wt(\omega G)$ is the number of zeros. On the other hand, the summand $x \zeta^{Tr(\langle \omega, x \rangle)}$ in (13) is not equal to 0 if and only if there exists a proportional vector of x as a column in G . If $f_G(x) \neq 0$, for some $x \neq 0$, then $f_G(x) = f_G(\alpha x)$ for all $\alpha \in \mathbb{F}_q \setminus \{0\}$. For each column x of the matrix G it holds that

$$\sum_{\alpha \in \mathbb{F}_q \setminus \{0\}} \tau_\omega(\alpha x) = \sum_{\alpha \in \mathbb{F}_q \setminus \{0\}} \zeta^{Tr(\langle \omega, \alpha x \rangle)} = \begin{cases} q-1, & \text{if } \langle \omega, x \rangle = 0, \\ -1, & \text{if } \langle \omega, x \rangle \neq 0. \end{cases} \quad (17)$$

Let $\langle \omega, x \rangle \neq 0$. Then $\{\alpha \langle \omega, x \rangle \mid \alpha \in \mathbb{F}_q, \alpha \neq 0\} = \mathbb{F}_q \setminus \{0\}$. The absolute trace Tr is a linear map onto $\mathbb{F}_p$ with a kernel of p^{m-1} elements. Hence, the sum (17) has exactly $p^{m-1} - 1$ additives $\zeta^0 = 1$ and exactly p^{m-1} additives ζ^λ for each $\lambda \in \mathbb{F}_p, \lambda \neq 0$. Further

$$\sum_{\alpha \in \mathbb{F}_q \setminus \{0\}} \zeta^{Tr(\langle \omega, \alpha x \rangle)} = p^{m-1} - 1 + p^{m-1} \sum_{\lambda \in \mathbb{F}_p \setminus \{0\}} \zeta^\lambda = -1 + p^{m-1} \sum_{\lambda \in \mathbb{Z}_p} \zeta^\lambda = -1.$$

Because the code has full length, then $f_G(0) = 0$. The summands in (13) might be rearranged in such a way that any of the columns of G corresponds exactly to one sum of the type (17). Then

$$\hat{f}_T(\omega) = (q-1)(n - wt(\omega G)) - wt(\omega G),$$

which leads to (16). $\square$

It was proved in [13] that if the characteristic of the field is 2 and the used basis of $\mathbb{F}_q$ is self-dual, the matrix T_1 is a $q \times q$ Sylvester-Hadamard matrix (this matrix is denoted by Λ in [13] (Proposition 2.1)). We give an example with a quaternary code with two characteristic vectors—one of them with respect to a self-dual basis and another one with respect to the ordinary polynomial basis $1, x$. Recall that the elements $\beta_1, \dots, \beta_m \in \mathbb{F}_q$, where $q = p^m$ for a prime p , form a self-dual basis of the field if $Tr(\beta_i \beta_j) = \delta_{ij}$ for the Kronecker δ .

Example 3. Let $q = 4$ and $\mathbb{F}_4 = \{0, 1, x, x+1 = x^2\}$. We consider the $[4, 2]$ quaternary code with a generator matrix

$$G = \begin{pmatrix} 1 & 0 & 1 & x^2 \\ 0 & 1 & x^2 & x \end{pmatrix}.$$

If we take the ordering $0 < 1 < x < x+1$, then the lexicographic ordering of the quaternary vectors of length 2 is

$$(0, 0) < (0, 1) < (0, x) < (0, x+1) < (1, 0) < \dots < (x+1, x) < (x+1, x+1),$$

and the characteristic vector of the code is

$$f_G^1(C) = (0, 1, 1, 1, 1, 0, 0, 2, 1, 2, 0, 0, 1, 0, 2, 0).$$

In this case, the trace transform matrix is

$$T_2 = \otimes^2 T_1 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \end{pmatrix}$$

Then $T_2 \cdot f'_G(C) = (12, 0, 0, 0, 0, -4, 4, -4, 0, -4, -4, 4, 0, 4, -4, -4)$ and after applying (16), we obtain that the weight distribution of the given code is $W_C(z) = 1z^0 + 3z^2 + 6z^3 + 6z^4$.

The quaternary field has a self-dual basis, namely $\beta_1 = x$, $\beta_2 = x + 1$. Then we have the following ordering of the elements $0 < \beta_2 < \beta_1 < 1$, since $\beta_2 = (0, 1)$, $\beta_1 = (1, 0)$, and $1 = \beta_1 + \beta_2 = (1, 1)$. In this case, the lexicographic ordering of the quaternary vectors of length 2 is

$$(0, 0) < (0, \beta_2) < (0, \beta_1) < (0, 1) < (\beta_2, 0) < \dots < (1, \beta_1) < (1, 1).$$

The new characteristic vector is

$$f''_G(C) = (0, 1, 1, 1, 1, 0, 2, 0, 1, 0, 0, 2, 1, 2, 0, 0).$$

Now $T_1 = \otimes^2 H_1$ and so T_2 has the following structure:

$$T_2 = \otimes^2 T_1 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \end{pmatrix}$$

After the corresponding calculations, the same weight distribution of the code is obtained, but the algorithm is much faster, as we can use the fast Walsh transform, presented in Section 4.

In the general case, if the trace transform could not be reduced to the Walsh–Hadamard transform, the matrix T_1 is of size $q \times q$, i.e., $p^m \times p^m$. Then the matrix T_k is a $p^{mk} \times p^{mk}$

matrix, so the computation of $TT_{\hat{f}_T}$ requires p^{2mk} multiplications and the same number of additions, $2p^{2mk}$ operations over all. Let T_k be factorized into k sparse $p^{mk} \times p^{mk}$ matrices. Each of them has at most $p^m \times p^m$ nonzero elements, so its multiplications by a vector consist of $2p^{mk}p^m$ multiplications and additions. The total amount of operations to obtain the same result is at most $2kp^{m(k+1)}$.

We finish this section with a proposition showing that the trace transform can be reduced to a Vilenkin–Chrestenson transform in some cases. Let $q = p^m$, where p is an odd prime and $m > 1$ is an odd integer. According to Theorem 1, there exists a self-dual basis $\beta_1, \dots, \beta_m$ of $\mathbb{F}_q$ over $\mathbb{F}_p$. Let $\mathbb{F}_q = \{\alpha_0 = 0, \alpha_1, \dots, \alpha_{q-1}\}$, where the elements are ordered lexicographically with respect to the self-dual basis. Therefore,

$$\alpha_s = \lambda_1(\alpha_s)\beta_1 + \dots + \lambda_m(\alpha_s)\beta_m, \quad s = 0, 1, \dots, q-1, \quad (18)$$

and $\lambda(\alpha_s) = (\lambda_1(\alpha_s), \dots, \lambda_m(\alpha_s)) \in \mathbb{F}_p^m$. Because of the lexicographic ordering, we have the following representation of the index s in the p -ary numeral system

$$s = \lambda_1(\alpha_s)p^{m-1} + \dots + \lambda_{m-1}(\alpha_s)p + \lambda_m(\alpha_s).$$

Let $V_m(p)$ be the Vilenkin–Chrestenson matrix for the p th root of unity ζ , considered in this section. Then, the following statement holds.

Proposition 1. For $\mathbb{F}_q = \{\alpha_0 = 0, \alpha_1, \dots, \alpha_{q-1}\}$, the matrices T_1 and $V_m(p)$ coincides.

Proof. According to the linearity of the trace function, we have

$$\begin{aligned} Tr(\alpha_j \alpha_{j'}) &= Tr\left(\sum_{s=1}^m \lambda_s(\alpha_j) \beta_s \sum_{s'=1}^m \lambda_{s'}(\alpha_{j'}) \beta_{s'}\right) \\ &= Tr\left(\sum_{s=1}^m \sum_{s'=1}^m \lambda_s(\alpha_j) \lambda_{s'}(\alpha_{j'}) \beta_s \beta_{s'}\right) \\ &= \sum_{s=1}^m \sum_{s'=1}^m Tr(\lambda_s(\alpha_j) \lambda_{s'}(\alpha_{j'}) \beta_s \beta_{s'}) \\ &= \sum_{s=1}^m \sum_{s'=1}^m \lambda_s(\alpha_j) \lambda_{s'}(\alpha_{j'}) Tr(\beta_s \beta_{s'}). \end{aligned}$$

Since the basis is self-dual, it holds that

$$\lambda_s(\alpha_j) \lambda_{s'}(\alpha_{j'}) Tr(\beta_s \beta_{s'}) = \begin{cases} \lambda_s(\alpha_j) \lambda_s(\alpha_{j'}), & \text{if } s = s', \\ 0, & \text{if } s \neq s'. \end{cases}$$

Then

$$Tr(\alpha_j \alpha_{j'}) = \sum_{s=1}^m \lambda_s(\alpha_j) \lambda_s(\alpha_{j'}) = \langle \lambda(\alpha_j), \lambda(\alpha_{j'}) \rangle.$$

Hence

$$\tau_\omega(x) = \zeta^{\langle \lambda(\alpha_j), \lambda(\alpha_{j'}) \rangle} = v_{\lambda(\omega)} \lambda(x)$$

and so

$$\hat{f}_T(\omega) = \hat{f}_{VC}(\lambda(\omega)),$$

where the $\hat{f}_{VC}$ is the Vilenkin–Chrestenson transform with the p -th root of unity ζ . Transferring this result to the transform matrices, we obtain $T_1 = V_m(p)$. $\square$

7. Conclusions

Computing the weight distribution of a linear code is not an easy problem, especially when the considered finite field is composite. Most algorithms that solve this problem

are based on searching in large amounts of generated data. Therefore, other different techniques, such as Fourier-type transforms, are also being investigated and applied.

In this paper, three fast discrete transforms are studied, namely the Walsh–Hadamard, the Vilenkin–Chrestenson, and the trace transform, mainly considering the problem of representing their matrices as a product of sparse matrices. Using these transforms, the weights of a linear code could be computed without generating the codewords themselves. Algorithms for implementing the given fast discrete transforms have not only better complexity but they could also be easily parallelized.

Author Contributions: Conceptualization, I.B. and P.P.; methodology, I.B.; validation, I.B., M.D.-S. and P.P.; formal analysis, I.B., M.D.-S. and P.P.; investigation, I.B., M.D.-S. and P.P.; resources, I.B. and P.P.; writing—original draft preparation, I.B., M.D.-S. and P.P.; writing—review and editing, I.B. and M.D.-S.; visualization, M.D.-S.; supervision, I.B.; project administration, I.B.; funding acquisition, I.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially supported by Bulgarian National Science Fund grant number KP-06-H62/2/13.12.2022.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article, because it describes entirely theoretical research.

Acknowledgments: The authors would like to thank Stefka Bouyuklieva for her valuable discussions and suggestions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Elliot, D.F.; Rao, K.R. *Fast Transforms: Algorithms, Analyses, Applications*; Academic Press: Orlando, FL, USA, 1982.
2. Karpovsky, M.G.; Stanković, R.S.; Astola, J.T. *Spectral Logic and Its Applications for the Design of Digital Devices*; John Wiley & Sons Ltd.: Hoboken, NJ, USA, 2008.
3. Joux, A. *Algorithmic Cryptanalysis*; Chapman & Hall/CRC: Boca Raton, FL, USA, 2009.
4. Maitra, S.; Sarkar, P. Cryptographically Significant Boolean Functions with Specified Five Valued Walsh Spectra. *Theor. Comp. Sci.* **2002**, *276*, 133–146. [[CrossRef](#)]
5. Uyan, E.; Çalik, Ç.; Doğanaksoy, A. Counting Boolean Functions with Specified Values in Their Walsh Spectrum. *J. Comp. Appl. Math.* **2014**, *259*, 522–528. [[CrossRef](#)]
6. Bouyukliev, I.; Bouyuklieva, S. Dual transform and projective self-dual codes. *Adv. Math. Commun.* **2024**, *18*, 328–341. [[CrossRef](#)]
7. Kaski, P.; Östergård, P. *Classification Algorithms for Codes and Designs*; Springer: Berlin/Heidelberg, Germany, 2006.
8. Betten, A.; Braun, M.; Frippertinger, H.; Kerber, A.; Kohnert, A.; Wassermann, A. *Error-Correcting Linear Codes. Classification by Isometry and Applications*; Springer: Berlin/Heidelberg, Germany, 2006.
9. Cheng, W.; Liu, Y.; Guilley, S.; Rioul, O. Towards finding best linear codes for side-channel protections. In Proceedings of the 10th International Workshop on Security Proofs for Embedded Systems (PROOFS’2021), Beijing, China, 17 September 2021.
10. Hannusch, C.; Major, S.R. Torch: Software Package For The Search Of Linear Binary Codes. In Proceedings of the 2022 IEEE 2nd Conference on Information Technology and Data Science (CITDS), Debrecen, Hungary, 16–18 May 2022; pp. 103–106. [[CrossRef](#)]
11. Karpovsky, M.G. On the Weight Distribution of Binary linear codes. *IEEE Trans. Inform. Theory* **1979**, *IT-25*, 105–109. [[CrossRef](#)]
12. Karpovsky, M.G. Weight distribution of translates, covering radius, and perfect codes correcting errors of given weights. *IEEE Trans. Inform. Theory* **1981**, *27*, 462–472. [[CrossRef](#)]
13. Piperkov, P.; Bouyukliev, I.; Bouyuklieva, S. An algorithm for computing the weight distribution of a linear code over composite Finite Field with characteristic 2. In *Recent Topics in Differential Geometry and Its Related Fields*; Adachi, T., Hashimoto, H., Eds.; World Scientific Publishing Company: Singapore, 2019; pp. 163–181.
14. Piperkov, P.; Bouyukliev, I.; Bouyuklieva, S. An algorithm for computing the covering radius of a linear code based on Vilenkin–Chrestenson transform. In *New Horizons in Differential Geometry and Its Related Fields*; Adachi, T., Hashimoto, H., Eds.; World Scientific Publishing Company: Singapore, 2022; pp. 105–123.
15. Bikov, D.; Bouyukliev, I. Parallel Fast Walsh Transform Algorithm and Its Implementation with CUDA on GPUs. *Cybern. Inf. Technol.* **2018**, *18*, 21–43. [[CrossRef](#)]
16. van Loan, C. *Computational Frameworks for the Fast Fourier Transform*; The Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 1992; ISBN 978-0-898712-85-8.
17. Henderson, K.W. Some Notes on the Walsh Functions. *IEEE Trans. Electron. Comput.* **1964**, *EC-13*, 50–52. [[CrossRef](#)]
18. Walsh, J.L. A Closed Set of Normal Orthogonal Functions. *Am. J. Math.* **1923**, *45*, 5–24. [[CrossRef](#)]
19. Good, I.J. The Interaction Algorithm and Practical Fourier Analysis. *J. R. Stat. Soc.* **1958**, *20*, 361–372. [[CrossRef](#)]
20. Chrestenson, H.E. A class of generalized Walsh functions. *Pac. J. Math.* **1955**, *5*, 17–31. [[CrossRef](#)]

21. Vilenkin, N.Y. On a class of complete orthonormal systems. *Izv. Akad. Nauk SSSR Ser. Mat.* **1947**, *11*, 363–400.
22. Bespalov, M.S. Discrete Chrestenson transform. *Probl. Inf. Transm.* **2010**, *46*, 353–375. [[CrossRef](#)]
23. Farkov, Y.A. Discrete wavelets and the Vilenkin-Chrestenson transform. *Math. Notes* **2011**, *89*, 871–884. [[CrossRef](#)]
24. Lidl, R.; Niederreiter, H. *Introduction to Finite Fields and their Applications*; Cambridge University Press: Cambridge, UK, 1986.
25. Mullen, G.L.; Panario, D. *Handbook of Finite Fields*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2013; pp. 2742–33487.
26. Seroussi, G.; Lempel, A. Factorization of Symmetric Matrices and Trace-orthogonal Bases in Finite Fields. *SIAM J. Comput.* **1980**, *9*, 758–767. [[CrossRef](#)]
27. Assmus, E.F.; Mattson, H.F. Coding and combinatorics. *SIAM Rev.* **1974**, *16*, 349–388. [[CrossRef](#)]
28. MacWilliams, F.J.; Sloane, N.J.A. *The Theory of Error-Correcting Codes*; Elsevier Science Publishers: Amsterdam, The Netherlands, 1977.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.