Mathematical Problems of Computer Science 34, 2010.

Orthogonal Transforms for Digital Signal and Image Processing

Hakob Sarukhanyan

Institute for Informatics and Automation Problems of NAS RA

Abstract

In this report there are presented some primary results obtained in Digital Signal and Image Processing laboratory of the Institute for Informatics and Automation Problems of NAS RA.

Fast Discrete Orthogonal Transforms: The computation of unitary transforms is a complicated and time-consuming task. However, it would not be possible to use the orthogonal transforms in signal and image processing applications without effective algorithms to calculate them. Note that both complexity issues–efficient software and circuit implementations are the heart of the most applications. An important question in many applications is how to achieve the highest computation efficiency of the discrete orthogonal transforms (DOT) [1]. The suitability of unitary transforms in each of the above applications depends on the properties of their basis functions as well as on the existence of fast algorithms, including parallel ones. A fast DOT is an efficient algorithm to compute the DOT and its inverse with an essentially smaller number of operations than direct matrix multiplication.

Recall that the DOT of the sequence $f(n)$ is given by $Y[k]=\frac{1}{\sqrt{N}}\sum_{n=0}^{N-1} f[n]\phi_n[k],\ k=\overline{0,N-1},$ where $\{\phi_n[k]\}$ is an orthogonal system. It follows that the determination of each $Y[k]$ requires N multiplication and $N-1$ addition. Because we have to evaluate $Y[k]$ for $k=\overline{0,N-1},$ it follows that the direct determination of DFT requires $N(N-1)$ operations, which means that the number of multiplication and additions/subtractions is proportional to N^2, i.e., the complexity of DFT is $O(N^2)$.

General Concept in Design of the Fast DOT Algorithms: A fast transform $T_N f$ may be achieved by factoring the transform matrix T_N by the multiplication of k sparse matrices. Typically, $N=2^n$ and $T_{2^n}=F_nF_{n-1}\cdots F_2F_1,$ where F_i are very sparse matrices so that the complexity of multiplying by F_i is $O(N),\ i=\overline{1,n}.$

Fast Fourier Transform: Particularly, the Fourier matrix F_N of order $N=2^n$ can be represented as $F_N=B_nA_{n-1}B_{n-1}\cdots A_2B_2A_1B_1,$ where A_r and A_i are sparse matrices of the following form

$$A_r = I_{2^{r-1}} \otimes \left(I_{2^{n-r}} \oplus W^0_{2^{n-r+1}} \oplus W^1_{2^{n-r+1}} \oplus \cdots \oplus W^{2^{n-r}-1}_{2^{n-r+1}}\right), \quad r = \overline{1, n-1},$$
$$B_i = I_{2^{i-1}} \otimes \left(H_2 \otimes I_{2^{n-i}}\right), \quad i = \overline{1, n}.$$

$H_2 = \begin{bmatrix} + & + \\ + & - \end{bmatrix}$, $\otimes$ is a sign of Kronecker product, and $W^k_M = \exp\left(-j\dfrac{2\pi}{M}k\right)$.Therefore the fast Fourier transform has the complexity $O(N\log_2 N)$.

The $N = 2^n$-point Fourier transform of a $x = \{x_0, x_1, \ldots, x_{N-1}\}$ with assumption $x[-1] = x[N-1]$ can be represented as (see also [2,3])

$$X[n] = \sum_{k=0}^{N/2-1} x[4k]W^{nk}_{N/2} + W^n_N \sum_{k=0}^{N/4-1} x[4k+1]W^{nk}_{N/4} + W^{-n}_N \sum_{k=0}^{N/4-1} x[4k-1]W^{nk}_{N/4}, \quad n = \overline{0, N-1}.$$

By introducing the notations $Y_0[n] = \sum_{k=0}^{N/2-1} x[4k]W^{nk}_{N/2}$, $n = \overline{0, N/2-1}$; $Y_1[n] = \sum_{k=0}^{N/4-1} x[4k+1]W^{nk}_{N/4}$, $Y_2[n] = \sum_{k=0}^{N/4-1} x[4k-1]W^{nk}_{N/4}$, $n = \overline{0, N/4-1}$, FFT can be realized as follows

$$\begin{aligned}
X[n] &= Y_0[n] + \left(W^n_N Y_1[n] + W^{-n}_N Y_2[n]\right), \\
X[n + N/4] &= Y_0[n + N/4] - j\left(W^n_N Y_1[n] - W^{-n}_N Y_2[n]\right), \\
X[n + 2N/4] &= Y_0[n] - \left(W^n_N Y_1[n] + W^{-n}_N Y_2[n]\right), \\
X[n + 3N/4] &= Y_0[n + N/4] + j\left(W^n_N Y_1[n] - W^{-n}_N Y_2[n]\right), \quad n = \overline{0, N/4-1}.
\end{aligned}$$

The number of operations for a realization of FFT given below

$$C^+_N = \frac{8}{3}N\log_2 N - \frac{16}{9}N - \frac{2}{9}(-1)^{\log_2 N} + 2,$$
$$C^\times_N = \frac{4}{3}N\log_2 N - \frac{38}{9}N + \frac{2}{9}(-1)^{\log_2 N} + 6.$$

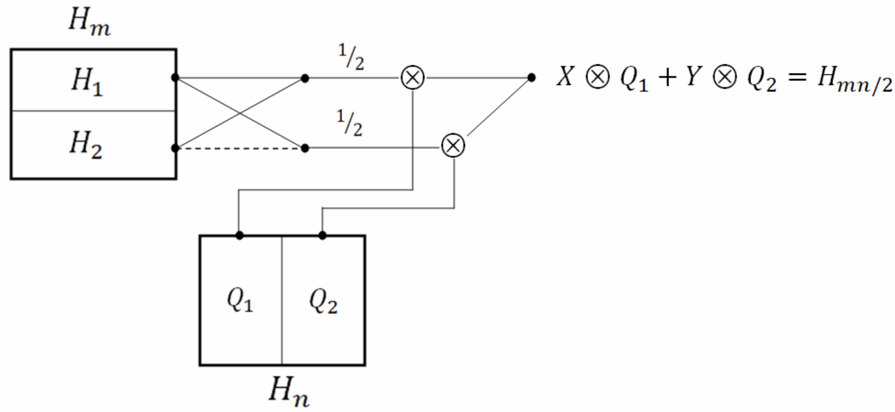
Fast Hadamard Transform: The Hadamard matrix H_N of order $N = 2^n$ can be factorized as follows: $H_N = F_n F_{n-1} \cdots F_2 F_1$, where $F_i = I_{2^{i-1}} \otimes \left(H_2 \otimes I_{2^{n-i}}\right)$, $i = 1,2,\ldots,n$. It is not difficult to show that $N = 2^n$-point fast Hadamard transform (FHT) has the complexity $O(N\log_2 N)$ (Note that FHT requires only the additions and subtraction operations). Later, in [4] were developed more general FHT which have the complexity $D = n\log_2 k + n\left(\frac{n}{k} - 1\right)$ with assumption that Hadamard matrix H_n of order n has the following representation

$$H_n = v_1 \otimes A_1 + v_2 \otimes A_2 + \cdots + v_k \otimes A_k$$

where v_i are k-size (-1,+1) mutualy orthogonal vectors, and A_j, $j = \overline{1, k}$ are $\frac{n}{k} \times n$ size (0,-1,+1) matrices with the following conditions

$$A_i * A_j, \quad i \neq j, \quad i, j = 1,2,\ldots,k,$$
$$\sum_{i=1}^{k} A_i \quad is\ a\ (+1,-1) - matrix,$$
$$\sum_{i=1}^{k} A_i A_i^T = \tfrac{n}{k} I_n,$$
$$A_i^T A_j = 0, \quad i \neq j, \quad i, j = 1,2,\ldots,k,$$
$$A_i^T A_i = \tfrac{n}{k} I_{n/k}, \quad i = 1,2,\ldots,k.$$

A-S theorem and its generalization: In 1981 Agaian and Sarukhanyan [5] have offered a construction method of Hadamard matrix (H-matrix) of order mn/2 proceeding from existence H-matrices of order m and n (see Figure below).



Later, this result has been named by multiplicative theorem A-S (the multiplicative theorem of Agaian-Sarukhanyan) [6]. In the same place, the existence of H-matrix of order mnpq/16 is proved based on A-S theorem. Further, by using of the Multiplicative method there have been developed methods of construction hybrid orthogonal matrices and corresponding fast transform algorithms. Last years A-S methodwas successfully applied at construction of Lapped transforms and Filter Banks, and also at construction Antipodal Paraunitary matrices and their application in orthogonal frequency division multiplexing (OFDM) systems [7, 8].

References

1. N. Ahmed and K. R. Rao, *Orthogonal Transforms for Digital Signal Processing*. Springer-Verlag, New York (1975).
2. R. Yavne, "An economical method for calculating the discrete Fourier transform," Proc. AFIPS Fall Joint Computer Conf., vol. 33, 1968, pp. 115–125.
3. M. Frigo and S. G. Johnson, "A modified split-radix FFT with fewer arithmetic operations," IEEE Transactions on Signal Processing, vol. 55, pp. 111-119, 2007.
4. H. Sarukhanyan. Decomposition of the Hadamard Matrices and Fast Hadamard Transform. Computer Analysis of Images and Patterns, Lecture Notes in Computer Science, vol. 1296, 1997, pp. 575-581
5. S. Agaian, H. Sarukhanyan. Recurrence Formulae Construction of Williamson's Type Matrices. Math. Notes, vol. 30, No.3- 4, 1981, pp. 796-804
6. J. Seberry and M. Yamada, "Hadamard matrices, sequences, and block designs," in Contemporary Design Theory: A Collection of Surveys, J. H. Dinitz and D. R. Stinson, Eds. New York: Wiley, 1992.
7. See-May Phoong, Yuan-Pei Lin. Lapped Hadamard Transforms and Filter Banks, IEEE, ICASSP 2003, vol. 6, 2003, pp. 509-512.
8. See-May Phoong, Kai-Yen Chang. Antipodal Paraunitary Matrices and Their Application to OFDM Systems, IEEE Trans. on Signal Processing, vol. 53, No. 4, 2005, pp. 1374-1386.