ISSN: 2053-3578 I.F. 12.34

MODIFIED FAST HAAR TRANSFORM ALGORITHM

Ibragimov Sanjarbek Salijanovich
t.f.f.d. (PhD)
Andijon davlat texnika instituti
e-mail: sanjari07@yahoo.com

Annotation. In this paper, modified version of the Fast Haar Transform algorithm is developed, and its impact on computational efficiency and memory requirements is analyzed. Unlike the traditional Fast Haar Transform algorithm, which performs calculations based on two-node structures, the improved algorithm utilizes four-node structures simultaneously to determine spectral coefficients. This enhanced approach reduces the number of iterations, optimizes redundant computations, and minimizes memory consumption. The modified Fast Haar Transform algorithm is represented in both graph-based and matrix-based forms, and a comparative analysis with the conventional Fast Haar Transform is provided.

Keywords: Haar Transform, Fast Transform, Spectral Coefficient, Number of Iterations, Algorithm Optimization.

Introduction

One of the fundamental stages in digital signal and image processing is transforming the data into its spectral representation and subsequently extracting the required components. Fast transform algorithms serve as essential tools for these processes. The Fast Haar Transform (FHT) is widely used in digital processing systems due to its simplicity and computational speed. However, certain limitations of the conventional Fast Haar Transform—particularly redundant computations and high memory requirements—necessitate improvements. In this paper, a Modified Fast Haar Transform (MFHT) developed by the author is proposed. Unlike the traditional two-node approach, the proposed method increases efficiency by performing computations simultaneously based on four-node structures [1-5,8].

Main Part

In the standard Fast Haar Transform algorithm, computations are performed using two-node structures. That is, the sum of two consecutive signal values yields the next iteration's approximation (sum) coefficient, while their difference gives the spectral (detail) coefficient. This study explores the implementation of the Haar transform using four-node structures to compute spectral coefficients. This method reduces the number of redundant operations and memory requirements during the spectral coefficient computation. Additionally, some intermediate sum coefficients can be bypassed. For signals with large input values, the computation of spectral coefficients via the Haar transform does not require approximation coefficients except in the final iteration [3,6,9].

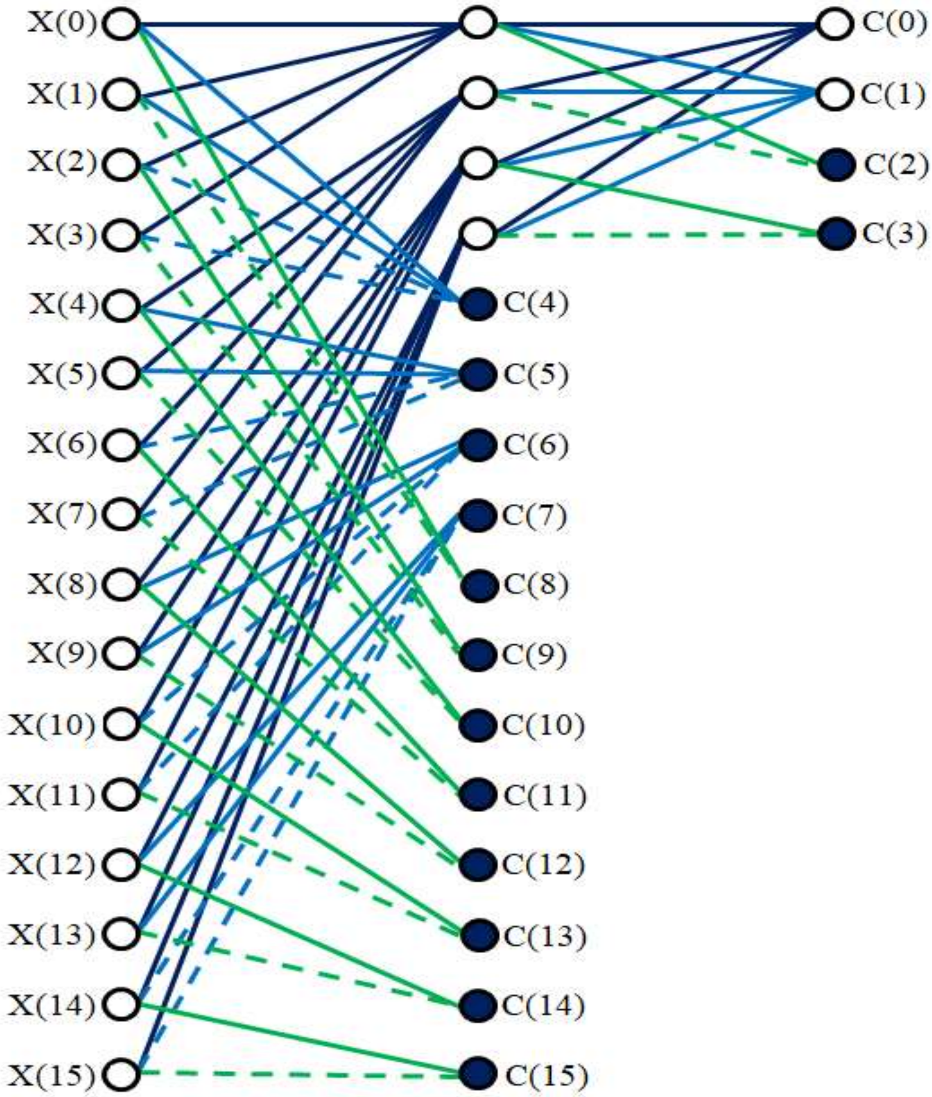

Figure 1. Graph of the Modified Fast Haar Transform

Figure 1 presents the graph of the MFHT developed by the author for a signal of length N = 16, utilizing four-node structures simultaneously during computation. In this graph, the number of iterations is defined by the formula $k = \log_4 N$.

The MFHT can also be implemented using a matrix formulation as follows [7,9]:

$$\begin{pmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \\ X(6) \\ X(7) \\ X(8) \\ X(9) \\ X(10) \\ X(11) \\ X(12) \\ X(13) \\ X(14) \\ X(15) \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix} = \begin{pmatrix} C(0) \\ C(1) \\ C(2) \\ C(3) \\ C(4) \\ C(5) \\ C(6) \\ C(7) \\ C(8) \\ C(9) \\ C(10) \\ C(11) \\ C(12) \\ C(13) \\ C(14) \\ C(15) \end{pmatrix}$$

In general, for a signal with N input values, the MFHT algorithm can be defined as follows.

For the coefficients at each iteration:

$$X_j(i) = X_{j-1}(4i) + X_{j-1}(4i+1) + X_{j-1}(4i+2) + X_{j-1}(4i+3)$$

$$C\left(i + \frac{N}{4}\right) = X_{j-1}(4i) + X_{j-1}(4i+1) - X_{j-1}(4i+2) - X_{j-1}(4i+3)$$

$$C\left(i + \frac{N}{2}\right) = X_{j-1}(4i) - (4i+1)$$

$$C\left(i + \frac{3N}{4}\right) = X_{j-1}(4i+2) - (4i+3)$$

Here is $i = 0,1, \dots, 4^{p-1}$; $p = k, \dots, 1$; $j = 0, 1, \dots, k$; $k = \log_4 N$

If the value of N is divisible by 4, all required coefficients can be obtained at each iteration step by applying a four-node computation algorithm. If N is only divisible by 2, a similar approach can be used, but a two-node computation algorithm must be applied in the final iteration step [8].

Results

Compared to the standard FHT algorithm, the MFHT algorithm reduces the number of iterations by half. This reduction in the number of iterations leads to a decrease in the number of redundant calculations required to compute the Haar coefficients. In the standard FHT, the number of redundant operations is equal to (N – 1), whereas in the MFHT, it is reduced to (N – 1) / 3.

Table 1.

Comparison table of fast transform algorithms

№	Type of Algorithm	Number of iterations of the algorithm		
		N	One-dimensional N=1024	Two-dimensional N=1024x1024
1	FHT	N	1024	2 095 104
2	MFHT	N/3	341	698 368

Table1 compares the number of iterations performed by the FHT and the MFHT algorithms for one-dimensional signals (N = 1024) and two-dimensional signals (N = 1024 × 1024).

The FHT is a conventional algorithm that requires a full 1024 iterations for a signal with 1024 elements. In the two-dimensional case, the number of iterations reaches 2,095,104.

In contrast, the MFHT is an enhanced algorithm that operates based on four-node structures. It reduces the number of iterations by nearly a factor of three - requiring only 341 iterations instead of 1024, and 698,368 iterations in the two-dimensional case.

Conclusion

The MFHT algorithm proposed in this study has shown a significant improvement in computational efficiency compared to the conventional FHT. As demonstrated in Table 1, for a one-dimensional signal of size N = 1024, the MFHT reduced the number of required iterations from 1024 to 341. Similarly, for two-dimensional signals of size 1024 × 1024, the number of iterations decreased from 2,095,104 to 698,368. This reflects an approximate threefold reduction in computational workload. Such a reduction directly impacts the number of redundant operations required for computing spectral coefficients, thereby optimizing execution time and lowering memory usage. The modified algorithm achieves this efficiency by replacing the traditional two-node structure with a four-node structure in each iteration, effectively reducing the depth of iteration and improving data throughput.

Furthermore, the algorithm's flexibility is enhanced through its graph-based representation and matrix formulation, which make it highly adaptable for software and hardware implementations. This also facilitates easier integration into embedded systems and real-time signal processing platforms. In particular, when the number of input signal elements is divisible by 4, the Modified Fast Haar Transform achieves its maximum performance, making it especially well-suited for applications in digital signal and image processing, real-time filtering, compression, and feature extraction tasks.

In summary, the MFHT algorithm provides a computationally efficient, scalable, and practical solution that addresses the limitations of traditional FHT and is highly applicable in time-critical and resource-constrained signal processing systems.

References

1. Воробьев В.А. Об эффективности параллельных вычислений // Автометрия. – 2011. - № 6, С. 50-58.

2. Гадзиковский В.И. Цифровая обработка сигналов.–М.: Солон-Пресс, 2013. – 766 с.

3. Гергель В.П. Высокопроизводительные вычисления дл многоядерных многопроцессорных систем. Учебное пособие – Нижний Новгород; Изд-во ННГУ им. Н.И.Лобачевского, 2010 – 420с.

4. Зайнидинов Х.Н. Методы и средства цифровой обработки сигналов в кусочно-полиномиальных базисах. // Монография, Академия государственного управления при Президенте РУз. Т: «Fan va texnologiyalar», 2014, -С. 192

5. Сюзев В.В. Операторы преобразования спектров в базисах Фурье и Уолша. // Наука и образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2017. № 2. С. 138-156.

6. Сюзев В.В. Основы теории цифровой обработки сигналов. Учебное пособие:–М. Издательство: «РТСофт», 2014. – 752с.

7. J. Nurmurodov, I. Yusupov, M. Gofurjonov, S. Qobilov, S. Ibragimov. Calculation of Spectral Coefficients of Signals on The Basis of Xaar By The Method Of Machine Learning. 14th International Conference on Intelligent Human Computer Interaction IHCI, October 20-22, 2022. p.547-558

8. Zaynidinov H.N., Ibragimov S.S., Tojiboyev G‘.O., Nurmurodov J.N. Efficiency of Parallelization of Xaar Fast Transform Algorithm in Dual-Core Digital Signal

Processors. 2021 8th International Conference on Computer and Communication Engineering (ICCCE). 22-23 June 2021, Kuala Lumpur, Malaysia (SCOPUS). p. 7-12

9. Zaynidinov H.N., Ibragimov S.S., Yusupov I., Tojiboyev G‘.O. Algorithm and Program for Parallel Calculating of Xaar Fast Transform in Dual-Core Special Processors. Journal of Multidisciplinary Engineering Science and Technology. ISSN: 2458-9403 (Online) Vol. 7 Issue 9, September – 2020. p. 12696-12703