

Improvement of image processing speed of the 2D Fast Complex Hadamard Transform

Yasuhito Fujita and Ken-ichi Tanaka

Department of Electronic and Bioinformatics, School of Science and Technology
Meiji University
1-1-1 Hiagshi-Mita, Tama, Kasawaki, Kanagawa, JAPAN
E-mail: {ce77073, tanaken}@isc.meiji.ac.jp

ABSTRACT

As for Hadamard Transform, because the calculation time of this transform is slower than Discrete Cosine Transform (DCT) and Fast Fourier Transform (FFT), the effectiveness and the practicality are insufficient. Then, the computational complexity can be decreased by using the butterfly operation as well as FFT.

We composed calculation time of FFT with that of Fast Complex Hadamard Transform by constructing the algorithm of Fast Complex Hadamard Transform. They are indirect conversions using program of complex number calculation, and immediate calculations. We compared calculation time of them with that of FFT.

As a result, the reducing the calculation time of the Complex Hadamard Transform is achieved. As for the computational complexity and calculation time, the result that quadrinomial Fast Complex Hadamard Transform that don't use program of complex number calculation decrease more than FFT was obtained.

Keywords: image encoding, Complex Hadamard Transform, High-speed operation, butterfly operation

1. Introduction

Currently, Discrete Cosine Transform (DCT) and Fast Fourier Transform (FFT) are used in the image encoding method. However, the computational complexity increases because calculation types both DCT and FFT are using the trigonometric function. Then, we are researching image processing using the Complex Hadamard Transform for image encoding and watermarking. In the Complex Hadamard Transform, there is no investigation the faster processing method that similar to DCT and FFT. And, it is considered that the effectiveness and the practicality are insufficient. Then, the computational complexity of Complex Hadamard Transform can be decreased to $N^2 \log_2 N$ or $N^2 \log_4 N$ from N^3 by using the butterfly operation as well as FFT. In other words, it is considered that the computing time can be reduced by using the butterfly operation, too. In this paper, we propose that the Fast Complex Hadamard Transform that uses real number and imaginary number.

2. Hadamard Transform

2.1 Hadamard Transform

Hadamard Transform can be performed by sum total of the product of the Hadamard matrix and input image. Hadamard matrix H is described by following Eq(1) and Eq(2);

$$H(0) = [1] \quad (1)$$

$$H(n) = \begin{bmatrix} H(n-1) & H(n-1) \\ H(n-1) & -H(n-1) \end{bmatrix} \quad (2)$$

where $n=1,2,3,\dots$. Hadamard matrix of $N \times N (N=2^n)$ is made from Eq(1) and Eq(2). At this time, the size of the input image should be the same as that of Hadamard matrix. Moreover, one of the characteristic of Hadamard matrix is that the retrogression row of Hadamard Transform is described by follows;

$$F = \frac{1}{n} HfH^T \quad (3)$$

where F is output, H is Hadamard matrix, f is input and H^T is transposed matrix of Hadamard matrix. Inverse transform can be performed by changing Eq(3). Inverse Hadamard Transform is described by follows;

$$f = nHFH^T \quad (4)$$

2.2 Complex Hadamard Transform

Complex Hadamard Transform is conversion that use the matrix that enhances it even to imaginary part by adding $\pm i$ to ± 1 of Hadamard Transform. Complex Hadamard Transform is described by follows;

$$H(0) = [1] \quad (5)$$

$$H(n) = \begin{bmatrix} H(n-1) & iH(n-1) \\ iH(n-1) & H(n-1) \end{bmatrix} \quad (6)$$

Complex Hadamard Transform is performed as much as Eq(3). In the case of Complex Hadamard Transform, the conjugate matrix is used instead of the transposed matrix. In Complex Hadamard matrix, conjugate matrix and inverse matrix are the same. Conversion is executed by using this property. Complex

Hadamard Transform and Inverse Complex Hadamard Transform are shown below;

$$F = \frac{1}{n} HfH^* \quad (7)$$

$$f = nHFH^* \quad (8)$$

where H^* is conjugate matrix of Complex Hadamard matrix.

3. Fast Hadamard Transform

As a method of reducing calculation cost us Hadamard Transform, the butterfly operation will be used. In this research, we propose the binomial butterfly operation and quadrinomial butterfly operation. And we made the program based on them. It becomes unnecessary to make the Hadamard matrix by using this method. It becomes unnecessary to carry out calculation using the Hadamard matrix in connection with it. And, since the computational complexity can be decreased to $N^2 \log_2 N$ or $N^2 \log_4 N$ from N^3 , it is considered that the calculation time can be reduced. Comparison of the computational complexity in Complex Hadamard Transform and Fast Complex Hadamard Transform is shown in Table.1.

Table.1 Comparison of the computational complexity

	normal	binomial	quadrinomial
N	N^3	$N^2 \log_2 N$	$N^2 \log_4 N$
2	8	4	2
4	64	32	16
8	512	192	96
16	4096	1024	512
32	32768	5120	2560
64	262144	24576	12288
128	2097152	114688	57344
256	16777216	524288	262144
512	134217728	2359296	1179648
1024	1073741824	10485760	5242880
2048	8589934592	46137344	23068672
4096	68719476736	201326592	100663296
8192	549755813888	872415232	436207616
16384	4398046511104	3758096384	1879048192
32768	35184372088832	16106127360	8053063680
65536	281474976710656	68719476736	34359738368

3.1 Fast Hadamard Transform

The figure of the binomial butterfly operation is shown below.

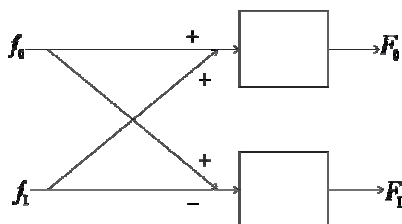


Fig.1 binomial butterfly operation (Fast Hadamard Transform)

If the expression is derived from Fig.1, it is as follows.

$$\begin{cases} x_N(k) = x_{N-1}(k) + x_{n-1}(k + 2^{N_{\max}-N}) \\ x_N(k) = x_{N-1}(k - 2^{N_{\max}-N}) - x_{n-1}(k) \end{cases} \quad (9)$$

where N_{\max} is height or width o input image ($N_{\max} = 2^{N_{\max}}$) and $N = 1, 2, \dots, N_{\max}$. At this time, the upper expression and the lower expression of Eq(9) are calculated by turns at interval of $k = 2^{N_{\max}-N}$. Thereafter, it thinks based on this butterfly operation.

3.2 Fast Complex Hadamard Transform

3.1 described the Fast Hadamard Transform. In this research, because we proposed Hadamard Transform instead of FFT, we describe the improvement of processing speed of Complex Hadamard Transform that added the imaginary part as well as FFT in this section. The algorithm of Fast Complex Hadamard Transform is that of Fast Hadamard Transform are same. Therefore, Eq(10) is obtained by transforming Eq(9) in consideration of the imaginary part.

$$\begin{aligned} x_N(k) &= x_{N-1}(k) + x_{n-1}(k + 2^{N_{\max}-N})i \\ x_N(k) &= x_{N-1}(k - 2^{N_{\max}-N})i + x_{N-1}(k) \end{aligned} \quad (10)$$

where N_{\max} is height or width o input image ($N_{\max} = 2^{N_{\max}}$) and $N = 1, 2, \dots, N_{\max}$. At this time, the upper expression and the lower expression of Eq(9) are calculated by turns at interval of $k = 2^{N_{\max}-N}$. The figure of Eq(10) is shown in below.

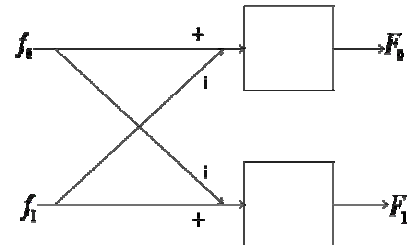


Fig.2 binomial butterfly operation (Fast Complex Hadamard Transform)

3.3 quadrinomial high-speed operation

Section 3.1 and Section 3.2 described about binomial calculations by using butterfly operation. In this section, quadrinomial high-speed operations of Hadamard Transform and Complex Hadamard Transform are described. In this case of quadrinomial high-speed operation, high-speed operation is achieved by taking the same type of 4×4 Hadamard matrix or 4×4 Complex Hadamard matrix. 4×4 Hadamard matrix and 4×4 Complex Hadamard matrix are shown in Eq(11) and Eq(12).

$$H(2) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \quad (11)$$

$$H(2) = \begin{bmatrix} 1 & i & i & -1 \\ i & 1 & -1 & i \\ i & -1 & 1 & i \\ -1 & i & i & 1 \end{bmatrix} \quad (12)$$

Next, quadrinomial high-speed operation expressions are shown in Eq(13) and Eq(14). Eq(13) is Fast Hadamard Transform and Eq.14 is Fast Complex Hadamard Transform. In this high-speed operation, decreasing the computational complexity to $N^2 \log_4 N$ from N^3 becomes possible. As a result, it is considered that a more high-speed operation than binomial calculations becomes possible. The figure of Eq(13) and Eq(14) are shown in Fig.3 and Fig.4.

$$\begin{aligned} x_N(k) &= x_{N-1}(k) + x_{N-1}(k + 2^{N_{\max} - N}) + x_{N-1}(k + 2 \cdot 2^{N_{\max} - N}) + x_{N-1}(k + 3 \cdot 2^{N_{\max} - N}) \\ x_N(k) &= x_{N-1}(k - 2^{N_{\max} - N}) + x_{N-1}(k) + x_{N-1}(k + 2^{N_{\max} - N}) + x_{N-1}(k + 2 \cdot 2^{N_{\max} - N}) \\ x_N(k) &= x_{N-1}(k - 2 \cdot 2^{N_{\max} - N}) + x_{N-1}(k - 2^{N_{\max} - N}) + x_{N-1}(k) + x_{N-1}(k + 2^{N_{\max} - N}) \\ x_N(k) &= x_{N-1}(k - 3 \cdot 2^{N_{\max} - N}) + x_{N-1}(k - 2 \cdot 2^{N_{\max} - N}) + x_{N-1}(k - 2^{N_{\max} - N}) + x_{N-1}(k) \end{aligned} \quad (13)$$

$$\begin{aligned} x_N(k) &= x_{N-1}(k) + x_{N-1}(k + 2^{N_{\max} - N})i + x_{N-1}(k + 2 \cdot 2^{N_{\max} - N})i - x_{N-1}(k + 3 \cdot 2^{N_{\max} - N}) \\ x_N(k) &= x_{N-1}(k - 2^{N_{\max} - N})i + x_{N-1}(k) - x_{N-1}(k + 2^{N_{\max} - N}) + x_{N-1}(k + 2 \cdot 2^{N_{\max} - N})i \\ x_N(k) &= x_{N-1}(k - 2 \cdot 2^{N_{\max} - N})i - x_{N-1}(k - 2^{N_{\max} - N}) + x_{N-1}(k) + x_{N-1}(k + 2^{N_{\max} - N})i \\ x_N(k) &= -x_{N-1}(k - 3 \cdot 2^{N_{\max} - N}) + x_{N-1}(k - 2 \cdot 2^{N_{\max} - N})i + x_{N-1}(k - 2^{N_{\max} - N})i + x_{N-1}(k) \end{aligned} \quad (14)$$

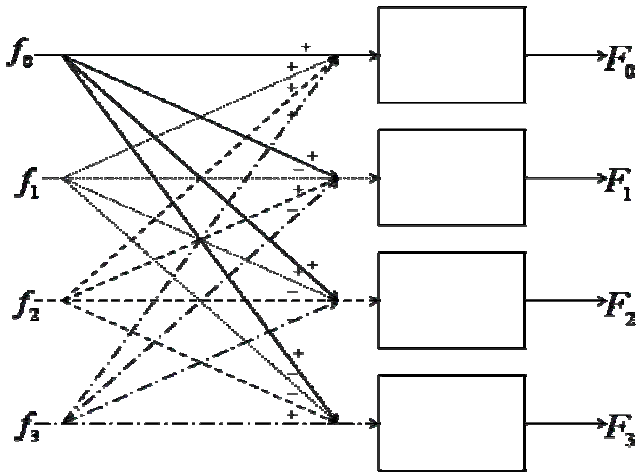


Fig.3 quadrinomial butterfly operation (Fast Hadamard Transform)

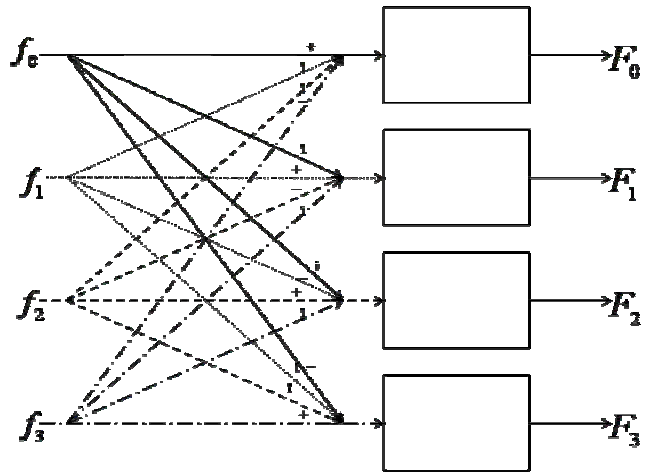


Fig.4 quadrinomial butterfly operation (Fast Complex Hadamard Transform)

4. Simulation result

4.1 Fast Complex Hadamard Transform

Fast complex Hadamard Transform described in Chapter 3 is actually performed to the image. And, we compared the images transformed by Fast Complex Hadamard Transform with the image transformed by Complex

Hadamard Transform. In this research, sample image is 256×256 pixel, gray scale, and shown in Fig.5. Moreover, the image performed Complex Hadamard Transform which doesn't use high-speed operation, the image performed binomial Fast Complex Hadamard Transform, the reproduction images of each transformed images are shown in Fig.6 to Fig.9.



Fig.5 Sample image



Fig.8 reproduction image (Complex Hadamard Transform)

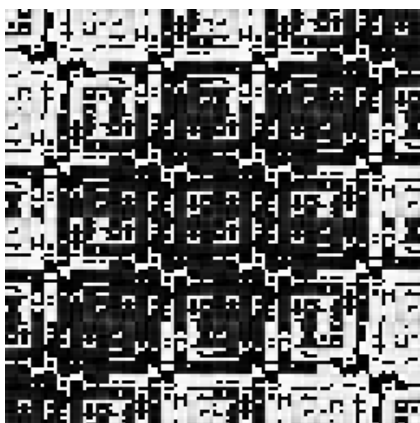


Fig.6 Conversion image (Complex Hadamard Transform)



Fig.9 reproduction image (Fast Complex Hadamard Transform)

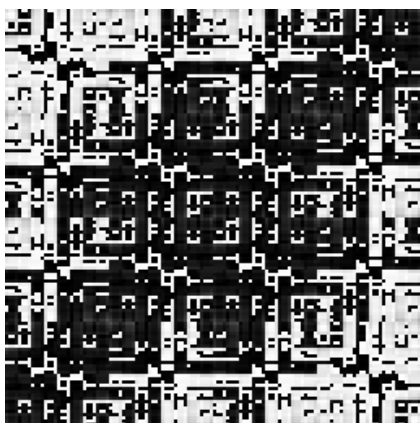


Fig.7 Conversion image (Fast Complex Hadamard Transform)

From the processing result, the image converted by Complex Hadamard Transform and the image converted by Fast Complex Hadamard Transform was same. That is, it is considered that binomial Fast Complex Hadamard Transform was able to be performed normally. Moreover, the conversion image by quadrimonial Fast Complex Hadamard Transform was able also to obtain the comparable result. Therefore, it is considered that

quadrimonial Fast Complex Hadamard Transform was also to be performed normally. But, quadrimonial Fast Complex Hadamard Transform was not able to be applied in the image of 128×128 pixel. In the case of quadrimonial Fast Complex Hadamard Transform, it is because the image depends on the power of four.

Table.2 Comparison of the time which the conversion took

Trial	CHT	FCHT	FCHT	FCHT	FCHT	FFT
1	0.968	0.031	0.031	0.016	0	0.016
2	0.969	0.046	0.032	0.032	0	0.016
3	0.984	0.047	0.063	0.015	0.015	0.016
4	0.968	0.031	0.047	0.016	0.015	0.016
5	0.984	0.031	0.031	0.015	0.016	0.015
6	0.968	0.031	0.031	0.016	0.016	0.016
7	0.969	0.046	0.047	0.015	0	0.015
8	0.984	0.047	0.047	0.015	0.015	0.016
9	0.968	0.047	0.031	0.016	0.015	0.016
10	0.969	0.031	0.031	0.016	0.016	0
11	0.969	0.046	0.047	0.032	0.015	0.016
12	0.968	0.032	0.032	0.016	0.016	0.016
13	0.984	0.031	0.047	0.016	0.016	0.015
14	0.985	0.047	0.047	0.015	0.015	0.015
15	0.984	0.032	0.032	0.016	0.016	0.016
16	0.969	0.047	0.047	0.016	0.016	0.016
17	0.984	0.047	0.032	0.015	0.015	0.015
18	0.969	0.047	0.031	0.016	0	0.015
19	0.984	0.031	0.032	0.031	0	0.016
20	0.968	0.047	0.031	0.031	0.016	0.016
Total	19.495	0.795	0.769	0.376	0.233	0.298
Average	0.97475	0.03975	0.03845	0.0188	0.01165	0.0149
MAX	0.985	0.047	0.063	0.032	0.016	0.016
MIN	0.968	0.031	0.031	0.015	0	0
Rank	6	5	4	3	1	2

- CHT Normal
- FCHT Use program of complex number calculation(binomial)
- FCHT Use program of complex number calculation(quadrinomial)
- FCHT Not use program of complex number calculation(binomial)
- FCHT Not use program of complex number calculation(quadrinomial)
- FFT Fast Fourier Transform

4.2 Comparison at calculation time

In this section, we compare the calculation time of Fast Complex Hadamard Transform with the calculation time of FFT. There are four kinds of Fast Hadamard Transform. They are indirect conversions using program of complex number calculation, and immediate conversion that don't use it. There are each binomial calculations and quadrinomial calculations. We performed each transforms 20 times, and took the average of calculation time, and compared and inquired about it. The result of comparison

is shown in Table.1. 0 sec in table.2 shows that processing was performed at the time of 0.000 or less. From table.2, it turns out that the Fast Complex Hadamard Transform has improved calculation time sharply. And Fast Complex Hadamard Transform was able to be processed earlier than Complex Hadamard Transform. Three kinds didn't have early than the calculation time of FFT. But the result that only quadrinomial Fast Complex Hadamard Transform that don't use program of complex number calculation is shorter than the calculation time of FFT was able to be obtained.

5. Conclusion

In this paper, the calculation speed of Complex Hadamard Transform was made quickly by using butterfly operation. First of all, we constructed the algorithm of high-speed operation by using butterfly operation and proposed four kinds of algorithms. They are binomial Fast Complex Hadamard Transform and quadrinomial Fast Complex Hadamard Transform. There are indirect conversions using the program of complex number calculations and immediate conversion that don't use it. These transform was performed to the image. We compared as the conversion image of Complex Hadamard Transform and the conversion images of Fast Complex Hadamard Transform. The image similar to the image of Complex Hadamard Transform was able to be obtained.

Next, we compared as the calculation times of Fast Complex Hadamard Transform, that of Complex Hadamard Transform and that of FFT. Improvement in the speed of Complex Hadamard Transform was possible. Three kinds didn't have early than the calculation time of FFT. But the result that only quadrinomial Fast Complex Hadamard Transform that don't use the program of complex number calculation is shorter than the calculation time of FFT was able to obtain. Moreover, though the calculation time of binomial Fast Complex Hadamard Transform that doesn't use complex number calculation was slightly slower than that of FFT, the calculation time of that was able to obtain a result almost equal to the calculation time of FFT. Since binomial Fast Complex Hadamard Transform is simple for a program better than quadrinomial Complex Hadamard Transform, it can expect convenience and practicality comparable as quadrinomial Fast Complex Hadamard Transform.

As future tasks, as the coding system replaced with FFT, we want to catch up about the creation of a hologram and the possibility of practical use etc.

6. REFERENCES

- [1] Zen-ichi Kiyasu, "The Hadamard matrix and its application -from the composition theory to the Walsh function-", IEICE, 1983
- [2] Chanta K's, "Compression algorithm -mounts by the principle and C language of coding-", Softbank Publishing, 2005
- [3] Takeshi Agui / Masayuki Nakajima, "Computer Image Processing", Morikita Publishing Co.,Ltd, 1994
- [4] Shigeo Tsuji-i / Kazuo kamata, "Digital Signal Processing", SHOLODO.co.LTD, 2004
- [5] Bogdan J.FALOWSKI / Susanto RAHARDJA, "Complex Hadamard Transforms: Properties, Relations and Architecture", IEICE TRASN.FUNDAMENTALS, VOL.E87-A, NO.8 AUGUST 2004