Fast 2-Dimensional Hadamard Transform

M. L. Hsien 謝明倫
Department of Information Engineering
I- Shou University
Kaohsiung, Taiwan, R.O.C
m873006m@csa500.isu.edu.tw

J. H. Jeng 鄭志宏
Department of Information Engineering
I-Shou University
Kaohsiung, Taiwan, R.O.C
jjeng@csa500.isu.edu.tw

Abstract

The Hadamard transform (HT) has been used extensively in image and speech applications. The advantage of HT is that the procedure of the transform is simple and fast, because there are only +/- operations involved. In this paper, a fast 2-dim HT is presented, which utilizes less amount of +/- operations than the direct method. The fast algorithm is done by successive Harr wavelet transforms (Harr WT). For both methods, the exact amounts of +/- operations are counted. The fast algorithm is verified using C++ programs. Using this new method, the transform speed can be enhanced effectively.

Key Words : ***Hadamard transform, Wavelet transform, Harr.***

Introduction

The 2-dim HT is often used in image processing, image coding and other relative applications [1 , 2, 4]. For instance, one can divide the original image into a set of small blocks and use 2-dim HT to decompose the blocks into independent components to reduce the correlation, then, use these properties for further applications. The most important feature of HT is the computational efficiency. To demonstrate this, let $f(x,y)$ be an image block and $F(s,t)$ be the transformed block given as

$$F = H_k^t * f * H_k \tag{1}$$

where H_k is the $k*k$ Hadamard matrix, H_k^t is the transpose of H_k and $k = 2^m$ for same m. The matrix H_k only contains +1 and −1. So, there are only +/- operations performed on the pixel values of f. Therefore, HT is easy to implement both in software and hardware. In this paper, a fast 2-dim HT (FHT) is proposed. The amount of +/- operation can be reduced using the high and low band mixed matrix based on Harr WT, so that, the computation speed can be improved.

The Fast Hadamard Transform

The lowest order of Hadamard matrix is of order 2, which is given as

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2}$$

Let H_k be the Hadamard matrix of order k. Then the next order Hadamard matrix H_{2k} can be derived recursively from H_k by

$$H_{2k} = \begin{bmatrix} H_k & H_k \\ H_k & -H_k \end{bmatrix} \tag{3}$$

Thus, H_4 can be derived from H_2 as

$$H_4 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \tag{4}$$

Now, one will use the low band and high band mixed matrix Q_k based on the Harr WT to represent the Hadamard matrix H_k. Let $Q_4^{(l)}$ and $Q_4^{(h)}$ be the low band and high band wavelet matrix generated from Harr WT, i.e.

$$Q_4^{(l)}=\begin{bmatrix}1&0\\1&0\\0&1\\0&1\end{bmatrix},\quad Q_4^{(h)}=\begin{bmatrix}1&0\\-1&0\\0&1\\0&-1\end{bmatrix} \tag{5}$$

Define $Q_4=\left[Q_4^{(l)},Q_4^{(h)}\right]$, which is a $4*4$ matrix given as

$$Q_4=\begin{bmatrix}1&0&1&0\\1&0&-1&0\\0&1&0&1\\0&1&0&-1\end{bmatrix} \tag{6}$$

Then, by a simple calculation, H_4 can be rewritten as

$$Q_4*Q_4=\begin{bmatrix}1&0&1&0\\1&0&-1&0\\0&1&0&1\\0&1&0&-1\end{bmatrix}\begin{bmatrix}1&0&1&0\\1&0&-1&0\\0&1&0&1\\0&1&0&-1\end{bmatrix}=\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}=H_4 \tag{7}$$

In general, let $k=2^m$, the mixed Harr WT matrix is of size $k*k$, which is of the form

$$Q_k=\begin{bmatrix}1&0&\dots&0&1&0&\dots&0\\1&0&\dots&0&-1&0&\dots&0\\0&1&\dots&0&0&1&\dots&0\\0&1&\ddots&0&0&-1&\dots&0\\\vdots&\vdots&\dots&\ddots&\vdots&\vdots&\dots&0\\\vdots&\vdots&\dots&\vdots&\ddots&\vdots&\dots&0\\0&0&\dots&1&0&0&\dots&1\\0&0&\dots&1&0&0&\dots&-1\end{bmatrix} \tag{8}$$

It can be calculated that the m-th power of Q_k equals to the HT matrix H_k, i.e.

$$H_k=Q_k^{\,m} \tag{9}$$

Thus, from (9), the 2-dim HT of an image block $f(x,y)$ in (1) can be expressed as

$$\begin{aligned}F&=H_k^t\,f\,H_k\\&=(\underbrace{Q_k^t\ldots(Q_k^t(Q_k^t}_{\text{m times}}\,f\,\underbrace{Q_k)Q_k)\ldots Q_k}_{\text{m times}})\end{aligned} \tag{10}$$

The 2-dim HT in (1) is performed directly using matrix multiplication, while the formula (10) is done by m successive operations of matrix multiplication of the form $Q_k^t f\,Q_k$. Although the HT is performed for m times, there are fewer +/- operations required in each step. The total amount of +/- operations of (1) and (10) are counted as follows.

Let $f(x,y)$ be a 2-dim image of the form

$$f(x,y)=\begin{bmatrix}f_{1,1}&f_{1,2}&\cdots&f_{1,k}\\f_{2,1}&f_{2,2}&\cdots&f_{2,k}\\\vdots&\vdots&\ddots&\vdots\\f_{k,1}&f_{k,2}&\cdots&f_{k,k}\end{bmatrix} \tag{11}$$

In (1), let $F=H_k^t g=H_k^t f H_k$. Since each entry in H_k is either 1 or –1, thus it requires $(k-1)$ +/- operations to obtain $g_{i,j}$ for each i,j. Thus there are total $k^2(k-1)$ +/- operations required to calculate g. Similar there are $k^2(k-1)$ +/- operations required to obtain F. Thus, the total number of +/- operations required to perform the HT in (1) is $2k^2(k-1)$.

For the formula in (10), there are m operations of matrix multiplication, which are

Step 1 : $p_1=Q_k^t\,f\,Q_k$

Step 2 : $p_2=Q_k^t\,p_1\,Q_k$

$\vdots$

Step m : $F=p_m=Q_k^t\,p_{(m-1)}\,Q_k$

Assume, in step 1, $p_1=Q_k^t g=Q_k^t f Q_k$. From (8), there are only two nonzero entries 1 and –1 in each column and row. Thus, there is only one +/- operation required to obtain $g_{i,j}$ for each i,j. Thus there are total k^2 +/- operations required to calculate g. Similar there are k^2 +/- operations required to obtain p_1. Thus, the total number of +/- operations required in step1 is $2k^2$, step2 is the same as step1 and so on. Thus, the total number of +/- operations required to perform the FHT in (10) is $2k^2\log_2 k$

Obviously, the amount of +/- operations is less than that of the direct method in (1), which is $2k^2(k-1)$.

Table 1. The amount of +/- operations and the times of 1000 computations for HT and FHT

Transforms	Image Size	Transform Iteration Times	+/- Operations	Computation Time (ms)
HT	8*8	1000	896 * 1000	10
	16*16	1000	7680 * 1000	80
FHT	8*8	1000	384 * 1000	5
	16*16	1000	2048 * 1000	31

Experimental Result

Both of the original HT and the FHT proposed in this paper are implemented using C++ programs on the Intel Pentium-II 400 PC. The algorithm of FHT is verified by comparing the output of FHT to that of the original HT method. To demonstrate the computation speed, image blocks of size 8*8 and 16*16 are tested using the two methods. The amount of +/- operations and the computation time are given in Table1. The time is in mini-second, which is the time of 1000 computations. Obviously, the 2-dim FHT is faster than the original HT.

This work was supported by the National Science Council, ROC, under Grant NSC 89-2213-E-214-031.

References

[1] W. K. Pratt, J. Kane, and H. C. Andrews, "Hadamard transform image coding," Proc. IEEE, vol. 57, pp. 58-68, 1969.

[2] S. S. Agaian, Lecture Notes in Mathematics 1168, Hadamard matrices and their Applications. New York: Springer-Verlag, 1980

[3] Chen Anshi, Li Di and Zhou Renzhong, "A research on fast Hadamard transform (FHT) digital systems" IEEE TENCON'93, vol 3, pp. 541-545, 1993

[4] Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", pp. 136-147, © 1992 by Addison-Wesley Publishing Company, inc.

[5] 戴顯權, "資料壓縮", 松崗圖書公司, 第 10 章, pp. 18-20, 民國 85 年