Performance analysis of different orthogonal transform for image processing application

Rita Rani

Abstract
In modern communication technologies, the demand of image data compression increasing rapidly. This paper presents different methods for image compression like DCT, Walsh, Hadamard and Walsh-Hadamard. Main objective of this paper is to achieve the higher compression ratio and lower the Mean square error. Different test images over different orthogonal transform are used to achieve the performance and coding efficiency. Hadamard transform gives less improvement in PSNR and large value of MSE as compare to walsh transform which gives better PSNR and Lesser MSE. Walsh-Hadamard enhances its performance and finally DCT gives least MSE and high PSNR.

Keywords: Walsh, Hadamard, Walsh-Hadamard, DCT, MSE and PSNR

1. Introduction
Digital image compression techniques are used to reduce the size of an image. Suppose if we upload the 16MB image then it will take four minutes to download by the use of 64kbps channel, but if we use the compression techniques its size will reduced to 800 kb and the time that it will take to download is 12 seconds. Hence the demand of image compression becomes necessary [1].

Image compression is divided into two categories: Transform coding and spatial coding [2]. In transform coding we used different techniques like DFT, DCT, Walsh, Hadamard, and Walsh-Hadamard [3]. In discrete Fourier transform computational complexity is very large as we have to calculate the sine and cosine terms. The remaining transforms method are orthogonal transform method in which we use forward transform kernel as well as inverse transform kernel. If we compress the image using forward kernel then it is necessary to use the inverse kernel to recover the original image.

Transform method are very useful for digital image compression. Image compression technique can be divided into two categories (1) Lossless (2) Lossy technique [4]. Lossless techniques compressed the image without losing the information but in case of lossy technique some information lost during compression. Hence above compression technique share common architecture. It’s become difficult to improve the performance of coding under these architecture. Hence to achieve high compression performance more techniques are induced in image coding and compression.

DCT (Discrete Cosine Transform) in past decade was very popular for image compressing the image, because it gives optimum performance and its implementation cost was very low. Compression techniques like JPEG [5] and MPEG [6] are based on DCT. In the image compression techniques main stages are transform and quantization, modelling and ordering and last stage is entropy coding. Compression encoding method is based upon DCT in which quantization table is used and which is determined by various quantization steps [7]. These quantization steps makes quantization table much complex. DCT, Walsh, Hadamard and Walsh-Hadamard Transform brought forward in this paper which can be applied for image compression. In recent times, much of the research activities in image coding have been focused on the DWT, which has become a standard tool in image compression applications because of their data reduction capability [8]. Image compression methods are also based on the use of non-orthogonal filters such as Gabor wavelet transform [9].
Aim of this paper is to analyze the performance of different orthogonal transform. Walsh Hadamard transform and DCT gives the better performance as compare to Walsh, Hadamard Transform.

2. Walsh Transform

The Walsh Transform is an orthogonal transform where we use the forward transform kernel as well as inverse transform kernel. Forward Transform kernel is used to compress the image and if we want to recover that image then Inverse kernel is required. In Walsh Transform computational complexity is less as compared to DFT and DCT because there are no cosine and sine term as only addition and subtraction make the computation easy. The Walsh transform uses square-waves and these vary from -1 to +1. The advantage of the Walsh transform is, it does not require floating-point math or transcendental functions. The inverse Walsh kernel is same as forward kernel.

Forward Kernel and inverse kernel are given below in 1-D form:

\[ g(x, u) = \frac{1}{N} \prod_{i=0}^{n-1} (-1)^{[b_i(x)h_{n-1-i}(u)]} \]  
\[ h(y, v) = \frac{1}{N} \prod_{i=0}^{n-1} (-1)^{[b_i(y)h_{n-1-i}(v)]} \]  

Forward and inverse kernel in 2-D form are given below:

\[ g(x, y, u, v) = \frac{1}{N} \prod_{i=0}^{n-1} (-1)^{[b_i(x)h_{n-1-i}(u)+b_i(y)h_{n-1-i}(v)]} \]  
\[ h(x, y, u, v) = \frac{1}{N} \prod_{i=0}^{n-1} (-1)^{[b_i(x)h_{n-1-i}(u)+b_i(y)h_{n-1-i}(v)]} \]  

2.1. Hadamard Transform

\[ H_n = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \]

\[ H_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \]

\[ H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \]

\[ H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \]

\[ H_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 \end{bmatrix} \]

Fig 1: Hadamard Transform Matrix

The Hadamard transform is same as Walsh because it uses square waves of -1 to +1 in amplitude. The transform is easily calculated from multiply the image with Hadamard matrix: The Hadamard transform \(H_n\) is a \(2^n \times 2^n\) matrix the Hadamard matrix transforms \(N=2^n\) real numbers into \(2^n\) real numbers which is shown in figure 1:

Hadamard transform have a computational complexity of \(O(N^2)\). The hadamard transform as show in figure 2 given below, requires only \(N \log N\) additions or subtractions.

2.2. Walsh Hadamard Transform

For representing sequency components which are contained in the signal in low-to-high order, we can re-ordering the rows and columns of Hadamard matrix \((H)\) according to their sequencies. In order to convert a sequency \((S)\) into their corresponding index number \((k)\) in Hadamard order is a three-step process:

Representation \(S\) in binary form:

\[ S = (S_{n-1}, S_{n-2}, \ldots, S_0) = \sum_{i=0}^{n-1} S_i \]

1. Next step is to convert \(S\) into Gray code:

\[ g_i = S_i \oplus S_{i+1} \]

2. Next step is to reverse the \(g_i\) to get \(k_i\):

\[ k_i = g_{n-1-i} \oplus S_{n-i} \]

Now \(k\) can be calculated as:

\[ k = (k_{n-1} k_{n-2} k_{n-3} \ldots k_0) = \sum_{i=0}^{n-1} S_{n-i} \oplus S_{n-1} \]

Where \(j=n-1-i\) or equivalently

\[ i=n-1-j \]

\[ \log_2 N = \log_2 8 = 3 \]

Let’s take an example,

\[
\begin{align*}
\begin{array}{cccccccc}
\text{binary} & 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\
\text{Gray code} & 000 & 001 & 011 & 010 & 110 & 111 & 101 & 100 \\
\text{bit-reverse} & 000 & 100 & 110 & 101 & 010 & 011 & 111 & 110 \\
k & 0 & 4 & 6 & 2 & 3 & 7 & 5 & 1
\end{array}
\end{align*}
\]

We can calculated the Walsh–Hadamard matrix as

\[
W = \frac{1}{\sqrt{8}} \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 \\
1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 \\
1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 \\
1 & -1 & -1 & 1 & -1 & 1 & -1 & 1
\end{bmatrix}
\]
The first column on the right of the matrix is for the sequency \( S \) of the corresponding row, which is the index for the sequency ordered matrix, and the second column is the index \( K \) of the Hadamard ordered. We see that this matrix is still symmetric:
\[
W[k, m] = W[m, k]
\]

**Table 1**: Original Test images and their reconstructed images

<table>
<thead>
<tr>
<th>Processing Images</th>
<th>Processing Images</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Original image</strong></td>
<td><strong>Reconstruced image</strong></td>
</tr>
<tr>
<td>“rice.png” with Hadamard</td>
<td>“cameraman.tif” with Hadamard</td>
</tr>
<tr>
<td>“rice.png” with Walsh</td>
<td>“cameraman.tif” with Walsh</td>
</tr>
<tr>
<td>“rice.png” with Walsh-Hadamard</td>
<td>“cameraman.tif” with Walsh-Hadamard</td>
</tr>
</tbody>
</table>
Table 2: The MSE & PSNR measurement of test images

<table>
<thead>
<tr>
<th>Database Images</th>
<th>Image Compression Transforms</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hadamard</td>
</tr>
<tr>
<td>camaraman.tif</td>
<td>MSE</td>
</tr>
<tr>
<td></td>
<td>148.7362</td>
</tr>
<tr>
<td>rice.png</td>
<td>76.0221</td>
</tr>
</tbody>
</table>

3. Result Discussion
We have applied the different orthogonal transform like DCT, Walsh, Hadamard and Walsh Hadamard on two different test images. The Table 1 shows the original image and reconstructed image in matlab by applying all these transforms separately over two different images camaraman.tif and rice.png. Table 2 represents the peak signal to noise ratio (PSNR) and mean square error (MSE) values for different orthogonal transform for two different images. From Hadamard to Walsh Transform MSE reduces 68.88% and PSNR improves 16.09% and in case of Walsh to Walsh Hadamard MSE reduces 0.35% and PSNR improves 0.049% and from Walsh Hadamard to DCT MSE reduce 65.46% and PSNR improves 12.77%.

4. Conclusion
From the above discussion we can concluded that any orthogonal transform applied should have better PSNR and lesser MSE. Walsh Transform that we have studied have a PSNR of 31.5114 and MSE is 46.2742, but in case of Hadamard Transform PSNR gives a value of 26.4406 and MSE gives a value of 148.7362. Hence Performance of DCT transform improves as compare to other orthogonal transforms. We can also calculate compression ratio of these transform to evaluate their performance.

5. References