

Performance Analysis of Advanced Transforms in 2-D Signals

Ms. K. Prasuna

Asst. Prof,

Department of ECE,

Vijaya Institute of Technology for Women,

Vijayawada, AP

Abstract:-Data compression is an important tool in digital image processing to reduce the burden on the storage and transmission systems. Fast 1D-2D Framelet Transform algorithm for computing advance transforms are proposed. For a 2D framelet transformation, the algorithm is applied in x-direction first and then in y-direction. The propose method reduces heavily processing time for decomposition of video sequences keeping or overcoming the quality of reconstructed sequences In addition, it cuts heavily the memory demands. In this paper, Discrete Hartley type transform (CCT) and Fast Fourier transforms (FFT) methods are used for both data compression and decompression process. Algorithms are developed and tested using the two methods on different images. A comparison with respect to mean square error with the original image is also presented. The main advantage of discrete Hartley type transform is, it is a real transform. Discrete cosine transform also has similar performance as that of discrete Hartley type transform. An algorithm for compression using FFT method and Framelet is also presented.

Keywords-Wavelet transform, Framelet, FFT,DCT

I. INTRODUCTION

Though standard DWT is a powerful tool for analysis and processing of many real-world signals and images, it suffers from three major disadvantages, Shift- sensitivity, Poor directionality and Lack of phase information. These disadvantages severely restrict its scope for certain signal and image processing applications [1].

Other extensions of standard DWT such as Wavelet Packet Transform (WP) and Stationary Wavelet Transform (SWT) reduce only the first disadvantage of shift-sensitivity but with the cost of very high redundancy and involved computation. Recent research suggests the possibility of reducing two or more above-mentioned disadvantages using different forms of Wavelet Transforms [2-4] with only limited (and controllable) redundancy and moderate computational complexity. Data compression before transmission reduces the channel bandwidth requirement and memory space [1] to [7]. There are different transforms available for data compression. These are Karhunen – Loeve Transform, Discrete Cosine

Transform, and Sampling and Interpolation process using Fast Fourier Transform.

In this paper Discrete Hartley type transform is used for data compression in two dimensions, which is directly applicable to digital image processing of RPV photographed images. The performance is compared with the existing Fast Fourier Transform. The Hartley transform (HT) was developed as a substitute to Fourier transform (FT) in applications where the data is in real domain. [6], [7]. The HT was also defined in two dimensions in which kernel is not separable . Another modified separable Hartley type transform, named the CAS-CAS transform (CCT) was also defined. CAS stands for Cosine plus Sine. CCT for two dimensional cosine plus sine transform. Algorithms are developed for data compression and decompression using the two transforms namely CCT and FFT in two dimensions. Comparison of compressed and decompressed image with the original image with respect to mean square error is carried out.

II. CCT METHOD

Discrete Hartley type transform in two dimensions as separable transform is the CAS-CAS transform and is called as CCT. Hartley transform relations in 2-D in which the kernel is not separable are

$$H(u,v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m,n) \text{cas} \left(\frac{mu}{M} + \frac{nv}{N} \right) \quad (1)$$

and reverse transform is

$$X(m,n) = \frac{1}{M \times N} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} H(u,v) \text{cas} 2\pi \left(\frac{mu}{M} + \frac{nv}{N} \right) \quad (2)$$

$\text{cas} 2\pi \left(\frac{mu}{M} + \frac{nv}{N} \right)$ is not a separable kernel. But for 2-D DFT the kernel is separable. Hence the row column decomposition method [14] of computing 2-D transform from 1-D fast Fourier transform algorithm can be used. In

case of 2-D Hartley transform row column can not be applied directly. To take advantage of the fast 1-D HT algorithms, a separable Hartley like transform namely CAS-CAS transform that is CCT is shown.

$$T(u,v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m,n) \text{cas} \frac{2\pi mu}{M} \text{cas} \frac{2\pi nv}{N} \tag{3}$$

$$X(m,n) = \frac{1}{M \times N} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} T(u,v) \text{cas} \frac{2\pi mu}{M} \text{cas} \frac{2\pi nv}{N} \tag{4}$$

Where

$$\text{cas}(\alpha) = \cos(\alpha) + \sin(\alpha) \quad \text{and}$$

$$H(u,v) = \frac{1}{2} [T(u,v) + T(-u,v) + T(u,-v) - T(-u,-v)] \tag{5}$$

In this way 2-D Discrete Hartley transform can be computed. For the compression and decompression process only CCT is used instead of Hartley transform.

A. Algorithm for Compression and De-Compression Using CCT

- a) The analog Image is scanned to get digital image with a size of pixels (256 x 256) as reference data matrix A. Instead of particular size of the digital image data, general size of the original image matrix A is considered as (M, N).
- b) CCT of this matrix A is Computed using equation (3) as Matrix B which is of same size as A. Important observation is that the CCT coefficient matrix is having larger amplitudes at the four corners and the value diminishes, negligibly small towards central region.
- c) Based on the above principle, the CCT coefficients matrix B is modified to reduce its size to (M/2, N/2), to achieve a compression factor of 4 by neglecting small amplitude coefficients. Construct a new matrix C (u, v) from B (u, v) which is reduced in size as follows

$$\begin{aligned} C(u,v) &= B(u,v) && \text{for } u = 0, 1, \dots, \left(\frac{M}{4}\right) - 1 \\ & && \text{for } v = 0, 1, \dots, \left(\frac{N}{4}\right) - 1 \\ &= B(u, v + \frac{N}{2}) && \text{for } u = 0, 1, \dots, \left(\frac{M}{4}\right) - 1 \\ & && \text{for } v = \left(\frac{N}{4}\right), \dots, \left(\frac{3N}{4}\right) - 1 \\ &= B(u + \frac{M}{2}, v) && \text{for } u = \left(\frac{M}{4}\right), \dots, \left(\frac{3M}{4}\right) - 1 \\ & && \text{for } v = 0, 1, \dots, \left(\frac{N}{4}\right) - 1 \\ &= B(u + \frac{M}{2}, v + \frac{N}{2}) && \text{for } u = \left(\frac{M}{4}\right), \dots, \left(\frac{3M}{4}\right) - 1 \\ & && \text{for } v = \left(\frac{N}{4}\right), \dots, \left(\frac{3N}{4}\right) - 1 \end{aligned}$$

(u, v) is matrix of size (M/2, N/2), a compression ratio of 4 with respect to original size of the image matrix A of size (M, N) is achieved.

$$\begin{aligned} D(u,v) &= C(u,v) && \text{for } u = 0, 1, \dots, \left(\frac{M}{4}\right) - 1 \\ & && \text{for } v = 0, 1, \dots, \left(\frac{N}{4}\right) - 1 \\ &= 0 && \text{for } u = 0, 1, \dots, \left(\frac{M}{4}\right) - 1 \\ & && \text{for } v = \left(\frac{N}{4}\right), \dots, \left(\frac{3N}{4}\right) - 1 \\ &= C(u, v - \frac{N}{2}) && \text{for } u = 0, 1, \dots, \left(\frac{M}{4}\right) - 1 \\ & && \text{for } v = \left(\frac{3N}{4}\right), \dots, (N-1) \\ &= 0 && \text{for } u = \left(\frac{M}{4}\right), \dots, \left(\frac{3M}{4}\right) - 1 \\ & && \text{for } v = 0, 1, \dots, (N-1) \\ &= C(u - \frac{M}{2}, v) && \text{for } u = \left(\frac{3M}{4}\right), \dots, (M-1) \\ & && \text{for } v = 0, 1, \dots, \left(\frac{N}{4}\right) - 1 \\ &= 0 && \text{for } u = \left(\frac{3M}{4}\right), \dots, (M-1) \\ & && \text{for } v = \left(\frac{N}{4}\right), \dots, \left(\frac{3N}{4}\right) - 1 \\ &= C(u - \frac{M}{2}, v - \frac{N}{2}) && \text{for } u = \left(\frac{3M}{4}\right), \dots, (M-1) \\ & && \text{for } v = \left(\frac{3N}{4}\right), \dots, (N-1) \end{aligned}$$

- d) This reduced CCT coefficient matrix is to be transmitted instead of the original matrix, so that memory space is saved for storage and burden on the transmission channel is reduced by a factor of 4.
- e) At the receiver, after receiving the modified CCT matrix C (u, v), is appended with zeros at respective regions, to make it to reference image size as follows
- f) Each D (u, v) is multiplied by the square of the compression factor.
- g) Inverse CCT is Performed using equation (4) on the sequence D (u, v) of size (M, N) to obtain the replica of Perform the original image A.

III. FAST FOURIER TRANSFORM METHOD

V. FRAMELETS

The Fourier transform relations in 2-D

$$\text{are } X(u, v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m, n) e^{-j2\pi\left(\frac{mu}{M} + \frac{nv}{N}\right)} \tag{6}$$

where $m = 0, 1, \dots, M-1$, $n = 0, 1, \dots, N-1$

$$x(m, n) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} X(u, v) e^{-j2\pi\left(\frac{mu}{M} + \frac{nv}{N}\right)} \tag{7}$$

where $u = 0, 1, \dots, M-1$, $v = 0, 1, \dots, N-1$

A. Algorithm for Compression and Decompression Using FFT

The procedure is similar to that of Hartley type transform for data compression and decompression using Fast Fourier transform. The algorithmic steps given above for Hartley transform, are to be used for this also by replacing CCT with FFT. FFT magnitude coefficients are also concentrated at the corners as that of CCT.

IV. ALGORITHM FOR COMPRESSION AND DECOMPRESSION USING DCT

The Discrete Cosine Transform relations in 2-D are

$$X(u, v) = \alpha_u \alpha_v \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m, n) \cos\left(\frac{\pi(2m+1)u}{2M}\right) \cos\left(\frac{\pi(2n+1)v}{2N}\right) \tag{8}$$

Where $m = 0, 1, \dots, M-1$; and $n = 0, 1, \dots, N-1$

$$\alpha_u = \frac{1}{\sqrt{M}} \text{ For } u = 0$$

$$\alpha_u = \sqrt{\frac{2}{M}} \text{ For } u = 1, \dots, M-1$$

$$\alpha_v = \frac{1}{\sqrt{M}} \text{ For } v = 0$$

$$\alpha_v = \sqrt{\frac{2}{M}} \text{ For } v = 1, \dots, N-1$$

And 2-D inverse discrete cosine Transform is

$$x(m, n) = \alpha_u \alpha_v \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X(u, v) \cos\left(\frac{\pi(2m+1)u}{2M}\right) \cos\left(\frac{\pi(2n+1)v}{2N}\right) \tag{9}$$

Where $u = 0, 1, \dots, M-1$, and $v = 0, \dots, N-1$

Equation (8) is forward transform and equation (9) is reverse transform.

Framelet are very similar to wavelets but have some important differences. In particular, whereas wavelets have an associated scaling function $F(t)$ and wavelet function $y(t)$, framelets have one scaling function $F(t)$ and two wavelet functions $y1(t)$ and $y2(t)$.

The scaling function $F(t)$ and the wavelets $y1(t)$ and $y2(t)$ are defined through these equations by the low-pass (scaling) filter $h0(n)$ and the two high-pass (wavelet) filters $h1(n)$ and $h2(n)$.

To compute a single level for 1-D signal the next steps should be followed:

- (a) Checking input dimensions: Input vector should be of length N , where N must be even and $N > =$ length of analysis filters
- (b) Construct a transformation matrix: using transformation matrices given in (10) and (11)
- (c) Transformation of input vector, which can be done by apply matrix multiplication to the $3N/2 \times N$ constructed transformation matrix by the $N \times 1$ input vector

$$Y = [W]_{\frac{3N}{2} \times N} \cdot [X]_{N \times 1}$$

Where

$$X_L = Y \left[0 : \frac{N}{2} - 1 \right]$$

$$X_{H1} = Y \left[\frac{N}{2} : N - 1 \right]$$

$$X_{H2} = Y \left[N : \frac{3N}{2} - 1 \right]$$

2-D Framelet transform: The choice of the transform to be used depends on a number of factors, in particular, computational complexity and coding gain.

- (a) Checking input dimensions: Input matrix should be of length $N \times N$, where N must be even and $N > =$ length of analysis filters
- (b) For an $N \times N$ matrix input 2-D signal, X , construct a $3N/2 \times N$ transformation matrix, W , using transformation matrices given in (10) and (11)
- (c) Apply Transformation by multiplying the transformation matrix by the input matrix by the transpose of the transformation matrix.

This multiplication of the three matrices result in the final discrete framelet transformed matrix.

$$Y = W \cdot X \cdot W^T$$

VI. RESULTS AND CONCLUSIONS

Two methods are tried for different compression factors. The mean square error estimation of resultant images with respect to the original image data (figure 1) are presented in table 1.

TABLE. 1.

Compression factor	CCT	FFT
4	0.0019	0.0018
16	0.0050	0.0048

The framelet transform is extended to 2D by iterating the 1D over sampled filter bank on the rows and then on the columns of an image, as is usually done for separable 2D-framelet transform. At a given level in the iterated filter bank, this separable extension produces nine 2D subbands. These subbands are showed in Fig. 2. Since L is a lowpass filter ($h_0(n)$) while both H1 and H2 are highpass filters ($h_1(n)$ and $h_2(n)$), the H2H2, H2H1, H1H2, H1H1 subbands each have a frequency-domain support comparable to that of the HH subband in a DWT. A similar scheme creates the H1L, H2L (LH1, LH2) subband with the same frequency-domain support as the corresponding HL (LH) subband of the DWT, but with twice as many coefficients. Finally, note that there is only one subband LL with the same frequency-domain support as the LL subband in a DWT (Fig.3&4).

REFERENCES

- [1] W. H. Chen, C. H. Smith, and S. Fralick, "A fast computational algorithm for the discrete cosine transform," IEEE Trans. Commun. Vol. com- 25, pp. 1004-1009, Sep. 1977
- [2] P. Yip and K. R. Rao, "A fast computational algorithm for the discrete cosine transform," IEEE Trans. Commun., vol. com- 38, pp. 304-307, Feb. 1980
- [3] K.R. Rao and R. Yip, Discrete cosine transform algorithm, Advantages and applications, Newyork, academic 1990
- [4] K. Rose, A. Heiman and I. Dintein, "DCT/DST Alternate transform image coding", IEEE Trans. Com. Vol. com-38, pp. 94-101, Jan. 1990
- [5] T.P. O'Rourke and R.L. Stevenson, "Improved image decompression for reduced transform coding artifacts, IEEE

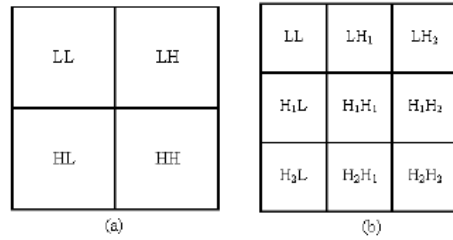


Fig.1 Image Subbands after a single-level decomposition, for (a): Scalar Wavelets and (b): Framelets

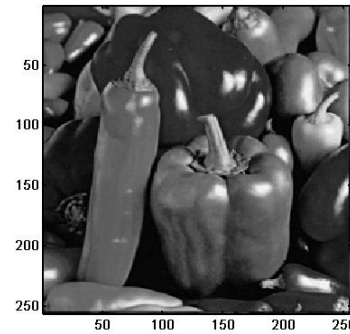


Fig2. .After Input column transform

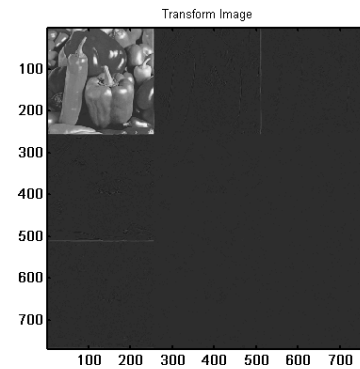


Fig.3.Upper leftmost transform, LL subband

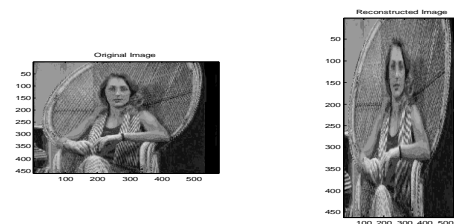


Fig 4. Original and reconstructed images MSE=65.025