Sparse Matrix based Computational Overhead Reduction in UMRT for N a power of 2

Preetha Basu,
Dept. of ECE,
TKM College of Engineering,
Kollam, Kerala, India

R. Gopikakumari
Division of Electronics, SOE
Cochin University of Science and Technology
Kochi, Kerala, India

ABSTRACT

Unique Mapped Real Transform (UMRT) is a transform which helps in frequency domain analysis of signals in the real domain. Different algorithms are developed for the computation of the unique MRT coefficients for N a power of 2 and for N an even number. They identify and place the UMRT coefficients in the form of an $N \times N$ UMRT matrix. The basis matrices of this transform are observed to be sparse in nature. In this paper a new technique is proposed to reduce the computational overhead in UMRT, the size N being a power of two, exploiting the sparse nature of the basis matrices.

General Terms

Frequency domain analysis, Sparse representation.

Keywords

UMRT, Basis matrix, Frequency domain analysis, Sparse Basis Matrix.

1. INTRODUCTION

Transform theory plays a fundamental role in signal and image processing. A transform maps data into a different mathematical space via a transformation equation [1]. Analysis of the properties of the signal is easier with the transform of that signal. Discrete Fourier Transform (DFT) is an important tool in signal processing applications to map data from time domain to frequency domain [2]. FFT is the most popular algorithm to implement DFT which is highly efficient for 1-D signals. In most of the algorithms for DFT implementation, the input data will generally be real valued which is converted to complex form and the computations are done in complex domain. 2_D Mapped Real Transform (2-D MRT), evolved from 2-D Discrete Fourier Transform (2-D DFT), maps 2-D data into frequency domain without any complex operations but in terms of real additions alone [3]. Originally, the transform mapped a $N \times N$ data matrix into M redundant matrices of size $N \times N$, M=N/2. Algorithms were developed to identify and place, the unique MRT coefficients present in the M matrices for N a power of 2 or for N an even number, in the form of an $N \times N$ UMRT matrix. In [3], all the MRT coefficients are computed, the unique coefficients are identified and arranged in an $N \times N$ matrix whereas in [4] the basic DFT coefficients are identified and the corresponding MRT coefficients are computed and placed them in an $N \times N$ UMRT matrix. These algorithms are effectively utilized for image compression applications [5] and for texture studies [6]. The algorithm in [7] computes and places the UMRT coefficients directly from the data, without computing MRT.

The Haar transform is another signal transform that converts real input to real output, and has been used extensively in signal and image processing. MRT is a recently developed transform which also uses the real-to-real conversion property of the Haar transform. Relationships between these two transforms are studied in [8]. MRT is shown to have directional properties which is utilised for orientation estimation. A subset of global patterns of a 16x16 MRT is used to estimate the orientation field of fingerprint images[9]. Discrete transforms are performed based on specific functions, called the basis functions [2]. The discrete version of 2-D basis function is called basis matrices (or basis images). The process of transforming the image data into another domain involves projecting the image onto the basis images. The mathematical term for this projection process is called an inner product. A new technique is proposed in this paper to reduce the computational overhead in UMRT exploiting the sparse nature of the basis matrices.

2. 2-D Unique Mapped Real Transform (2-D UMRT)

Let $x_{n1,n2}$, $0 \le n1, n2 \le N-1$ be the elements of $N \times N$ data matrix. The 2-D MRT coefficients $Y^{p}_{k1,k2}$, are expressed as [3]

$$Y^{p}_{k1,k2} = \sum_{\forall (n1,n2)|z=p} x_{n1,n2} - \sum_{\forall (n1,n2)|z=p+M} x_{n1,n2}$$

where $0 \le k1, k2 \le N-1$, $0 \le p \le M-1$

The MRT maps a $N \times N$ data matrix into M matrices of size $N \times N$ using real additions. The visual representation of MRT coefficients [10] show similarities that can be exploited in different ways for reducing computational requirement. The redundant coefficients present in MRT are removed to derive UMRT. The algorithm in [7] computes the UMRT coefficients directly from the data, without going through the MRT as in [3] or the basic DFT coefficients as in [4]. The placement scheme suggested in [7] places the UMRT coefficients in positions where redundancy occurs, obtained by finding out the m number of co_prime integers of N/dm, defined as $Co\_prime(m)$. Thus MRT coefficients corresponding to the frequency index (k_1, k_2) was computed selecting the data based on the condition

$z = p$ or $z = p + M$ and placed in the position $(u_1, u_2) = (((k_1.co\_prime(m)))_N, ((k_2.co\_prime(m)))_N)$

3. COMPUTATIONAL OVERHEAD IN 2-D UMRT

In most of the existing transforms like Walsh-Hadamard, Haar, DFT etc., all the input data contribute to each transform coefficient. The visual representation of MRT coefficients [8] shows that the actual computation of a UMRT coefficient involves selected set of data only. The actual number of input data participating in the computation of a particular UMRT coefficient $Y_{u1,u2}$ in terms of addition/subtraction is given by *2.N.dm* where $dm = gcd(k1, k2.M)$, divisor of M. There are N^2 UMRT coefficients and hence a total of $2N^3$.dm addition/subtraction of data elements are involved in a UMRT computation. But the position of the data, to be added or subtracted, is identified by computing the parameter $z = ((n1k1 + n2k2))_N$ and verifying whether its value is *p or p+M*. Thus z is to be calculated N^2 times to compute a particular UMRT coefficient even though only *2.N.dm* data are involved. This causes an overhead in UMRT computation. Computational overhead can be reduced by exploiting the properties present in visual pattern of UMRT coefficients.

3.1 Interpretation using basis matrix

The UMRT computation can be represented as

$$Y(u_1,u_2) = \sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1} B(n_1,n_2;u_1,u_2)X(n_1,n_2)$$

The mapping between data and UMRT is many to many. The basis matrices for mapping the data to UMRT coefficients show sparse nature. Since the UMRT computation involves computational overhead due to the parameter z, the concept of basis matrix is introduced here to reduce the overhead.

Basis matrix(UB), given below in matrix form, transforms the 2-D input data matrix to the transform domain.

$$\text{UB} = \begin{bmatrix} B_{0,0} & B_{0,1} & B_{0,2} & \cdots & B_{0,N-1} \\ B_{1,0} & B_{1,1} & B_{1,2} & \cdots & B_{1,N-1} \\ B_{2,0} & B_{2,1} & B_{2,2} & \cdots & B_{2,N-1} \\ . & . & & & . \\ . & . & & & . \\ B_{N-1,0} & B_{N-1,1} & B_{N-1,2} & & B_{N-1,N-1} \end{bmatrix}$$

1	1	1	1	1	0	-1	0	1	-1	1	-1	0	1	0	-1
1	1	1	1	1	0	- 1	0	1	-1	1	-1	0	1	0	-1
1	1	1	1	1	0	-1	0	1	-1	1	-1	0	1	0	-1
1	1	1	1	1	0	-1	0	1	-1	1	-1	0	1	0	-1
1	1	1	1	1	0	-1	0	1	-1	1	-1	0	1	0	-1
0	0	0	0	0	-1	0	1	0	0	0	0	-1	0	1	0
-1	-1	-1	-1	-1	0	1	0	-1	1	-1	1	0	-1	0	1
0	0	0	0	0	1	0	-1	0	0	0	0	1	0	-1	0
1	1	1	1	1	0	-1	0	1	-1	1	-1	0	1	0	-1
-1	-1	- 1	- 1	-1	0	1	0	-1	1	-1	1	0	-1	0	1
1	1	1	1	1	0	-1	0	1	-1	1	-1	0	1	0	-1
-1	-1	-1	-1	-1	0	1	0	-1	1	-1	1	0	-1	0	1
0	0	0	0	1	0	-1	0	0	0	0	0	0	1	0	-1
1	1	1	1	0	1	0	-1	1	-1	1	-1	1	0	-1	0
0	0	0	0	-1	0	1	0	0	0	0	0	0	-1	0	1
-1	-1	-1	-1	0	-1	0	1	-1	1	-1	1	-1	0	1	0

Figure 1: Basis matrices of 4-point 2-D UMRT

Each element $B_{u1,u2}$ in UB is a basis matrix of size NxN corresponding to the UMRT coefficient $Y_{u1,u2}$. The elements $b_{n1,n2}$ of the basis matrix $B_{u1,u2}$ is given by

$$b_{n1,n2} = \begin{cases} 1, if\ z = p \\ -1, if\ z = p + M \\ 0, else \end{cases}$$

3.2 Number of non-zero elements in the Basis matrix $B_{u1,u2}$.

The 2-D MRT coefficient $Y^{p}_{k1,k2}$ maps the N x N data matrix onto *p* twiddle factor axes in the frequency domain[2]. The number of p values depends on the frequency index (k_1,k_2) and is given by N/2dm or M/dm. The total number of elements in basis matrix $B_{u1,u2} = N^2$. The number of p values =*M/dm.*

The total number of non-zero elements in basis matrix $B_{u1,u2} = N^2 \div M/dm = 2.N.dm$

When $N = 8$ and $k1=0, k2=0, dm=gcd(0,0,4)=4$ and $M/dm=1$. Thus p has only one value, that is p=0. Since p has only one value that is zero, all the 64 inputsamples are mapped onto p=0 axis and hence elements of basis matrix $B_{0,0}$ are all 1.

When $N = 8$ and $k1=0, k2=1, dm=gcd(0,1,4)=1$ and $M/dm=4$, p has 4 values, ie p=0,1,2,3. Since p has four values, 64 input samples of $x_{n1,n2}$ are divided in 4 groups of 16 each. Correspondingly the basis matrix $B_{0,1}$ and three other associated basis matrices will have sixteen non-zero elements each.

4. Modified algorithm for UMRT computation exploiting sparse basis matrix.

The algorithm is developed based on the observation of the patterns present in the visual representation of MRT coefficients and basis matrices of UMRT coefficients. Initially, the row and column indices $(n1, n2)$ of non-zero basis elements are identified for computing a particular UMRT coefficient. Data elements $x_{n1,n2}$ in those positions are added together without computing z. The sign of $x_{n1,n2}$ can be found from $-1^{((n1k1+n2k2)-p))_N}$. Although the sparse nature of basis matrix is exploited, there is no need to create a basis matrix in the present implementation. The transform coefficients are categorized into three as $k1 = 0, k2 = 0$ and others.

When $k1 = 0$,all the rows of the basis matrix has elements. Since there are $2N.dm$ non-zero elements in a basis matrix, each row has $2N.dm / N = 2dm$ elements. The row index is incremented by $2dm$ from 0 to N. Thus $2N.dm$ row indices are formed. Column index occurs in an increment of $N / 2dm$ in the interval 0 to N and is repeated N times for N rows. Thus $N \div N / 2dm$ is repeated N times which is equal to $2N.dm$ column indices. It is seen from inspection that when p increases column index is incremented once. For different p's, each column index is incremented by p/dm times. Row index is retained as such.

When $k2 = 0$, all the columns of the basis matrix has elements. The column index is incremented by $2dm$ from 0 to N. Thus $2N.dm$ column index is formed. Row index is incremented from 0 to N in steps of $N / 2dm$ and is repeated N times. That is $N \div N / 2dm$ is repeated N times which is equal to $2N.dm$ column indices. For different p's, each row index is incremented by p / dm times. Column index is retained as such.

When $k1 \,\&\, k2$ not equal to zero, the row index is incremented from 0 to N , $2dm$ times. Thus $2N.dm$ row indices are formed. Column index is incremented from 0 to N in steps of $N / 2dmk2$, where dmk_2 is $\gcd(M, k2)$ and is repeated $N.dm / dmk2$ times. Thus $N.dm / dmk2 \div N / 2dmk2$ is repeated N times which is equal to $2N.dm$ column indices. From the visual representation of basis matrices it is seen that, with each increment in p, the row and column indices are incremented in a special pattern. For different p's, each row index is incremented by $p.r / dm$ times, each column index is decremented by p.c/dm times. Constant r and c depends on the coefficients k_1,k_2 and satisfies the equations, $((k1.r))_{k2} = dm$ and $k1.r = dm + k2.c$.

Various steps involved in the algorithm are depicted using a Flow Chart in figure 3. The sub processes of finding out the row and column indices of sparse basis matrices for various *(k1,k2)* (Block A and block B) are shown separately in figure 4.

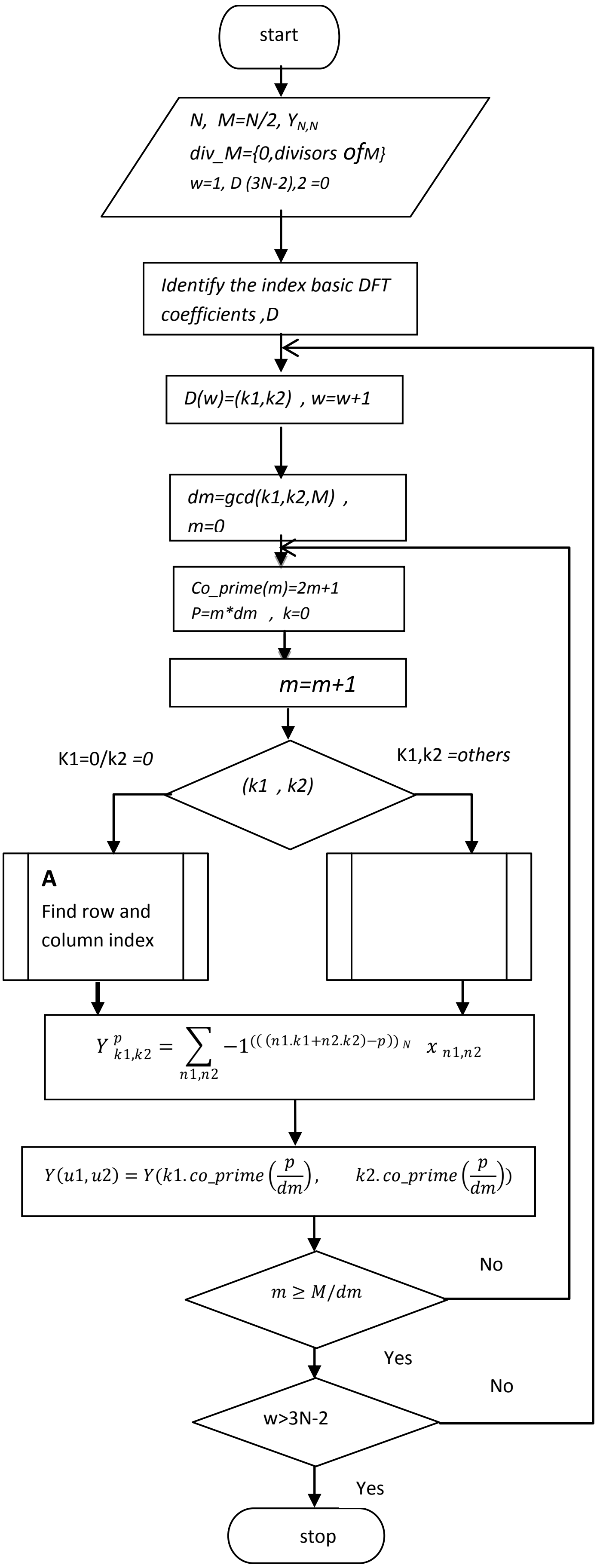


Figure 2: Flowchart for finding UMRT in sparse basis matrix method

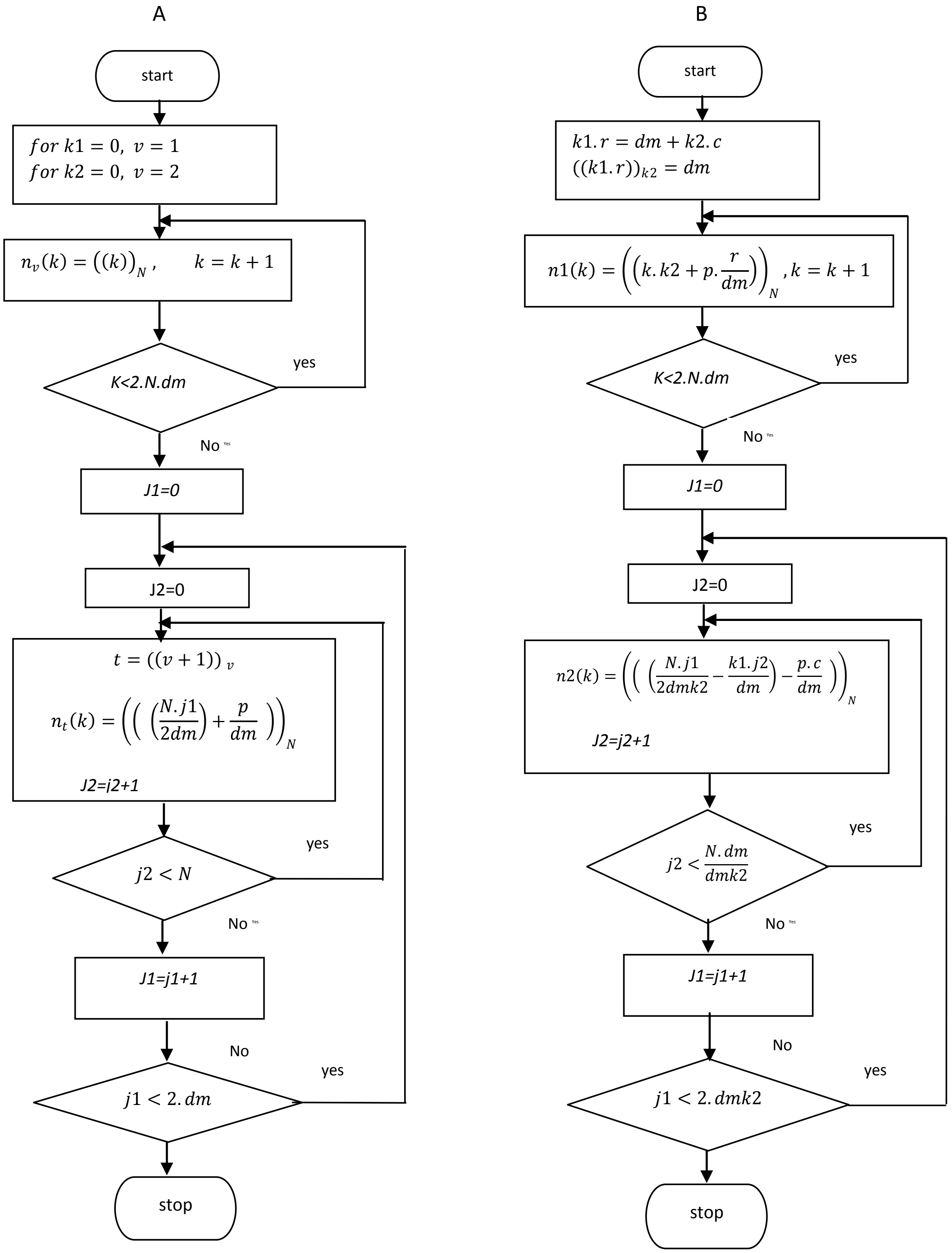


Figure 3: Flowchart of the sub processes A and B

The Algorithm

1. Initialize $N,\ M = N/2,\ Y_{N,N} = 0$ find divisors *of M,* div_m ={0 and divisors of M}.

2. Identify frequency indices $(k1, k2)$ of basic DFT coefficients.

3. Find p's for each $(k1, k2)$

4. For each $(k1, k2, p)$, find row and column indices (n1 and n2) of $x_{n1,n2}$ added to find $Y_{u1,u2}$

5. The transform coefficients

$$Y^{\,p}_{\,k1,k2} = \sum_{n1,n2} -1^{(((n1.k1+n2.k2)-p))_N} \;\; x_{\,n1,n2}$$

6. UMRT coefficients

$$Y(u1,u2) = Y\left(k1.co_{prime\left(\frac{p}{dm}\right)},\ k2.co_{prime\left(\frac{p}{dm}\right)}\right)$$

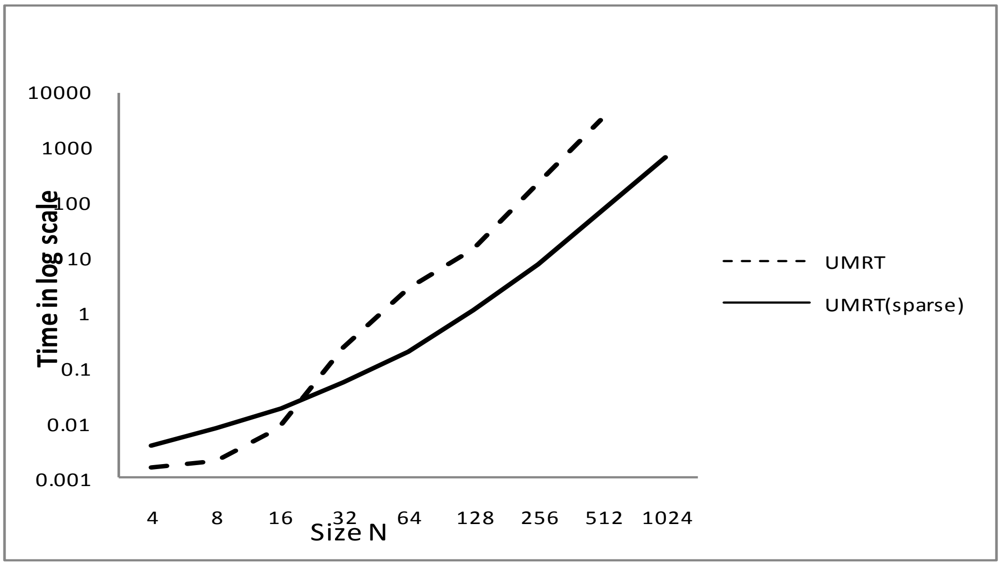
TABLE I:TIME TAKEN FOR UMRT COMPUTATION

Size N	Computation time (sec)	
	UMRT	UMRT(sparse)
4	0.0014	0.0035
8	0.0019	0.0070
16	0.0073	0.0163
32	0.2112	0.0480
64	2.5972	0.1745
128	14.32	1.02
256	193.21	7.44
512	3123.11	62.22
1024	_	642.70

5. RESULTS AND ANALYSIS

The proposed algorithm for the UMRT computation is implemented on Intel core i5 machine with clock speed 2.4 GHz and 4GB RAM. Lena image is used to compare the performance of the present algorithm with the previous algorithm[7]. The table I shows the results of comparison performed. A sharp increase of computational time saving is attained for different values N. Fig 4 gives a graphical representation of the comparison made, time being in logarithmic scale.

Figure 4: Graph showing comparison of time taken in log scale as a function of N.



6. CONCLUSION

The result shows a considerable amount of saving in time especially when the size of the data matrix increases. The proposed algorithm is faster compared to the earlier algorithms. Although the computation time is comparable or slightly higher for small values of N, as N increases the sparse matrix method of implementation performs exceedingly faster.

Thus if the UMRT computation exploiting the sparsity of basis matrices proposed in this paper is used in frequency domain analysis of 2-D signals, especially for applications like video processing and image enhancement where size of the image taken is large, the computation time will be drastically reduced. An alternate placement approach for arranging the unique MRT coefficients in the order of sequencies is proposed in [11] named as SMRT. Since the basis matrices are the same, the proposed algorithm can be used to reduce the computational complexity of SMRT computation also.

7. REFERENCES

[1] A K Jain, "Fundamentals of digital image processing", Prentice Hall New Delhi, India, 2003.

[2] Alexander D. Poularikas, "Transforms and Applications Handbook", Third Edition,CRC press, 2010.

[3] Rajesh Cherian Roy and R. Gopikakumari "A new transform for 2-D signal representation (MRT) and some of its properties", 2004 IEEE International Conference on Signal Processing and Communications.

[4] Bhadran. V, R.C.Roy, R.Gopikakumari, "Algorithm to identify basic coefficients of 2-D DFT for any even N". in proc. ICVCOM, Saintgits college of engineering,Kottayam,Apr 16-18,2009.

[5] M. S. Anish Kumar, Rajesh Cherian Roy and R. Gopikakumari, "A new image compression and decompression technique based on 8x8 MRT", International Journal on Graphics, Vision and Image Processing, Volume 6, July 2006, pp. 51-53 .

[6] Meenakshy K., "Development & Implementation of a CAD System to predict the Fragmentation of Renal Stones Based on Texture Analysis of CT Images", Ph. D Dissertation, Cochin University of Science & Technology, Kochi, 2010.

[7] Preetha Basu, Bhadran.v., Gopikakumari.R, "A new algorithm to compute forward and inverse 2-D UMRT for N — A power of 2",IEEE International Conference on Power, Signals, Controls and Computation (EPSCICON), 2012 .

[8] Rajesh Cherian Roy and R. Gopikakumari "Relationship between the Haar transform and the MRT", 8th International Conference on Information, Communication and Signal processing(ICICS), 2011.

[9] Rajesh Cherian Roy and R. Gopikakumari, "A Transform(MRT) naturally suited for Directional Pattern Analysis", Proc. SPIE8760, January 28, 2013.

[10] Bhadran. V, R. C. Roy, R. Gopikakumari, "Visual Representation of 2-D DFT in terms of 2x2 data, A pattern analysis", proceedings of international conference on computing, communication and Networking (ICCCN '08), Chettinad college of engineering and technology, karur, India, dec 18-20, 2008 .

[11] Jaya. V. L, Preetha Basu, Gopikakumari.R, "A new Placement approach of 2-D unique MRT Coefficients for N a power of 2", Annual IEEE India Conference (INDICON), 2012.