- ***Complexity assessment in terms of execution time and energy consumption***

The results, in terms of computation cycles per 8 × 8 blocks for various 2-D transform are given in Table 3.4. From this Table, it is clear that the proposed transform requires less number of cycles compared to the other techniques. Table 3.5 shows the processing time and energy consumption obtained using different transforms. As expected, because of the substantial reduction in the number of arithmetic operations, the proposed transform is more efficient in speed and energy. On average, it managed to save 14%, 41% and 75% on both the processing time and energy consumption, compared with the methods presented in [68], [67] and [66] respectively. This would considerably increase the lifetime of the wireless sensor node.

So, results indicate that the proposed transform not only reduces the execution time and energy consumption, but also provides superior results in terms of PSNR performance compared with recent works.

Transform	Cycles
IJG float [65]	580106
Lecuire et al [66]	2734
Phamila et al [67]	1840
Kouadria et al [68]	1113
Proposed [75]	1036

Table.3.4: *Computation cycles per 8 × 8 block obtained by different 2-D transforms on the ATmega128L platform*

Transform	Execution time (ms)	Energy consumption (µJ)
IJG float [65]	72.51	1595.22
Lecuire et al [66]	0.37	8.53
Phamila et al [67]	0.23	5.06
Kouadria et al [68]	0.15	3.47
Proposed [75]	0.13	2.99

Table.3.5: *Execution time and energy consumption per 8 × 8 block obtained by different transforms on the ATmega128L platform.*

The proposed DCT approximation has a low complexity which combines the rounded DCT with a pruned approach. The proposed method requires 16 and 192 addition operations for 1D and 2D respectively. Experimental comparisons with recent works, using Atmel Atmega128L platform, show that the proposed scheme reduces the energy consumption, processing time and provide a better performance in terms of PSNR metric. This makes it quite suitable to be used in WISNs with a prospect to increase the lifetime of the sensor nodes. We believe that a hardware implementation in ASIC/FPGA will provide an even better performance, and will be considered in the future work.

Second proposed 8 by 8 DCT approximation

3.6.2.1 Mathematical concept

The methods presented in [83] and [85] are simple and prominent. They consist in applying an integer function to the standard DCT matrix in order to obtain its entries in the set $\{-1,0,+1\}$. Our proposed transform is obtained in a similar way. First, the standard 8×8 DCT matrix ($DCT_{8\times8}$) is scaled by 3, then its elements are fed into a *round* function. This results in the matrix $T_{8\times8} = round\{3 * DCT_{8\times8}\}$ that is given as follows:

$$T_{8\times8} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix} \qquad (3.28)$$

Where the function $T_{8\times8} = round\{\bullet\}$ rounds the elements of the matrix to the nearest integer, and is defined as follows:

$$round\{x\} = sign(x).floor(|x| + \tfrac{1}{2}) \qquad (3.29)$$

The matrix $T_{8\times8}$ is not orthogonal. It can be orthogonalized according to [109] using the diagonal adjustment matrix $D_{8\times8} = \sqrt{(T_{8\times8}.T^{t}_{8\times8})^{-1}}$ This procedure of orthogonalization leads to the following proposed DCT approximation matrix $C_{8\times8} = D_{8\times8} \cdot T_{8\times8}$ It can be verified that this transformation matrix is orthogonal (i.e. $C^{-1}_{8\times8} = C^{t}_{8\times8}$), where t denotes the matrix transpose operation.

A fast algorithm for the matrix $T_{8\times8}$ was derived and is depicted in Figure 3.9. According to this proposed algorithm the computation of the proposed DCT approximation requires 24 and 384 additions for the 1-D and the 2-D versions respectively.

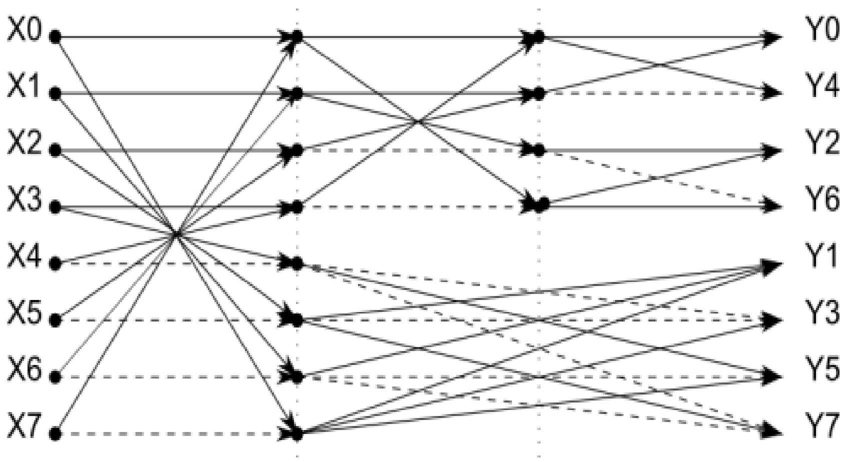


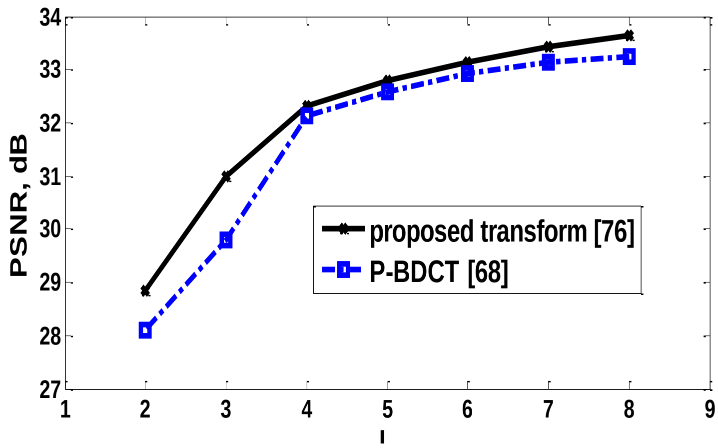
Figure 3.*9* Signal flow graph for the proposed matrix $T_{8\times8}$. Xi (i = 0,1,…7) represents input data relates to output data Yj (j = 0, 1…7) according to $Y_{8\times1} = T_{8\times8} \cdot X_{8\times1}$. Dashed arrows are multiplications by -1

To further reduce computational complexity of the proposed DCT approximation, we used the 2-D pruning approach described before (section 3.4.2).

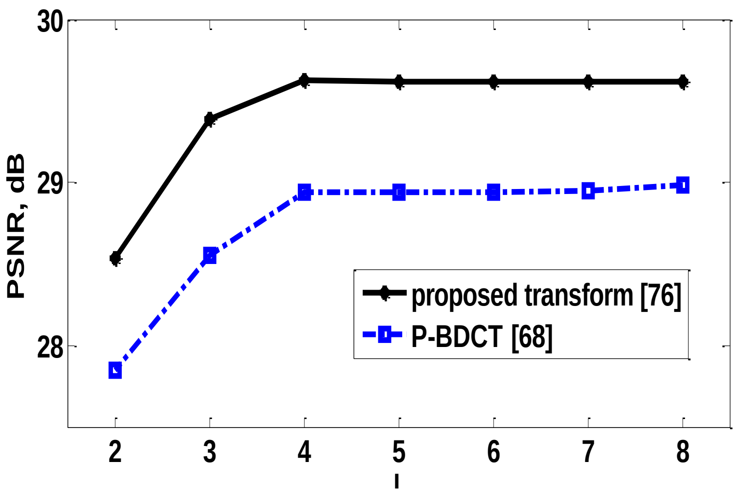
3.6.2.2 Performance Evaluation

3.6.2.2.1 Results in terms of image quality

To evaluate the performance of the proposed transform in image compression, we used it as part of the JPEG chain. This JPEG chain includes standard quantization table and Huffman encoder that are given in the JPEG standard [79]. For this purpose, we considered a set of 512×512 8-bit standard grayscale images taken from [80].



(a)



(b)

Figure. 3.10 *PSNR against L parameter for P-BDCT [68] and proposed transform[76]. Results are given for the image Lena. Fig.2a and Fig.2b correspond to bit rate of 0.5 bpp and 0.2 bpp respectively*

Table 3.6 provides the resulting PSNR for the proposed transform and P-BDCT [68] for the same complexity (i.e. L = 8) and bitrate = 0.5 bpp. It is clear from this table that the proposed transform leads to superior performances in terms of image quality. An average of 0.3 dB in PSNR gain was obtained.

Method	Barbara	Lena	Baboon	Boat	Peppers
Proposed [76]	27,01	33,64	23,46	29,62	32,18
P-BDCT [68]	26,71	33,25	23,24	29,35	31,82

Table.3.6: *PSNR obtained by the proposed transform [76] and P-BDCT [68]. Results are given for the same complexity (L= 8) and bit rate = 0.5 bpp.*

Figure 3.10a and Figure 3.10b show the PSNR against L using the two transforms at a bit rate of 0.5 and 0.2 bpp respectively. It can be easily seen that the proposed transform outperforms P-BDCT [68] in terms of PSNR whatever the values of L and bitrate are. The average PSNR gain is about 0.56 dB. Note that, for a large set of test images, the method maintains the same performance and doesn't depend on the image type.

3.6.2.2.2 Results in terms of number of operations, execution time and energy consumption

Complexity assessment in terms of number of operations

Table.3.7 summarises the number of additions required to compute an 8×8 block using the proposed 2-D transform compared to P-BDCT [68]. It can be seen from Table 3.6 that the proposed 8x8 DCT approximation is of less complexity than the transform P-BDCT [68]. It requires about 9 % less arithmetic operations. Therefore, the proposed transform is expected to perform better in terms of energy consumption at the WISN nodes.

L	P-BDCT [68]	Proposed[76]
L = 8	384	384
L = 7	345	330
L = 6	308	294
L = 5	273	247
L = 4	240	216
L = 3	187	165
L = 2	140	120

Table.3.7: *Computation complexity analysis. results are given for 2-D*

- ***Complexity assessment in terms of execution time and energy consumption***

The main concern in resource constrained WISNs is to reduce energy consumption as much as possible. To assess the energy efficiency of the proposed 8×8 DCT approximation in sensor nodes, we perform an experiment similar to that in [68] and [66]. For this purpose, we adopt the well-known Atmel Atmega128L platform. Results, in terms of computation cycles, execution time and energy consumption per 8×8 blocks for the proposed transform and for P-BDCT [68] are given in Table 3.8. From this table, it is clear that the proposed transform requires less cycles than the transform P-BDCT [68]. As expected, because of the reduction in the number of arithmetic operations, the proposed transform is more efficient in energy. On average, it managed to save about 10% on both the execution time and energy consumption as compared to the work in [68]. This will certainly result in a tangible increase in the lifetime of the WISN nodes.

	P-BDCT [68]			Proposed [76]		
	Cycles	**Time (µs)**	**Energy (µj)**	**Cycles**	**Time (µs)**	**Energy (µj)**
L =8	1850	250	5,77	1843	249	5,73
L = 7	1666	225	5,18	1584	214	4,92
L = 6	1450	196	4,52	1381	187	4,3
L = 5	1274	172	3,97	1136	154	3,54
L =4	1113	150	3,47	993	134	3,08
L = 3	890	120	2,78	775	105	2,41
L = 2	700	94	2,18	552	74	1,7

Table.3.8: *Computation cycles, execution time and energy consumption obtained by the proposed transform [76] and P-BDCT [68] on the Atmega128 L platform*

The proposed transform not only reduces the execution time and energy consumption, but also provides superior results in terms of PSNR. The obtained experimental results show that the proposed scheme reduces the energy consumption, when compared to P-BDCT, while at the same time provides a better performance in terms of PSNR metric.

3.6.3 Proposed Pruned DTT approximation

3.6.3.1 Mathematical concept

The discrete Tchebichef transform (DTT) is a novel polynomial-based orthogonal transform. It exhibits interesting properties, such as high energy compaction, optimal decorrelation and direct orthogonality, and hence is expected to produce good transform coding results. Advances in the areas of image and video coding have generated a growing interest in discrete transforms. The demand for high quality with a limited use of computational resources and improved cost benefits has led to experimentation with novel transform coding methods. In this part, we will use an approximation of the Discrete Tchebichef transform in JPEG baseline for image compression.

The discrete Tchebichef transform (DTT) is an orthogonal transformation drifted from the discrete Tchebichef polynomials (DTP) [72].

1D Forward and inverse transforms for a block of 8×8 are given by: $Y_{8\times8} = C_{8\times8} \cdot X_{8\times8}$ and $X_{8\times8} = C_{8\times8}^{-1}.X_{8x8} = C_{8X8}^{t}.X_{8x8}$

Where $X_{8\times8}$ and $Y_{8\times8}$ are the input and transformed signals, respectively. $C_{8\times8}$ can be described by the following equation: $C_{8\times8} = D_{8\times8} \cdot T_{8\times8}$, where $D_{8\times8}$ is a diagonal matrix and $T_{8\times8}$ is an integer matrix. From [74] $T_{8\times8}$ is given by:

$$T_{8\times8} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -7 & -5 & -3 & -1 & 1 & 3 & 5 & 7 \\ 7 & 1 & -3 & -5 & -5 & -3 & 1 & 7 \\ -7 & 5 & 7 & 3 & -3 & -7 & -5 & 7 \\ 7 & -13 & -3 & 9 & 9 & -3 & -13 & 7 \\ -7 & 23 & -17 & -15 & 15 & 17 & -23 & 7 \\ 1 & -5 & 9 & -5 & -5 & 9 & -5 & 1 \\ -1 & 7 & -21 & 35 & -35 & 21 & -7 & 1 \end{bmatrix} \quad (3.30)$$

The diagonal matrix $D_{8\times8}$ is given by equation (3.21):

$$D_{8\times8} = \frac{1}{2}.diag\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{42}}, \frac{1}{\sqrt{42}}, \frac{1}{\sqrt{66}}, \frac{1}{\sqrt{154}}, \frac{1}{\sqrt{546}}, \frac{1}{\sqrt{66}}, \frac{1}{\sqrt{858}}\right) \quad (3.31)$$

The matrix $D_{8\times8}$ can be merged into the quantization step as suggested in many works in literature [73], [74], [110] and [85], and thus it does not introduce any computational overhead. Hence the matrix $T_{8\times8}$ is the only source of computational complexity. As it is described in [74], the fast algorithm for computing the matrix $T_{8\times8}$ requires 44 additions and 29 shift operations.

- *Pruned Discrete Tchebichef Transform (P-DTT)*

The proposed method is kind of approximation that is based on pruned approach and the integer DTT presented in [74]. Our main focus is to reduce the computational complexity of the exact DTT. For that raison, we use the 2-D pruning approach introduced previously.

Since most of the block energy is compacted in the first 4×4 low frequency coefficients, in this work we adopted L=4. By doing so, the forward and inverse proposed 2D P-DTT will be as it is mentioned in formulas (3.25) and (3.26) respectively, and the diagonal matrix is given by:

$$D_{4\times4} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{42}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{42}} & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{66}} \end{bmatrix} \tag{3.32}$$

- *Fast algorithm of the proposed P-DTT:*

Based upon [74] and the pruning approach presented in subsection A, we have deducted a fast algorithm of the proposed P-DTT. Table 3.9 presents the different steps for the calculation of this algorithm. Where X=[x_0, x_1, …, x_7] is an 8-point input vector and Y=[y_0, y_1, y_2, y_3] is a 4-point output vector

Step 1	$u_0 = x_0 + x_7$ $\quad v_0 = x_0 - x_7$ $u_1 = x_1 + x_6$ $\quad v_1 = x_1 - x_6$ $u_2 = x_2 + x_5$ $\quad v_2 = x_2 - x_5$ $u_3 = x_3 + x_4$ $\quad v_3 = x_3 - x_4$
Step 2	$z_0 = v_0 + v_3$ $k_0 = u_0 + u_2$ $\quad z_1 = v_1 - v_2$ $k_1 = u_1 + u_3$ $\quad z_2 = v_1 + v_2$
Step3	$m_0 = k_0 + k_1$ $m_2 = 2(3u_0 - 2u_2)$ $\quad w_0 = -(z_1 + z_0)$
Step4	$l_0 = m_0 + m_2$ $l_4 = w_0 - 6v_0$ $\quad l_5 = 2(2v_3 + 3z_2)$
Step5	$y_0 = m_0$ $\quad y_2 = l_0 - 6u_3$ $y_1 = l_4 - 4z_2$ $\quad y_3 = l_4 + l_5$

Table.3.9: *Fast algorithm for the proposed 1D P-DTT*

Since all the involved multiplication coefficients are integer, it can be implemented by a series of shift and addition operations. These operations are presented in Table 3.10, where the shift is designed by <<.

$$m \times 2 =<< m$$
$$m \times 3 =<< m + m$$
$$m \times 4 =<<<< m$$
$$m \times 6 =<< (<< m + m)$$

Table.3.10: *Multiplication through add and shift operations*

3.6.3.2 Performance Evaluation

3.6.3.2.1 Results in terms of image quality

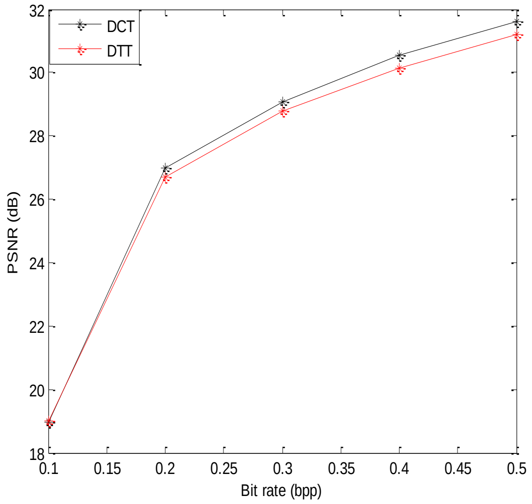
This section provides a comparative analysis of the performance of our proposed method with the integer DTT in [74] and the popular Loeffler DCT algorithm presented in [50]. For this purpose we used a set of standard 512×512 8-bit gray-level images taken from [80], see Figure 3.1. We performed a JPEG like simulation that includes standard quantization table and Huffman encoder that are given in the JPEG standard [79].

As the PSNR and MSE have been proved to be less inconsistent with human eye perception, we also adopted the structural similarity index (SSIM) [80], [109]. This latter is best known for his consistent with the human visual system (HVS)

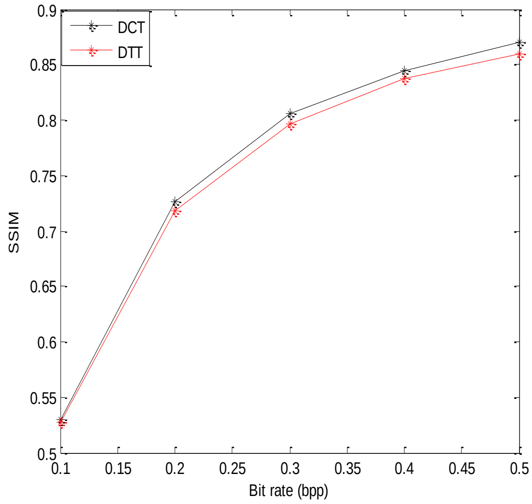
- *Loeffler DCT and integer DTT comparison:*

Figure.3.11 presents the PSNR performance of the Loeffler DCT algorithm [50] and the integer DTT [74]. In this graph; we consider the average PSNR of all tested images cited in section (3.3.1) as figure of merit. According to [85], average calculation may provide more robust results than results obtained from each particular image.

Comparison of the curves clearly shows that the DTT has almost the same performances in PSNR as the DCT in low bitrates, and it is slightly lower in high level of bitrates.



(a)



(b)

Figure.3.11: *(a) Average PSNR, (b) average SSIM , for several compression ratios for: the exact DCT and the exact DTT*

- *Loeffler DCT, integer DTT and proposed P-DTT for different values of L comparison:*

PSNR and SSIM values computed from the reconstructions of tested images using: DCT, exact DTT and proposed method for different values of L are plotted in Figure. 3.12 and Figure. 3.13 respectively.

From PSNR results, it can be easily seen that the proposed method has the same performance in low bitrates that the DCT and the DTT for various input images. The proposed method tends to be slightly inferior in high bitrates that corresponds to low image compression ratios.

It is evident from the SSIM results shown in Figure 3.13 that the difference between the proposed P-DTT and the mentioned transforms get closer for the whole range of selected bitrates.

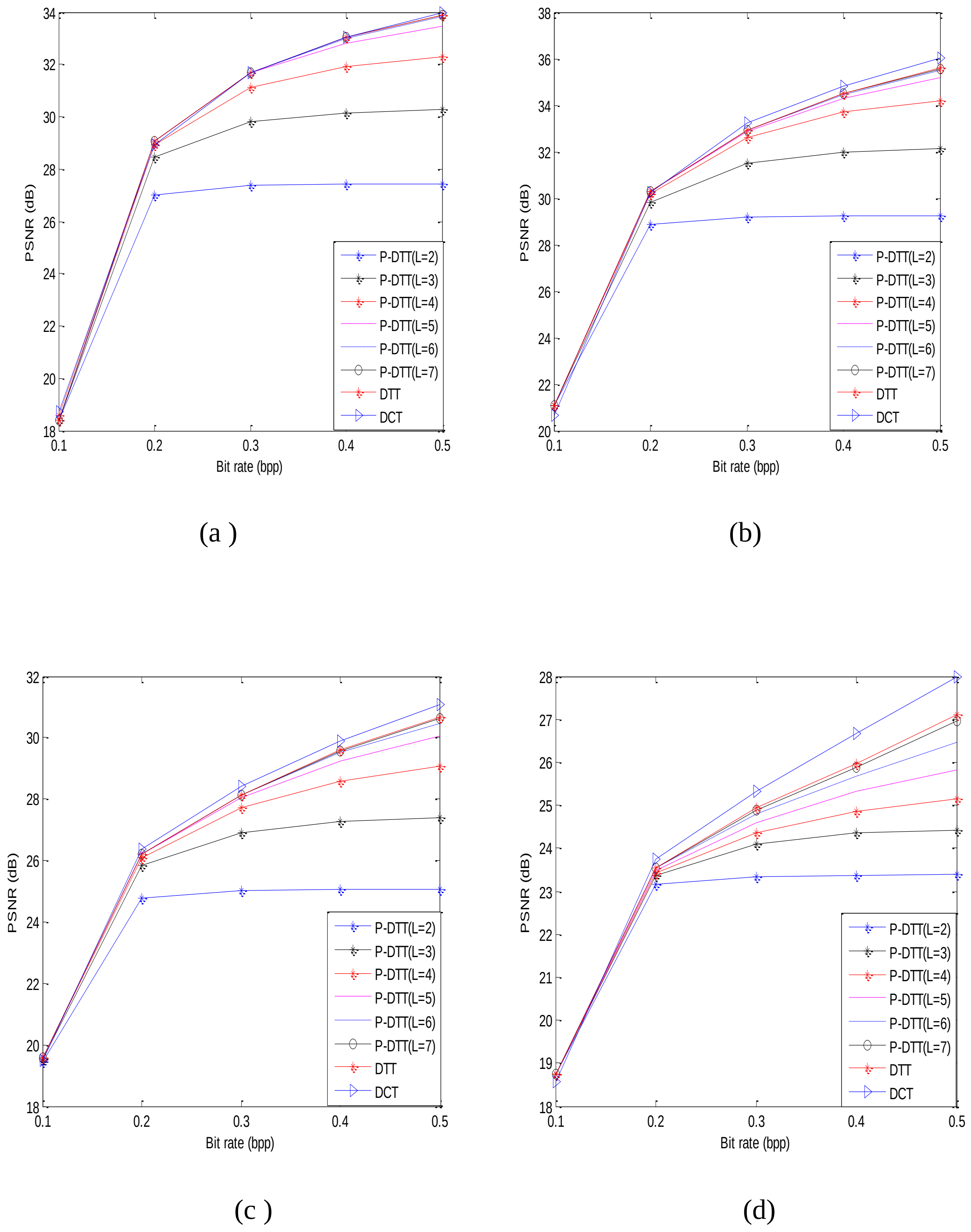
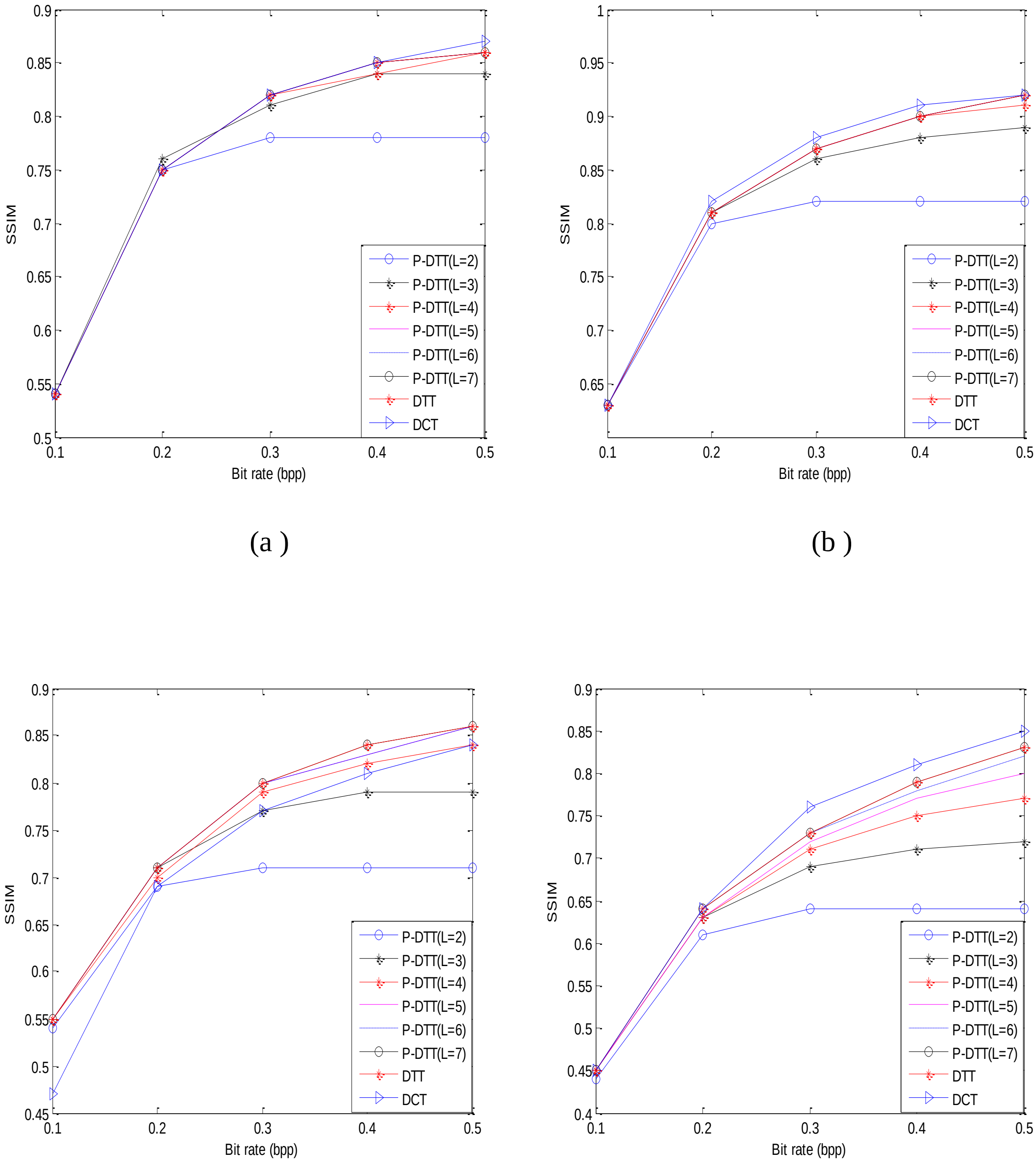


Figure. 3.12 *Rate-Distortion results of images: (a) peppers, (b) Lena,(c) boat, (d) barbara for different bit-rates*



(a) (b)

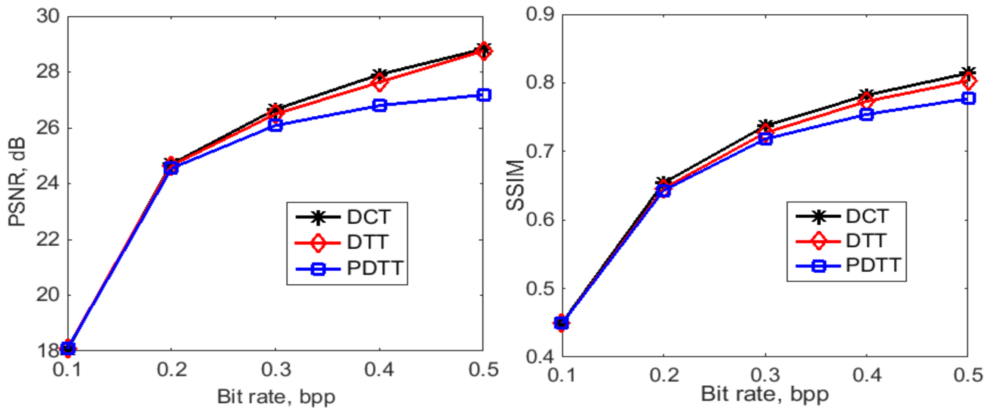
(c) (d)

Figure. 3.13. *Structural Similarity Index (SSIM) results against bitrate For : (a) peppers, (b) Lena, ,(c) boat, (d) barbara for different bit-rates*

- *Loeffler DCT, integer DTT and proposed P-DTT for L=4 (adopted) comparison:*

Multimedia contents require extensive bandwidth and energy for transmission. Due to the limited available bandwidth and sensor power an image captured by a WSN node typically needs to be highly compressed before transmission. The appropriate bpp range for

WMSNs depends on the application in which the WMSN is involved. But, generally, WSNs require high compression ratio (small bpp < 0.5 bpp). Consider that issue; we measure the compression efficiency, at a range of [0-0.5] bpp, using the PSNR and the structural similarity index (SSIM) as figure of merits. Figure.3.14 presents the rate-distortion (R-D) curves of aforementioned transforms. The average PSNR and SSIM values (averaged over all test images) of the reconstructed images are given in Figure.3.14 (a) and (b), respectively. Average calculation may provide more robust results than results obtained from each particular image.



a *b*

Figure.3.14. *rate-distortion (R-D) curves of the DCT [50],DTT [74] and PDTT[77]*
(a) Average PSNR, *(b)* Average SSIM

From PSNR results in Figure.3.14 (a), it can be seen that up to 0.2 bpp the P-DTT (L=4) has the same performance than the DCT and the DTT. However, at bitrates higher than 0.2 bpp, the P-DTT performance tends to be slightly lower. Note that for perfect reconstruction of the image, the PSNR value is equal to infinity. However, in a context of lossy compression, a reconstructed image with a PSNR of 40 dB or higher, presents negligible and invisible degradations to the naked eye. If it is in the range of 25-30 dB then the image quality is typically acceptable. And this is acceptable especially in WMSNs when high quality imagery is not a relevant requirement. From SSIM results shown in Figure.3.14 (b) we observe that the difference between the three transforms (P-DTT (L=4), DTT and DCT) get closer for the whole range of selected bitrates.

- *Visual Comparison:*

PSNR=28.52dB, SSIM=0.77 (DCT [50])   PSNR=28.14dB, SSIM=0.76 (DTT[74])   PSNR=27.79dB, SSIM=0.76 (P-DTT[77])

PSNR=33.10dB, SSIM=0.88 (DCT [50])   PSNR=32.82dB, SSIM=0.87 (DTT[74])   PSNR=32.63dB, SSIM=0.87 (P-DTT[77])

PSNR=25.31dB, SSIM=0.75 (DCT [50])   PSNR=24.82dB, SSIM=0.73 (DTT[74])   PSNR=24.36dB, SSIM=0.71 (P-DTT[77])

PSNR=31.67dB, SSIM=0.82 (DCT [50])   PSNR=31.70dB, SSIM=0.82 (DTT[74])   PSNR=31.04dB, SSIM=0.82 (P-DTT[77])

Figure.3.15. Reconstructed *images using Loeffler DCT [50], exact DTT [74] and proposed P-DTT[77]. Results are given for bitrate = 0.3 bpp*

A motivating aspect of the proposed P-DTT in image compression is that the visual quality of reconstructed images is competitive compared to the other methods. To illustrate this feature, Figure 3.15 presents a visual comparison of some test images compressed at 0.3 bpp using P-DTT when L is equal to 4, Loeffler DCT [50] and integer DTT algorithms [74].

3.6.3.2.2 Results in terms of number of operations and energy consumption

- ***Results in terms of number of operations:***

Table 3.11 summarizes the number of operations required to compute the aforementioned transform for 1D and 2D.

	Calculation complexity 1D			**Calculation complexity 2D**		
Transforms	**Add**	**Mult**	**Shift**	**Add**	**Mult**	**Shift**
DCT	28	11	0	448	176	0
DTT	44	0	29	704	0	464
P-DTT L=7	38	0	24	608	0	384
P-DTT L=6	35	0	21	560	0	336
P-DTT L=5	30	0	17	480	0	272
P-DTT L=4	25	0	14	400	0	224
P-DTT L=3	22	0	11	352	0	176
P-DTT L=2	17	0	5	272	0	80

Table.3.11: *Arithmetic complexity in terms of number of operations for 1D and 2D*

Note that the Loeffler algorithm [50] reach the theoretical limit of the complexity of the exact 8-point DCT and DTT [74] is the only fast algorithm archived in literature for computing the exact 8-point DTT.

From this table, it is obviously shown that the proposed transform requires lesser complexity than the other transforms. When L is equal to 4 which we have adopted in our proposed method in [77], it requires 41 % and 52 % less additions and shift operations, respectively compared with the integer DTT [74]. Furthermore, it does not involve any multiplications, which are computationally intensive, as this is needed for Loeffler algorithm [50]. Therefore, the proposed transform yields to a considerable energy saving, thing that is suitable for embedded systems and resource constrained WSNs.

- ***Results in terms of energy consumption***

Table 3.12 represents a comparison between the energy consumption of the DTT and P-DTT when L=4, we adopted the parameters which refer to the characteristics of Mica2 platform. Mica2 platform is a well-known sensor node which contains a low-power microcontroller namely Atmel Atmeg128. As expected from the complexity analysis, we notice from the table 3.12 that the P-DTT gives better results in terms of energy efficiency. It saves about 58% of energy compared to DTT. These savings are for one still image and when the application requires videos, which demand several images per second, the energy reduction will be considerable. Hence, the lifetime of the battery-powered sensor node will be highly prolonged.

	Energy for 8x8 blocs $E_{8\times8}$(L)(**µJ**)	Energy for 512x512 Image $\mathrm{E}_{512\times512}$(L)(**mJ**)
DTT	3,85	15,77
P-DTT L=4	1,58	6,47

Table.3.12: *Comparison between the energy consumption of DTT and P-DTT L=4*

The P-DTT computes only the upper 16 coefficients, whereas the DCT and the DTT compute the entire 64 coefficients, which require more data to be stored and processed in the working memory (RAM). Therefore, it is evident that the P-DTT requires much less memory than the other transforms. Moreover, we should note that this reduction propagates across the quantization and the encoding steps.

3.7 Conclusion:

In this chapter, we have proposed two low complexity DCTs approximations which have a good tradeof quality /complexity. The first proposed DCT approximation combines the rounded DCT with a pruned approach. The proposed method requires 16 and 192 addition operations for 1D and 2D respectively. Experimental comparisons with recent works, using Atmel Atmega128L platform, show that the proposed scheme reduces the energy consumption, processing time and provide a better performance in terms of PSNR metric. The second proposed DCT approximation is scaled and rounded version of the exact DCT matrix. The obtained experimental results show that the proposed scheme reduces the energy consumption, when compared to P-BDCT, while at the same time provides a better performance in terms of PSNR metric. These results make them quite suitable to be used in WISNs with a prospect to increase the lifetime of sensor nodes.

In the other hand, we have proposed a pruned version of DTT (P-DTT) transform which requires 41 % and 52 % less additions and shift operations respectively, when compared with Loeffler DCT and the exact DTT. Moreover, the P-DTT shows competitive performance in image compression in terms of PSNR and SSIM metrics. These features make it able to be an alternative transform of DCT and it could be used in resource-limited wireless image sensor network and other digital signal processing applications that need the same constraints. These results make us believe that a hardware implementation in ASIC/FPGA will provide an even better performance, and will be considered in the future work.