Chapter-III

WATERMARKING USING DECIMAL SEQUENCES
3.1. Introduction

In this chapter robust blind watermarking algorithm using decimal sequences is presented. Like other pseudorandom noise sequences (PN Sequence), decimal sequences (d-sequence) may be used in designing a Code Division Multiple Access (CDMA) system. The properties of d-sequence allow it to use in error correcting codes and in many applications in communications. The advantage of using d-sequence over PN sequence is that one can select the required sequence, because the d-sequence is generated from a prime number which provides a greater flexibility in the choice of sequence. This shows the flexibility of the system. The operation of embedding and extraction using correlation of the watermark is done in the frequency domain. The frequency transforms used are Discrete Cosine Transform (DCT) and Integer Discrete Cosine Transform (Int DCT). The proposed watermarking algorithm is more robust, since it can withstand many attacks including compression and filtering. Experimental results are evaluated with Correlation (CORR), Peak Signal to Noise Ratio (PSNR) and Weighted Peak Signal to Noise Ratio (WPSNR). This algorithm is shown to be more robust than the other watermarking algorithms using d-sequences.

In this chapter, the technique of watermarking using only decimal sequences is introduced. Decimal sequences have better autocorrelation properties for some specific shifts as compared to the pseudo noise sequences [66], so the results of watermark recovery is expected to be better than the method using PN sequences.

This chapter is organized as follows: Section 2 of a general description of decimal sequences. Section 3 gives brief introduction about the transforms DCT and IntDCT that are used in this research work to develop the watermarking algorithm using d-sequences. Section 4 explains the developed Watermarking Scheme. Section 5 presents the experimental results for the proposed work. Finally, section 6 gives information about conclusions of the algorithm.

3.2 Introduction to Decimal Sequences

Decimal sequences are obtained when a number is represented in a decimal form in a base \( r \) and they may terminate, repeat or be aperiodic. As these sequences
are periodic their randomness needs to be checked only in one period. For a certain class of decimal sequences of \( \frac{1}{q} \), where \( q \) prime, the digits spaced half a period apart add up to \( r-1 \), where \( r \) is the base in which the sequence is expressed. These properties of decimal sequences have made it possible to establish an upper bound on the autocorrelation function. Decimal sequences are also known to have good cross correlation properties and they can be used in applications involving PN sequences [66, 68]. In the previous section a few properties of decimal sequences and their generation using feedback shift registers that allow carry and their application to watermarking are explained. The function for generating the decimal sequences is

\[
dseq = \left[ r' \mod q \right] \mod r
\]

Where \( r \) is the radix and \( q \) is the base of the prime number.

Decimal sequences are watermarked in a similar manner as that of the PN sequences by spread spectrum watermarking. The \( d \)-sequences are added to the cover image either by a circular shift or a random shift. The decimal sequence spread-spectrum watermarking scheme is shown in Figure 4.1. The prime \( q \) drives the decimal sequence generator, produces the chip sequence \( u \), which has zero mean and whose elements are equal of \( -\sigma_u \) or \( +\sigma_u \). The chip sequence \( u \) is either added or subtracted from the signal \( x \) depending on the value of the watermark bit \( b \), which takes values \( \{+1, -1\} \). The signal, \( s \) is the watermarked signal and \( n \) is the noise introduced into the system.

![Figure 3.1: Decimal sequence watermarking scheme.](image)
3.3. Different Transforms used in this Proposed Work

In this proposed work two watermarking algorithms have been developed: the first one using DCT and the later one using IntDCT. It is shown that even though the clarity of the retrieved images is more in the case of DCT in comparison with IntDCT, in situations where the computationally complexity must be very less the IntDCT is preferred over DCT. IntDCT helps to reduce blocking artifacts caused during the processing. On the whole it is observed that IntDCT has similar properties to that of DCT, based on the application the transform is preferred.

3.3.1. Discrete Cosine Transform (DCT)

A transformation function which transforms image from spatial domain to frequency domain which makes analysis of a signal simple. Watermarking in the DCT domain is usually performed on the lower or the mid-band frequencies, as higher frequencies are lost during image compression. DCT Watermarking is done by using direct application of transform to entire image or block wise. The watermark is embedded in perceptually significant components, without any distortion of visual representation of host image.

Formulæ of the 2-D DCT:

\[ F(u, v) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C(u)C(v)f(i, j)\cos\left(\frac{\pi(2i+1)u}{2N}\right) \]

Formulæ of the 2-D Inverse DCT:

\[ f(i, j) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)F(u, v)\cos\left(\frac{\pi(2i+1)u}{2N}\right) \]

which

\[ C(u), C(v) = \begin{cases} \frac{1}{\sqrt{N}}, u, v = 0 \\ \frac{1}{\sqrt{2}}, u, v = 1, 2, \ldots, N - 1 \end{cases} \]

55
Even though computational complexity of DCT is more; it is preferred in many compression techniques, since the quality of the image after processing is greater compared to other transforms. An image is subdivided into 8x8 blocks. To each of these 8x8 block of the original image, DCT is applied as a result of it we get 64 coefficients, representing different horizontal and vertical intensities.

3.3.2. Integer Discrete Cosine Transform (IntDCT)

In this proposed work to develop watermarking algorithm the IntDCT is used, which is generally used in H.264 video standard [92]. H.264 uses this new 4x4 transform to help reduce blocking artifacts caused by existing video coding standards by using 8x8 DCT. IntDCT basically has the same properties as the original DCT. But there are some fundamental differences. First of all, it is an integer transform. All operations can be carried out with integer arithmetic, without loss of accuracy. The core part of the transform is free from multiplications. It only requires additions and shifts. It does not need floating-point and fixed-point multiplications required by DCT. This reduces the computational complexity and it is much easier for hardware implementation.

To develop this integer DCT, let us examine the 4x4 DCT of a 4x4 matrix X:

\[
Y = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} \times \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix}
\]

(3.3)

Where

\[
a = 1/2, \quad b = 1/2 \cos (\pi / 8), \quad c = 1/2 \cos (3\pi / 8)
\]

The matrix multiplication can be factorized to the following equivalent form

\[
Y = (B^*X*B^T) \times Q
\]

(3.4)
Where

\[
\begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & d & -d & -1 \\
1 & -1 & -1 & 1 \\
d & -1 & 1 & 1
\end{bmatrix}, \quad Q = \begin{bmatrix}
a^2 & ab & a^2 & ab \\
ab & b^2 & ab & b^2 \\
a^2 & ab & a^2 & ab \\
ab & b^2 & ab & b^2
\end{bmatrix}
\] (3.5)

\((B^* X^* B^T)\) is a "core" 2-D transform. \(Q\) is a matrix of scaling factors and the symbol \(\otimes\) indicates that each element of \((B^* X^* B^T)\) is multiplied by the scaling factor in the same position in matrix \(Q\). \(d\) is \(c/b\) approximately (0.414).

To simplify the implementation of the transform, \(d\) is approximated by 0.5. To ensure that the transform remains orthogonal, \(b\) also needs to be modified so that:

\[a = 1/2, \quad b = 2/5, \quad d = 1/2\]

The 2\textsuperscript{nd} and 4\textsuperscript{th} rows of matrix \(B\) and the 2\textsuperscript{nd} and 4\textsuperscript{th} columns of matrix \(B^T\) are scaled by a factor of 2 and the post-scaling matrix \(Q\) is scaled down form. This avoids multiplications by 1/2 in the core transform \((B^* X^* B^T)\), which would result in loss of accuracy using integer arithmetic.

The final forward 4x4 IntDCT becomes

\[Y = (B_f^* X^* B_f^T) \otimes Q_f\] (3.6)

Where

\[
\begin{bmatrix}
1 & 1 & 1 & 1 \\
2 & 1 & -1 & -2 \\
1 & -1 & -1 & 1 \\
1 & -2 & 2 & -1
\end{bmatrix}, \quad Q_f = \begin{bmatrix}
a^2 & ab & a^2 & ab/2 \\
ab/2 & b^2/4 & ab/2 & b^2/4 \\
a^2 & ab & a^2 & ab/2 \\
ab/2 & b^2/4 & ab/2 & b^2/4
\end{bmatrix}
\] (3.7)

This transform is an approximation to the 4x4 DCT.

The inverse transform is given by

\[X^* = B_i^T \ast (Y \otimes Q_i) \ast B_i\] (3.8)
Where

\[
 B = \begin{bmatrix}
 1 & 1 & 1 & 1 \\
 1 & \sqrt{2} & -\sqrt{2} & -1 \\
 1 & -1 & -1 & 1 \\
 \sqrt{2} & -1 & 1 & -\sqrt{2}
\end{bmatrix}
, \quad Q_i = \begin{bmatrix}
 a^2 & ab & a^2 & ab \\
 ab & b^2 & ab & b^2 \\
 a^2 & ab & a^2 & ab \\
 ab & b^2 & ab & b^2
\end{bmatrix}
\]  

(3.9)

The matrix \( Y \) is pre-scaled by multiplying each coefficient by the appropriate weighting factor from matrix \( Q_i \), +/− \((1/2)\) in the matrices \( B \), and in its transposed matrix can be implemented by a right-shift without a significant loss of accuracy because the coefficients \( Y \) are pre-scaled. In H-264 scaling matrices \( Q_f \) and \( Q_i \) are combined into quantization.

3.4 Watermarking Algorithms using Decimal Sequence

Embedding and Recovery using Decimal Sequences

A decimal sequence is generated in Matlab using the function

\[
dseq = \lfloor r' \mod q \rfloor \mod r
\]  

(3.10)

Where, \( r \) is the radix or the base and \( q \) is the prime number.

The addition of the \( d \)-sequences to the cover image is done in a manner similar to that applied for the PN sequences. There is a trade off between the robustness and the quality of the image as the gain \( K \) is increased. But, the decimal sequence gives an option of experimenting with various prime numbers, keeping the gain constant, until a satisfactory result for both encryptions of data and its retrieval is observed. For retrieval of the encrypted message the decimal sequences are generated again and then correlated with the watermarked image. The \( d \)-sequences may be added to the cover image by either a circular shift or a random shift.
3.4.1 Proposed Watermarking Algorithm using DCT

It has been noted earlier that the PN sequences based watermarking produces noise due to high autocorrelation values as the period of generated PN sequences is too large when compared to the size of the cover image. To improve the recovery in certain cases the original cover image needs to be high pass filtered but this affects the quality of the watermarking. Since d-sequences have zero cross correlation for some prime numbers [66], one would obtain superior performance if different d-sequences are used in the watermark. But, if the same d-sequence is used, the autocorrelation can be as high as 33% during certain shifts of the sequence, but since many other shifts have zero autocorrelation, it is required to use selectively these shifts to produce better results [68].

As d-sequences have zero cross correlation for some prime numbers, superior performance is obtained if different d-sequences are used in the watermark [74, 79]. Using the decimal sequences also has the flexibility of trying out various prime numbers until satisfactory results of embedding and recovery is obtained.

A. Watermark Embedding Process

The original or host image is first subdivided into sub blocks of size 8x8 pixels, then DCT is applied to each sub block. In each sub block one optimal location to have best correlation, psnr and wpsnr is located and based on watermark bit d-sequence is embedded.

The scaling factor 'k' denotes the strength of the watermark and it can be used to control the overall robustness and the quality of the image. The scaling factor varies from image to image. Increasing this value increases the strength of the watermark but introduces a gradual visible distortion while decreasing this value would result in better hiding of the watermark and hence better quality but decreases the strength of the watermark. So there must be trade-off between strength of the watermark and quality of the watermark, Hence an optimal value needs to be decided depending on the watermark and the d-sequence before embedding. Upon
inverse transformation the watermark will be scattered over the entire image and we obtain the watermarked image.

The digital watermark embedding process of the proposed algorithm is divided into 7 steps and described as below

**Step1**: The original or host image (I) of size 512x512 (M x N) is block processed with 8x8 DCT and it divided into Macro blocks of size 32x32 (B x B) square block.

**Step2**: According to the proposed scheme Watermark image (W) which is logo image will be of size (M/B x N/B) is converted into watermark vector of 1-D sequence.

**Step3**: The macro block of size 32x32 which is further subdivided into sub blocks of size 8x8 pixels, then DCT are applied to each sub block. In each sub block on Optimal location to have best correlation, psnr and wpsnr is located and based on watermark bit d-sequence is embedded.

**Step4**: Now the prime number which acts as a key is selected to generate a d-sequence to the length of \( (B/8 \times B/8) = 32/8 \times 32/8 = 4 \times 4 = 16 \). So the d-sequence length is 16.

**Step5**: In each and every block insert the d-sequence by the following method. If 

\[
(\text{Watermark bit} = 0)
\]

Sub block \((b1, b2) = \text{Sub block } (b1, b2) + k_{\text{gain}} \times \text{d-sequence}\)

else

Sub block \((b1, b2) = \text{Sub block } (b1, b2) - k_{\text{gain}} \times \text{d-sequence}\)

end

Where \((b1, b2)\) location indicates the optimal location of the 8x8 sub block to have best Correlation, PSNR and WPSNR.

**Step6**: The process is repeated from step3 to step5 up to the length of the watermark vector.

**Step7**: Apply the Inverse 8x8 block DCT to produce watermarked image.
B. Watermark Extraction Process

In order to recover the watermark, the Correlation-based watermark detection scheme is used to extract the watermark. The d-sequence is generated using the same prime number used in watermark embedding, and compared with temporary sequence generated after processing the watermarked image. Selecting a threshold 'T' in the process of correlation filters out the unwanted noise. We must be aware that determining the presence of a watermark through correlation is a statistical test and hence there is a possibility of obtaining detection errors. Errors can be of two types, '0' that is falsely detected as '1' and '1' that is falsely detected as '0'. The setting of the threshold is considered as a decision based on the need to minimize errors, such as those mentioned above, in watermark detection.

The digital watermark extraction process of the proposed algorithm is divided into 5 steps and described as below:

Step1: Apply the DCT block processing to the watermarked image; bring out the macro block from which 8x8 sub blocks are identified.

Step2: Regenerate the d-sequence using the same seed used in the watermark embedding.

Step3: By specifying the optimal location the temporary sequence is generated from each 8x8 block. This is obtained from resultant image, and then correlation \([C(b)]\) is calculated between d-sequence and generated temporary sequence.

Step4: The threshold value 'T' (\(\geq 0.5\)) is considered as decision based on the need to minimize the errors. And recover watermark bit is taken as follows:

\[
\text{Recovered watermark bit} = \begin{cases} 
0 & \text{if } C(b) > T \\
1 & \text{if } C(b) < T 
\end{cases}
\]

Reshape the recovered bits as same as the watermark image size.

Step5: The results are evaluated based on Correlation, PSNR and WPSNR for the resultant images. Correlation is calculated between watermark image and extracted watermark image. PSNR and WPSNR are the measurements between watermarks image and watermarked image.
C. Experimental Results

For evaluation of the developed algorithm, experiments are conducted using gray scale images. The host image used is Lena as shown in the Figure.3.2.a of size 512x512. The prime number that is used to generate the d-sequence is ‘17’ and on using the appropriate scaling factor and threshold it is noticed that the watermark is recovered perfectly well. As mentioned earlier the scaling factor is set according to the quality of the watermarked image. The greater the scaling factor, the better is the watermark detection however reducing the overall quality of the image. In accordance with the host image Lena and watermark an optimum value of ‘k’ chosen as 20. Here, a watermark of size 16x16 pixels has been used as shown in Figure.3.2.b.

![Host image. Lena](image1)

Figure.3.2.a: Host image. Lena

![Watermark](image2)

Figure.3.2:b: Watermark

![Watermarked image](image3)

Figure.3.3.a: Watermarked Image

![Retrieved Watermark](image4)

Figure.3.3.b: Retrieved Watermark

In this proposed algorithm CORR, PSNR and WPSNR are calculated for all the locations of sub block (8x8). Among all the locations three best locations are chosen for which the three parameters are optimum. Experimental analysis is done for various host and watermark images. Consider the watermark bit to be embedded,
and it is observed that the location in which watermark bit is to be embedded varies based on the content features (intensity, luminance and texture) of the images. The algorithm is also experimented with various combinations of primes to verify the robustness of the proposed scheme and found that in almost all the combinations the detection of the watermark was highly satisfactory.

The proposed algorithm is based on the d-sequence. Since the watermark bit is embedded in the best location, hence the rate of watermark withstanding capacity is very high; the quality of the watermark image is good in terms of perceptibility, PSNR and WPSNR. This method is superior to all other watermarking algorithms using d-sequence in terms of both PSNR and robustness for different attacks shown in the Table 3.1.

Table 3.1: Performance results of proposed method for without and with attacks.

<table>
<thead>
<tr>
<th>ATTACK</th>
<th>((b_1,b_2)=(4,1))</th>
<th>((b_1,b_2)=(4,3))</th>
<th>((b_1,b_2)=(4,5))</th>
<th>((b_1,b_2)=(8,8))</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CORR</td>
<td>PSNR &amp; WPSNR</td>
<td>CORR</td>
<td>PSNR &amp; WPSNR</td>
</tr>
<tr>
<td>No Attack</td>
<td>0.8700</td>
<td>46.5547, 0.9163</td>
<td>0.8700</td>
<td>46.5547, 0.9163</td>
</tr>
<tr>
<td></td>
<td>34.3694</td>
<td>45.4008</td>
<td>34.3694</td>
<td>45.4008</td>
</tr>
<tr>
<td>Median</td>
<td>0.7864</td>
<td>37.1131, 0.8050</td>
<td>0.7864</td>
<td>37.1131, 0.8050</td>
</tr>
<tr>
<td>Cropping</td>
<td>0.6756</td>
<td>27.3761, 0.8329</td>
<td>0.6756</td>
<td>27.3761, 0.8329</td>
</tr>
<tr>
<td></td>
<td>18.6944</td>
<td>27.3677, 0.8515</td>
<td>18.6944</td>
<td>27.3677, 0.8515</td>
</tr>
<tr>
<td>Rotation</td>
<td>0.8315</td>
<td>27.2089, 0.9163</td>
<td>0.8315</td>
<td>27.2089, 0.9163</td>
</tr>
<tr>
<td></td>
<td>27.2227</td>
<td>32.4591, 0.9441</td>
<td>27.2227</td>
<td>32.4591, 0.9441</td>
</tr>
<tr>
<td>Gaussian Blur</td>
<td>0.4835</td>
<td>36.1610, 0.5607</td>
<td>0.4835</td>
<td>36.1610, 0.5607</td>
</tr>
<tr>
<td></td>
<td>12.0748</td>
<td>36.1678, 0.3478</td>
<td>12.0748</td>
<td>36.1678, 0.3478</td>
</tr>
<tr>
<td>Gaussian Noise</td>
<td>0.4601</td>
<td>36.2093, 0.4289</td>
<td>0.4601</td>
<td>36.2093, 0.4289</td>
</tr>
<tr>
<td></td>
<td>21.6831</td>
<td>29.2200, 0.3326</td>
<td>21.6831</td>
<td>29.2200, 0.3326</td>
</tr>
<tr>
<td>Row Column Copy</td>
<td>0.8515</td>
<td>43.8890, 0.9070</td>
<td>0.8515</td>
<td>43.8890, 0.9070</td>
</tr>
<tr>
<td></td>
<td>38.8166</td>
<td>43.2588, 0.9348</td>
<td>38.8166</td>
<td>43.2588, 0.9348</td>
</tr>
<tr>
<td>Row Column Blan</td>
<td>0.8388</td>
<td>38.9279, 0.9070</td>
<td>0.8388</td>
<td>38.9279, 0.9070</td>
</tr>
<tr>
<td></td>
<td>31.6701</td>
<td>38.7095, 0.9348</td>
<td>31.6701</td>
<td>38.7095, 0.9348</td>
</tr>
<tr>
<td>JPEG 65</td>
<td>0.8885</td>
<td>39.6590, 0.9163</td>
<td>0.8885</td>
<td>39.6590, 0.9163</td>
</tr>
<tr>
<td></td>
<td>30.7020</td>
<td>39.4288, 0.8608</td>
<td>30.7020</td>
<td>39.4288, 0.8608</td>
</tr>
<tr>
<td>JPEG 45</td>
<td>0.8700</td>
<td>38.7403, 0.9070</td>
<td>0.8700</td>
<td>38.7403, 0.9070</td>
</tr>
<tr>
<td></td>
<td>30.2605</td>
<td>38.3295, 0.1256</td>
<td>30.2605</td>
<td>38.3295, 0.1256</td>
</tr>
</tbody>
</table>

63
D. Conclusions

Images for some of the attacks is shown in the following Figures and values are shown in Table 3.1. The proposed algorithm is shown to be more robust for median filter attack, compared to all other d-sequence algorithms [66, 67, 68, 74 and 79], this is the exclusive feature of this algorithm. This algorithm is also tested for salt & pepper noise based on the chosen random values the correlation, PSNR and WPSNR will be varied. This indicates that an embedded watermark is still recoverable even after the common image processing operations on the watermarked image. And hence this algorithm is highly suitable for the Copyright protection, Owner identification and Copy control.

Figure.3.4 shows the result of watermarked image along with retrieved watermark image for Rotation attack, Similarly Figure.3.5 shows Cropping attack, Figure.3.6 shows Median filter attack and Figure.3.7 shows Jpeg compression attack respectively.

![Figure 3.4: Rotation Attack](image1)

![Figure 3.5: Cropping Attack](image2)

![Figure 3.6: Med filter Attack](image3)

![Figure 3.7: Jpeg compression Attack](image4)

3.4.2 Modification to Proposed Algorithm using DCT and IntDCT
The proposed algorithm which is described in the above section can still be further developed or modified to increase the data payload by maintaining the trade-off among the three parameters (robustness, imperceptibility and capacity) by considering 4x4 block processing.

1. Modification of Algorithm using DCT

a. Embedding the watermark

The original or host image is first divided into Macro block of size 32x32 which is further subdivided into sub blocks of size 8x8 DCT block processing, then it is further subdivided into 4x4 super sub blocks. In this proposed method among four super sub blocks of 8x8 block only one super sub block is selected. And in the super sub block one optimal location to have trade-off among Correlation, PSNR and WPSNR is located and based on the watermark bit d-sequence is spread throughout the sixteen 8x8 blocks of macro block. Now the prime number which acts as a key is selected to generate a d-sequence to the length of 16.

The scaling factor 'k' denotes the strength of the watermark and it can be used to control the overall robustness and the quality of the image and it is selected accordingly. Upon inverse transformation the watermark will be scattered over the entire image and the watermarked image is obtained.

b. Watermark Extraction

In order to recover the watermark, the Correlation-based watermark detection scheme is used to extract the watermark which is similar to the proposed watermarking algorithm using decimal sequences.
Table 3.2: Performance results of proposed method using DCT without and with attacks.

<table>
<thead>
<tr>
<th>ATTACK</th>
<th>(b1,b2)=(4,1)</th>
<th>(b1,b2)=(4,3)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CORR</td>
<td>PSNR</td>
</tr>
<tr>
<td>No Attack</td>
<td>0.8700</td>
<td>46.5547</td>
</tr>
<tr>
<td>Median filter</td>
<td>0.7864</td>
<td>37.1131</td>
</tr>
<tr>
<td>Cropping</td>
<td>0.6756</td>
<td>27.3761</td>
</tr>
<tr>
<td>Rotation</td>
<td>0.8315</td>
<td>32.4667</td>
</tr>
<tr>
<td>Gaussian Blur</td>
<td>0.4835</td>
<td>36.1610</td>
</tr>
<tr>
<td>Gaussian Noise</td>
<td>0.4601</td>
<td>29.2093</td>
</tr>
<tr>
<td>Row Column</td>
<td>0.8515</td>
<td>43.8890</td>
</tr>
<tr>
<td>Copying</td>
<td>0.8388</td>
<td>38.9279</td>
</tr>
<tr>
<td>Row Column</td>
<td>0.8885</td>
<td>39.6590</td>
</tr>
</tbody>
</table>

For different attacks the CORR (Correlation), PSNR and WPSNR values are shown in the Table 3.2. This algorithm can be used when data payload required is more. This algorithm is less robust compared proposed algorithm.

2. Modification of Algorithm using IntDCT

a. Embedding the Watermark

The original or host image is first divided into Macro block, which is further subdivided into sub blocks of size 4x4 pixels, then IntDCT is applied to each sub block. In each sub block one optimal location is chosen to have best Correlation, PSNR and WPSNR is located and based on watermark bit d-sequence is embedded.
An optimal value 'k' the scaling factor value is chosen. Upon inverse transformation the watermark will be scattered over the entire image and the watermarked image is obtained.

b. Watermark Extraction

In order to recover the watermark, the Correlation-based watermark detection scheme is used to extract the watermark. The d-sequence is generated using the same prime number used in watermark embedding, and compared with temporary sequence generated after processing the watermarked image.

Table 3.3: Performance results of proposed method using Int DCT

<table>
<thead>
<tr>
<th>ATTACK</th>
<th>((b_1,b_2)=(2,3))</th>
<th>((b_1,b_2)=(2,4))</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CORR</td>
<td>PSNR</td>
</tr>
<tr>
<td>No Attack</td>
<td>0.9834</td>
<td>40.6296</td>
</tr>
<tr>
<td>Median filter</td>
<td>0.7805</td>
<td>37.1528</td>
</tr>
<tr>
<td>Cropping</td>
<td>0.8978</td>
<td>27.2955</td>
</tr>
<tr>
<td>Rotation</td>
<td>0.9611</td>
<td>32.3836</td>
</tr>
<tr>
<td>Gaussian Blur</td>
<td>Very poor</td>
<td>40.6296</td>
</tr>
<tr>
<td>Gaussian Noise</td>
<td>0.5161</td>
<td>40.6296</td>
</tr>
<tr>
<td>Row Column Copying</td>
<td>0.9834</td>
<td>40.6296</td>
</tr>
<tr>
<td>Row Column Blanking</td>
<td>0.9834</td>
<td>40.6296</td>
</tr>
<tr>
<td>JPEG (65)</td>
<td>0.9611</td>
<td>40.6296</td>
</tr>
</tbody>
</table>
3.5. Results

Evaluation of developed algorithms that is experimental analysis on the proposed work is done using gray scale images. The host image used is Lena of size 512x512 as shown in Figure.3.2.a. The prime number that is used to generate the d-sequence is 17 and on using the appropriate scaling factor and threshold the watermark is recovered perfectly well. As mentioned earlier the scaling factor is set according to the content and the quality of the watermarked image. The greater the value of scaling factor or gain factor ‘k’ reduces the overall quality of watermarked image. Hence an optimum value is chosen accordingly and it is set to 20 for Lena. Here, a watermark of size 32x32 pixels has been used in the proposed algorithm using IntDCT. Figure.3.8 shows watermark, watermarked image and recovered watermark respectively.

In this process Correlation, PSNR and WPSNR are calculated for all the locations of sub block (4x4) and the two best locations are chosen for which there is trade-off among the three parameters. Based on the watermark bit, d-sequence is embedded in the chosen location. The location in which watermark bit is embedded varies based on the content features of the host image. Experimental analysis is carried out with various combinations of primes to verify the robustness of over scheme and found that in almost all the combinations the detection of the watermark was highly satisfactory.
3.6. Conclusions

The proposed algorithm is based on the d-sequence. Since the watermark bit is embedded in the best location, the rate of watermark withstand capacity is very high, the quality of the watermark image is good in terms of perceptibility, PSNR and WPSNR. This method is superior to all other watermarking algorithms using d-sequence in terms of both PSNR and robustness for different attacks shown in the Table 3.2 and 3.3. The proposed algorithm is shown to be more robust for median filter attack, compared to all other d-sequence algorithms. This is the exclusive feature of this algorithm. This algorithm is also tested for salt & pepper noise based on the chosen random values the correlation, PSNR and WPSNR will be varied. This indicates that an embedded watermark is still recoverable even after the common image processing operations on the watermarked image. And hence this algorithm is highly suitable for the “copyright protection”.

In this proposed work it is observed how data payload can be increased is shown. Computational complexity is more in algorithms using DCT and can be reduced using IntDCT. But clarity or perceptibility is more in the algorithms using DCT.