Chapter 2

A Hybrid Semi-Blind Digital Image Watermarking Technique

2.1 Introduction

There has been an upsurge in broadcasting media since the beginning of this century because of many technical innovations in this field. Media security concerns are copyright protection, broadcast monitoring and owner identification. Digital watermarking is the greatest bet to address these concerns. The ease of distribution of documents through the web may transgress protection laws against unauthorized copies and make fidelity questionable. Digital watermarking has been proposed as a solution against these practices. Digital watermarking is a labeling technique of digital data with secret information that can be extracted in the receptor. The image in which this data is inserted is called cover image or host. The watermarking process has to be resilient against all possible attacks, keeping the content of the watermark readable in order to be recognized when extracted. Features like robustness and fidelity are essentials for a watermarking system, however the size of the embedded information has to be considered since data becomes less robust as its size increases. Therefore a trade-off of these features must be considered. In this thesis, we show a classification of watermarks, propose a basic model for watermarking and explain efficient algorithms for image watermark embedding and extraction.

2.2 Detection Types

This classification determines which resources are necessary for the analysis to extract the watermark from the cover image.
1) Blind: In this detection type the original image and mark data is not available to the receiver. For example: Copy control applications must send different watermarks for each user and the receiver must be able to recognize and interpret these different marks.

2) Non-Blind: In this case, the receiver needs the original data, or some derived information from it, for the detection process [1]. This data will also be used in the extraction algorithm.

The lifting wavelet transform (LWT) is a recent approach to wavelet transform and singular value decomposition (SVD) is a valuable transform technique for robust digital watermarking. While LWT allows generating an infinite number of discrete biorthogonal wavelets starting from an initial one, singular values (SV) allow us to make changes in an image without affecting the image quality much. This paper presents an approach which tries to amalgamate the features of these two transforms to achieve a hybrid and robust digital image watermarking techniques. Certain performance metrics are used to test the robustness of the method against common image processing attacks. Copyright Protection is a major issue as far as content transfer over the worldwide web is concerned. There is a growing concern over multimedia content protection with the growing accessibility and usability of internet. Major issue of concern here is the images, audio or video transmitted. Various methods address this issue of multimedia content protection, one of them being digital image watermarking[77-100]. Watermarking (data hiding) [1, 2, 3] is the process of embedding data into a multimedia element such as image, audio or video. This embedded data can later be extracted from, or detected in, the multimedia for security purposes[101-110].

Digital watermarking is the process of possibly irreversibly embedding information into a digital signal. The signal may be audio, pictures or video, for example. If the signal is copied, then the information is also carried in the copy. In visible watermarking, the
information is visible in the picture or video. Typically, the information is text or a logo which identifies the owner of the media. In invisible watermarking, information is added as digital data to audio, picture or video, but it cannot be perceived as such. The digital watermarking is intended to complement cryptographic process. Access control or authenticity verification has been addressed by digital watermarking as well as by biometric authentication [4].

2.3 Watermark Embedding

The method used to embed the watermark influence both the robustness against attacks and the detection algorithm, but some methods are very simple and cannot meet the application requirements. El-Gayyar and von zur Gathen[111] showed that designing a watermark should consider a trade-off among the basic features of robustness, fidelity and payload. There are two approaches for the embedding process:

1) **Spatial Domain**: These watermarks insert data in the cover image changing pixels or image characteristics. Watermark is embedded using LSB, Statistical, Feature based and Block based techniques. Spatial-domain techniques work with the pixel values directly. The algorithms should carefully weight the number of changed bits in the pixels against the possibility of the watermark becoming visible. These watermarks have been used for document authentication and tamper detection. Generally, spatial domain watermarking is easy to implement from a computational point of view, but too fragile to resist numerous attacks [5].

2) **Transform Domain**: These algorithms hide the watermarking data in transform coefficients, therefore spreading the data through the frequency spectrum making it hard to detect and strong against many types of signal
processing manipulations. The most used transforms are: Discrete cosine transform (DCT), Discrete Fourier Transform (DFT), discrete wavelet transform (DWT) and discrete lifting transform (LWT) like the ones suggested in [6,7].

2.4 Transform Domain Watermarking Algorithms

Transform-domain techniques employ various transforms, either local or global. In order to have more promising techniques, researches were directed towards watermarking in the transform domain, where the watermark is not added to the image intensities, but to the values of its transform coefficients. Then to get the watermarked image, one should perform the transform inversely.

In case of frequency domain watermarking schemes, there has to be a trade-off between robustness and invisibility. When a watermark is embedded in the most significant components, it becomes robust to attacks but the watermark becomes difficult to hide. Whereas, when we embed a watermark in the lesser significant components, it is easier to hide it but the scheme is least resistant to attacks.

The wavelet transform is one type of transform domain technique. Wavelet based transforms gained popularity recently because of the property of multi-resolution analysis that it provides. Wavelets can be orthogonal or bi-orthogonal. Most of the wavelets used in watermarking are orthogonal wavelets. A new approach to wavelet transform is the lifting wavelet transform [20]. In this paper, the fusion of LWT and SVD approaches i.e. LWT-SVD scheme is proposed, where an image is watermarked using other image for the purpose of validation.
2.5 Watermarking technique using IWT and SVD

The decomposition of a signal in terms of a wavelet basis is termed as wavelet transform. The basic idea of wavelet transforms is to exploit the correlation structure present in most real life signals to build a sparse approximation. A new mathematical formulation proposed by Swelden [20], based on spatial construction of the wavelets is called the lifting-based wavelet transform. The underlying principle of this approach [19,20] is to break up the high-pass and the low-pass wavelet filter into a sequence of smaller filters that in turn can be converted into a sequence of alternating upper and lower triangular matrices and a diagonal matrix with constants. The factorization is obtained by using an extension of the Euclidean algorithm. The resulting formulation can be implemented by means of banded matrix multiplications.

Let $h(z)$ and $g(z)$ be the low pass and high pass analysis filters and $h(z)$ and $g(z)$ be the low pass and high pass synthesis filters. The polyphase representation of the filter $h$ is expressed as depicted in equation 1 and Figure 2.1.

$$h(z) = h(z^2) + z^{-1} h(z^2)$$ (1)

First subsample into even and odd, then apply the dual polyphase matrix. For the inverse transform, First apply the polyphase matrix and then join even and odd.
The lifting scheme [20] is an easy relationship between perfect reconstruction filter pairs that have the same low-pass or high-pass filter. One can then start from the Lazy wavelet and use lifting to gradually build one's way up to a multiresolution analysis with particular properties. The lifting technique used in the present approach is Primal Lifting, which lifts the low-pass sub-band with the help of high-pass sub-band.

The Singular Value Transform (SVD), was explored a few years ago for watermarking purposes. In recent years, SVD has been used in watermarking as a different transform as it is one of the most powerful tools of linear algebra with several applications in image compression [8,9,10,11,12,13], watermarking[14,15,16,17]. Singular values are the luminance values of SVD image layer, changing these values slightly do not affect the image quality much [18]. The purpose of singular value decomposition is to reduce a dataset containing a large number of values to a dataset containing significantly fewer values, but which still contains a large fraction of the variability present in the original data. SVD analysis results in a more compact representation of these correlations, especially with multivariate datasets and can provide insight into spatial and temporal variations exhibited in the fields of data being analyzed.
The embedding is directly related with the extraction algorithm. The embedding algorithm is basically a combination of the watermark with the chosen media, so the result is equivalent to:

\[ I_w = E(I,W) \]  \hspace{1cm} (2)

where \( I \) is the original media, \( W \) the watermark, \( E \) is the embedding function and \( I_w \) the watermarked media. The function depends on the algorithm and the analyzed domain. SVD is a numeric analysis of linear algebra which is used in many applications in image processing. It is used to decompose a matrix with a little truncate error according to the equation below:

\[ A = U S V^T \]  \hspace{1cm} (3)

Where \( A \) is the original matrix, \( U \) and \( V \) are orthogonal matrices with dimensions \( m \times m \) and \( n \times n \) respectively, \( S \) is a diagonal matrix of the Eigenvalues of \( A \) and \( T \) indicates matrix transposition. After the decomposition of the cover image the watermark is added using a scale coefficient \( \alpha \) to get the following equation:

\[ S + \alpha W = U_w S_w V_w^T \]  \hspace{1cm} (4)

Multiplying matrices \( U, V^T \) and \( S_w \) result in the marked image \( A_w \):

\[ A_w = U S_w V^T \]  \hspace{1cm} (5)

This was possible due to the high stability of singular values (SV) of SVD. This method improves watermark robustness and resistance against many kinds of attacks.

The full singular value decomposition of an \( m \)-by-\( n \) matrix involves an \( m \)-by-\( m \) \( U \), an \( m \)-by-\( n \) \( S \), and an \( n \)-by-\( n \) \( V \). In other words, \( U \) and \( V \) are both square and \( S \) is the same size as \( A \). The singular value decomposition is the appropriate tool for analyzing a mapping from one vector space into another vector space, possibly with a different dimension.
2.6 Proposed Technique

The section describes the proposed watermarking scheme which is carried out in two phases, the watermark embedding phase and the watermark extraction phase. The watermark embedding scheme is illustrated in Figure 2.2.

![Figure 2.2: The proposed watermark embedding scheme.](image)

The watermark extraction scheme is illustrated in Figure 2.3.

![Figure 2.3: The proposed watermark extraction scheme.](image)
2.7 Algorithm I - Watermark Embedding

Let A be the cover image, B be the watermark and k the embedding coefficient then watermarked image I is obtained by the function I= WaterMark_Embed(A,B, k); as per the following algorithm. The matlab illustration for the transformation is also given.

1. Apply first level Lifting Wavelet Transform on Cover Image [ca1,ch1,cv1,cd1]=lwt2(A);
2. Apply second level Lifting Wavelet Transform on Cover Image [ca2,ch2,cv2,cd2]=lwt2(ca1);
3. Decompose to [U,S,V] by applying SVD [U,S,V]=svd(ch2);
4. Apply first level Lifting Wavelet Transform on Watermark Image [wa1,wh1,wv1,wd1]=lwt2(B);
5. Apply second level Lifting Wavelet Transform on Watermark Image [wa2,wh2,wv2,wd2]=lwt2(wa1);
6. Decompose to [P,Q,R] by applying SVD [P,Q,R]=svd(wh2);
7. Compute embedding (D=S+k*Q)
8. Decompose to [U1,D1,V1] by applying SVD. ([U1,D1,V1]=svd(D) )
9. Compute Watermarked Image matrix (CA2=U*D1*V)
10. Apply first level Inverse LWT on Watermarked Cover Image ca1=ilwt2(ca2,ch2,cv2,cd2);
11. Apply second level LWT on Watermarked Cover Image I=ilwt2[ca1,ch1,cv1,cd1];
12. I=Watermarked Image

2.8. Algorithm 2 - Watermark Extraction

Let I be the watermarked image, B be the actual watermark and k the embedding coefficient then extracted watermark I’ is obtained by the function I’= Watermark_Extract(I,B,k);
as per the following algorithm. The matlab illustration for the transformations is also given.

1. Apply first level LWT on Watermarked Cover Image \([c_{a1}, c_{h1}, c_{v1}, c_{d1}] = \text{lwt2}(I)\);
2. Apply second level LWT on Watermarked Cover Image \([c_{a2}, c_{h2}, c_{v2}, c_{d2}] = \text{lwt2}(c_{a1})\);
3. Decompose to \([X, Y, Z]\) by applying SVD. \([X, Y, Z] = \text{svd}(c_{h2})\);
4. Compute inverse embedding \(B = (Y - S)/k\);
5. Compute watermark image matrix \((w_{h2} = P^tB^tR_s)\)
6. Apply Inverse LWT on Recovered Watermark \((I' = \text{lwt2}(w_{a2}, w_{h2}, w_{v2}, w_{d2}))\)
7. \(I'\) = Recovered Watermark

2.9 Experimental Work and Results

2.9.1 Experimental Measurement Metrics

We have used two quality measurements to quantify the error between images namely, Peak Signal to Noise Ratio (PSNR), and Mean Structural Similarity Index Measure (MSSIM) [21].

\[
\text{PSNR} = 10 \log_{10} \frac{255^2}{MSE} \quad \text{and,} \\
\text{MSE} = \frac{\sum_{i=1}^{n} (I(i) - I'(i))^2}{n}
\]

Where \(I\) and \(I'\) are the original and watermarked images respectively, \(n\) is the total number of pixels. 255 refers to the maximum possible pixel value in an eight bit image. The SSIM index is a full reference metric, in other words, the measuring of
image quality based on an initial uncompressed or distortion-free image as reference. SSIM is designed to improve on traditional methods like peak signal-to-noise ratio (PSNR) and mean squared error (MSE), which have proved to be inconsistent with human eye perception. SSIM is a new paradigm for quality assessment, based on the hypothesis that the HVS is highly adapted for extracting structural information. The measure of structural similarity compares local patterns of pixel intensities that have been normalized for luminance and contrast. In practice, a single overall index is sufficient enough to evaluate the overall image quality; hence a mean SSIM (MSSIM) index is used as the quality measurement metric.

\[
SSIM (x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{\mu_x^2 + \mu_y^2 + \sigma_x^2 + \sigma_y^2 + C_2}
\]

\[
MSSIM (X, Y) = \frac{1}{M} \sum_{j=1}^{M} SSIM (x_j, y_j)
\]

The Lena image of size 512×512 has been selected as the cover image and for the watermark the cameraman image of size 512×512 has been used.
Table 2.1: Watermarked Images And Recovered Watermarks For Various Values Of Embedding Factor(K).

<table>
<thead>
<tr>
<th>Factor(K)</th>
<th>PSNR = 38</th>
<th>PSNR = 52</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10</td>
<td>MSSIM = 0.94567</td>
<td>MSSIM = 0.997063</td>
</tr>
<tr>
<td>0.20</td>
<td>MSSIM = 0.927186</td>
<td>MSSIM = 0.999873</td>
</tr>
<tr>
<td>0.5</td>
<td>MSSIM = 0.878370</td>
<td>MSSIM = 0.999881</td>
</tr>
</tbody>
</table>
Table 2.2: List And Summary Of Attacks Simulated On The Watermarked Image.

<table>
<thead>
<tr>
<th>Attack /Transform</th>
<th>Options</th>
<th>Survives the attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cropping</td>
<td>% of original image: 1, 2</td>
<td>YES</td>
</tr>
<tr>
<td>Histogram Equalization</td>
<td></td>
<td>YES</td>
</tr>
<tr>
<td>Median Filtering</td>
<td>Filter size 3 x 3</td>
<td>YES</td>
</tr>
<tr>
<td>Salt n pepper</td>
<td>Filter size 3 x 3</td>
<td>YES</td>
</tr>
<tr>
<td>Sharpening</td>
<td>Filter size 3 x 3</td>
<td>YES</td>
</tr>
<tr>
<td>Scaling</td>
<td>Default value for number of levels is 64</td>
<td>YES</td>
</tr>
<tr>
<td>Weiner Filtering</td>
<td>Noise density: 0.02 to 0.05</td>
<td>YES</td>
</tr>
<tr>
<td>Gaussian noise</td>
<td>Gaussian white noise of mean 0 and variance 0.01</td>
<td>YES</td>
</tr>
<tr>
<td>JPEG Compression</td>
<td>Compression ratio: 30, 50, 70, 90</td>
<td>YES</td>
</tr>
</tbody>
</table>
2.9.2 Results

Table 2.1 gives a pictorial representation of the effect of varying the embedding factor on the PSNR and the recovered watermarks. To test the robustness of the technique, the results in the present study were tested against the attacks listed in Table 2.2 whereas Table 2.3 shows the extracted watermarks after the watermarked image was subjected to the above mentioned attacks. The technique was also tested against JPEG compression with different quality factors i.e. different compression ratios. Table 2.4 shows recovered watermarks after simulation of JPEG Compression for different compression ratios.
Table 2.3: Watermarked Images And Recovered Watermarks After Simulation Of Various Processing Attacks On The Watermarked Image.

- **Cropping**
  - PSNR = 50
  - MIN / MAX = 0.095647

- **Histogram Equalization**
  - PSNR = 57
  - MIN / MAX = 0.725906

- **Gaussian Noise**
  - PSNR = 26
  - MIN / MAX = 0.841495

- **Salt and Pepper Noise**
  - PSNR = 23
  - MIN / MAX = 0.66536

- **Median Filtering**
Table 2.4: Recovered Watermarks After Simulation Of Jpeg Compression Attack For Different Compression Ratios.

<table>
<thead>
<tr>
<th>Compression</th>
<th>PSNR</th>
<th>MSSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>90%</td>
<td>57</td>
<td>0.999306</td>
</tr>
<tr>
<td>70%</td>
<td>51</td>
<td>0.997820</td>
</tr>
<tr>
<td>50%</td>
<td>44</td>
<td>0.991676</td>
</tr>
<tr>
<td>30%</td>
<td>53</td>
<td>0.998113</td>
</tr>
</tbody>
</table>
2.9.3 Effect Of Using Different Wavelets

The lifting wavelet transform was implemented using different wavelet families and their corresponding effect on the watermarked image was observed. While Figure 2.4 presents a graphical overview of the results obtained, Table 2.5 depicts the same results in an analytical manner. Some wavelets show better performance than others. After testing the technique with wavelet families like db1, bior1.1, rbio 5.5 and rbio1.1, it was observed that rbio 5.5 wavelets shows higher performance than others for lower embedding capacity. But when capacity was increased and tested rbio1.1 shows a better performance. Hence, we have used rbio 1.1 as the wavelet in lifting scheme as these are compactly supported biorthogonal spline wavelets for which symmetry and exact reconstruction are possible with FIR filters as the filters used for decomposition and reconstruction are different, thus reducing the interference and providing better reconstruction.

Figure 2.4: Graphical representation of embedding factor (k) versus PSNR for various wavelet families.
Table 2.5: The Effect Of Various Wavelet Families On PSNR Of The Watermarked Image.

<table>
<thead>
<tr>
<th>k</th>
<th>rbio1.1 (in dB)</th>
<th>rbio5.5 (in dB)</th>
<th>Db1 (in dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>38</td>
<td>31</td>
<td>36</td>
</tr>
<tr>
<td>0.2</td>
<td>36</td>
<td>26</td>
<td>32</td>
</tr>
<tr>
<td>0.5</td>
<td>34</td>
<td>21</td>
<td>32</td>
</tr>
<tr>
<td>1.0</td>
<td>29</td>
<td>19</td>
<td>21</td>
</tr>
<tr>
<td>1.5</td>
<td>26</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>2.0</td>
<td>24</td>
<td>18</td>
<td>17</td>
</tr>
</tbody>
</table>

2.9.4 The Gain Factor Effect

The watermark is embedded into the cover image using different embedding factors (k). In the embedding process, the singular component are multiplied by a gain factor, and then embedded in the host image coefficients. Therefore, changing the value of the gain factor has an obvious effect on both, the watermarked image, and the watermark extracted from it. The results in the proposed technique were tested for values of k in the range of 0.10 to 2.0. Table 2.1 shows the effect of changing the gain factor on the Peak Signal-To-Noise Ratio (PSNR) of the watermarked images and the recovered images. Figure 2.5 represents the same effect for the rbio1.1 wavelet being used in the current approach.

![Graphical representation of embedding factor (k) versus PSNR](image)

Figure 2.5: Graphical representation of embedding factor (k) versus PSNR
2.9.5 The Decomposition Level Effect

In order to embed the watermark into the host image, one should perform LWT to the host image and obtain the required coefficients for embedding. The coefficients needed for embedding can be obtained from one level (scale) of LWT or more. In this study, we perform tests that include one-level LWT, and other tests that include two-level LWT.

To show the effect of the level, a gain factor of 0.2 was selected in embedding and the tests were performed on Lena image and the watermark. The effect of the decomposition level is shown in Table 2.6. The table shows the results of two tests, the first test embeds in the approximation subband of the first level decomposition, and the second embeds in the approximation sub-band (second level) that is obtained from the first level decomposition.

It has been observed that embedding in 1\textsuperscript{st} level results in a lesser value of the quality metrics as the recovered watermark is not very clear whereas in case of 2\textsuperscript{nd} level, the watermarked image as well as the recovered watermark has higher value for both the quality metrics i.e. PSNR and MSSIM.

The results in the present study were tested against the attacks listed in Table 2.3, to test the robustness of the technique.

\textbf{Table 2.6: Effect Of Varying The Level Of Decomposition}

<table>
<thead>
<tr>
<th>LWT Level</th>
<th>Watermarked Image</th>
<th>Recovered Watermark</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>MSSIM</td>
</tr>
<tr>
<td>Level 1</td>
<td>36</td>
<td>0.9381</td>
</tr>
<tr>
<td>Level 2</td>
<td>38</td>
<td>0.9456</td>
</tr>
</tbody>
</table>
2.10 Conclusion

The difference expansion watermarking methods usually embed the data bit by bit into the cover image. We have proposed an algorithm to embed bytes directly into the difference as watermark. We have analyzed the performance of various wavelets for a given capacity and we have also studied how embedding capacity varies for a given image using various wavelet decompositions by varying the payload. Some wavelets, though they seem to perform better at lower capacity, they are not able to embed like rbio1.1 smoothly at different embedding rates.