CHAPTER 4
A PROPOSED NOVEL BIOMETRIC WATERMARKING APPROACH USING LWT- SVD

4.1 Introduction

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 chapter 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.

4.2 Theoretical Concepts Used

4.2.1 Lifting Wavelet Transform

4.2.1.1 Basics of Orthogonal Discrete Wavelets

For many purposes it is useful to expand a given function as a linear combination of some particular basis functions. A well known example of this is Fourier analysis, where the basis is \( \{1, \sin(nx), \cos(nx) : n \in \mathbb{N}\} \). This basis is orthonormal under the inner product

\[
<u, v> = \frac{1}{\pi} \int_0^{2\pi} u(x)v(x)dx
\]  

(4.1)

The expansion of a \( 2\pi \) periodic function \( f \) into the Fourier basis is thus

\[
f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos(nx) + b_n \sin(nx))
\]  

(4.2)

Where \( a_0 = \langle f, 1 \rangle = \frac{1}{\pi} \int_0^{2\pi} f(x)dx \)  

(4.3)

\[
a_n = \langle f, \cos(nx) \rangle = \frac{1}{\pi} \int_0^{2\pi} f(x)\cos(nx)\ dx
\]  

(4.4)

\[
b_n = \langle f, \sin(nx) \rangle = \frac{1}{\pi} \int_0^{2\pi} f(x)\sin(nx)\ dx
\]  

(4.5)
Fourier analysis is an invaluable tool for the analysis of periodic signals such as vibrations. However, if \( f \) is non-periodic or is of compact support (i.e. \( f \) is zero outside a finite interval), it is less appropriate. This is because the Fourier basis functions have no ‘spatial localisation’ they do not decay at all. Thus, if we require cancellation outside a finite domain, convergence of the expansion is slow. In such cases, a better approach is to use wavelets. These are basis functions of compact (spatial) support, and are generated from a single function \( \psi(x) \), the ‘mother wavelet’, by integer translation,

\[
\psi(x) > \psi(x - 1) \tag{4.6}
\]

and dyadic dilation,

\[
\psi(x) \rightarrow \psi(2x) \tag{4.7}
\]

It is important to note that while the Fourier basis is perfectly localised in frequency space, consisting of sinusoids, the wavelet basis is not. There is a reciprocal relationship between a function’s spatial and frequency localisation, similar in nature to the Heisenberg uncertainty relation. The price paid for the compact support of wavelets is a loss of perfect localisation in frequency space.

The aim here is to analyse a function on differing scales. The mathematical formalism for this is encapsulated in the following definition:

A *Multiresolution Analysis* is a sequence of closed subspaces \( \ldots, V_{-1}, V_0, V_1, \ldots \) of \( L^2(\mathbb{R}) \) (the square integrable functions) such that

\[
V_n \in V_{n-1} \tag{4.8}
\]

\[
\bigcup_{n \in \mathbb{Z}} V_n = L^2(\mathbb{R}) \text{ and } \bigcap_{n \in \mathbb{Z}} V_n = \{0\} \tag{4.9}
\]

\[
f(x) \in V_n \iff f(2x) \in V_{n-1} \tag{4.10}
\]

\[
f(x) \in V_0 \Rightarrow f(x - k) \in V_0 \text{ for all } k \in \mathbb{Z} \tag{4.11}
\]

Consider a *scaling function* \( \phi(x) \) such that the set of integer translates of form an orthonormal set, that is

\[
< \phi(x - j), \psi(x - k) = \delta_{j,k} \tag{4.12}
\]
Let $V_0$ span $\{\phi(x-k) : k \in \mathbb{Z}\}$. We see that $\phi(x)$ generates a multiresolution analysis. Hence we have

\[
\phi(x) \in V_0 \Rightarrow \phi(x) \in V_{-1}, \text{ since } V_0 \subset V_{-1}
\]

\[
\Rightarrow \phi\left(\frac{x}{2}\right) \in V_0
\]

Since $\{\phi(x-k): k \in \mathbb{Z}\}$ is a basis for the vector space $V_0$, we must have

\[
\phi\left(\frac{x}{2}\right) = \sum_k c_k \phi(x-k) \quad (4.13)
\]

or, rescaling the variable as

\[
\phi(x) = \sum_k c_k \phi(2x-k) \quad (4.14)
\]

where only a finite number $N$ of the $c_k$ are nonzero since it needs to be of compact support. Equation (4.14) is called the *scaling relation*, and is a two scale *dilational equation*. The numbers $c_k$ are called *filter coefficients*. The number of nonzero filter coefficients affects the properties of the scaling function, notably its differentiability properties. It can be shown that the function which satisfies this relation will be zero outside the range $[0, N - 1]$. This is seen by considering iteration of the map $x \rightarrow 2x - k$ for $k = 0, \ldots, N - 1$.

For example, if there are four nonzero coefficients $-c_0, c_1, c_2, c_3$ then graphing $2x - k$:

Fig 4.1: Plot of Wavelet Coefficients Given in the Example.
we see that only points in the range \([0,3]\) have fixed points under the transformation \(x \rightarrow 2x - k\), where \(k = 0,1,2,3\). Points outside \([0,3]\) diverge to \(\pm \infty\) under iteration of this transformation. Therefore, for a point \(x_0\) outside this range, the scaling relation implies that \(\phi(x_0)\) can be expressed as a linear combination of \(\phi(x_i)\), where the \(x_i\) have arbitrarily large magnitude. Thus, since we want the scaling function to have compact support, we conclude that the value of the function outside this range must be zero. A similar argument can be made in the general case, so the scaling function is zero outside the range \([0,N - 1]\).

### 4.2.1.2 Conditions on the Filter Coefficients

It is clear that the scaling relation \(\phi\) determines only up to a multiplicative constant. It is necessary to impose some conditions on \(\phi\) in order to uniquely determine the filter coefficients. Firstly we require \(\phi\) to be normalised, that is, we set \(\int \phi(x) \, dx = 1\) (Where limits of integration are omitted, the integration is understood to range from \(-\infty\) to \(+\infty\)). The scaling relation then determines a condition on the filter coefficients:

\[
\int \phi(x) \, dx = \sum_k c_k \int \phi(2x - k) \, dx = \frac{1}{2} \sum_k c_k \int \phi(y) \, dy
\]  

(4.15)

Where

\[
y = 2x - k \Rightarrow \sum_k c_k = 2
\]  

(4.16)

This is the normalisation condition. The filter coefficients are generally scaled by a factor of \(\sqrt{2}\) and denoted \(h_k\), where \(h_k = \frac{c_k}{2}\). The normalisation condition is then \(\sum_k h_k = \sqrt{2}\).

The second condition to be imposed is that the integer translates of the scaling function are orthonormal, since the multiresolution analysis was generated on this premise. With

\[
< f(x), g(x - k) = \int_{-\infty}^{\infty} f(x)g(x) \, dx
\]

we require

\[
< \phi(x), \phi(x - k) >= \delta_{0,k}, \text{ but}
\]

\[
\int \phi(x)\phi(x - k) \, dx = \sum_l c_l \sum_m c_m \int \phi(2x - l)\phi(2x - 2k - m) \, dx
\]

\[
= \frac{1}{2} \sum_l c_l \sum_m c_m \int \phi(y)\phi(y + l - 2k - m) \, dy
\]

\[
= \frac{1}{2} \sum_l c_l \sum_m c_m \delta_{l,2k+m} = \frac{1}{2} \sum_l c_l c_{l-2k}
\]  

(4.17)
So, if \( k = 0 \), we have the square normalisation condition:

\[ \frac{1}{2} \sum_k c_k^2 = 1 \Rightarrow \sum_k c_k^2 = 2 \]

and, if \( k \neq 0 \), we have the orthonormality condition:

\[ \sum_k c_k c_{k-2m} = 0 \text{ for all } m \neq 0. \]

Thus we have \( \frac{N}{2} \) more conditions on the coefficients the - square normalisation condition, and the orthonormality condition where \( m \) ranges from 1 to \( \frac{N}{2} - 1 \).

The next condition we shall impose on the scaling function concerns the degree \( p \) to which the monomials \( 1, x, x^2, \ldots \) can be represented exactly by a basis consisting of translates of the scaling function. That is,

\[ x^{p-1} = \sum_k \beta_k \phi(x + k) \text{ where} \]

\[ \beta_k = \int x^{p-1} \phi(x + k) \, dx \]

by orthonormality of the translates of \( \phi \). Strang [90] showed that this can be related to a condition on the Fourier transform of the scaling function, but a simpler method of finding the condition on the coefficients is to define wavelets. We define the mother wavelet in terms of the scaling function as

\[ \psi(x) = \sum_k (-1)^N c_k \phi(2x + k - N + 1) = \sum_k (-1)^{N-k-1} c_{N-k-1} \phi(2x - k) \quad (4.18) \]

This definition has been constructed so that the mother wavelet is orthogonal to the scaling function and its translates that is, \( \langle \psi(x) \phi(x - k) \rangle = 0 \forall k \in \mathbb{Z} \). Note also that it is normalised and its mean is zero, that is \( \langle \psi, \psi \rangle = 1 \) and \( \langle \psi, 1 \rangle = 0 \). Since the monomials \( 1, x, x^2, \ldots, x^p \) are to be exactly represented as a linear combination of these translates, the monomials must also be orthogonal to the mother wavelet. That is

\[ \int x^{p-1} \psi(x) \, dx = 0 \quad (4.19) \]

So the approximation condition is equivalent to the condition that the first \( p \) moments of the mother wavelet are zero. This can be shown to be equivalent to requiring that the coefficients satisfy the relation
\[ \sum_{k} -1^k k^{p-1} c_k = 0 \] (4.20)

where \( p = 1, 2, \ldots, \frac{N}{2} \), yielding another \( \frac{N}{2} \) conditions.

### 4.2.1.3 The Wavelet Basis

From the mother wavelet, we generate the remaining basis functions. We define

\[ \psi_{j,k}(x) = 2^{j/2} \psi(2^j x - k), k \in \mathbb{Z} \] (4.21)

The index \( j \) indicates the resolution we are at, and \( k \) indicates the translation. The prefactor is needed for normalisation, that is, so that

\[ < \psi_{j,k}, \psi_{j,k} > = 1 \forall j, k. \]

The basis thus consists of the scaling function (and its translates), the mother wavelet, and the remaining wavelets \( \psi_{j,k} \). Since the mother wavelet has zero mean, each \( \psi_{j,k} \) has zero mean. The 'average' or 'DC' part of a function, when expanded into a wavelet basis, is therefore contained in the coefficient of the scaling function. So, if we wish to expand a function \( f \) into a wavelet basis, the expansion is given by

\[ f(x) = a_k \phi(x - k) + b_{j,k} \psi_{j,k} \] (4.22)

and, since the basis is orthonormal, the expansion coefficients are given by

\[ a_k = \int f(x) \phi(x - k) dx \]
\[ b_{j,k} = \int f(x) \psi_{j,k} dx \] (4.23)

The wavelet bases defined in the above analysis are known as DN bases. This notation honours Ingrid Daubechies [91], the creator of orthogonal wavelets of compact support with \( N \) here denoting the number of nonzero coefficients.

### 4.2.1.4 The Biorthogonal Wavelets

The Biorthogonal wavelet transform is an invertible transform which provides additional degrees of freedom so that both perfect reconstruction and linear-phase filters can be realized simultaneously. There exist two sets of lowpass filters (for reconstruction), and highpass filters (for decomposition). One set is the dual of the other.
The decomposition and reconstruction filters are obtained from two distinct scaling functions associated with two multiresolution analysis in duality for generation of biorthogonal wavelets. Another advantageous property of Biorthogonal over orthogonal wavelets is that they have higher embedding capacity if they are used to decompose the image into different channels. All the above mentioned properties make Biorthogonal wavelets promising in the watermarking domain.

In the Biorthogonal case, there are two scaling functions, which may generate different multiresolution analysis, and accordingly two different wavelet functions. The scaling sequences must satisfy the following biorthogonality condition.

For orthogonal wavelets, the scaled translate of the scaling function φ and mother wavelet ψ are denoted by

\[ φ(x) = \sqrt{2} \sum_{k} h_k φ(2x - k) \]  \hspace{1cm} (4.24)

\[ ψ(x) = \sqrt{2} \sum_{k} g_k φ(2x - k) \]  \hspace{1cm} (4.25)

For Biorthogonal wavelet, rather than a single scaling function there is a dual scaling function and mother wavelet.

\[ φ^ω_n(x) = 2^\frac{n}{2} φ(2^n x - k) \]  \hspace{1cm} (4.26)

\[ ψ^ω_k(x) = 2^\frac{n}{2} ψ(2^n x - k) \]  \hspace{1cm} (4.27)

In case of normal DWT decomposition, if the embedding rate becomes high, data imperceptibility becomes lower and robustness performance is also decreased. Interference may also occur as different sets of spreading codes (used for different watermark messages) are added with the decomposed cover image signal using single scaling function. Moreover, the decomposition does not always yield low correlation with the code patterns and high robustness may not be achieved.

This problem can be solved to a great extent, if image signal is decomposed properly in different directions, so that low correlation value with the code patterns can be satisfied. The correlation between the code patterns is calculated and the image decomposition coefficients obtained using several DWT and Biorthogonal DWT. It is observed that the Biorthogonal DWT provides lower correlation with the code patterns. This is possibly due to the
complementary information present in two wavelet systems that offers better directional selectivity compared to classical wavelet transform.

![Wavelet Decomposition Diagram]

**Fig 4.2:** Second Level Decomposition. LL Represents the Approximation Sub-Band, HL Represents the Horizontal Sub-Band, LH Represents the Vertical, and HH Represents Diagonal Sub-Band.

### 4.2.1.5 Lifting Wavelet Transform Using Biorthogonal Wavelets

As seen in the previous section, traditional wavelet constructions use the Fourier transform to build the space-frequency localization.

A new mathematical formulation proposed by Swelden [92], based on spatial construction of the wavelets is called the lifting-based wavelet transform. The underlying principle of this approach [92,93,94,95] 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 \( \tilde{h}(z) \) and \( \tilde{g}(z) \) 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

\[
h(z) = h_e(z^2) + z^{-1} h_o(z^2)
\]  

(4.28)

where \( h_e \) contains the even filter coefficients and \( h_o \) contains the odd filter coefficients of the FIR filter \( h \).

We can define the polyphase matrix for the filter \( h \) as
\[ P(z) = \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix} \]  \hspace{1cm} (4.29)

Similarly, the polyphase representation of the filters \( g(z), h(z) \) and \( \tilde{g}(z) \) is expressed as follows:

\[ g(z) = g_e(z^2) + z^{-1}g_o(z^2) \]

\[ \tilde{h}(z) = \tilde{h}_e(z^2) + z^{-1}\tilde{h}_o(z^2) \hspace{1cm} (4.30) \]

\[ \tilde{g}(z) = \tilde{g}_e(z^2) + z^{-1}\tilde{g}_o(z^2) \]

Based on the formulation in 4.30, the polyphase matrix representation of the filters can be given as follows:

\[ \tilde{P}(z) = \begin{bmatrix} \tilde{h}_e(z) & \tilde{h}_o(z) \\ \tilde{g}_e(z) & \tilde{g}_o(z) \end{bmatrix}, \quad P(z) = \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix} \]  \hspace{1cm} (4.31)

The two matrices i.e. \( P(z) \) and \( \tilde{P}(z) \) are called dual of each other. The two polyphase matrices must satisfy the following condition for perfect reconstruction:

\[ P(z)\tilde{P}(z^{-1})^T = I \hspace{1cm} (4.32) \]

**Fig 4.3:** Polyphase Representation of Wavelet Transform: 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.
When the determinant of $P(z)$ is unity, the synthesis filter bank $(h, g)$ is called complementary and so is the analysis filter pair $(\tilde{h}, \tilde{g})$. However, when $h(z) = \tilde{h}(z) = g(z) = \tilde{g}(z) = 1$, the DWT simply splits an input signal into two sub sequences, one with all the odd samples and one with all the even sequences. This is called the lazy wavelet transform. The lifting wavelet transform essentially means first applying the lazy wavelet transform on the input stream, then executing primal lifting and finally scaling the output streams to produce low pass and high pass subbands.

The lifting wavelet transform used in the present context has been strengthened by using biorthogonal wavelet transform. Designing wavelets using biorthogonal filters allows more degrees of freedoms than orthogonal wavelets. One additional degree of freedom is the possibility to construct symmetric wavelet functions. As discussed in the previous section, the property of perfect reconstruction and symmetric wavelet functions that exists in biorthogonal wavelets provide it with higher embedding capacity if they are used to decompose the image into different channels.

The correlation is calculated between the embedded watermarks and the image decomposition coefficients obtained using classical LWT and Biorthogonal LWT. The Biorthogonal LWT provides lower correlation with the embedded watermarks due to the complementary information present in two wavelet systems that offers better directional selectivity compared to classical transform.

### 4.2.2 Singular Value Decomposition

The Singular Value Decomposition was discovered over 120 years ago independently by Eugenio Beltrami and Camille Jordan as described in [98, 107-109]. The development in the 1960’s of practical methods of computing SVD transformed the field of numerical algebra. Since the 1970’s, the SVD has been used in an overwhelming number of applications. Theorem 2 as will be explained later in this section is one of the most important features of the SVD, as it is extremely useful in least squares approximations and Principal Component Analysis.

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 watermarking.
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 can be looked at from three mutually compatible points of view. On the one hand, it can be seen as a method for transforming correlated variables into a set of uncorrelated ones that better expose the various relationships among the original data items. At the same time, SVD is a method for identifying and ordering dimensions along which data points exhibit the most variations. This ties in to the third way of viewing SVD, which is that once points with maximum variation are identified, it’s possible to find the best approximation of the original data points using fewer dimensions. Hence, SVD can be seen as a method for data reduction.

As an illustration of these ideas, consider the 2-dimensional data points in fig 4.4 as shown using the blue line.

![Figure 4.4: Best-Fit Regression Line Reduces Data From Two Dimensions To One.](image)

The regression line running through them (depicted in orange color) shows the best approximation of the original data with a 1-dimensional object (a line). It is the best approximation in the sense that it is the line that minimizes the distance between each original point and the line. If a perpendicular line from each point is drawn to the regression line, and the intersection of those lines is taken as the approximation of the original datapoint, a
reduced representation of the original data that captures as much of the variation as possible is obtained. It is noteworthy that there is a second regression line perpendicular to the first, shown in Figure 4.5. This line captures as much variation as possible along the second dimension of the original data set. It does poorer job of approximating the original data because it corresponds to a dimension exhibiting less variation to begin with.

It is possible to use these regression lines to generate set of uncorrelated data points that will show subgroupings in the original data not necessarily visible at first glance.

![Figure 4.5: Regression Line Along Second Dimension Captures Less Variation In Original Data.](image)

These are the basic ideas behind SVD: taking a high dimensional, highly variable set of data points and reducing it to a lower dimensional space that exposes the substructure of the original data more clearly and orders it from the most variation to the least. One can take advantage of this fact and simply ignore variation below a particular threshold to massively reduce the data and at the same time preserving the main relationships of interest. 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.
4.2.2.1 Mathematical Explanation of Full Singular Value Decomposition

SVD is based on a theorem from linear algebra which performs optimal matrix decomposition in a least square sense packing the maximum signal energy into a few coefficients as possible [99].

**Theorem 1** Any matrix $A \in \mathbb{R}^{m \times n}$ can be factored into singular value decomposition

$$AV = US.$$  
(4.33)

$$A^T U = VS.$$  
(4.34)

$$A = USV^T.$$  
(4.35)

Where the matrix $U \in \mathbb{R}^{m \times m}$ is an orthogonal matrix ,

$$U = [u_1, u_2, \ldots, u_r, u_{r+1}, \ldots, u_m]$$  
(4.36)

column vectors $u_i$ , for $i = 1, 2, \ldots, m$ form an orthonormal set

$$u_i^T u_j = \delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$  
(4.37)

and matrix $V$ is an $n \times n$ orthogonal matrix

$$V = [v_1, v_2, \ldots, v_r, v_{r+1}, \ldots, v_n]$$  
(4.38)

column vectors $v_i$ , for $i = 1, 2, \ldots, n$ form an orthonormal set

$$v_i^T v_j = \delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$  
(4.39)

Here, $S$ is an $m \times n$ diagonal matrix with singular values $SV$ on the diagonal. The $p$ diagonal entries of matrix $S$ are usually denoted by $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_p \geq \sigma_p$.

Equation 4.35 can be rewritten as a sum of rank-1 matrices:

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$  
(4.40)

where $\sigma_i$ is the $i$th singular value and $u_i$ and $v_i$ are the $i$th columns of $U$ and $V$. The above
equation is useful when one wants to estimate $A$ using a matrix of lower rank

**Theorem 2** (Eckart-Young) Let the SVD of $A$ be given by equation 4.35. If $k < r = \text{rank}(A)$ and $A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$, then

$$\min_{\text{rank}(B)=k} ||A-B||_2 = ||A-A_k||_2 = \sigma_{k+1}$$

(4.41)

The proof of the theorem is presented in [100].

The full singular value decomposition of an $(M \times N)$ matrix involves an $(M \times M)$ $U$, an $(M \times N)$ $S$, and an $(N \times N)$ $V$. 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. Speaking specifically from image viewpoint, singular values are the luminance values of SVD image layer, changing these values slightly, do not affect the image quality much.

**4.2.2.2 Properties of the SVD**

- The singular values $\sigma_1, \sigma_2, \ldots, \sigma_r$ are unique, however, the matrices $U$ and $V$ are not unique.

- Since $AA^T = V \sigma^T \sigma V^T$, so $V$ diagonalizes $AA^T$, it follows that $v_i \sigma_i$ are the eigenvectors of $AA^T$.

- Since $AA^T = U \sigma \sigma^T U^T$, so it follows that $U$ diagonalizes $AA^T$ and that $u_i \sigma$ are the eigenvectors of $AA^T$.

- If the rank of matrix $A$ is $r$ then $v_j, v_{j+1}, \ldots, v_r$ form an orthonormal basis for range space of $A^T$, $R(A^T)$, and $u_j, u_{j+1}, \ldots, u_r$ form an orthonormal basis for range space $A, R(A)$.

- The rank of matrix $A$ is equal to the number of nonzero singular values.
4.3 Proposed Technique

An image comprises of certain high frequency components (edges) known as the detailed coefficients and low frequency components (smooth areas) known as the approximation coefficients. Most of the previous SVD and DWT-based watermarking techniques treat different parts of the image in the same way. Therefore, the edges and the smooth areas of the image, related to different sub-bands, accept similar effects. The HVS is less sensitive to noise on edges, hence making similar changes to perceptually significant and insignificant areas of the image consequently lead to noticeable alternation in smooth areas, thereby causing a significant degradation to the image quality.

This chapter proposes a novel biometric watermarking technique with imperceptible image quality alteration. Additional advantages of the presented technique could be highlighted as high capacity and robustness of the method against different types of common attacks. Since LWT provides high redundancy in transform domain, the high capacity of the transformed host has been utilized as the beneficial point to scatter the watermark data within the host data.

4.3.1 Biometric Feature Processing

To employ offline handwritten signature as watermark, the preprocessing algorithm as explained in chapter 2 is applied on the signature image. Initially, the signature image is binarized and resized to an image of 300 pixels x 200 pixels. This is to isolate single stroke or a cluster of separated strokes of a handwritten signature from the background. Median filter is applied to this binary image to eliminate noise which might be present in the form of speckles, smears, scratches etc. that might thwart feature extraction.

Hough transform (HT) is then applied to the signature image for projection into feature space. The step is followed by applying KL transform to compress the feature space generated by HT without losing the significant attributes. Lastly, KL features are statistically discretized into binary representation signature code.
4.3.2  Watermark Embedding

The following steps explain the embedding phase.

i. Let $I_{\text{original}}$ be the host image of size $N \times N$.

ii. The Lifting Wavelet Transform $I_{\text{lwt}(i,j)}$ of the host image is calculated according to the selected decomposition level (L), sub-bands of size $\frac{N}{2^L} \times \frac{N}{2^L}$ can be achieved.

iii. Let $S_{\text{original}}$ be the original offline handwritten signature of $m \times n$ where $m \leq n$. Resize the signature image such that size ($I_{\text{original}}$) = size ($S_{\text{original}}$)

iv. Calculate $S_{\text{lwt}(i,j)}$, the corresponding wavelet transform of the signature image.

v. At L= 2, apply SVD to the horizontal detailed sub-band of the cover image as well as to the signature image.

vi. The singular values of the cover image sub-band are modified with the singular values of the signature sub-band obtaining modified LWT coefficient at the 2nd level.

vii. \[
[l_{\text{wm}}(2, h)]_{\text{singular}} = \left[ I_{\text{original}}(2, h) \right]_{\text{singular}} + k * \left[ S_{\text{original}}(2, h) \right]_{\text{singular}}.
\]

(4.42)

viii. Embedding at this level is described as

ix. \[
I_{\text{lwt}}(2, j) = \begin{cases} 
S_{\text{lwt}}(2, h) \\
I_{\text{lwt}}(2, j)
\end{cases}.
\]

(4.43)

x. Using the inverse wavelet transformation the final watermarked image $I_{\text{watermark}}$ will be constructed.

4.3.3  Watermark Extraction

Since the SVs of the original images are needed in the extraction phase, the proposed technique is non-blind as it uses the singular vector matrices of the original signature image as the keys. The extraction phase is explained by the following steps.

i. Compute the Lifting Wavelet Transform of the watermarked image according to the selected decomposition level (L)

ii. Locate the embedded coefficients and extract the singular values of the corresponding sub-band of the signature image through Equation 10.

iii. \[
\Sigma S_{\text{wm}} = (l_{\text{wm}} - I_{\text{original}})/k
\]

(4.44)
iv. Combine the SVs thus obtained to recover the 2\textsuperscript{nd} level approximation coefficient.

v. Perform 2–level Inverse LWT to obtain the watermark.

### 4.3.4 Template Matching Based Authentication

This extracted watermark is fed as an input to the biometric feature processing algorithm for template matching. The database contains 250 offline handwritten signatures collected from 50 users at different times to capture the intrapersonal differences in signing by a single user. Initially all the steps mentioned in biometric feature processing are applied to the entire signature database to generate a feature vector comprising the feature vectors corresponding to each signature image. These steps are applied to the recovered signature image to extract its features. The Euclidean distance between the feature vector of the recovered signature and the feature vectors of all the signatures in the database is calculated according to the formula as given by.

\[
\text{dist}(x, y)(a, h) = \sqrt{(x - a)^2 + (y - h)^2}.
\] (4.45)

The database image with the least Euclidean distance with the extracted image is the corresponding template and hence the verification of the signature of the user.

### 4.4 Significance Measures

#### 4.4.1 Peak Signal to Noise Ratio (PSNR)

The proposed algorithm has been tested for various signal processing attacks like median filtering, salt and pepper noise addition, histogram equalization, Gaussian noise and JPEG compression. The experimental results have been gauged using Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR) which have been given below.

\[
\text{MSE} = \frac{1}{mn} \sum_{x=-0}^{m-1} \sum_{y=-0}^{n-1} (I(x, y) - W(x, y))^2.
\] (4.46)

where I and W are the original and the watermarked images having a resolution of m×n.

\[
\text{PSNR} = 10 \log_{10} \frac{\text{max}^2}{\text{MSE}}.
\] (4.47)
4.4.2 Embedding Factor Optimization

The embedding factor $k$ is of vital significance to any watermarking scheme. Its value has a direct implication on the following two factors

- Robustness: As $k$ increases, the algorithm becomes more resilient to attacks and hence more robust.

- Imperceptibility: As $k$ increases, the image becomes more distorted and hence imperceptibility is not achieved.

Due to this trade-off, a multi objective optimization technique is used to select appropriate strength factor so as to maintain imperceptibility with improved attack resilience. To show the effect of $k$ on image distortions, the performance evaluation of the method has been done using image quality determination metric Structural Similarity Index Measure (SSIM) [89]. While PSNR is a commonly used metric, SSIM index is an advanced method for measuring the similarity between two images. SSIM is designed to improve on traditional methods like PSNR and MSE by modelling image distortion as a combination of three factors considering the properties of Human Visual System (HVS). These three factors are loss of correlation, luminance distortion and contrast distortion.

4.4.2.1 Structural Similarity Index Measure (SSIM)

Let $I_{original} = x$ be the original image and $I_{wm} = y$ be the manipulated (either watermarked or attacked image). If one of the signals has perfect quality, then the similarity measure can serve as a quantitative measurement of the quality of the second signal. The system separates the task of similarity measurement into three comparisons: luminance, contrast and structure. First, the luminance of each signal is compared. Assuming discrete signals, this is estimated as the mean intensity:

$$
\mu_x = \frac{1}{N} \sum_{t=1}^{N} x_t
$$

(4.48)

The luminance comparison function $l(x, y)$ is then a function $\mu_x$ and $\mu_y$. Second, we remove the mean intensity from the signal. In discrete form, the resulting signal $x - \mu_x$ corresponds to the projection of vector $x$ onto the hyperplane defined by
\[ \sum_{i=1}^{N} x_i = 0 \] (4.49)

The standard deviation (the square root of variance) is used as an estimate of the signal contrast. An unbiased estimate in discrete form is given by

\[ \sigma_x = \left( \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \mu_x)^2 \right)^{1/2} \] (4.50)

The contrast comparison \( c(x, y) \) is then the comparison of \( \sigma_x \) and \( \sigma_y \).

Third, the signal is normalized (divided) by its own standard deviation; so that the two signals being compared have unit standard deviation. The structure comparison \( s(x, y) \) is conducted on these normalized signals \( (x - \mu_x)/\sigma_x \) and \( (y - \mu_y)/\sigma_y \).

Finally, the three components are combined to yield an overall similarity measure:

\[ S(x, y) = f(l(x,y); c(x,y); s(x,y)) \] (4.51)

An important point is that the three components are relatively independent. For example, the change of luminance and/or contrast will not affect the structures of images. In order to complete the definition of the similarity measure in Eq. 4.51, the three functions \( l(x,y), c(x,y), s(x,y) \), as well as the combination function \( f(.) \) need to be defined.

For luminance comparison, we define

\[ l(x,y) = \frac{2\mu_x \mu_y + C_1}{\mu^2_x + \mu^2_y + C_1} \] (4.52)

where the constant \( C_1 \) is included to avoid instability when \( \mu^2_x + \mu^2_y \) is very close to zero. Specifically,

\[ C_1 = (K_1 L)^2 \] (4.53)

where \( L \) is the dynamic range of the pixel values (255 for 8-bit grayscale images), and \( K_1 \ll 1 \) is a small constant.

HVS is sensitive to the relative luminance change, and not the absolute luminance change.
Letting $R$ represent the size of luminance change relative to background luminance, the luminance of the distorted signal can be represented as $\mu_y = (1 + R)\mu_x$. Substituting this into 4.52 gives

$$
I(x, y) = \frac{2(1 + R)}{1 + (1 + R)^2 + \frac{C_1}{\mu_x^2}}
$$

(4.54)

If we assume $C_1$ is small enough (relative to $\mu_x^2$) to be ignored, then $I(x, y)$ is a function only of $R$, qualitatively consistent with Weber's law.

The contrast comparison function takes a similar form:

$$
c(x, y) = \frac{2 \sigma_x \sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}
$$

(4.55)

where $C_2 = (K_2 L)^2$, and $K_2 \ll 1$.

The structure comparison function is defined as follows:

$$
s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x \sigma_y + C_3}
$$

(4.56)

Finally, the three comparisons of Eqs. (4.54), (4.55) and (4.56) and name the resulting similarity measure the Structural SIMilarity (SSIM) index between signals $x$ and $y$:

$$
SSIM(x, y) = [I(x, y)]^\alpha [c(x, y)]^\beta [s(x, y)]^\gamma
$$

(4.57)

The SSIM quality index is defined as

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

(4.58)

$\mu_x, \mu_y, \sigma_x^2, \sigma_y^2$ are the mean value and variances of $x$ and $y$ respectively and

$$
\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y)
$$

(4.59)

$N$ = Number of samples. Parameters $C_1$ and $C_2$ are defined as
$l$ is the dynamic range of the pixel values (for gray scale images it ranges from 0-255) and $K_1, K_2 << 1$ are small constants. The standard value for $K_1$ is taken as 0.01 while for $K_2$, it is 0.03. The image perceptual comparison is best when the size of $x$ and $y$ is the same. In practice, one usually requires a single overall quality measure of the entire image. We use a mean SSIM (MSSIM) index to evaluate the overall image quality.

$$MSSIM = \frac{1}{M} \sum_{i=1}^{M} SSIM(x_i, y_i)$$ \hspace{1cm} (4.60)

Thus the parameter optimization will be achieved using two objective functions $f_i(k) = 1 - SSIM(k)$ and $f_r(k)$. $f_i(k)$ shows the effect of $k$ on imperceptibility while $f_r(k)$ shows the impact of $k$ on attack resilience and is calculated as the error probability function described as

$$P_{error} = \frac{1}{2} P_{error|0} + \frac{1}{2} P_{error|1}$$ \hspace{1cm} (4.61)

Both the functions have a contrasting nature while the imperceptibility increases monotonically with $k$, robustness is monotonically decreasing. Thus the aim is to find an optimum value of $k$ so as to minimize both the functions. The goal attainment method given in [96] provides a solution to find optimum $k$ value according to SSIM which gives satisfactory level of attack resilience. In this approach the value $0.5 < k < 0.7$ is found to be optimal for most of the images.

### 4.5 Results and Discussions

Table 4.1 shows that the PSNR between the original and the recovered watermark varies between 53 dB to 60 dB for the signatures of 30 users when the watermarked image is not subjected to any attack. In Table 4.2, it can be seen that the PSNR value of recovered watermarks (for 5 users) after the watermarked image is subjected to various attacks varies between 50 dB to 55 dB. The watermarked image was subjected to non-geometric attacks such as cropping, histogram equalization, median filtering (with varying filter lengths), salt and pepper noise (with varying variance), Gaussian filtering, Weiner filtering, sharpening and JPEG compression. In all the cases even after the attack, the watermarks could be recovered because the algorithm embeds the watermark to the most significant image coefficients thus ensuring that degradation to detailed image coefficients does not affect the watermarks.
### Table 4.1 Extracted Watermarks

<table>
<thead>
<tr>
<th>PSNR, MSSIM</th>
<th>PSNR, MSSIM</th>
<th>PSNR, MSSIM</th>
<th>PSNR, MSSIM</th>
<th>PSNR, MSSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>54, 0.99886</td>
<td>53, 0.99826</td>
<td>54, 0.99909</td>
<td>60, 0.99918</td>
<td>55, 0.99903</td>
</tr>
<tr>
<td><img src="image1.png" alt="Signature 1" /></td>
<td><img src="image2.png" alt="Signature 2" /></td>
<td><img src="image3.png" alt="Signature 3" /></td>
<td><img src="image4.png" alt="Signature 4" /></td>
<td><img src="image5.png" alt="Signature 5" /></td>
</tr>
<tr>
<td>54, 0.9983</td>
<td>54, 0.99863</td>
<td>53, 0.99929</td>
<td>55, 0.9991</td>
<td>54, 0.99818</td>
</tr>
<tr>
<td><img src="image6.png" alt="Signature 6" /></td>
<td><img src="image7.png" alt="Signature 7" /></td>
<td><img src="image8.png" alt="Signature 8" /></td>
<td><img src="image9.png" alt="Signature 9" /></td>
<td><img src="image10.png" alt="Signature 10" /></td>
</tr>
<tr>
<td>59, 0.99949</td>
<td>54, 0.99969</td>
<td>54, 0.9994</td>
<td>55, 0.9997</td>
<td>54, 0.99937</td>
</tr>
<tr>
<td><img src="image11.png" alt="Signature 11" /></td>
<td><img src="image12.png" alt="Signature 12" /></td>
<td><img src="image13.png" alt="Signature 13" /></td>
<td><img src="image14.png" alt="Signature 14" /></td>
<td><img src="image15.png" alt="Signature 15" /></td>
</tr>
<tr>
<td>55, 0.9995</td>
<td>55, 0.99949</td>
<td>57, 0.9998</td>
<td>53, 0.99928</td>
<td>53, 0.99865</td>
</tr>
<tr>
<td><img src="image16.png" alt="Signature 16" /></td>
<td><img src="image17.png" alt="Signature 17" /></td>
<td><img src="image18.png" alt="Signature 18" /></td>
<td><img src="image19.png" alt="Signature 19" /></td>
<td><img src="image20.png" alt="Signature 20" /></td>
</tr>
<tr>
<td>54, 0.99922</td>
<td>53, 0.99853</td>
<td>54, 0.99914</td>
<td>54, 0.99942</td>
<td>55, 0.99877</td>
</tr>
<tr>
<td><img src="image21.png" alt="Signature 21" /></td>
<td><img src="image22.png" alt="Signature 22" /></td>
<td><img src="image23.png" alt="Signature 23" /></td>
<td><img src="image24.png" alt="Signature 24" /></td>
<td><img src="image25.png" alt="Signature 25" /></td>
</tr>
<tr>
<td>PSNR=55</td>
<td>PSNR=55</td>
<td>PSNR=54</td>
<td>PSNR=55</td>
<td>PSNR=53</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>MSSIM=0.9994</td>
<td>MSSIM=0.99874</td>
<td>MSSIM=0.99806</td>
<td>MSSIM=0.99946</td>
<td>MSSIM=0.99713</td>
</tr>
<tr>
<td>![Signature 1]</td>
<td>![Signature 2]</td>
<td>![Signature 3]</td>
<td>![Signature 4]</td>
<td>![Signature 5]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PSNR=55</th>
<th>PSNR=60</th>
<th>PSNR=53</th>
<th>PSNR=56</th>
<th>PSNR=55</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSSIM=0.9998</td>
<td>MSSIM=0.99948</td>
<td>MSSIM=0.99844</td>
<td>MSSIM=0.9997</td>
<td>MSSIM=0.9999</td>
</tr>
<tr>
<td>![Signature 6]</td>
<td>![Signature 7]</td>
<td>![Signature 8]</td>
<td>![Signature 9]</td>
<td>![Signature 10]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PSNR=53</th>
<th>PSNR=54</th>
<th>PSNR=56</th>
<th>PSNR=54</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSSIM=0.99791</td>
<td>MSSIM=0.99886</td>
<td>MSSIM=0.99953</td>
<td>MSSIM=0.99885</td>
</tr>
<tr>
<td>![Signature 11]</td>
<td>![Signature 12]</td>
<td>![Signature 13]</td>
<td>![Signature 14]</td>
</tr>
</tbody>
</table>

Table 4.2: Recovered Watermarks (For 5 Users) After The Watermarked Image Is Subjected To Various Attacks.

<table>
<thead>
<tr>
<th>CROPPING</th>
<th>HISTOGRAM</th>
<th>MEDIAN</th>
<th>SALT &amp; PEPPER</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR=51</td>
<td>PSNR=28</td>
<td>PSNR=30</td>
<td>PSNR=29</td>
</tr>
<tr>
<td>MSSIM=0.99808</td>
<td>MSSIM=0.87914</td>
<td>MSSIM=0.97102</td>
<td>MSSIM=0.82041</td>
</tr>
<tr>
<td>![Cropping 1]</td>
<td>![Histogram 1]</td>
<td>![Median 1]</td>
<td>![Salt &amp; Pepper 1]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GAUSSIAN</th>
<th>WEINER</th>
<th>SHARPENING</th>
<th>JPEG</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR=26</td>
<td>PSNR=36</td>
<td>PSNR=24</td>
<td>PSNR=44</td>
</tr>
<tr>
<td>MSSIM=0.76158</td>
<td>MSSIM=0.99583</td>
<td>MSSIM=0.82852</td>
<td>MSSIM=0.98957</td>
</tr>
<tr>
<td>![Gaussian 1]</td>
<td>![Weiner 1]</td>
<td>![Sharpening 1]</td>
<td>![JPEG 1]</td>
</tr>
</tbody>
</table>

(a) User 1
<table>
<thead>
<tr>
<th>CROPPING</th>
<th>HISTOGRAM</th>
<th>MEDIAN</th>
<th>SALT &amp; PEPPER</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR=51</td>
<td>PSNR=30</td>
<td>PSNR=32</td>
<td>PSNR=30</td>
</tr>
<tr>
<td>MSSIM=0.99814</td>
<td>MSSIM=0.88515</td>
<td>MSSIM=0.97036</td>
<td>MSSIM=0.83164</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GAUSSIAN</th>
<th>WEINER</th>
<th>SHARPENING</th>
<th>JPEG</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR=27</td>
<td>PSNR=38</td>
<td>PSNR=26</td>
<td>PSNR=46</td>
</tr>
<tr>
<td>MSSIM=0.76587</td>
<td>MSSIM=0.99145</td>
<td>MSSIM=0.83901</td>
<td>MSSIM=0.99247</td>
</tr>
</tbody>
</table>

(b) User 2

<table>
<thead>
<tr>
<th>CROPPING</th>
<th>HISTOGRAM</th>
<th>MEDIAN</th>
<th>SALT &amp; PEPPER</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR=51</td>
<td>PSNR=28</td>
<td>PSNR=31</td>
<td>PSNR=29</td>
</tr>
<tr>
<td>MSSIM=0.99781</td>
<td>MSSIM=0.85022</td>
<td>MSSIM=0.96537</td>
<td>MSSIM=0.79077</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GAUSSIAN</th>
<th>WEINER</th>
<th>SHARPENING</th>
<th>JPEG</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR=26</td>
<td>PSNR=37</td>
<td>PSNR=24</td>
<td>PSNR=44</td>
</tr>
<tr>
<td>MSSIM=0.70474</td>
<td>MSSIM=0.98374</td>
<td>MSSIM=0.78463</td>
<td>MSSIM=0.99399</td>
</tr>
</tbody>
</table>

(c) User 3
<table>
<thead>
<tr>
<th><strong>CROPPING</strong></th>
<th><strong>HISTOGRAM</strong></th>
<th><strong>MEDIAN</strong></th>
<th><strong>SALT &amp; PEPPER</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR=50</td>
<td>PSNR=30</td>
<td>PSNR=30</td>
<td>PSNR=30</td>
</tr>
<tr>
<td>MSSIM=0.99948</td>
<td>MSSIM=0.97916</td>
<td>MSSIM=0.99067</td>
<td>MSSIM=0.97404</td>
</tr>
<tr>
<td><img src="image1" alt="Image" /></td>
<td><img src="image2" alt="Image" /></td>
<td><img src="image3" alt="Image" /></td>
<td><img src="image4" alt="Image" /></td>
</tr>
<tr>
<td><strong>GAUSSIAN</strong></td>
<td><strong>WEINER</strong></td>
<td><strong>SHARPENING</strong></td>
<td><strong>JPEG</strong></td>
</tr>
<tr>
<td>PSNR=28</td>
<td>PSNR=37</td>
<td>PSNR=26</td>
<td>PSNR=44</td>
</tr>
<tr>
<td>MSSIM=0.95668</td>
<td>MSSIM=0.99871</td>
<td>MSSIM=0.96388</td>
<td>MSSIM=0.99755</td>
</tr>
<tr>
<td><img src="image5" alt="Image" /></td>
<td><img src="image6" alt="Image" /></td>
<td><img src="image7" alt="Image" /></td>
<td><img src="image8" alt="Image" /></td>
</tr>
</tbody>
</table>

(d) User 4

<table>
<thead>
<tr>
<th><strong>CROPPING</strong></th>
<th><strong>HISTOGRAM</strong></th>
<th><strong>MEDIAN</strong></th>
<th><strong>SALT &amp; PEPPER</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR=52</td>
<td>PSNR=28</td>
<td>PSNR=30</td>
<td>PSNR=29</td>
</tr>
<tr>
<td>MSSIM=0.99834</td>
<td>MSSIM=0.86367</td>
<td>MSSIM=0.96734</td>
<td>MSSIM=0.80566</td>
</tr>
<tr>
<td><img src="image9" alt="Image" /></td>
<td><img src="image10" alt="Image" /></td>
<td><img src="image11" alt="Image" /></td>
<td><img src="image12" alt="Image" /></td>
</tr>
<tr>
<td><strong>GAUSSIAN</strong></td>
<td><strong>WEINER</strong></td>
<td><strong>SHARPENING</strong></td>
<td><strong>JPEG</strong></td>
</tr>
<tr>
<td>PSNR=26</td>
<td>PSNR=36</td>
<td>PSNR=24</td>
<td>PSNR=45</td>
</tr>
<tr>
<td>MSSIM=0.74643</td>
<td>MSSIM=0.99469</td>
<td>MSSIM=0.81329</td>
<td>MSSIM=0.99002</td>
</tr>
<tr>
<td><img src="image13" alt="Image" /></td>
<td><img src="image14" alt="Image" /></td>
<td><img src="image15" alt="Image" /></td>
<td><img src="image16" alt="Image" /></td>
</tr>
</tbody>
</table>

(e) User 5
**Fig. 4.6:** PSNR versus Embedding Factor for Various Cover Images.

**Fig. 4.7:** MSSIM versus Embedding Factor for Various Cover Images.
Figures 4.6 and 4.7 show the effect of varying the embedding factor, the significance of which has been explained in section 4.4.2. It is evident that lesser detailed is the image like peppers and cameraman, more is the visible distortion as we keep increasing the embedding factor. In such cases watermark begins to become a part of the approximation coefficients and it’s effect becomes more pronounced than that in images with more details like Lena and Mandril.

![Embedding at level 1 wavelet decomposition](image)

<table>
<thead>
<tr>
<th></th>
<th>LENA</th>
<th>CAMERAMAN</th>
<th>MANDRIL</th>
<th>PIRATE</th>
<th>PEPPERS</th>
<th>GOLDHILL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 1</td>
<td>45</td>
<td>48</td>
<td>53</td>
<td>45</td>
<td>42</td>
<td>41</td>
</tr>
</tbody>
</table>

(a)

![Embedding at level 2 wavelet decomposition](image)

<table>
<thead>
<tr>
<th></th>
<th>LENA</th>
<th>CAMERAMAN</th>
<th>MANDRIL</th>
<th>PIRATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 2</td>
<td>54</td>
<td>51</td>
<td>57</td>
<td>55</td>
</tr>
</tbody>
</table>

(b)

Fig. 4.8: PSNR versus Various Wavelet Decomposition Levels (a) level 1 (b) level 2
Figures 4.8 and 4.9 show the effect of varying the decomposition level on various host images. As the decomposition level is increased i.e when the image is first decomposed into one approximation and three detailed bands, and then the approximation band is further decomposed to obtain second level approximation and detailed coefficients, the PSNR as well as MSSIM improves because the watermark is now embedded to further significant coefficients thus reducing the overall degradation to the image quality.

4.6 Conclusion

In this chapter, a novel biometric watermarking scheme using LWT-SVD for offline handwritten signature has been discussed. The proposed technique shows superior results as compared to the existing technique. The work can be further expanded by incorporating the latest signature verification techniques so as to reduce the FAR or FRR of the proposed system and also amalgamate the two areas of biometric watermarking and signature authentication/verification.