CHAPTER 5

IMPROVING ATTACK RESILIENCE OF BIOMETRIC (IRIS) WATERMARKING AUTHENTICATION THROUGH ZERNIKE FEATURE EXTRACTION

5.1 Introduction

This chapter presents an extension to the work presented in previous chapter, where offline handwritten signature was used as the biometric and minimum distance classifier was used for template matching.

The chapter extends the algorithm proposed previously by improving attack resilience through Zernike moment based feature extraction for template matching. The use of Iris as a watermark provides a further secure level of authentication as it includes distinct physical characteristics of a person. Signature is a behavioral biometric, learned and acquired over time. In a sense, the genetically and environmentally determined muscle dexterity of the hand is translated into a visual and machine-readable token which is more prone to change with time, mood, age, illness or emotion as well. Iris on the other hand is a physiological biometric rich enough that a onetime sample may suffice for comparing biometric identifiers. It provides a fine edge as its fine and intricate structure allows a high level of accuracy thus lowering the false matching rates. Iris has been purported to be a universal identifier with good discriminating properties.

5.2 Theoretical Concept Used

5.2.1 Zernike moments

The statistical expectation of certain power function of a random variable is called moment, the most common being the mean. Other generally used moments are variance, standard deviation etc. Khotanzad and Hong [97] introduced the use of Zernike moments for the recognition of shapes in digital binary images. The orthogonality property of the Zernike moments, which simplifies the process of image reconstruction, make the suggest feature selection approach practical. The basic reason of using Zernike Moment here is due to its significant advantage of being robust to noise, rotation invariance and higher computational
efficiency. Zernike moments are calculated by mapping an image function onto a basis set of orthogonal functions which is nothing but a unit disc in the polar coordinates. The basis function of Zernike Moment is calculated by

\[ V_{n,m}(x, y) = R_{n,m}(\rho)e^{im\theta} \]  \hspace{1cm} (5.1)

where we have \( n \) as the order, \( m \) as the repetition, both \( m \) and \( n \) are non-negative integers following the condition \( n - |m| \) is even and greater than equal to 0, \( \rho \) is the vector length of \((x, y)\) from the origin and \( \theta \) is the angle between the vector \( \rho \) and the \( x \)-axis taken counterclockwise. \( R \) is the radial polynomial given by

\[ R_{n,m}(\rho) = \sum_{k=|m|, (n-k)=even}^{n} \frac{(-1)^{n-k/2} \frac{n+k}{2} \frac{k+m, k-m}{2}}{2} \]  \hspace{1cm} (5.2)

Following this, we calculate Zernike moment as

\[ Z_{n,m} = \frac{n+1}{\pi} \sum \sum_{x^2+y^2 \leq 1} f(x, y)V_{n,m}^*(x, y) \]  \hspace{1cm} (5.3)

### 5.3 Proposed Technique

The Human Visual System (HVS) due to its limited sensitivity does not clearly see the periphery regions and is unable to discriminate between signals with an infinite precision. Exploiting this fact, the section describes the method which embeds the biometric watermark (Iris) in such a manner that it does not affect the perceptible image quality and the user remains unaware of the presence of the watermark.

#### 5.3.1 Watermark Embedding

A generalized block diagram that is representative of the embedding algorithm is seen in Figure 5.1.
Consider the host image $H_{original}$ of size $N \times N$ to be watermarked with Iris image of the user $I_{user}$ of size $P \times Q$. After resizing the iris image to $N \times N$, a 2-level 2D Lifting Wavelet Transform decomposition using biorthogonal filters is performed on the host as well as the Iris image $H_{lwt}(i,j)$ and $I_{lwt}(i,j)$ where $i$ denotes the decomposition level and $j$ denotes the wavelet sub-band. Both the images are first subjected to the first level LWT to obtain one approximate ($H_{original}(1,a)$) and three detailed ($H_{original}(1,c), H_{original}(1,v)$ and ($H_{original}(1,d)$) sub-bands for the host image and similarly, $I_{user}(1,a), I_{user}(1,c), I_{user}(1,d), I_{user}(1,v)$ for the Iris image. The LWT approximate band represents the coarse region comprising of the low frequency components of an image whereas the detailed coefficients denote the finest domain that is occupied by middle and high frequency coefficients. For the purpose of further sub-sampling, ($H_{original}(1,a)$) and $I_{user}(1,a)$ are selected, as the further sub-sampling of detailed coefficients is prohibited. On decomposing the images second time, the detailed coefficients $H_{original}(2,c)$ and $I_{user}(2,c)$ are used for further processing. SVD is applied independently to these sub-bands of the cover image as well the Iris image.

\[
[H_{original}(2,c)]_{singular}, [H_{original}(2,c)]_{singular}, [H_{original}(2,c)]_{singular}
\]
\[ SVD(H_{\text{original}}(2,c)) \]  
(5.4)

\[ [I_{\text{user}}(2,c)]_{\text{singular}}, [I_{\text{user}}(2,c)]_{\text{singular}}, [I_{\text{user}}(2,c)]_{\text{singular}} = SVD(I_{\text{user}}(2,c)) \]  
(5.5)

Next the singular values of the cover image sub-band are modified with the singular values of the Iris image sub-band, thus obtaining the modified LWT coefficient at the 2nd level using an embedding factor \( k \) which determines the extent of embedding.

\[ D = [H_{\text{original}}(2,c)]_{\text{singular}} + k \cdot [I_{\text{user}}(2,c)]_{\text{singular}}. \]  
(5.6)

\[ [H_{\text{intermediate}}(2,c)]_{\text{singular}}, [H_{\text{intermediate}}(2,c)]_{\text{singular}}, [H_{\text{intermediate}}(2,c)]_{\text{singular}} = SVD(D) \]  
(5.7)

\[ H_{\text{watermarked}}(2,c) = \]  
\[ ([H_{\text{original}}(2,c)]_{\text{singular}} \cdot [H_{\text{intermediate}}(2,c)]_{\text{singular}} [H_{\text{original}}(2,c)]_{\text{singular}})^{-1} \]  
(5.8)

\[ H_{\text{watermarked}}(1,a) = \]  
\[ llwt(H_{\text{original}}(2,a), H_{\text{watermarked}}(2,c), H_{\text{original}}(2,v), H_{\text{original}}(2,d)) \]  
(5.9)

Using the inverse wavelet transformation the final watermarked image \( H_{\text{watermark}} \) will be constructed.

\[ H_{\text{watermarked}} = llwt(H_{\text{watermarked}}(1,a), H_{\text{original}}(1,c), H_{\text{original}}(1,v), H_{\text{original}}(1,d)) \]  
(5.10)

### 5.3.2 Watermark Extraction

This scheme of watermarking is non-blind as it requires the original host image as a key for extraction of the watermark. The extraction scheme is described in Figure 5.2.

The extraction process begins by applying the lifting wavelet transform on the watermarked image to obtain it’s corresponding sub-bands

\[ (H_{\text{watermarked}}(1,a), H_{\text{watermarked}}(1,c), H_{\text{watermarked}}(1,v), H_{\text{watermarked}}(1,d)) = lwt2((H_{\text{watermarked}})) \]  
(5.11)

\[ (H_{\text{watermarked}}(2,a), H_{\text{watermarked}}(2,c), H_{\text{watermarked}}(2,v), H_{\text{watermarked}}(2,d)) = lwt2(H_{\text{watermarked}}(1,a)) \]  
(5.12)

Then the singular value decomposition of the 2nd level horizontal detailed coefficient is performed to obtain the corresponding singular values.
\[
([H_{\text{watermarked}}(2,c)]_{\text{singular}}', [H_{\text{watermarked}}(2,c)]_{\text{singular}} , [H_{\text{watermarked}}(2,c)]_{\text{singular}}') = \text{SVD} \ (H_{\text{watermarked}}(2,c))
\] (5.13)

Using the same embedding factor as was used for embedding, an intermediate value is calculated

\[
B = ([H_{\text{watermarked}}(2,c)]_{\text{singular}} - [H_{\text{original}}(2,c)]_{\text{singular}}') / k
\] (5.14)

The watermark is recovered by applying inverse transform

\[
l_{\text{rec.user}}(2,c) = [l_{\text{user}}(2,c)]_{\text{singular}} * B * [l_{\text{user}}(2,c)]_{\text{singular}}'
\] (5.15)

\[
l_{\text{rec.user}}(1,a) = i\text{lwt2}((l_{\text{user}}(2,a), l_{\text{rec.user}}(2,c), l_{\text{user}}(2,d), l_{\text{user}}(2,v))
\] (5.16)

\[
l_{\text{rec.user}} = i\text{lwt2}((l_{\text{rec.user}}(1,a), l_{\text{user}}(1,c), l_{\text{user}}(1,d), l_{\text{user}}(1,v))
\] (5.17)

The image is finally resized to it's original size

\[
l_{\text{rec.user}} = \text{resize}(l_{\text{rec.user}}, [P, Q])
\] (5.18)

---

**Fig 5.2: Watermark Extraction Scheme**
5.3.3 Zernike Feature Extraction Based Matching Technique

The central idea to use Zernike as a feature extractor is to extract the features from a constraint set of image values i.e.

\[
g(\rho, \theta) = \{l_{\text{original}} = f(x, y): \rho = \sqrt{x^2 + y^2} \text{ and } \theta = \tan^{-1}\left(\frac{y}{x}\right)\} \text{ then}
\]

\[Z_{n,m} = \{h(g(\rho, \theta)): \rho \leq 1\}\]  

(5.19)

The proposed algorithm for calculating Zernike Moment is as follows:

- The image is converted to a binarized image.
- The resultant image is mapped onto a unit disc in polar coordinates in such a way so that the center of the image coincides with the center of the unit disc. Our calculations are limited to the area within this unit circle. The function used here will be mapped as

\[
g(\rho, \theta) = \{f(x, y): \rho = \sqrt{x^2 + y^2}\}
\]

(5.20)

- The image is then normalized by re-sampling the image to a preset size.
- If \(\max(g(\rho, \theta)) = m_1\) and \(\min(g(\rho, \theta)) = m_2\), then the normalization function \(n(\rho, \theta)\) will be

\[
n(\rho, \theta) = (g(\rho, \theta) - m_2)/m_1
\]

(5.21)

The basis function is calculated with the help of radial polynomial. Zernike Moment is obtained by projecting image function on this polynomial and the magnitudes thus obtained represent the features of the input image.

The extracted iris image is subjected to the procedure of extraction of features for template matching. The database used here contains the iris images of 100 persons, 6 images of each-3 of left eye and 3 of right eye. The procedure as described in previous chapter is used for matching using Euclidean distance classifier. Euclidean distance between the feature vectors of two images is calculated by the formula:

\[
dist(a, b)(c, d) = \sqrt{(a - c)^2 + (b - d)^2}
\]

(5.22)
5.4 Results and Discussion

5.4.1 Robustness Test

There can be two types of operations to check for robustness. First is a gray scale manipulation like filtering, noise adding, lossy compression, gamma correction, color quantization, etc. and second is geometric transformations like scaling, cropping, rotation, affine transforms, etc. This algorithm proposes a technique to check the image robustness against a variety of robustness attacks. Figure 5.3 shows the PSNR between the original and the recovered watermark varies between 60 dB to 66 dB for the watermarked images watermarked with iris images of 12 users when the watermarked image is not subjected to any attack.

![PSNR Graph](image1)

![MSSIM Graph](image2)

**Fig 5.3:** PSNR and SSIM Values Calculated for Watermarked Lena, Baboon and Peppers using Iris Images of Different Persons as Watermarks.
In Figure 5.4, it can be seen that the PSNR value of recovered watermarks (for 3 users) after the watermarked image is subjected to various attacks varies between 40 dB to 45 dB which is quite a significant improvement as compared to the results obtained in the previous chapter where no special feature extraction technique was used for template matching.

![ Attacks](image)

**Fig 5.4:** Percentage Survival of the Watermarked Images when subjected to Attacks.

![ Lossy JPEG Compression](image)

**Fig 5.5:** Percentage Survival of Image when subjected to JPEG Compression Attack with Quality Factor Varying from 10 to 50.
Figure 5.5 shows that even after subjecting the watermarked image to a JPEG compression ratio ranging between 90% to 30%, the watermark recovery is pretty good. For various noise ratios between 10% to 20%, the PSNR varies between 30 dB to 40 dB.

5.5 Conclusion

The algorithm discussed in the present chapter has shown significant improvement in watermark identification as compared to the method discussed in chapter 4. The use of lifting wavelet transform provides a higher correlation between the recovered watermark and the original watermark. The authentication algorithm can be further strengthened using neural networks, owing to its ability to compare features of images more accurately.