Geometrically invariant image watermarking using Polar Harmonic Transforms

Leida Li a,*, Shushang Li a, Ajith Abraham b, Jeng-Shyang Pan c

a School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China
b Machine Intelligence Research Labs, Scientific Network for Innovation and Research Excellence, Auburn, WA 98071-2259, USA
c Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung 807, Taiwan

A R T I C L E   I N F O

Article history:
Received 31 January 2011
Received in revised form 24 December 2011
Accepted 26 February 2012
Available online 8 March 2012

Keywords:
Digital watermark
Invariant moment
Polar Harmonic Transform

A B S T R A C T

This paper presents an invariant image watermarking scheme by introducing the Polar Harmonic Transform (PHT), which is a recently developed orthogonal moment method. Similar to Zernike moment (ZM) and pseudo-Zernike moment (PZM) approaches, PHT is defined on a circular domain. The magnitudes of PHTs are invariant to image rotation and scaling. Furthermore, the PHTs are free of numerical instability, so they are more suitable for watermarking. In this paper, the invariant properties of PHTs are investigated. During embedding, a subset of the accurate PHTs are modified according to the binary watermark sequence. Then a compensation image is formatted by reconstructing the modified PHT vector. The final watermarked image is obtained by adding the compensation image to the original image. In the decoder, the watermark can be retrieved from the magnitudes of the PHTs directly. Experimental results illustrate that the proposed scheme outperforms ZM/PZM based schemes in terms of embedding capacity and watermark robustness and is also robust to both geometric and signal processing based attacks.

© 2012 Elsevier Inc. All rights reserved.

1. Introduction

Information hiding plays an important role in multimedia content protection. During the past few years, several information hiding techniques have been investigated extensively and great achievements have been obtained, including robust watermarking [15,22], fragile watermarking [6], reversible watermarking [7,34] and steganography [19,33]. This paper focuses on robust watermarking. A good watermarking scheme should be able to retrieve the watermark even when the image is attacked by geometric distortions, such as image rotation and scaling. In order to achieve this goal, extensive research has been done, which can be grouped into the following three categories [24,40]: (1) Inverse transform. The idea is to transform the distorted image into a form, which has the same size and orientation with the original image. The watermark is then detected from the aligned image. Extensive search and template based methods belong to this category [23,27]. In practice, extensive search based methods are computationally expensive, and template based embedding is subject to template removal attacks. (2) Feature based embedding. The basic principle is to associate the watermark signal with image features. While the salient image features are invariant to geometric distortions, invariance of the watermark is achieved. This method has been addressed extensively these years [18,22,29]. The robustness of this kind of method depends on the invariance of the image features. Watermark capacity is often limited, because the watermark is embedded into the feature based on local regions. (3) Invariant domain embedding. The image is first processed to form a representation invariant to the general
geometric transforms. Watermark embedding and extraction are implemented in the invariant domain. The proposed scheme falls into this category.

Invariant image representation is crucial for many pattern recognition tasks, such as character recognition [2,5], face recognition [17,20], and texture/shape classification [13,14]. Digital watermarking resembles these applications in which geometric invariance is desired. In practical applications, the watermarking scheme should be able to handle both traditional signal processing operations and geometric distortions. Invariant domain based watermarking is a kind of method that can achieve this goal [24,40]. The popular invariant transforms reported in the literature include, but not limited to, image normalization [3,11], Fourier–Mellin transform [25,28,41], Tchebichef moment [26,38], Krawtchouk moment [36,39], and Zernike moment (ZM)/pseudo-Zernike moment (PZM) [9,12,16,21,30–32]. The idea of image normalization based method is to align the image using geometric moments before watermark embedding and extraction [3]. While invariance can be easily achieved, the inverse normalization process during watermark embedding introduces interpolation error to the watermarked image, which can be seen as a kind of signal processing attack. This problem also arises in Fourier–Mellin based methods, due to the log polar mapping and inverse log polar mapping operations [28]. ZM and PZM based methods are extensively investigated because the magnitudes of ZMs/PZMs are invariant to image rotation. Farzam and Shirani first introduced ZM into watermarking [12]. The original image is first divided into sub-blocks and the watermark is then embedded by modifying ZMs of each sub-block. Kim and Lee [16] embedded a watermark pattern into the original image by modifying the ZMs with orders less than 5. Xin et al. embedded a binary watermark sequence by modifying the accurate ZMs/PZMs using dither modulation [31]. One problem of the ZM/PZM based method is that the capacity of the embedded watermark is limited, due to the high computation complexity and numerical instability issues. The computation of ZM/PZM involves many factorial terms, so high order moments are hard to calculate. Furthermore, the moments with orders higher than a certain value, approximately 44 for ZM while 23 for PZM [31], are not accurate, thus are not suitable for watermarking.

The Polar Harmonic Transform (PHT) is a new kind of orthogonal moment defined on the circular domain [35]. The magnitudes of PHTs are invariant to image rotation and scaling. Compared to ZMs/PZMs, the computation cost of PHTs is extremely low. Besides, the PHTs are free of numerical instability issues so that high order moments can be obtained accurately. As a result, we believe PHTs are more suitable for watermarking. In this paper, PHT is introduced for watermarking, aiming to achieve both geometric invariance and high capacity data hiding. The properties of PHTs are investigated and the accurate moments are selected. Then a binary watermark sequence is embedded by quantizing the magnitudes of PHTs. In the decoder, the embedded watermark can be easily recovered from the magnitudes of the PHTs. Simulation results show that the proposed scheme can resist both geometric attacks and signal processing attacks. Compared to ZM/PZM based methods, the proposed scheme can achieve better robustness and higher capacity.

The rest of this paper is organized as follows: Section 2 introduces the principles of PHT and the proposed method is presented in Section 3. Section 4 presents the experimental results followed by the conclusions in Section 5.

2. Polar Harmonic Transform

Harmonic analysis is the technology that studies the representation of signals as the superposition of basic waves. It has been widely used for rotation invariant pattern recognition [1,4,10]. The Polar Harmonic Transform, also can be called Circular Harmonic Transform (CHT), is a kind of orthogonally invariant moment defined on the unit circle.

The general orthogonal moments are defined by projecting the image onto the orthogonal kernel function, which is denoted by $H_{nl}(r, \theta)$ in this paper. In the unit circle domain, the orthogonal kernel consists of a radial component and a circular component.

$$H_{nl}(r, \theta) = R_{n}(r)e^{i\theta}$$

where $R_{n}(r)$ is the radial part, $e^{i\theta}$ is the circular part. Furthermore, the kernels satisfy the following orthonormal condition.

$$\int_{0}^{2\pi} \int_{0}^{1} H_{np}(r, \theta)H_{pq}(r, \theta)^{*}rdrd\theta = \delta_{np}\delta_{pq}$$

where * is the complex conjugate, $\delta$ is the Kronecker delta defined as

$$\delta_{kk} = \begin{cases} 1, & k = k' \\ 0, & k \neq k' \end{cases}$$

With the kernel function, the orthogonal moments of an image $f(r, \theta)$, can be computed by

$$M_{nl} = \Omega \int_{0}^{2\pi} \int_{0}^{1} [H_{nl}(r, \theta)]^{*}f(r, \theta)rdrd\theta$$

where $\Omega$ is a constant, $(n, l)$ is the order of the moment. When different kernels are used, different orthogonal moments can be produced.
2.1. Definition of PHT

PHT is a generalized name of Polar Complex Exponential Transform (PCET), Polar Cosine Transform (PCT) and Polar Sine Transform (PST) [35]. They have been grouped under the name PHT because the kernels of them are basic waves, i.e. trigonometric functions.

For PCET, the kernel function is

\[ H_{n}(r, \theta) = R_{n}(r)e^{j\theta} = e^{2\pi i n r} e^{j\theta} \]  

(4)

where \( R_{n}(r) = e^{2\pi i n r} \) is the radial component. For image \( f(r, \theta) \), the PCET of order \( n \) with repetition \( l \) is

\[ M_{nl} = \frac{1}{\pi} \int_{0}^{2\pi} \int_{0}^{1} [e^{2\pi i n r} e^{j\theta}] f(r, \theta) r \, dr \, d\theta \]  

(5)

where \( |n|, |l| = 0, 1, \ldots, \infty \). The orthogonality condition of the kernel is [35]

\[ \int_{0}^{2\pi} \int_{0}^{1} H_{n}(r, \theta)[H_{nl}(r, \theta)]^{*} r \, dr \, d\theta = \pi \delta_{mn} \delta_{ll} \]  

(6)

Eq. (5) is defined for analog images. For a digital image \( f(x, y) \), the PCET computed on the discrete domain is

\[ M_{nl} = \frac{4}{\pi N^{2}} \sum_{x=-N}^{N-1} \sum_{y=-N}^{N-1} [H_{n}(x, y)]^{*} f(x, y) \]  

(7)

where \( N \times N \) is the size of the image.

Due to the orthogonal property of PCET, an image can be expressed in terms of the moments, namely image reconstruction from PCETs.

\[ f(r, \theta) = \sum_{n=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} M_{nl} H_{nl}(r, \theta) \]  

(8)

Similar to PCET, PCT and PST are defined as

\[ M_{nl}^{pc} = \Omega_{n} \int_{0}^{2\pi} \int_{0}^{1} [H_{n}^{pc}(r, \theta)]^{*} f(r, \theta) r \, dr \, d\theta \]  

(9)

with \( |n| = 0, 1, \ldots, \infty \).

\[ M_{nl}^{ps} = \Omega_{n} \int_{0}^{2\pi} \int_{0}^{1} [H_{n}^{ps}(r, \theta)]^{*} f(r, \theta) r \, dr \, d\theta \]  

(10)

with \( n = 1, \ldots, \infty \), \( |l| = 0, 1, \ldots, \infty \).

\[ \Omega_{n} = \begin{cases} \frac{1}{\pi}, & n = 0 \\ \frac{2}{\pi}, & n \neq 0 \end{cases} \]

The kernels of PCT and PST are defined by

\[ H_{n}^{pc}(r, \theta) = R_{n}^{c}(r)e^{j\theta} = \cos(\pi n r) e^{j\theta} \]  

(11)

\[ H_{n}^{ps}(r, \theta) = R_{n}^{s}(r)e^{j\theta} = \sin(\pi n r) e^{j\theta} \]  

(12)

2.2. Invariant property

**Rotation invariance.** For an image \( f(r, \theta) \), if it is rotated clockwise by \( \phi \) degrees, the rotated moment \( M_{nl}^{rot} \) and the original one \( M_{nl} \) satisfy the following equation.

\[ M_{nl}^{rot} = M_{nl} \cdot e^{-j\phi} \]  

(13)

It is known from Eq. (13) that \( |M_{nl}^{rot}| = |M_{nl}| \). Therefore, the magnitudes of the PHTs are invariant to image rotations.

**Scale invariance.** The magnitudes of PHTs are invariant to scaling if the computation area can be made to cover the same content. In practice, this condition is met because the PHTs are defined on the unit disk. Namely, the image is stretched or shrunk to a domain \( (x_{s}, y_{t}) \in [-1, 1] \times [-1, 1] \) [35],

\[ x_{s} = \frac{s - N/2}{N/2}, \quad y_{t} = \frac{t - N/2}{N/2} \]  

(14)

where \( s, t = 0, 1, \ldots, N - 1 \).
3. Proposed scheme

The proposed algorithm operates as follows. For an image, the PHTs are first computed. Then the accurate moments are selected. A secret key is then employed to randomly select some moments, and a binary watermark sequence is encoded by modifying the magnitudes of the selected moments. For the proposed method, the following issues are important, so they are discussed in further detail.

3.1. Accurate moment selection

The PHTs are defined for analog images. For digital images, the moments can only be obtained approximately. Accordingly, the invariance of the moments is achieved approximately. For watermarking, only the accurate moments should be used. Xin et al. pointed out that ZMs/PZMs with different repetitions have different accuracies [31]. In implementation, we find that this also holds for PHTs. Let us look at an image with constant gray scale $f(r, h) = C$, namely $f(r, h) = C$. According to the definitions of PHTs, we have the following results for PCET and PCT

$$M_{nl} = \begin{cases} C, & n = l = 0 \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

and for PST

$$M_{nl}^s = \begin{cases} \frac{C}{l}, & n = 1, 3, 5, \ldots, l = 0 \\ 0, & \text{otherwise} \end{cases} \quad (16)$$

In our implementations, we find that the PHTs with repetitions $l = 4m, m \in \mathbb{Z}$, are inconsistent with the above results. In order to demonstrate this point, we compute the PHTs on a 256 x 256 image with constant gray scale 168. The magnitudes of some PCETs, PCTs and PSTs are listed in Tables 1–3. In this experiment, we restrict the orders and repetitions by $|n| + |l| \leq 9$ for PCET, $n + |l| \leq 9$ for PCT and PST.

From the tables, it could be concluded that the magnitudes of PHTs with repetitions $l = 4m, m \in \mathbb{Z}$, are not zero. Some of them even have significant values, which is not consistent with Eqs. (15) and (16). Generally, we have the following results.

**Theorem 1.** Given a constant image $f(r, h) = C$, the PCETs computed via Eq. (5) and the PCTs computed via Eq. (9) are

$$M_{nl} = \begin{cases} C, & n = l = 0 \\ \text{nonzero,} & (n, l) \neq (0, 0), \quad l = 4m, \quad m \in \mathbb{Z} \\ 0, & \text{otherwise} \end{cases}$$

**Table 1**

<table>
<thead>
<tr>
<th>$l = 0$</th>
<th>$n = 0$</th>
<th>$n = 1$</th>
<th>$n = 2$</th>
<th>$n = 3$</th>
<th>$n = 4$</th>
<th>$n = 5$</th>
<th>$n = 6$</th>
<th>$n = 7$</th>
<th>$n = 8$</th>
<th>$n = 9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>166.59</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.1224</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0705</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.0996</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.1224</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0705</td>
</tr>
<tr>
<td>0.0996</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.1224</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.0997</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.1225</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.0997</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.1226</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.0998</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.1227</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0999</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.1227</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1001</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.1227</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1002</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.1227</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1003</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 2**

<table>
<thead>
<tr>
<th>$l = 0$</th>
<th>$n = 0$</th>
<th>$n = 1$</th>
<th>$n = 2$</th>
<th>$n = 3$</th>
<th>$n = 4$</th>
<th>$n = 5$</th>
<th>$n = 6$</th>
<th>$n = 7$</th>
<th>$n = 8$</th>
<th>$n = 9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>166.59</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.1224</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0705</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.1992</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.2031</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.1410</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.1992</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.2448</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.1992</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.2449</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.1993</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.2032</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1993</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.2032</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1993</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.2032</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1993</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.2032</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1994</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.2032</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1994</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.2032</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4. L. Li et al. / Information Sciences 199 (2012) 1–19
The PSTs, computed via Eq. (10), are
\[ M_{nl} = \begin{cases} \frac{4C}{n\pi}, & n = 1, 3, 5, \ldots, l = 0 \\ \text{nonzero}, & (n, l) \notin \{(1, 0), (3, 0), (5, 0), \ldots\}, \ l = 4m, \ m \in \mathbb{Z} \\ 0, & \text{otherwise} \end{cases} \]

See Appendix for the proof of this theorem, which is based on Xin’s work [31]. Based on Theorem 1, the PHTs with repetitions \( l = 4m, m \in \mathbb{Z} \), are not accurate, so they cannot be used to encode watermark bits. As a result, the accurate moments that can be used for watermark embedding is denoted by \( S = \{M_{nl}, l \neq 4m, m \in \mathbb{Z}\} \).

In order to evaluate the invariance of the PHTs, we performed some simulations. For rotation, the image is rotated by angles from 0° to 180° with the interval 10°. Then the average value \( \mu \) and the standard deviation \( \sigma \) of the magnitudes are computed. Then \( \sigma/\mu \) is calculated to evaluate the invariance of the moments, which indicates the percentage of spread of the magnitudes from their average values. For scaling, the image is scaled with factors from 0.5 to 2 with interval 0.1. Furthermore, the simulation results on ZMs/PZMs are also given for comparisons. Figs. 1 and 2 show the invariance of the moments, where about 50 lower order moments are employed in the experiment.

From Figs. 1 and 2, we know that the invariance of PHTs is better than that of ZMs/PZMs. For PHTs, \( \sigma/\mu \) is almost evenly distributed for both low order and high order moments. For ZMs/PZMs, low order moments show better invariance than high order ones, which can be seen from the ascending bars. From this point of view, PHT based watermarking is expected to outperform ZM/PZM based watermarking.

### 3.2. Modification of PHTs

It is known from Eq. (8) that accurate image reconstruction can only be achieved when all PHTs are known. With limited number of moments, the image can only be obtained approximately. Fig. 3 shows an example of the reconstructed images using PHTs, with restrictions \( |n| + |l| \leq 32 \) for PCET, \( n + |l| \leq 32 \) for PCT and PST. In this case, 2113 PCETs, 1089 PCTs and 1024 PSTs are used for reconstruction, respectively.
It is known from Fig. 3 that although the orders of the moments are very high, the reconstructed images still degrade severely. As a result, we cannot embed the watermark by modifying the moments directly.

For a moment \( M_{nl} \), if we modify it by an amount \( k \), then the image reconstructed using the new moment \( M_{nl}^0 \) is

\[
\hat{f}(r, \theta) = \sum_n \sum_l M_{nl}^0 H_{nl}(r, \theta) = \sum_n \sum_l (M_{nl} + k) H_{nl}(r, \theta) = \sum_n \sum_l M_{nl} H_{nl}(r, \theta) + \sum_n k H_{nl}(r, \theta) = \hat{f}(r, \theta) + e(r, \theta)
\]

(17)

It is known from Eq. (17) that the reconstructed image consists of two parts, the original image \( \hat{f}(r, \theta) \) and a compensation image \( e(r, \theta) \). It should be noted that due to the conjugate property of PCETs, when \( M_{nl} \) is modified, its conjugate moment \( M_{-n,-l} \) should also be modified to obtain a real image. As a result, the compensation image, \( e(r, \theta) \), can be rewritten as

\[
e(r, \theta) = \sum_n \sum_l 2H_{nl}(r, \theta) = 2[H_{-n,-l}(r, \theta) + H_{nl}(r, \theta)] = \begin{cases} \lambda H_{00}(r, \theta), & n = l = 0 \\ \lambda[H_{-n,-l}(r, \theta) + H_{nl}(r, \theta)], & n \neq 0 \text{ or } l \neq 0 \end{cases}
\]

(18)

If we add \( e(r, \theta) \) to the original image and compute the moment again, then the new moment becomes

\[
M_{nl}^0 = M_{nl} + \lambda
\]

(19)

See Appendix for the proof of this equation. It is known from Eq. (19) that if we only modify \( M_{nl} \) and add the compensation image directly to the original image. Then the magnitudes of the moments will only change at \( (n, l) \), not affecting other moments.

Fig. 4 shows an example of moment modification at order \( (2, -2) \) and its conjugated one \( (-2, 2) \), with \( \lambda = 1 \). The two moments are modified to produce a compensation image \( e(r, \theta) \). Then it is added to the original image and the new moments \( M_{2,-2}^0 \) and \( M_{-2,2}^0 \) are computed. The magnitude differences are shown in Fig. 4. It can be seen that the added compensation image only affects the moments at \( (2, -2) \) and \( (-2, 2) \), and the magnitude errors are all zero for other moments. Therefore, if we embed a watermark bit by modifying the magnitude of \( M_{nl} \), then we are able to extract it from the magnitude of \( M_{nl}^0 \).
Case II: \( n \neq 0 \) or \( l \neq 0 \)

\[
M'_{nl} = \frac{1}{\pi} \int_0^{2\pi} \int_0^1 \left[ H_{nl}(r, \theta) \right] [f(r, \theta) + \epsilon(r, \theta)] r \, dr \, d\theta \\
= \frac{1}{\pi} \int_0^{2\pi} \int_0^1 \left[ H_{nl}(r, \theta) \right] f(r, \theta) r \, dr \, d\theta + \frac{1}{\pi} \int_0^{2\pi} \int_0^1 \left[ H_{nl}(r, \theta) \right] \epsilon(r, \theta) r \, dr \, d\theta \\
= M_{nl} + \frac{1}{\pi} \int_0^{2\pi} \int_0^1 \left[ H_{nl}(r, \theta) \right] \lambda [H_{n-l}(r, \theta) + H_{nl}(r, \theta)] r \, dr \, d\theta \\
= M_{nl} + \frac{\lambda}{\pi} \int_0^{2\pi} \int_0^1 \left[ H_{nl}(r, \theta) \right] H_{n-l}(r, \theta) r \, dr \, d\theta + \frac{\lambda}{\pi} \int_0^{2\pi} \int_0^1 \left[ H_{nl}(r, \theta) \right] H_{nl}(r, \theta) r \, dr \, d\theta
\]

(42)

Similar to Case I, \( M'_{nl} \) can be simplified as

\[
M'_{nl} = M_{nl} + \frac{\lambda}{\pi} \delta_{n-l} \delta_{l,l} + \frac{\lambda}{\pi} \delta_{n} \delta_{l,l}
\]

(43)

For Case II, either \( \delta_{n,-n} \) or \( \delta_{l,-l} \) is equal to zero. Therefore, \( M'_{nl} \) can be simplified as

\[
M'_{nl} = M_{nl} + 0 + \lambda = M_{nl} + \lambda
\]

(44)

To sum up, Eq. (19) holds.

References


