A robust image watermarking algorithm using SVR detection

Xiang-Yang Wang, a,b, *, Zi-Han Xub, Hong-Ying Yang b

a State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
b School of Computer and Information Technology, Liaoning Normal University, Dalian 116029, China

Abstract

Geometric distortion is known as one of the most difficult attacks to resist. Geometric distortion desynchronizes the location of the watermark and hence causes incorrect watermark detection. According to the Support Vector Regression (SVR), a new image watermarking detection algorithm against geometric attacks is proposed in this paper, in which the steady Pseudo-Zernike moments and Krawtchouk moments are utilized. The host image is firstly transformed from rectangular coordinates to polar coordinates, and the Pseudo-Zernike moments of the host image are computed. Then some low-order Pseudo-Zernike moments are selected, and the digital watermark is embedded into the cover image by quantizing the magnitudes of the selected Pseudo-Zernike moments. The main steps of watermark detecting procedure include: (i) some low-order Krawtchouk moments of the image are calculated, which are taken as the eigenvectors; (ii) the geometric transformation parameters are regarded as the training objective, the appropriate kernel function is selected for training, and a SVR training model can be obtained; (iii) the Krawtchouk moments of test image are selected as input vector, the actual output (geometric transformation parameters) is predicted by using the well trained SVR, and the geometric correction is performed on the test image by using the obtained geometric transformation parameters; (iv) the digital watermark is extracted from the corrected test image. Experimental results show that the proposed watermarking detection algorithm is not only robust against common signal processing such as filtering, sharpening, noise adding, and JPEG compression etc., but also robust against the geometric attacks such as rotation, translation, scaling, cropping and combination attacks, etc.

Corresponding author. Address: School of Computer and Information Technology, Liaoning Normal University, Dalian 116029, China. Tel.: +86 0411 85992415; fax: +86 0411 85992323.
E-mail address: wxy37@126.com (X.-Y. Wang).

1. Introduction

Due to the growth of the network (especially Internet) and multimedia technology, digital multimedia is widely used and accessed everywhere. However, digital multimedia can be copied, manipulated, and reproduced illegally, without quality degradation and protection. Therefore, copyright protection has become a social issue. Digital watermarking has drawn a lot of attention in the past decade because it is recognized as a potential solution to a number of issues related to multimedia works, such as authorship identification, content authentication, fingerprinting, and secondary data delivery (Barni, Cox, & Kalker, 2005).

With the development of watermarking technologies, attacks against watermarking systems have become more sophisticated. In general, the attacks on watermarking systems can be categorized into common signal processing and geometric distortions. While the common signal processing, such as lossy compression, sharpening, noise addition, lowpass filtering, etc., reduces watermark energy, geometric distortions induce synchronization errors between the original and the extracted watermark pattern and therefore can mislead the watermark detector. Most of the previous methods have shown robustness against common signal processing, and only a few specialized watermarking methods have addressed the geometric distortions. These few can be classified into invariant transform, template insertion and feature-based synchronization (Licks & Jordan, 2005; Sheng-He, Zhe-Ming, & Xia-Mu, 2004).

1.1. Invariant transform

The most obvious way to achieve resilience against geometric attacks is to use an invariant transform. In Kim and Lee (2003), Dong, Jovan, Nikolos, Yongyi, and Franck (2005), Parameswaran and Anbumani (2006), the watermark is embedded in an affine-invariant domain by using Fourier–Mellin transformation, generalized Radon transformation, and Zernike moment, respectively. Despite they are robust against global affine transformations, those techniques involving invariant domain suffer from implementation issues and are vulnerable to cropping.
1.2. Template insertion

Another solution to cope with geometric attacks is to identify the transformation by retrieving artificially embedded references. In Xiaojun (2004), the template is embedded in the discrete Fourier transform (DFT) domain as local peaks in predefined positions. The embedded local peaks are searched during watermark detection process in order to yield information about the affine transformations that the image has undergone (Lichtenauer, Setyawan, & Lagendijk, 2004). However, this kind of approach can be tampered by the malicious attack since anyone can access the peaks in the DFT and easily eliminate them.

1.3. Feature-based

The last category is based on media features. Its basic idea is that, by binding the watermark with the geometrically invariant image features, the watermark detection can be done without synchronization error. In Weinheimer (2004), the Mexican hat wavelet is used to extract feature points, and several copies of the watermark are embedded in the disks centered at the feature points. Lee, Lee, and Lee (2007) propose a new geometrically invariant watermarking method that uses circular Hough transform for watermark synchronization. Through circular Hough transform, the circular features are extracted that are invariant to geometric distortions. Seo and Yoo (2006) introduce a content-based image watermarking algorithm based on scale-space representation. Lee, Kim, and Lee (2006) propose a novel image watermarking method against geometric distortions by using the scale-invariant feature transform (SIFT). The feature-based image watermarking is better than others in terms of robustness. However, some drawbacks indwelled in current feature-based schemes restrict the performance of watermarking system. First, the feature point extraction is sensitive to image modification. Second, the computational complexity in calculating the features of an image before watermark detection is added. Third, the volume of watermark data is lesser.

In order to effectively resolve the problem of resisting geometric attacks, the support vector machine (SVM) theory is introduced to the image watermarking domain. Fu, Shen, and Lu (2004) first embed template and watermark into original image in the same way, then a SVM train model is obtained by using the template samples, and the output of SVM model is obtained and the watermark is extracted. In scheme (Wu & Xie, 2006), in order to obtain the rotation, scaling and translation (RST) parameters, the SVM are utilized to learn image geometric pattern represented by six combined low order image moments. The watermark extraction is carried out after watermarked image has been synchronized without original image. Tsai and Sun (2007) propose a novel watermarking technique called SVM-based color image watermarking (SCIW) for the authentication of color images. The SCIW method utilizes the set of training patterns to train the SVM and then applies the trained SVM to classify a set of testing patterns. Following the results produced by the classifier, the SCIW method retrieves the hidden watermark without the original image during watermark extraction. Li, Ling, and Lu (2007) introduce a novel semifragile watermarking scheme based on (SVM). This scheme first gives a definition of wavelet coefficient direction tree, then the relation model between the root node and its offspring nodes is established using SVM, and further watermark is embedded and extracted based on this relation model.

Based on a large number of theory analyses and experimental results, we can easily come to the conclusion that it is possible to resist geometric attacks by utilizing the advanced SVM, but the current SVM based image watermarking have shortcomings as follows:

- The features are not selected by combining the principle of invariability, which influences the working efficiency of SVM training and lowers the imperceptibility and robustness performance of watermarking.
- They are not very robust against geometric attacks like cropping, local random bending, and mixed attacks, etc.
- In watermark detection procedure, the original watermark signal is needed, so it is unfavorable to practical application.

In this paper, a new SVR based image watermarking detection algorithm against geometric attacks is proposed, in which the steady Pseudo-Zernike moments and Krawtchouk moments are utilized. The host image is firstly transformed from rectangular coordinates to polar coordinates, and the Pseudo-Zernike moments of the host image are computed. Then some low-order Pseudo-Zernike moments are selected, and the digital watermark is embedded into the cover image by quantizing the magnitudes of the selected Pseudo-Zernike moments. The main steps of watermark detection procedure include: (1) some low-order Krawtchouk moments of the image are calculated, which are taken as the eigenvectors; (2) the geometric transformation parameters are regarded as the training objective, the appropriate kernel function is selected for training, and a SVR training model can be obtained; (3) the Krawtchouk moments of test image are selected as input vector, the actual output (geometric transformation parameters) is predicted by using the well trained SVR, and the geometric correction is performed on the test image by using the obtained geometric transformation parameters; (4) the digital watermark is extracted from the corrected test image.

This paper is organized as follows. In Section 2, the Support Vector Regression (SVR) is briefly introduced. Section 3 presents the invariance properties of Pseudo-Zernike moments and Krawtchouk moments. Sections 4 and 5 describe our watermark embedding and extracting algorithm, respectively. Section 6 will be dedicated to the description of a variety of simulation experiments, which will illustrate the effectiveness of the proposed scheme. Finally, conclusions will be briefed in Section 7.

2. Support Vector Regression (SVR)

Support vector machine (SVM) is a universal classification algorithm proposed by Vapnik (1995) in the middle of 1990s, it is thought of a new innovation of learning machine, which uses the statistical learning theory.

The basic theory of SVM can be depicted by a typical two-dimensional case shown in Fig. 1. In Fig. 1, (●) and (■) denote two categories of samples. H is the separating hyperplane, $H_1$ and $H_2$ are parallel to $H$ (they have the same normal) and no training points fall between them; the margin of a separating hyperplane is defined as $H_1 + H_2$. The optimal separating hyperplane what you call not only can separate the two categories of samples exactly (the ratio of training errors is 0), but also has the maximal
margin. Thus the problem of optimal separating hyperplane can be transformed a constraints problem.

Support Vector Regression (SVR) is an application of support vector machine on regression learning. Next, we will briefly introduce the related theories about SVR.

For the training sets:
\[
\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}
\]

\(x \in \mathbb{R}^d, y \in \mathbb{R}\) to get the relation between the input \(x_i\) and output \(y_i\), it can seek a optimal regression function \(f(x)\) by SVR training, so that the difference between the output value between \(y_i\) and \(\hat{y}_i\), which is defined as:
\[
C(\mathbf{y}_i - \mathbf{y}_i) = \sqrt{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}
\]

The Pseudo-Zernike moments is as follows:
\[
\text{Pseudo-Zernike polynomial} = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} A_{nm} V_{nm}(x, y)
\]

The orthogonality and completeness of the Pseudo-Zernike polynomials is discussed in our scheme, it only needs replace the inner product operation of training samples in the optimal function seeking, for the nonlinear image data discussed in our scheme, it only needs replace the inner product operation mentioned above by kernel function \(k(x_i, x_j)\) to implement the nonlinear function regression.

\section{3. The Pseudo-Zernike moments and Krawtchouk moments}

\subsection{3.1. The Pseudo-Zernike moments}

Pseudo-Zernike moments consist of a set of complex polynomials (Khotanzad & Hong, 1990) that form a complete orthogonal set over the interior of the unit circle, \(x^2 + y^2 < 1\). If the set of these polynomials is denoted by \(\{V_{nm}(x, y)\}\), then the form of these polynomials is as follows:

\[
V_{nm}(x, y) = V_{nm}(\rho, \theta) = R_{nm}(\rho) \exp(jm\theta)
\]

where \(\rho = \sqrt{x^2 + y^2}\), \(\theta = \tan^{-1}(y/x)\). Here \(n\) is a nonnegative integer, \(m\) is restricted to be \(|m| \leq \bar{n}\) and the radial Pseudo-Zernike polynomial \(R_{nm}(\rho)\) is defined as the following:

\[
R_{nm}(\rho) = \frac{(-1)^{\bar{n} - m}}{s(n + \bar{n})!} \rho^{n - s} \left(1 - \rho^2\right)^{s/2}
\]

Like any other orthogonal and complete basis, the Pseudo-Zernike polynomial can be used to decompose an analog image function \(f(x, y)\):

\[
f(x, y) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} A_{nm} V_{nm}(x, y)
\]

where \(A_{nm}\) is the Pseudo-Zernike moments of order \(n\) with repetition \(m\), whose definition is:

\[
A_{nm} = \frac{n+1}{\pi} \int_{-1}^{1} \int_{-1}^{1} f(x, y) V_{nm}(x, y) dx dy
\]

It should be pointed out that in case of digital images, (8) cannot be applied directly, but rather, its approximate version has to be employed. For instance, given a digital image of size \(M \times N\), its Pseudo-Zernike moments are computed as:

\[
\hat{A}_{nm} = \frac{n+1}{\pi} \sum_{i=1}^{M} \sum_{j=1}^{N} h_{nm}(x_i, y_j) f(x_i, y_j)
\]

where the value of \(i\) and \(j\) are taken such that \(x_i^2 + y_j^2 \leq 1\), and

\[
h_{nm}(x_i, y_j) = \int_{x_i - \Delta x}^{x_i + \Delta x} \int_{y_j - \Delta y}^{y_j + \Delta y} V_{nm}(x, y) dx dy
\]

where \(\Delta x = \frac{\Delta x}{\Delta x}, \Delta y = \frac{\Delta y}{\Delta x}\). The Pseudo-Zernike moments of discrete images:

\[
\hat{A}_{nm} = \frac{n+1}{\pi} \sum_{i=1}^{M} \sum_{j=1}^{N} V_{nm}(x_i, y_j) f(x_i, y_j) \Delta x \Delta y
\]

The orthogonality and completeness of the Pseudo-Zernike yield the following formula for reconstructing the image function:

\[
f(x, y) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} A_{nm} V_{nm}(x, y)
\]

The reason we use Pseudo-Zernike moments for image watermarking is that they have some very important properties, i.e., their magnitudes are invariant under image rotation and image flipping. We now elaborate on these invariance properties.

If image \(f(r, \theta)\) is rotated \(\alpha\) degrees counterclockwise, the image under rotation is \(f(r, \theta) = f(r, \theta - \alpha)\), and the Pseudo-Zernike moments of \(f(r, \theta)\) can be expressed by:

\[
A_{nm} = \frac{n+1}{\pi} \int_{-\pi}^{\pi} \int_{0}^{r} f(r, \theta - \alpha) \left[R_{nm}(r) \exp(jm\theta)\right] r dr d\theta
\]

Assume \(\theta' = \theta - \alpha\), we can get

\[
A_{nm} = \frac{n+1}{\pi} \int_{-\pi}^{\pi} \int_{0}^{r} f(r, \theta') \left[R_{nm}(r) \exp(jm\theta')\right] r dr d\theta'.
\]

It can be shown that the Pseudo-Zernike moments of the resulting image are \(A_{nm} = A_{nm} \exp(-jmx)\), which leads to \(|A_{nm}'| = |A_{nm}|\). Therefore, if a watermark is inserted in the magnitudes of Pseudo-Zernike moments, it is robust to rotation.
It can also be shown that if an image \( f(x, y) \) is flipped horizontally, the Pseudo-Zernike moments of the resulting image are
\[
\begin{align*}
A_{nm}^{(h)} &= A_{nm}^c, & \text{if } m \text{ is even; } \\
A_{nm}^{(h)} &= -A_{nm}^c, & \text{if } m \text{ is odd. }
\end{align*}
\]

Similarly, if an image \( f(x, y) \) is flipped vertically, the Pseudo-Zernike moments of the result image \( A_{nm}^{(v)} = A_{nm}^s \) for \( m \) even or odd. In either case, the magnitudes of the Pseudo-Zernike moments do not change, indicating that the watermark hidden in the magnitudes of Pseudo-Zernike moments has perfect robustness to image flipping attacks.

The Pseudo-Zernike moments also have other advantages.

1. **Good robustness against noise.** Pseudo-Zernike moments have been proven to be superior to other moment functions such as Zernike moments in terms of feature representation capabilities. Pseudo-Zernike moments can offer more feature vectors than Zernike moments. There are \( (n+1)^2 \) linearly independent Pseudo-Zernike polynomials of order \( \leq n \) as compared to \( (n+1)(n+2)/2 \) of Zernike polynomials due to additional constraint of \( n - m = \text{even} \). Under the same order, Pseudo-Zernike moments have more low-order moments than Zernike moments, therefore Pseudo-Zernike moments are less sensitive to image noise than the conventional Zernike moments.

2. **Orthogonal property.** Pseudo-Zernike moments are orthogonal. Orthogonal moments have been proven to be more robust in the presence of noise, and they are able to achieve a near zero value of redundancy measure in a set of moment functions.

3. **Reconstruction property.** Pseudo-Zernike moments can be used to reconstruct the image.

4. **Multilayer expression.** For an image, the low-order moments of the Pseudo-Zernike moments can express the outline of the image; and the high-order moments of the Pseudo-Zernike moments can express the detail of the image.

5. **It is convenient to implement.** The higher order Pseudo-Zernike moments can be constructed arbitrarily, not like the Hu moments.

On the whole, Pseudo-Zernike moment is a good figure description operator and is suitable for watermark embedding.

### 3.2. The Krawtchouk moments

Krawtchouk moments (Yap, Paramesran, & Ong, 2003) are a set of moments formed by using Krawtchouk polynomials as the basis function set. Krawtchouk polynomials, introduced by Mikhail Krawtchouk, are a set of polynomials associated with the binomial distribution. In this section, the definitions of Krawtchouk and weighted Krawtchouk polynomials are first provided, followed by Krawtchouk moments and Krawtchouk moment invariants.

#### 3.2.1. Krawtchouk polynomials

The definition of the \( n \)th order classical Krawtchouk polynomial is defined as
\[
K_n(x;p,N) = \sum_{k=0}^{n} \binom{n}{k} \binom{N-k}{n-k} x^k / k!
\]

where \( x, n = 0, 1, 2, \ldots, N, N > 0 \). The parameter \( p \in (0,1) \), \( zF_1 \) is the hypergeometric function, defined as
\[
zF_1(a,b;c;z) = \sum_{k=0}^{\infty} \frac{(a)_k(b)_k}{(c)_k k!} z^k
\]

and \((a)_k\) is the Pochhammer symbol given by
\[
(a)_k = a(a+1)(a+2)\ldots(a+k-1) = \frac{\Gamma(a+k)}{\Gamma(a)}
\]

The set of \((N+1)\) Krawtchouk polynomials \( \{K_n(x;p,N)\} \) forms a complete set of discrete basis functions with weight function
\[
\omega(x;p,N) = \binom{N}{x} p^x (1-p)^{N-x}
\]

and satisfies the orthogonality condition
\[
\sum_{x=0}^{N} \omega(x;p,N)K_n(x;p,N)K_m(x;p,N) = \rho(n;p,N)\delta_{nm}
\]

where \( n, m = 0, 1, 2, \ldots, N \) and
\[
\rho(n;p,N) = (-1)^n \frac{(1-p)}{p} \cdot \frac{n!}{(-N)_n}
\]

#### 3.2.2. Weighted Krawtchouk polynomials

The conventional method of avoiding numerical fluctuations for moment computations is by means of normalization by the norm \( \|K_n(x;p,N)\| \). The normalized Krawtchouk polynomials with respect to the norm is defined as
\[
\tilde{K}_n(x;p,N) = K_n(x;p,N) / \sqrt{\rho(n;p,N)}
\]

In addition to normalizing the polynomials with the norm, the square root of the weight is also introduced as a scaling factor. The set of weighted Krawtchouk polynomials is defined by
\[
\mathcal{R}_n(x;p,N) = K_n(x;p,N) \sqrt{\frac{\omega(x;p,N)}{\rho(n;p,N)}}
\]

#### 3.2.3. Krawtchouk moments

Krawtchouk moments have the interesting property of being able to extract local features of an image. The Krawtchouk moments of order \( n + m \) in terms of weighted Krawtchouk polynomials, for an image with intensity function, \( f(x,y) \), is defined as
\[
Q_{nm} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \mathcal{R}_n(x;p_1,N-1) \mathcal{R}_m(y;p_2,M-1) f(x,y)
\]

where \( \mathcal{R}_n(x;p,N-1) \) is the \( n \)th order weighted Krawtchouk polynomial, which is defined as
\[
\mathcal{R}_n(x;p,N-1) = K_n(x;p,N-1) \sqrt{\frac{\omega(x;p,N-1)}{\rho(n;p,N-1)}}
\]

The weight function \( \omega(x;p,N-1) \) is given by
\[
\omega(x;p,N-1) = \binom{N-1}{x} p^x (1-p)^{N-1-x}
\]

and \( \rho(n;p,N-1) \) is the squared norm, which is given by
\[
\rho(n;p,N-1) = (-1)^n \frac{(1-p)}{p} \cdot \frac{n!}{(-N+1)_n}
\]

A set of Krawtchouk moments up to order \((N_{\text{max}}, M_{\text{max}})\) are given, then inverse moment transform can be computed using
\[
f(x,y) = \sum_{n=0}^{N_{\text{max}}} \sum_{m=0}^{M_{\text{max}}} Q_{nm} \mathcal{R}_n(x;p_1,N-1) \mathcal{R}_m(y;p_2,M-1)
\]

The above formula is used for reconstructing the image from a set of Krawtchouk moments up to order \((N_{\text{max}}, M_{\text{max}})\).
3.2.4. Krawtchouk moment invariants

A set of Krawtchouk moment invariants is given by

\[ \bar{Q}_{nm} = (\rho(n;p,1,N-1)\rho(m,p,2,M-1))^{1/2} \sum_{i=0}^{n} \sum_{j=0}^{m} a_{nm} \rho_{nm} \rho_{ij} \]

where \( \{a_{nm}\} \) are the coefficients determined by

\[ K_n(x;p,N) = \sum_{k=0}^{N-1} a_{n,p,k} x^k = 2F1 \left( -n, -N; \frac{1}{p} \right) \]

and \( \rho_{nm} \) are the geometric moment invariants of the image

\[ \sqrt{\alpha(x_1;1,N-1,1,0)(y_1;0,p,2,2-M)\eta(x,1,y)} \]

which are given by

\[ \bar{v}_{nm} = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \frac{NM}{M_0} \sqrt{\alpha(i,1,N-1,1,0)(j,0,p,2,2-M)\eta(i,j)} \]

\[ \left\{ (x_i - \bar{x}) \cos \theta + (y_i - \bar{y}) \sin \theta \right\} \sqrt{\frac{NM}{M_0} + N^2} \]

\[ \left\{ (y_i - \bar{y}) \cos \theta + (x_i - \bar{x}) \sin \theta \right\} \sqrt{\frac{NM}{M_0} + M^2} \]

where, \( \bar{x} = \frac{\sum_{i=0}^{N-1} x_i}{\sum_{i=0}^{N-1} 1}, \bar{y} = \frac{\sum_{j=0}^{M-1} y_j}{\sum_{j=0}^{M-1} 1}, M_{nm} = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} x_i^m y_j^n f(i,j), h_{mn} = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \left( x_i - \bar{x} \right)^m \left( y_j - \bar{y} \right)^n f(i,j) \]

The angle \( \theta \) calculated using the above expression is limited to \(-45^\circ \leq \theta \leq 45^\circ\), hence to obtain the exact angle \( \theta \) in the range from 0° to 360°.

4. Watermark embedding scheme

In this paper, a robust image watermarking scheme based on Pseudo-Zernike moments is proposed. Firstly, the Pseudo-Zernike moments of the host image are computed. Then, some low-order Pseudo-Zernike moments are selected, and the digital watermark is embedded into the cover image by quantizing the magnitudes of the selected Pseudo-Zernike moments. The diagram of our watermark embedding scheme is shown in Fig. 2.

Let \( I = \{f(x,y), 1 \leq x \leq M, 1 \leq y \leq N\} \) represent a host digital image (gray image), and \( f(x,y) \) denotes the pixel value at position \( (x,y) \). The main steps of the embedding procedure developed can be described as follows:

Step 1: Generating the watermark. A random sequence \( W = \{w_l, l=1, \ldots, L\} \) is generated by the secret key \( K_1 \), where \( L \) is the size of the sequence. The sequence values belong to the set \{0,1\}.

Step 2: Pseudo-Zernike moments computation. The Pseudo-Zernike moments of the host image \( I \) are computed (see Section 3.1).

Step 3: Pseudo-Zernike moments selection. Several low order Pseudo-Zernike moments are selected. The invariance property of Pseudo-Zernike moments of digital images have to be compromised to some extent due to approximation error of Pseudo-Zernike polynomial integration and interpolation error of the image rotation and scaling. As a result, some Pseudo-Zernike moments are computed more accurately, hence more suitable for data hiding than others. We consider two major factors in selection of moments for data hiding. Firstly, the moments with order higher than a certain value \( N_{\text{max}} \) cannot be obtained accurately, and thus have to be ruled out for data hiding. In our experiments, we set \( N_{\text{max}} = 20 \). Secondly, due to the deviation form orthogonality of sampled Pseudo-Zernike polynomials, the moments with repetition \( m = 4l (l=0,1,2) \) cannot be computed accurately, thus not suitable for data hiding.

Let \( S = \{A_{nm}, n \leq N_{\text{max}}, m = 0, m \neq 4i\} \), which is the set of all the eligible Pseudo-Zernike moments for date hiding. The cardinalities of \( S \) can readily be obtained as

\[ |S| = \begin{cases} 3N_{\text{max}} + N_{\text{max}}^2 & N_{\text{max}} = 4i \\ 3N_{\text{max}} + N_{\text{max}}^2 + 1 & N_{\text{max}} = 4i + 1 \\ 3N_{\text{max}} + N_{\text{max}}^2 + 2 & N_{\text{max}} = 4i + 2 \\ 3N_{\text{max}} + N_{\text{max}}^2 + 3 & N_{\text{max}} = 4i + 3 \\ \end{cases} \]  

(19)

where \( i \) is any nonnegative integer.

For the sake of security, we use a secret key \( K_2 \) to pseudo-randomly choose \( L \) Pseudo-Zernike moments from \( S \) to form a moment vector \( Z = (A_{p_1,q_1}, \ldots, A_{p_L,q_L}) \), where \( A_{p_1,q_1} (i=1,2,\ldots, L) \) is Pseudo-Zernike moments in \( S \).

Step 4: Digital watermark embedding. The digital watermark is embedded into the elements of moment vector \( Z \) via quantization. The magnitude of \( A_{p_1,q_1} \) is quantized, producing a new vector \( \bar{Z} = (\bar{A}_{p_1,q_1}, \ldots, \bar{A}_{p_L,q_L}) \), where \( \bar{A}_{p_1,q_1} \) is the modified version of \( A_{p_1,q_1} \), satisfying

\[ |\bar{A}_{p_1,q_1}| = \left\lceil \frac{|A_{p_1,q_1} - d_k(W_i)|}{\Delta} \right\rceil, \Delta = d_k(W_i), i = 1, \ldots, L \]  

(20)

where \( \lceil \cdot \rceil \) is the rounding operation, \( \Delta \) is the step size of quantization, and \( d_k(\cdot) \) is the dither function

\[ d_k(1) = \frac{A}{2} + d_k(0) \]

where \( d_k(0) \in [0,1] \) is pseudo-randomly generated with another key \( K_3 \), which is used to further increase the secrecy and security of the embedded signal. The modified Pseudo-Zernike moments are now readily calculated as

\[ \bar{A}_{p_1,q_1} = \frac{|\bar{A}_{p_1,q_1}|}{\Delta} A_{p_1,q_1}, i = 1, \ldots, L \]

It is worth noting that in quantizing each \( A_{p_1,q_1} \), if \( q_1 \neq 0 \), its conjugate \( A_{p_1,q_1}^* \) must be quantized simultaneously to ensure they
always have the same magnitudes, so that the reconstructed image is real.

Step 5: Difference image computation.

We can obtain the difference image \( G = [g(x, y), \ 1 \leq x \leq M, \ 1 \leq y \leq N] \) by

\[
g(x, y) = f(x, y) - f_2(x, y) \tag{21}
\]

where \( f(x, y) \) is the host image, and \( f_2(x, y) \) is the image components contributed by the selected moments before they are changed.

\[
\tilde{f}_2(\cdot) = \sum_{i=1}^{l} A_{h_{i}} V_{\alpha_{i}}(\cdot) + A_{h_{i} - q} V_{\alpha_{i} - q}(\cdot) \tag{22}
\]

Step 6: Obtaining the watermarked image.

The watermarked image

\[
l = \{ f(\cdot), \ 1 \leq x \leq M, \ 1 \leq y \leq N \}
\]

can be obtained by image combination

\[
f'(x, y) = g(x, y) + \tilde{f}_2(x, y)
\]

Here, \( \tilde{f}_2(x, y) \) is the image components contributed by those modified moments

\[
\tilde{f}_2(\cdot) = \sum_{i=1}^{l} A_{h_{i}} V_{\alpha_{i}}(\cdot) + A_{h_{i} - q} V_{\alpha_{i} - q}(\cdot) \tag{23}
\]

5. Watermark detection scheme

According to the Support Vector Regression (SVR), a robust image watermarking detection algorithm based on Krawtchouk moments is proposed, and it can be summarized as follows: (i) some low-order Krawtchouk moments of an arbitrary image are calculated, which are taken as the eigenvectors; (ii) the geometric transformation parameters are regarded as the training objective, the appropriate kernel function is selected for training, and a SVR training model can be obtained; (iii) the Krawtchouk moments of test image are selected as input vector, the actual output (geometric transformation parameters) is predicted by using the well trained SVR, and the geometric correction is performed on the test image by using the obtained geometric transformation parameters; (iv) the digital watermark is extracted from the corrected test image.

Let \( l = \{ f(x, y), \ 1 \leq x \leq M, \ 1 \leq y \leq N \} \) denote the test image, and \( f(x, y) \) denote the pixel value at position \( (x, y) \). The main steps of the watermark detecting procedure developed can be described as follows.

5.1. Eigenvector construction

Generally speaking, geometric attacks include various forms, such as rotation, scaling, translation, cropping, etc. In this paper, we chiefly discuss the familiar geometric attacks including rotation, scaling, and translation, etc. According to image theory, image energy chiefly accumulates on low order moments, and the edges or the fine details of the image usually appear on the high order moments. Considering that we will chiefly discuss global geometric attacks, so we select 4 low-order Krawtchouk moments \( Q_{200}, Q_{011}, Q_{010}, Q_{110} \) (we denote them as \( f_1, f_2, f_3, f_4 \) respectively) to reflect the global information of image (see Section 3.2).

5.2. SVR training

Support Vector Regression (SVR) provide a very promising classification technology that led to remarkable improvements in handwritten digit recognition, face detection in images, object recognition, text categorization and nonlinear time-series prediction.

SVR have been developed on a solid base of statistical learning theory and are especially designed to provide high flexibility for approximating class boundaries, yet to avoid over-fitting phenomena. SVR can be understood as novel learning algorithms for neural networks that completely avoid the problem of selecting the number of layers and the number of hidden units. In essence, SVR lead to linearly constrained quadratic programs that are solved by numerical methods.

In order to obtain the SVR training model, we firstly construct the training sample images \( I_b \) by moving (including X-axis and Y-axis), rotating and scaling an arbitrary image \( I \), and compute 4 low-order Krawtchouk moments \( (f_1^k, f_2^k, f_3^k, f_4^k) \) of training sample image \( I_b \). Then, the low-order Krawtchouk moments are regarded as features for training. Finally, the corresponding transformation (X-translation, Y-translation, rotation, scaling, etc.) parameters \( t^k_1, t^k_2, s^k, \theta^k \ (k = 1, 2, \ldots , K) \) are described as the training objective, and we can obtain the training samples as following

\[
\Omega = \{ f_1^k, f_2^k, f_3^k, f_4^k, t^k_1, t^k_2, s^k, \theta^k \ (k = 1, 2, \ldots , K) \}
\]

For the linear transformation like rotation, scaling, and translation, there is no coupling among the 4 outputs, so we adopt the MIMO system constructed by 4 SVR parallel structures which is with 4 inputs, and the SVR model can be obtained by training.

5.3. Geometric correction of test image

In order to extract accurately digital watermark, we firstly use SVR training model to predict the data of test image, and then correct the test image according to the predicted value for resisting the geometric attacks. The process of correcting test image based on SVR is as follows:

(1) Calculate the 4 low-order Krawtchouk moments \( (f_1^k, f_2^k, f_3^k, f_4^k) \) of test image \( l \), and let them be the input vectors.

(2) The actual output \( t^k_1, t^k_2, s^k, \theta^k \) (geometric transformation parameters) is predicted by using the well trained SVR.

(3) Correct the geometric attacks of test image \( l \) (that is inverse transformation such as rotation angle, translation parameters etc.) by using the obtained geometric transformation parameters \( t^k_1, t^k_2, s^k, \theta^k \) so that we can get the corrected test image \( l' \).

Fig. 3 gives geometric correction results for standard image Lena and Barbara.

5.4. Watermark extraction

The watermark detecting procedure in the proposed scheme neither needs the original host image nor any other side information. Let \( l' \) denote the corrected test image, the main steps of watermark extracting can be described as follows.

Step 1: The Pseudo-Zernike moments of the corrected test image \( l' \) are computed (see Section 3.1).

Step 2: With the same key \( K_2 \) as in the process of watermark embedding, we locate the moments that carry the watermark information, thus forming a moment vector

\[
Z' = (A_{h_{11}}, \ldots , A_{h_{10}})
\]

Step 3: The digital watermark is extracted from the Pseudo-Zernike moments of the corrected test image.

Firstly, two dither vectors \( d_0(0), d_1(1) \) are generated with the same key \( K_3 \) as in the embedder.

Then, by using the same quantizer as in (20), we quantize the magnitudes of each \( A_{h_{10}} \) with two corresponding dithers, respectively
\[ |A_{p,q}^i - A_{p,q}^j| = \left| \frac{A_{p,q}^i - d_k(j)}{A} \right| A + d_k(j), \quad j = 0, 1 \]

where \( i = 1, \ldots, L \), \( j = 0, 1 \) and \([-\cdot]\) is the rounding operation.

Finally, by comparing the distances between \( A_{p,q}^i \) and its two quantized versions, we obtain estimate of the watermark bit

\[ w_i^r = \arg \min_{0,1} \{ |A_{p,q}^i - A_{p,q}^j| \} \quad i = 1, \ldots, L \]

which is so-called minimum distance decoder. The above formula can be decomposed into the following steps:

1. The distances between \( A_{p,q}^i \) and its two quantized versions are respectively defined as

\[ \text{dis}0 = \left( (A_{p,q}^i)_0 - A_{p,q}^i \right)^2 \]

and

\[ \text{dis}1 = \left( (A_{p,q}^i)_1 - A_{p,q}^i \right)^2. \]

2. The difference between the two distances in ① is computed, note \( \tau = \text{dis}0 - \text{dis}1 \).

Fig. 3. The geometric correction results. (a) The test image (Lena). (b) The corrected test image (Lena). (c) The test image (Barbara). (d) The corrected test image (Barbara).

Fig. 4. The watermark embedding examples by using the proposed algorithm and scheme (Tsai & Sun, 2007). (a) Watermarked Lena with 64 bits using our algorithm. (b) Watermarked Mandrill with 64 bits using our algorithm. (c) Watermarked Barbara with 64 bits using our algorithm. (d) Watermarked Lena with 64 bits using scheme (Tsai & Sun, 2007). (e) Watermarked Mandrill with 64 bits using scheme (Tsai & Sun, 2007). (f) The absolute difference between origin image and watermarked image (Lena) for our algorithm. (g) The absolute difference between origin image and watermarked image (Mandrill) for our algorithm. (h) The absolute difference between origin image and watermarked image (Barbara) for our algorithm. (i) The absolute difference between origin image and watermarked image (Barbara) for scheme (Tsai & Sun, 2007). (j) The absolute difference between origin image and watermarked image (Lena) for scheme (Tsai & Sun, 2007). (k) The absolute difference between origin image and watermarked image (Mandrill) for scheme (Tsai & Sun, 2007). (l) The absolute difference between origin image and watermarked image (Barbara) for scheme (Tsai & Sun, 2007).
The watermark bits are decided by the results in (2):

\[
\text{if } t < 0 \text{ then } w_i = 0
\]
\[
\text{else } w_i = 1
\]

6. Simulation results

We test the proposed watermarking scheme on the popular test images 256 \times 256 8-bit Lena, Mandrill and Barbara. A pseudo-random sequence of size 64-bits is used as the watermark pattern.

The quantization step size \( \Delta = 1.8 \), the number of training sample is \( K = 80 \), and the radius-based function (RBF) is selected as the SVR kernel function.

Fig. 4 is the result of applying our algorithm and scheme (Tsai & Sun, 2007) for data embedding in an image. Fig. 4(a)–(c) are the watermarked image Lena, Mandrill and Barbara applying scheme (Tsai & Sun, 2007). Fig. 4(d)–(f) and Fig. 4(g)–(i) are the absolute difference between origin image and watermarked image for our algorithm and scheme (Tsai & Sun, 2007), respectively, multiplied by 50 for better display.

6.1. The quality of watermarked images

In this work, we use peak signal-to-noise ratio (PSNR) to measure the quality of watermarked images. It is defined as

\[
\text{PSNR} (I, I_0) = 10 \log \frac{255^2 \times M \times N}{\sum_{i=1}^{M} \sum_{j=1}^{N} [f(x_i, y_j) - f'(x_i, y_j)]^2}
\]

### Table 1

<table>
<thead>
<tr>
<th>Image</th>
<th>Proposed scheme</th>
<th>Scheme in Tsai and Sun (2007)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>44.31</td>
<td>39.64</td>
</tr>
<tr>
<td>Mandrill</td>
<td>45.18</td>
<td>41.53</td>
</tr>
<tr>
<td>Barbara</td>
<td>44.20</td>
<td>40.09</td>
</tr>
</tbody>
</table>

### Table 2

The watermark detection results for common signal processing (BER) (%).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>JPEG compression 70</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.40</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>50</td>
<td>0</td>
<td>1.31</td>
<td>0</td>
<td>1.75</td>
<td>0</td>
<td>1.07</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>5.76</td>
<td>0</td>
<td>6.26</td>
<td>0</td>
<td>4.57</td>
</tr>
<tr>
<td>Gaussian filter 0.55</td>
<td>0</td>
<td>2.85</td>
<td>0</td>
<td>3.55</td>
<td>0</td>
<td>2.83</td>
</tr>
<tr>
<td>Average filter</td>
<td>0.55</td>
<td>7.56</td>
<td>1.14</td>
<td>9.02</td>
<td>2.38</td>
<td>10.15</td>
</tr>
<tr>
<td>Gaussian noise 0.03</td>
<td>0</td>
<td>6.84</td>
<td>1.14</td>
<td>9.02</td>
<td>2.38</td>
<td>10.15</td>
</tr>
<tr>
<td>Salt and peppers noise</td>
<td>0</td>
<td>1.32</td>
<td>0</td>
<td>1.66</td>
<td>0</td>
<td>0.92</td>
</tr>
<tr>
<td>Salt and peppers noise + JPEG 70</td>
<td>0</td>
<td>3.73</td>
<td>0</td>
<td>5.72</td>
<td>0</td>
<td>1.30</td>
</tr>
<tr>
<td>Gaussian filter + JPEG 70</td>
<td>0</td>
<td>2.85</td>
<td>0</td>
<td>4.79</td>
<td>0</td>
<td>7.68</td>
</tr>
<tr>
<td>Gaussian noise + average filter</td>
<td>1.01</td>
<td>10.58</td>
<td>1.96</td>
<td>11.59</td>
<td>3.73</td>
<td>12.71</td>
</tr>
<tr>
<td>Average filter + sharpening</td>
<td>0.06</td>
<td>8.43</td>
<td>0</td>
<td>5.64</td>
<td>0</td>
<td>7.92</td>
</tr>
<tr>
<td>Gaussian noise + sharpening</td>
<td>1.04</td>
<td>12.17</td>
<td>1.01</td>
<td>13.85</td>
<td>3.08</td>
<td>14.58</td>
</tr>
</tbody>
</table>

### Table 3

The watermark detection results for geometric attacks (BER) (%).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotation 5°</td>
<td>0.56</td>
<td>5.25</td>
<td>0.76</td>
<td>6.37</td>
<td>1.12</td>
<td>6.81</td>
</tr>
<tr>
<td>45°</td>
<td>0.50</td>
<td>7.93</td>
<td>0.84</td>
<td>9.03</td>
<td>1.06</td>
<td>10.04</td>
</tr>
<tr>
<td>Scaling 0.5</td>
<td>0</td>
<td>1.81</td>
<td>0</td>
<td>3.50</td>
<td>0.54</td>
<td>3.34</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>3.03</td>
<td>0</td>
<td>5.73</td>
<td>0</td>
<td>5.64</td>
</tr>
<tr>
<td>Translation H 5</td>
<td>0</td>
<td>31.48</td>
<td>0</td>
<td>32.60</td>
<td>0</td>
<td>30.20</td>
</tr>
<tr>
<td>H 30</td>
<td>0</td>
<td>28.58</td>
<td>0</td>
<td>30.87</td>
<td>0</td>
<td>28.57</td>
</tr>
<tr>
<td>V 5</td>
<td>0</td>
<td>29.43</td>
<td>0</td>
<td>29.76</td>
<td>0</td>
<td>30.65</td>
</tr>
<tr>
<td>V 30</td>
<td>0</td>
<td>40.23</td>
<td>0</td>
<td>39.13</td>
<td>0</td>
<td>39.60</td>
</tr>
<tr>
<td>Flip vertically</td>
<td>0</td>
<td>3.82</td>
<td>0</td>
<td>2.65</td>
<td>0</td>
<td>3.25</td>
</tr>
<tr>
<td>Flip horizontally</td>
<td>0</td>
<td>5.34</td>
<td>0</td>
<td>3.04</td>
<td>0</td>
<td>3.80</td>
</tr>
<tr>
<td>Cropping 15%</td>
<td>2.32</td>
<td>6.09</td>
<td>3.65</td>
<td>6.45</td>
<td>3.15</td>
<td>7.06</td>
</tr>
<tr>
<td>Local random bending</td>
<td>4.78</td>
<td>47.95</td>
<td>3.07</td>
<td>44.65</td>
<td>5.56</td>
<td>44.73</td>
</tr>
<tr>
<td>Rotation 5° + cropping 10%</td>
<td>4.33</td>
<td>8.42</td>
<td>4.76</td>
<td>10.71</td>
<td>3.92</td>
<td>12.02</td>
</tr>
<tr>
<td>Scaling 2.0 + rotation 5°</td>
<td>3.59</td>
<td>17.82</td>
<td>4.11</td>
<td>14.26</td>
<td>3.64</td>
<td>15.02</td>
</tr>
<tr>
<td>Rotation 45° + translation (V 30)</td>
<td>4.36</td>
<td>41.72</td>
<td>4.61</td>
<td>43.33</td>
<td>4.57</td>
<td>39.84</td>
</tr>
<tr>
<td>JPEG 70 + Gaussian noise + average filter + Scaling 0.5</td>
<td>4.64</td>
<td>21.47</td>
<td>5.96</td>
<td>22.16</td>
<td>6.33</td>
<td>26.15</td>
</tr>
</tbody>
</table>
where $I$ is the original image and $I'$ is the watermarked version, both with dimensions $M \times N$.

The PSNR of a watermarked image is determined by two main factors. On the one hand, given a fixed number of bits to be embedded, the PSNR is determined by the quantization step size $\Delta$. A larger $\Delta$ leads to a stronger watermark, but results in a lower PSNR, and vice versa. On the other hand, given a fixed $\Delta$ or watermark strength, the number of bits to be embedded decides the PSNR of the watermarked image. The more bits embedded, the lower value of PSNR, and vice versa.

In all our experiments, we chose quantization step size $\Delta = 1.8$ such that the resulting PSNR $> 4$ dB, which guarantees a good watermark transparency. From Table 1, we can know that the proposed scheme is better than scheme (Tsai & Sun, 2007) in terms of the transparency.

6.2. Robustness to various attacks

The robustness of the proposed watermarking system and scheme (Tsai & Sun, 2007) has been tested against different kind of attacks. Simulation results for common signal processing and geometric attacks are shown in Tables 2 and 3, respectively.

In this study, reliability was measured as the bit error rate (BER) of extracted watermark, its definition is

$$\text{BER} = \frac{B}{M \times N} \times 100\%,$$

where $B$ is the number of erroneously detected bits, $M \times N$ is the watermark image dimensions.

7. Conclusion

In image watermarking, the watermark’s vulnerability to geometric attacks has long been a difficult problem. In this paper, a new SVR based image watermarking detection algorithm against geometric attacks is proposed, in which the steady Pseudo-Zernike moments and Krawtchouk moments are utilized. The features of the proposed scheme are as follows: (1) It exhibits high robustness to common signal processing and geometric attacks. (2) The training features (Krawtchouk moments) are selected according to the principle of invariability, which enhance the working efficiency of SVR training and watermarking scheme. (3) The proposed SVR based geometric correction method can be combined with any other watermarking algorithm, and can be used in many areas. In addition, the proposed scheme also has such merits as easy calculation, easily to implement, all these merit enhance the practicality and application value for digital image copyright protection.

Drawbacks of the proposed watermarking scheme are related to its lesser watermark volume. Future work will focus on eliminating these drawbacks.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant Nos. 60773031 and 60873222, the Open Foundation of State Key Laboratory of Networking and Switching Technology of China under Grant SKLNST-2008-1 01, the Open Foundation of State Key Laboratory of Information Security of China under Grant No. 03–06, Liaoning Research Project for Institutions of Higher Education of China under Grant No. 2008351, and the Open Foundation of State Key Laboratory for Novel Software Technology of China under Grant No. A200702.

References


