第 30 卷 第 5 期
2004 年 9 月

自 动 化 学 报
ACTA AUTOMATICA SINICA

Vol. 30, No. 5
Sept., 2004

Feature Based Fragile Image Watermarking Framework[1)]

LIU Fei-Long ZHU Xin-Shan WANG Yang-Sheng
(*Institute of Automation, Chinese Academy of Science, Beijing* 100080)
(E mail: xszhu@mail.pattek.com.cn)

Abstract A feature based fragile image watermarking framework is proposed to resist vector quantization (VQ) attack. Image features are extracted to process the original watermark or generate the watermark so that the embedded watermark is not only unknown to the attacker, but also dependent on the image. Therefore, different images have different embedded watermarks so that it is impossible for the attacker to build VQ codebook. At the same time, the watermark is embedded into the image under consideration of security and localization. The basic requirements for the embedding scheme are given for the trade-off between the security and localization. An actual moment invariants based fragile image watermarking scheme is presented to demonstrate our fragile watermarking framework. Analysis and experimental results demonstrate that our algorithm can simultaneously detect even one bit image alteration with graceful localization and resist vector quantization attack without the need of any unique keys or image indexes.

Key words Fragile watermarking, VQ attack, image feature, moment invariants, random block dependent

1 Introduction

With the rapid development of digital watermarking technique, fragile watermarking schemes have been presented for the purpose of authentication and verification of content integrity. A fragile watermark can be provided to decide whether the media has been tampered with. Normally, besides publicity, there are two basic requirements for fragile watermarking: 1) Localization, *i. e.*, fragile watermarking should detect tamper with graceful localization ability. 2) Security, *i. e.*, fragile watermarking should be able to detect any tamper under attacks with low error probability. Most of current proposed fragile watermarking schemes are block-wise independence based schemes.

Schyndel[1] embedded the digital watermark in the least significant bit plane directly. Wolfgang[2] extended Schyndel's work to implement a fragile watermarking by VW2D. Walton[3] hid the key-dependent check-sums of the seven most significant bits (MSBs) of grayscales in the least significant bits (LSBs) of pixels. Wong[4] described a scheme to divide the image into blocks and hide the hash function output of MSBs into the LSBs of blocks. Yeung[5] put forward a look-up table scheme to authenticate individual pixels. Although these schemes have good localization ability, they are not secure because they could not thwart the VQ attack successfully[6,7].

Our paper is organized as follows. In Section 2, we will introduce the background of VQ attack and some proposed schemes to impede VQ attack. Then, feature based fragile watermarking framework is presented in Section 3. An actual example of this fragile watermarking framework and experimental results are given in Section 4 to demonstrate the performance of our framework. Finally, conclusions are given in Section 5.

2 VQ attack

Holliman and Memon[7] proposed a counterfeiting attack on block-wise independent

1) Supported by the National 863 Plan(2003AA114020)
Received January 30, 2003; in revised form October 8, 2003
收稿日期 2003-01-30; 收修改稿日期 2003 10-08

watermarking schemes. Based on the same watermark and same key for each image, the watermark embedding function of block-wise independent watermarking schemes can be represented as

$$\varepsilon_K(X,W) = \varepsilon_{K_1}(X_1,W_1) \parallel \varepsilon_{K_2}(X_2,W_2) \parallel \cdots \parallel \varepsilon_{K_n}(X_n,W_n) \tag{1}$$

where K_1, K_2, $\cdots$, K_n are the keys derived solely from the watermark insertion key K, watermark $W=W_1 \parallel W_2 \parallel \cdots \parallel W_n$ with $\parallel$ denoting concatenation, $X=X_1 \parallel X_2 \parallel \cdots \parallel X_n$ means the image blocks and ε means watermarking embedding function. Then, the corresponding detection function is

$$\hat{\boldsymbol{W}} = D_{K_i}(\hat{\boldsymbol{X}}') = D_{K_i}(\hat{\boldsymbol{X}}_1') \parallel D_{K_i}(\hat{\boldsymbol{X}}_2') \parallel \cdots \parallel D_{K_i}(\hat{\boldsymbol{X}}_n') = \boldsymbol{W}_1 \parallel \boldsymbol{W}_2 \parallel \cdots \parallel \boldsymbol{W}_n \tag{2}$$

Given a key K, two image blocks X_i and X_j are K-equivalent if

$$D_K(X_i) = D_K(X_j) = W \tag{3}$$

The consequence of this property for a block-wise independent watermarking scheme is that a set of image blocks belong to the same K-equivalent class. This makes the VQ attack successfully.

To resist the VQ attack, the basic idea is to make the VQ codebook more difficult or impossible to construct. There are several proposed schemes[7~13].

1) Increasing block size or including the block indices in the signatures

The first proposed method[7] to resist VQ attack is to increase the block size and include the block indices into the signature. Although this method can make it more difficult to construct VQ codebook, they can not avoid VQ attack completely. When a large database of watermarked images is given, it is feasible to implement the VQ attack. In addition, increasing block size will damage the scheme's ability to locate the tamper.

2) Including image indices in the signature

An alternative scheme[7,8] is to allocate a unique index for a unique image, which makes different images have different watermarks or signatures. Therefore, it can effectively resist VQ attack because the VQ codebook can not be built. However, because of the necessity of index value during verification, these schemes are unsuitable for actual application when there are large number of images need to authenticate. To avoid this limitation, Wong and Memon[7,10] suggested to extracting the image indices from image itself. However, their suggested scheme, which extracts image indexes by the Hash function of the MSBs of the whole image, will completely impair the localization ability of the watermark even only one pixel's MSBs have been altered. So does Byun's scheme[9].

3) Neighborhood dependent blocks

The main idea to impede the VQ attack of neighborhood dependent blocks is to remove the block-wise independence of the watermark embedding. The watermark embedding in one block is dependent on both the block itself and its surrounding neighborhood blocks. Therefore, this method[7] can resist VQ attack successfully. However, it has two disadvantages. First, once a block $\boldsymbol{x}_i$ is tampered with, the detection will lead to a result that all those blocks that depend on $\boldsymbol{x}_i$ have been tampered with, which damages the localization of detection. Second, once a big block is attacked by collage attack, the detection result is the surrounding blocks of this big block have been tampered with while the inner of this big block is authentic, which make the detection impossible to distinguish the tamper in this block from the tamper of the surrounding blocks of this block. Paulo's scheme[11] has these two disadvantages too.

4) Hierarchical block based

A successful hierarchical block based fragile watermarking scheme has been proposed by Celik[12]. They divide the image into blocks in a multi-level hierarchy and calculate the block signature in this hierarchy. The signature in the low-level hierarchy is employed to detect the forgeries while the signature in high-level hierarchy is used to thwart the VQ at-

tack. This method can successfully resist VQ attack.

5) Random block dependent

To thwart the VQ attack and avoid the disadvantages of surrounding block dependent fragile watermarking, Feilong Liu[13] proposed an improved block dependent fragile watermarking. Image and watermark are decomposed into blocks and sort randomly correspondingly. Two signatures, which are generated by the image block itself and its prior block, are embedded into the LSBs of the image block. Therefore, watermark embedding in this scheme is dependent not only on the image block but also on its prior block. Decision strategy is used to detect where the image has been tampered with. This scheme can resist VQ attack successfully. Especially, this scheme can distinguish the simple tamper from the collage or VQ attack when only one kind attack happens.

In this paper, we impede the VQ attack by generateding a different watermark for each image so that the VQ codebook is impossible to build. Different from these proposed schemes, we extract the image indices from stable image features so that our indices are unique for individual images and can keep stable when the watermarked image has suffered moderate alteration.

3 Feature based fragile image watermarking framework

3.1 Watermark generation

The basic idea of our scheme to thwart the VQ attack is to generate unique and unknown watermark for different image to make K-equivalent impossible. Therefore, to implement this goal for our scheme, several requirements for watermark are needed.

1) Unique, *i.e.*, a different image should have a different watermark.

2) Stable. Because of the need for watermark in tamper verification, watermark should be stable after the image has been tampered with moderately.

3) Secure, *i.e.*, the watermark should not be easy to find by the attacker even the process to generate the watermark is known to the attacker.

As Fig. 1 displays, to generate the watermark, several steps are processed.

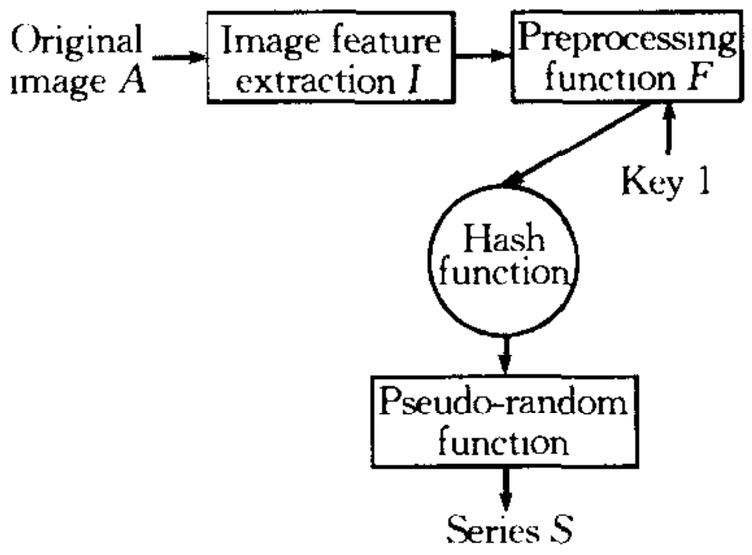

Fig. 1 Watermark generation

1) Image features are extracted firstly. To meet the watermark requirements of uniqueness and stability, how to choose image features is far more important for the success of our scheme. Two characteristics of image features are necessary: a) Image features should have different values for different images so that we can distinguish different images; b) Image features should be stable when images have suffered from some image processing such as median filtering, noise adding, altering moderately *etc*. For example, we extract the image moment invariants as the image features because the image moment invariants have these two characteristics, which have different values for different images and can keep stable after altering.

2) After the image features have been extracted, a preprocessing function is used to preprocess these features to yield the same value even after the image has been tampered with moderately. Therefore, the preprocess function should have the ability to endure some change of image features after the image has been suffered from moderate alteration.

3) Finally, these outputs of preprocessing function are used to generate series S, which can be used as watermark W or transitional data to process the original watermark to acquire the final embedded watermark W. To keep the watermark secure, we need to control the way to generate series S. For example, we can use a secret key to encrypt these outputs of preprocessing function so that the attacker could not find the data when

he does not know the key. After that, some functions such as Hash function are applied to make the output different from each other largely. Finally, the output of Hash function can be used as the key of random function to generate the series S.

3.2 Watermark embedding

To detect local tampering with low error probability, we need to harmonize the trade-off between the localization and security when we embed watermark. Suppose the M-by-N original image is $A=\{a_{i,j}\}\in F^{M\times N}$, $i=1,2,\cdots,M$; $j=1,2\cdots,N$, the watermark $W=\{w_{ij}\}\in F^{M\times N}$, $i=1,2,\cdots,M$; $j=1,2\cdots,N$, the watermarked image $A'=\{a_{i,j}\}\in F^{M\times N}$, $i=1,2,\cdots,M$; $j=1,2\cdots,N$, where F can be in the integral domain Z, real number domain R or the complex number domain C. Then the fragile image watermark embedding can be expressed as

$$\begin{aligned} A' &= f(A,W) \\ Q(A,A') &\leqslant 0 \end{aligned} \tag{4}$$

where $f(\ )$ is the scheme to embed watermark, $Q(\ )$ is the constraint function, which make the watermark imperceptible in watermarked image, such as visual constraint function or other signal constraint function such as $PSNR>40$db (where $Q(A,A')=40-PSNR$) *etc.*

Before choosing a watermarking embedding function, two factors are needed to consider. a) Localization. To detect the tampering locally, watermark should be embedded locally such as in one pixel or small image block. b) Low error detection probability. Once the watermarked image has been tampered with, even only a bit alteration, we should detect it with high probability. Moreover, the embedding method should have the ability to detect the tamper after attacked by some attack methods. However, these two factors contradict each other. To acquire good localization ability, watermark embedding should base on smaller block, which will impair the ability to detect the alteration with low error probability simultaneously.

The watermark embedding process in our framework can be described as Fig. 2. We first generate the watermark as Section 3.1 explains. Watermark is embedded into the original image by the watermark embedding scheme under the consideration of localization and low error probability. Finally, we can acquire the watermarked image A' as we express in formula 4.

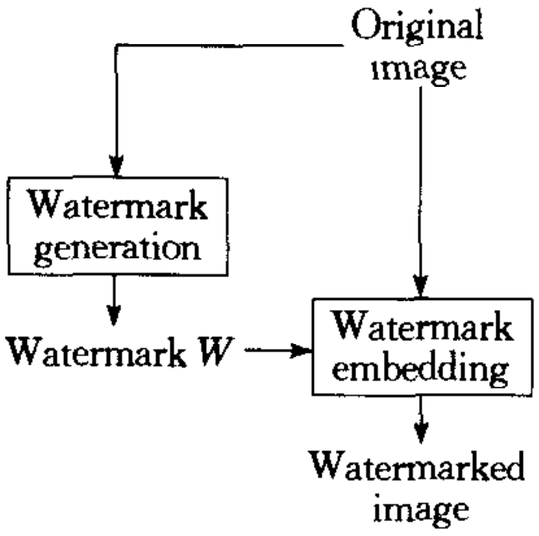

Fig. 2 Watermark embedding

3.3 Tamper detection

As a fragile watermarking scheme, tamper detection must be public. The tamper detection can be explained as follows.

$$\begin{aligned} W' &= f^{-1}(A_{\text{test}}) \\ W' &= W \end{aligned} \tag{5}$$

After comparing the extracted watermark with the original watermark, we can detect where the image has been tampered with. From the process of tamper detection shown in Fig. 3, we can know that choosing more stable image features is far more important for the correction of tamper detection in this feature based fragile watermarking framework.

4 An actual example

4.1 Watermark generation

To thwart vector quantization attack efficiently and completely, a very simple means is to have a unique watermark, which is unknown to the attacker, for a unique image. Most of the researchers generate this kind watermark by unique indexes or keys. However, this is impractical in applications. Therefore, we extract the index from the image it-

self by processing the image moment invariants.

The use of two dimensional moment invariants was highly recognized in the area of pattern recognition. Hu[14~16] has developed some moments that are invariant to geometric manipulations, namely, orthogonal and affine transformations. For completeness, we review his result as follows[17].

For an image $f(x,y)$, its geometric moments $m_{p\cdot q}$ is defined as

$$m_{p\cdot q} = \iint_{\Gamma} x^p y^q f(x,y) \tag{6}$$

and the central moments

$$\mu_{p\cdot q} = \iint_{\Gamma} (x-\bar{x})^p (y-\bar{y})^q f(x,y) \tag{7}$$

where Γ is the support of the image, the image centroids are

$$\bar{x} = m_{1,0}/m_{0,0}, \qquad \bar{y} = m_{0,1}/m_{0,0} \tag{8}$$

Define the normalized central moments $\eta_{p,q}$

$$\eta_{p,q} = \frac{\mu_{p,q}}{(\mu_{0,0})^{\gamma}}, \quad \gamma = (p+q+2)/2 \tag{9}$$

After that, the moment invariants, which have been developed by Hu[14~16], can be formulated as follows.

$$\begin{aligned}
\phi_1 &= \eta_{2,0} + \eta_{0,2} \\
\phi_2 &= (\eta_{2,0} - \eta_{0,2})^2 + 4\eta_{1,1}^2 \\
\phi_3 &= (\eta_{3,0} - 3\eta_{1,2})^2 + (\eta_{0,3} - 3\eta_{2,1})^2 \\
\phi_4 &= (\eta_{3,0} + \eta_{1,2})^2 + (\eta_{0,3} + \eta_{2,1})^2 \\
\phi_5 &= (\eta_{3,0} - 3\eta_{1,2})(\eta_{3,0} + \eta_{1,2})[(\eta_{3,0} + \eta_{1,2})^2 - 3(\eta_{0,3} + \eta_{2,1})^2] + \\
&\quad (\eta_{0,3} - 3\eta_{2,1})(\eta_{0,3} + \eta_{2,1})[(\eta_{0,3} + \eta_{2,1})^2 - 3(\eta_{3,0} + \eta_{1,2})^2] \\
\phi_6 &= (\eta_{2,0} - \eta_{0,2})[(\eta_{3,0} + \eta_{1,2})^2 - 3(\eta_{0,3} + \eta_{2,1})^2] + 4\eta_{1,1}(\eta_{3,0} + \eta_{1,2})(\eta_{0,3} + \eta_{2,1}) \\
\phi_7 &= (3\eta_{2,1} - \eta_{0,3})(\eta_{3,0} + \eta_{1\ 2})[(\eta_{3,0} + \eta_{1,2})^2 - 3(\eta_{0,3} + \eta_{2,1})^2] + \\
&\quad (\eta_{3,0} - 3\eta_{2,1})(\eta_{0,3} + \eta_{2,1})[(\eta_{0,3} + \eta_{2,1})^2 - 3(\eta_{3,0} + \eta_{1,2})^2]
\end{aligned} \tag{10}$$

$$\begin{aligned}
\psi_1 &= [\mu_{2,0}\mu_{0,2} - \mu_{1,1}^2]/\mu_{0,0}^4 \\
\psi_2 &= [\mu_{3,0}^2\mu_{0,3}^2 - 6\mu_{3,0}\mu_{2,1}\mu_{1,2}\mu_{0,3} + 4\mu_{3,0}\mu_{1,2}^3 + 4\mu_{0,3}\mu_{2,1}^3 - 3\mu_{1,2}^2\mu_{2,1}^2]/\mu_{0,0}^{10} \\
\psi_3 &= [\mu_{2,0}(\mu_{1,2}\mu_{0,3} - \mu_{1,2}^2) - \mu_{1,1}(\mu_{0,3}\mu_{3,0} - \mu_{2,1}\mu_{1,2}) + \mu_{0,2}(\mu_{3,0}\mu_{1,2} - \mu_{2,1}^2)]/\mu_{0,0}^7 \\
\psi_4 &= [\mu_{2,0}^3\mu_{0,3}^2 - 6\mu_{2,0}^2\mu_{1,1}\mu_{1,2}\mu_{0,3} - 6\mu_{2,0}^2\mu_{0,2}\mu_{2,1}\mu_{0,3} + 9\mu_{2,0}^2\mu_{0,2}\mu_{1,2}^2 + 12\mu_{0,2}\mu_{1,1}^2\mu_{2,1}\mu_{0,3} + \\
&\quad 6\mu_{2,0}\mu_{1,1}\mu_{0,2}\mu_{3,0}\mu_{0,3} - 18\mu_{2,0}\mu_{1,1}\mu_{0,2}\mu_{2,1}\mu_{1,2} - 8\mu_{1,1}^3\mu_{3,0}\mu_{0,3} - 6\mu_{2,0}\mu_{0,2}^2\mu_{3,0}\mu_{1,2} + \\
&\quad 9\mu_{2,0}\mu_{0,2}^2\mu_{2,1}^2 + 12\mu_{1,1}^2\mu_{0,2}\mu_{3,0}\mu_{1,2} - 6\mu_{1,1}\mu_{0,2}^2\mu_{3,0}\mu_{2,1} + \mu_{0,2}^3\mu_{3,0}^2]/\mu_{0,0}^{11}
\end{aligned} \tag{10}$$

In actual applications, these moment invariants are processed as

$$\begin{aligned}
\phi^* &= |\log_{10}(\phi)| \\
\psi^* &= |\log_{10}(\psi)|
\end{aligned} \tag{11}$$

From the experimental results in table 1 and Masoud[17], we can find that there are two special characteristics of the image moment invariants. One is that different image has different moment invariants' value, which is useful for engendering the different watermark for different image. The other is that these moment invariants not only are invariant to geometric manipulation, but also can keep stable when the image has been modified moderately. These characteristics are very important to devise a graceful fragile watermarking scheme.

As Fig. 4 explains, after image moment invariants[14] $\boldsymbol{I} = [\phi_1^*, \phi_2^*, \phi_3^*, \phi_4^*, \phi_5^*, \phi_6^*, \phi_7^*, \psi_1^*, \psi_2^*, \psi_3^*, \psi_4^*]$ are calculated, we use the preprocessing function F to process these moment invariants to produce a series array. How to choose function F is very important for the following two reasons: a) To avoid altering the watermark once the original image has

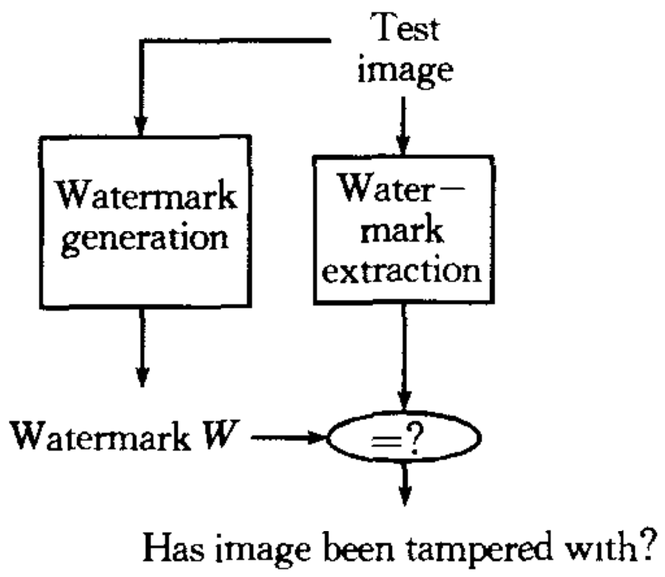

Fig. 3 Tamper detection

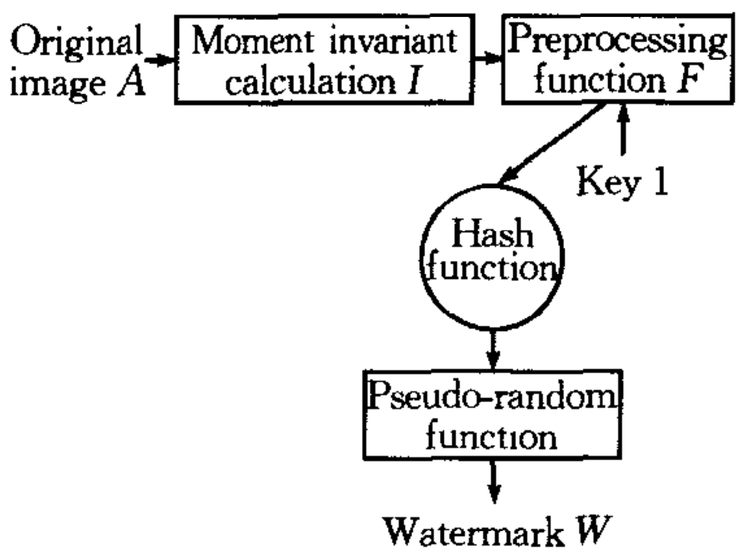

Fig. 4 Watermark generation

been tampered with moderately, function F should have the ability to keep the output same; b) To avoid speculating the watermark from function F, function F should have the ability to keep the output secure. In this paper, two steps are used to compose the function F. 1) we choose the simple quantization function to process these invariants first as follows

$$Q_I = F(I) = 2x \begin{cases} ceil(I/a) + 0.5, & \text{when } 0.25 < I/a - ceil(I/a) < 0.75 \\ ceil(I/a), & \text{when } 0 < I/a - ceil(I/a) < 0.25 \\ ceil(I/a) + 1, & \text{when } 0.75 < I/a - ceil(I/a) < 1 \end{cases} \tag{12}$$

where $ceil(I)$ functioan is to calculate the max integer of I which is smaller than I, a is the scale of quantization, which is chosen empirically by the changing area of these moment invariants when the watermarked image has suffered moderate alteration. 2) A key *key* 1 is used to permute the series, which is made up of the outputs of the preprocessing function of these image moment invariants. After that, this series is inputted into hash function to generate the output H. The output of hash function H is used as the key to generate pseudo-random single polar watermark W, which is made up of 0 and 1 series with the size of original image.

Table 1 Moment invariants of 6 512×512 gray images

	Baboon	Crowd	Lake	Lenna	Pepper	Woman
ϕ_1^*	2.8903	2.7341	2.9040	2.9103	2.8551	2.7784
ϕ_2^*	8.5079	7.5609	7.4780	8.4089	7.7766	7.2323
ϕ_3^*	12.1956	11.8196	11.0071	12.2636	10.6872	10.4646
ϕ_4^*	12.1378	10.5972	10.8983	11.2849	11.5577	11.6265
ϕ_5^*	24.5487	22.3391	21.9078	23.4721	22.7184	22.7106
ϕ_6^*	16.7002	14.3866	14.8832	15.5571	15.5210	15.3880
ϕ_7^*	24.8699	21.5378	22.0470	24.5220	22.7119	23.3343
ψ_1^*	6.3835	6.0737	6.4194	6.4237	6.3160	6.1680
ψ_2^*	24.9975	22.3999	22.4600	23.7023	23.5217	23.1557
ψ_3^*	16.2391	14.7533	14.9414	15.8292	14.9459	14.4213
ψ_4^*	21.7492	20.7753	20.4706	22.0099	20.1824	19.7592

4.2 Watermark embedding

A better fragile watermarking scheme should reach a good trade-off between localization and security. Suppose the M-by-N original image $A=\{a_{i,j}\}\in F^{M\times N}$, $i=1,2,\cdots,M$; $j=1,2,\cdots,N$, the watermark $W=\{w_{ij}=0 \text{ or } 1\}\in F^{M\times N}$, $i=1,2,\cdots,M$; $j=1,2,\cdots,N$, (which has been encrypted to keep it secure), the watermarked image $A'=\{a_{i,j}\}\in F^{M\times N}$, $i=1,2,\cdots,M$; $j=1,2,\cdots,N$, where F can be integer domain Z, real number domain R or the complex number domain C. As Fig. 5 shows, considering the trade-off between localization and security, 1) we firstly decompose original $M\times N$ image into k by l non-overlapping blocks $(A_{11},A_{12},\cdots,A_{mn})$ with $m=M/k$, $n=N/l$. So does watermark W. 2) A key

key 1 is used to sort these non-overlapping blocks randomly so that we can acquire the sorted blocks of original image ($B(1), B(2), \cdots, B(i,), \cdots, B(mn)$) and corresponding watermark blocks ($W(1), W(2), \cdots, W(i), \cdots, W(mn)$). After that, each corresponding watermark block $W(i)$ is decomposed into two halves $W_1(i)$ and $W_2(i)$. 3) Except the first block, each sorted block $B(i)$ and its prior sorted block $B(i-1)$ are processed to generate two series. The first bit stream series, which are composed by all bits of all the pixels in prior sorted block $B(i-1)$ and all 7 MSBs of all pixels in this sorted block $B(i)$ with random order that is encrypted by *key* 2, are inputted into MD5 hash function to generate 128bits output H. The 128bits output H is decomposed into $\boldsymbol{p}=128/(2kl)$ parts $H_1, \cdots, H_p$, and do *XOR* operation with $W_1(i)$, the first half of the corresponding watermark block $W(i)$, bit by bit to acquire the first series M_1 as

$$M_1 = H_1 \oplus \cdots \oplus H_p \oplus W_1(i) \quad (13)$$

Then, the second bit stream series, which are composed by all 7 MSBs of all pixels in this sorted block $B(i)$ with random order that is encrypted by *key* 3, are inputted into MD5 hash function to generate 128bits output HH. The 128bits output HH is decomposed into $\boldsymbol{p}=128/(2kl)$ parts $HH_1, \cdots, HH_p$, and do *XOR* operation with $W_2(i)$, the second half of the corresponding watermark block $W(i)$, bit by bit to acquire the second series M_2 as

$$M_2 = HH_1 \oplus \cdots \oplus HH_p \oplus W_2(i) \quad (14)$$

4) Finally, M_1, M_2 are embedded into the corresponding first half and second half of the LSBs of this block $B(i)$, respectively.

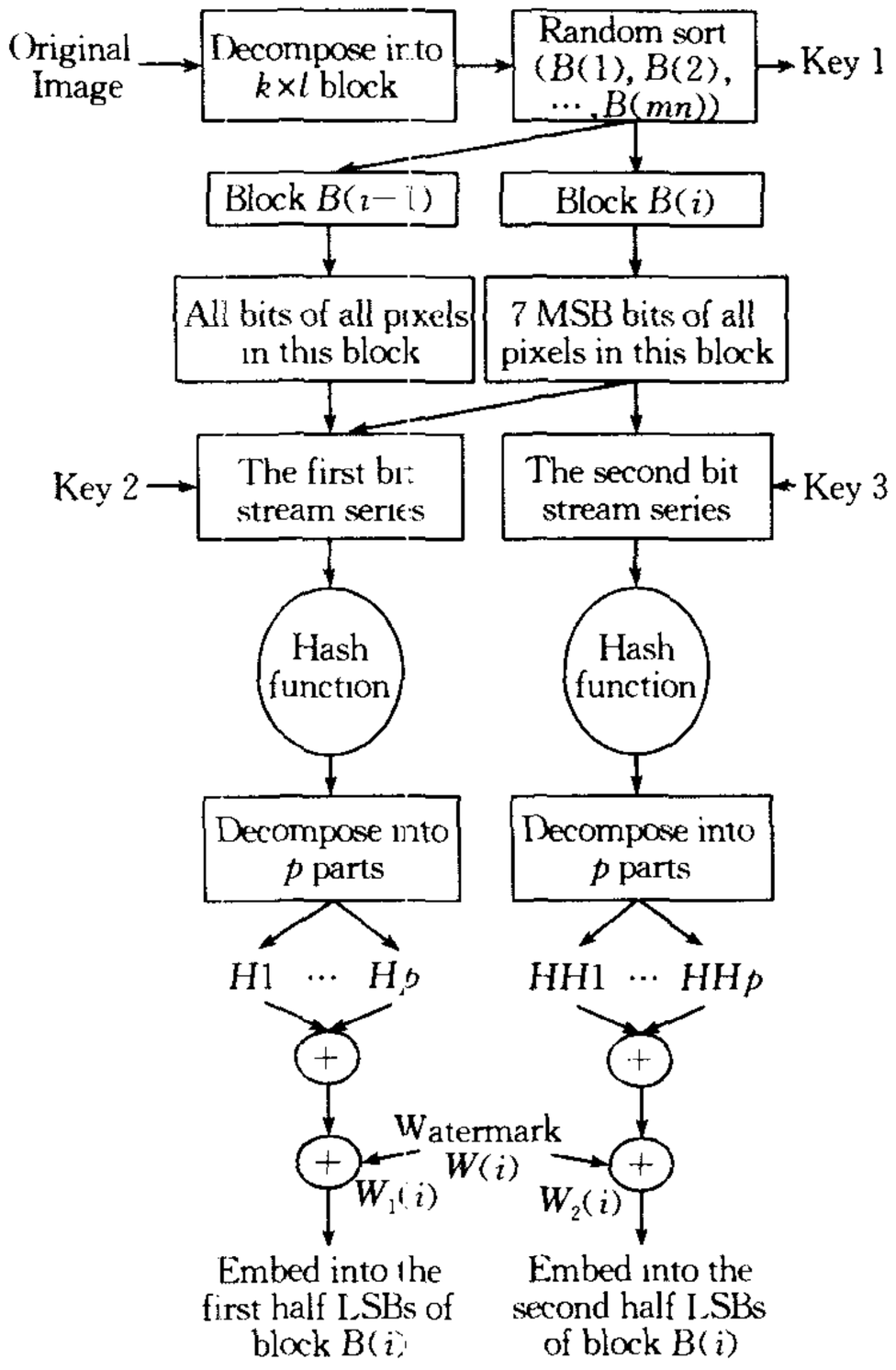

Fig. 5 Watermark embedding

From the view of watermarking embedding, which is not only dependent on the block itself, but also dependent on its prior block, it is impossible to identify the equivalent class at the different places in an image or at the same places in different watermarked images. Therefore, besides the different watermark, we apply the special watermarking embedding scheme to resist VQ attack too.

4.3 Tamper detection

As in Fig. 6, 1) We firstly generate these two series M_1, M_2 as watermarking embedding. 2) Then compare these two series with the corresponding first half and the other half LSBs of this image block to acquire the detection result of this image block. So do all other image blocks. 3) Finally, cooperated with detection result of its posterior block, we can decide whether the image block has been tampered with using following decision strategy.

Assuming that

a) Detection results of each block

$R_1(i)$: $M_1 = ?$ The first half LSBs

$R_2(i)$: $M_2 = ?$ The second half LSBs

i means the ith block, $R_j(i)=1$ if equal, $R_j(i)=0$ if unequal, $j=1,2$;

b) Output of tamper detection

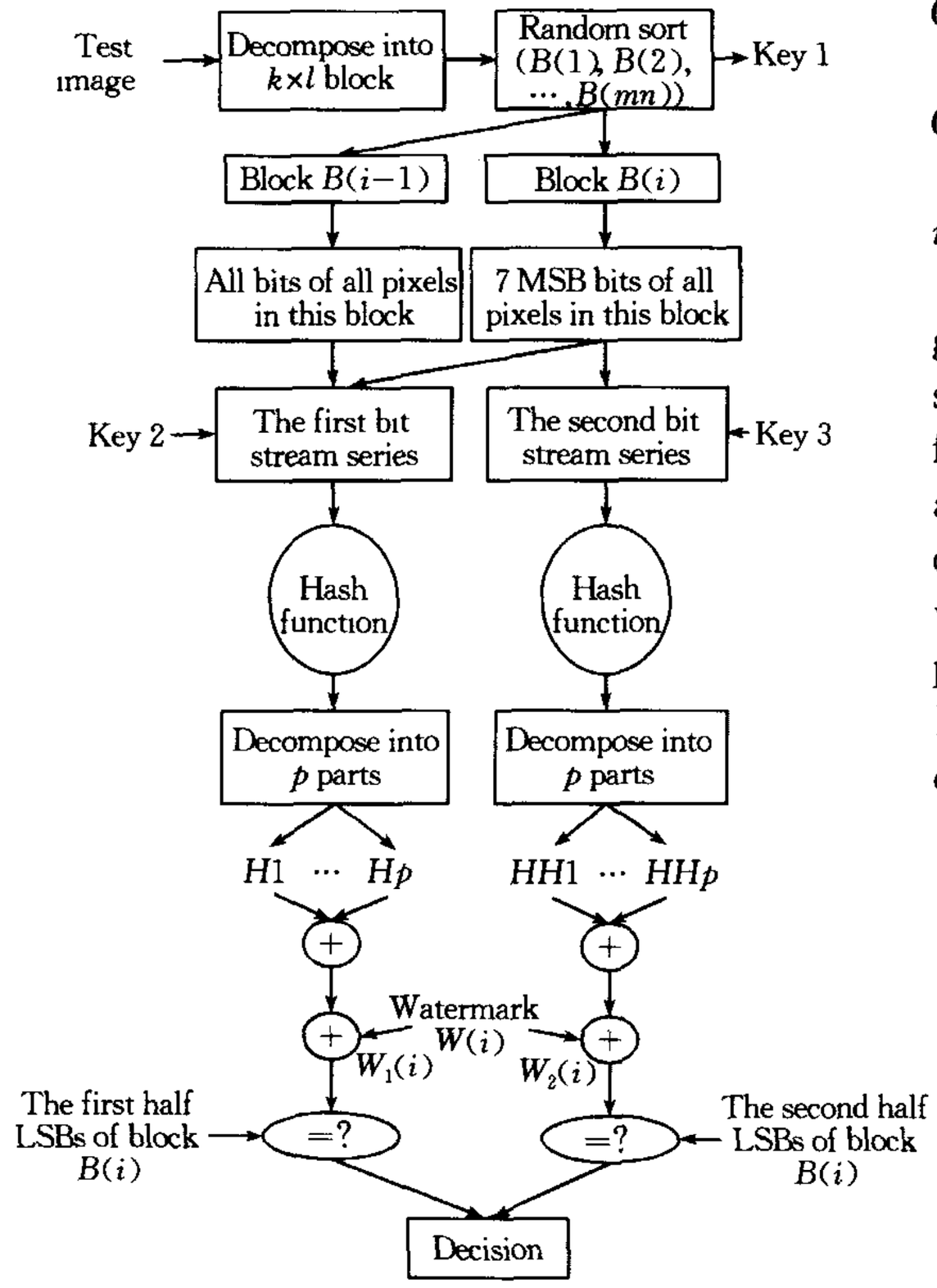

Fig. 6 Tamper detection

$O1$: $O(i)=1$, if the ith block is authentic

$O2$: $O(i)=0$, if the ith block is tampered with

i means the ith block.

Hash function has the property that given an input bit string and its corresponding output, it is computationally infeasible to find another input bit string of any length that will be hashed to the same output. Moreover, the output series will vary "widely" for any change to the input. Based on these characteristics of Hash function, we can use the following decision strategy:

1) if $R_1(i)=1$, then $O(i)=1$; end

2) if $R_1(i)=0$,

if $R_2(i)=0$, then $O(i)=0$; end

if $R_2(i)=1$,

if $R_1(i+1)=1$, then $O(i)=1$; end

if $R_1(i+1)=0$, then $O(i)=0$; end

end

end

From this decision strategy, we can acquire the decision that where the image has been tampered with and where the image is authentic. Moreover, we can distinguish the random altering from the collage attack or VQ attack by our decision strategy when only one kind tamper happens. Normally, random altering is to alter the pixels in image block directly so that $R_2(i)=0$, but the collage attack or VQ attack is to cut and copy image blocks from one watermarked image to another watermarked image, which could not alter the pixels in image directly so that $R_2(i)=1$. Based on this different characteristic between the random altering and collage attack or VQ attack and the characteristics of Hash function, we can distinguish the random altering from the collage attack or VQ attack by the following decision strategy when only one kind tamper happens.

For any tamper, $R_1(i)$ must be equal to $0(R_1(i)=0)$. Then,

1) when $R_1(i+1)=0$ and $R_2(i)=0$, random altering

2) when $R_1(i+1)=0$ but $R_2(i)=1$, collage attack or VQ attack.

A wrong decision may happen when $R_1(i)=0$, $R_2(i)=1$ and $R_1(i+1)=0$. Our decision is that block $B(i)$ has been tampered with. But, this decision can be acquired too when this block has not been tampered with while its posterior block has been altered. However, this situation can happen only when the prior block $B(i-1)$ and the posterior block $B(i+1)$ of this block $B(i)$ all have been altered simultaneously. Moreover, because these image blocks are sort randomly, the probability for this situation happens is small. Therefore, this wrong decision has little effect on our scheme.

4.4 Experimental results

To test the performance of our fragile watermarking algorithm, two kind tests have

been done. The first test we did is randomly altering the pixel value of image. The second is collage attack that cut and paste image blocks from itself or other watermarked image. The popular 512 by 512 gray-level image Lenna is used as the test image to be authenticated. 10 image moment invariants $I=[\phi_1^*, \phi_2^*, \phi_3^*, \phi_4^*, \phi_5^*, \phi_6^*, \psi_1^*, \psi_2^*, \psi_3^*, \psi_4^*]$ are chosen, (1, 2, 2, 2, 4, 3, 2, 4, 3, 4) as the scale of quantization for these 10 moment invariants[17]. The image is decomposed into 8×8 blocks. All these $512/8\times512/8$ image blocks are sorted randomly. Private key symmetric encryption function is used to encrypt the bits series, which is composed by the pixels of image blocks (Certainly, we can employ some more complicated encryption algorithms to encrypt the bits series to make the bits series more secure.). After that, MD5 Hash function is used to process this encrypted bits series. The experimental results are given in Fig. 7.

(a) Original image (b) Watermarked image (c) Randomly tampered watermarked image

(d) Detection result of randomly tampered watermarked image (e) Counterfeit watermarked image (f) Detection result of counterfeit watermarked image

Fig. 7 Experimental results (Darkest place in detection result means the places where image has been tampered)

Fig. 7(a) is the original 512×512 gray-level Lenna image. The watermarked image is given in Fig. 7(b). We alter this watermarked image randomly by only one pixel. The altered watermarked image, which could not give any visual effect, is presented in Fig. 7(c). The detection result is displayed in Fig. 7(d), which shows that our scheme can detect any small tamper. The next test result is given in Fig. 7(e) and Fig. 7(f). The counterfeit watermarked image, where we alter the hair of the girl on the back of her shoulder, is given in Fig. 7(e). The detection result is shown in Fig. 7(f), which indicates that our algorithm can successfully detect the counterfeit places. As our analysis in Section 3, when the tamper is randomly altering only, we can acquire that $R_1(i+1)=0$ and $R_2(i)=0$. When the tamper is collage and VQ attacks only, we can acquire that $R_1(i+1)=0$ but $R_2(i)=1$. But when these two kind of tampers happen simultaneously, the detection result is same as random altering where $R_1(i+1)=0$ and $R_2(i)=0$. Therefore, in this situa-

tion, we could not distinguish these two kinds of tampers. Besides these tests, the effectiveness of our proposed algorithm against VQ attack is obvious. The VQ attack codebook is impossible to construct so that the VQ attack is unable to implement.

The experimental results demonstrate that our algorithm can successfully detect the random altering and college attack. Because our watermarking embedding is dependent on the different unknown watermark, which is generated by the image moment invariants, and random block dependent, VQ attacker could not identify the equivalence classes. Therefore, it is impossible for VQ attack to implement so that our means can resist the vector quantization attack successfully.

5 Conclusion

In this paper, a feature based fragile image watermarking framework is proposed. The image features are employed to generate the watermark so that the VQ attack is unfeasible. The basic requirements for choosing image feature and watermarking embedding function have been given for this framework. An example, which is moment invariants based fragile image watermarking, demonstrates our framework.

References

1 van Schyndel R G, Tirkel A Z, Osborne C F. A digital watermark. In: Proceedings of the IEEE International Conference on Image Processing, Austin, Texas: IEEE Computer Society Press, 1994, **2**:86～90

2 Wolfgang R B, Delp E J. Fragile watermarking using the VW2D watermark. In: Proceedings of SPIE, Security and Watermarking of Multimedia Contents , San Jose, California: SPIE, Jan 25～27, 1999, 204～213

3 Walton S. Information authentication for a slippery new age. *Dr. Dobbs Journal*, 1995, **20**(4): 18～26

4 Wong P. A watermark for image integrity and ownership verification. In: Proceedings of IS & T PIC, Portland, Oregon, 1998,374～379

5 Yeung M, Mintzer F. An invisible watermarking technique for image verification. In: Proceedings of ICIP 97, Santa Barbara, California, 1997,680～683

6 Jiri Fridrich M Goljan, Memon N. Further attacks on Yeung-Mintzer fragile watermarking scheme. In: Proceedings of SPIE Photonic West, Electronic Imaging 2000, Security and Watermarking of Multimedia Contents, San Jose, California, 2000, 24～26

7 Holliman M, Memon N. Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes. *IEEE Transactions on Image Processing*, 2000,**9**(3): 432～441

8 Fridrich J, Goljan M, Baldoza A C. New Fragile Authentication Watermark for Images. ICIP 2000, Vancouver, Canada, 2000, 10～13

9 Byun S C, Lee L L. Shin T H, Ahn B H. A Public Key Based Watermarking for Color Image Authentication. In: Proceedings of ICASSP 2002

10 Wong P W, Memon N. Secret and public key authentication watermarking schemes that resist vector quantization attack. In: Proceedings of SPIE Security and watermarking of Multimedia, San Jose, USA: 2000,**3971**(40):1593～1601

11 Paulo S L M Barreto, Hae Yong Kim, Vincent Rijmen, Toward a secure public-key blockwise fragile authentication watermarking. In: Proceedings of ICIP 2001, Thessaloniki, Greece: 2001, 494～497

12 Celik M U, Sharma G, Saber E, Tekalp A M. A hierarchical image authentication watermark with improved localization and security. In:Proceedings of ICIP 2001, Thessaloniki, Greece: 2001, 502～505

13 Liu Fei-Long, Wang Yang-Sheng. An improved block dependent fragile watermarking. In: Proceeding of ICME 2003, Baltimove, MD: 2003,501～504

14 Hu Ming-Kuei. Pattern recognition by moment invariants. In:Proceedings of IRE, 1961,**49**:1428

15 Hu Ming-Kuei. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 1962, **8**:179～187

16 Flusser J, Suk T. Pattern recognition by affine moment invariants. *Pattern Recognition*, 1993,**26**(1):167～174

17 Alghoniemy M, Tewfik A H. Image watermarking by moment invariants. In:Proceedings of ICIP 2000,Vancouver, 2000, 73～76

LIU Fei-Long Received his bachelor degree from Institute of Gold Technology, Northeast University, P. R. China in 1995, and master degree from South China University of Technology, P. R. China, in 2000. Currently, he is working in the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include image and signal processing, digital watermarking, Pattern recognition, and wavelet analysis.

ZHU Xin-Shan Received his bachelor and master degrees from Harbin Institute of Technology in P. R. China, in 2000, and 2002, respectively. Currently, he is doing research in Institute of Automation, Chinese Academy of Sciences. His research interests include algorithms for audio, image and video coding and processing, multimedia security, data compression, and data hiding.

WANG Yang-Sheng Received his master and Ph. D. degrees from Huazhong University of Science and Technology, P. R. China, in 1984 and 1989, respectively. Nowadays, he is a professor in National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences.

基于特征的易碎图像水印框架

刘飞龙　　朱新山　　王阳生

（中国科学院自动化研究所　北京　100080）

（E-mail：xszhu@mail.pattek.com.cn）

摘　要　提出了一种基于特征的易碎图像水印框架，来阻止 VQ 攻击. 图像特征被提取出来，用来处理原始水印或生成水印，使得要嵌入的水印信息不仅不被攻击者知道，而且依赖于原始图像. 因此，不同的原始图像嵌入了不同的水印信息，从而使得攻击者无法建立 VQ 码表，因而无法实现 VQ 攻击. 同时，为了提高水印的安全性和篡改的局部检测性，本文给出了该框架下水印嵌入方法的基本要求. 根据该易碎图像水印框架，本文设计了一种基于图像矩不变量的易碎水印算法. 分析和实验结果表明，该算法不仅可以很好地、局部地检测图像中的篡改，即使图像中仅有一位被篡改时. 同时，在不需任何额外的密码或图像索引号的情况下，成功地抵抗 VQ 攻击.

关键词　易碎水印，VQ 攻击，图像特征，矩不变量，随机块依赖

中图分类号　TP391