Robust Hash-based Image Watermarking with Resistance to Geometric Distortions and Watermark-Estimation Attack

Chun-Shien Lu[a], Shih-Wei Sun[a,b], and Pao-Chi Chang[b]

[a]Institute of Information Science, Academia Sinica, Taipei, Taiwan 115, ROC
[b]Dept. of Electrical Engineering, National Central University, Chung-Li, Taiwan 320, ROC

ABSTRACT

Digital watermarking provides a feasible way for copyright protection of multimedia. The major disadvantage of the existing methods is their limited resistance to both extensive geometric distortions and watermark-estimation attack (WEA). In view of this fact, this paper aims to propose a robust image watermarking scheme that can withstand geometric distortions and WEA simultaneously. Our scheme is mainly composed of two components: (i) mesh generation and embedding for resisting geometric distortions; and (ii) construction of hash-based content-dependent watermark (CDW) for resisting WEA. Extensive experimental results obtained from standard benchmark confirm the ability of our method in improving robustness.

Keywords: Attack, Copyright protection, Embedding, Mesh, Hash, Robustness, Watermark

1. INTRODUCTION

Digital watermarking has been recognized as a helpful technology for applications of copyright protection, traitor tracing, and authentication during the last decade. No matter what kinds of applications are considered, robustness is the critical issue affecting the practicability of a watermarking system. In data hiding, robustness refers to the capability of resistance to attacks that are used to destroy or remove hidden watermarks. In,[19] attacks are classified into four categories: (1) removal attacks; (2) geometric attacks; (3) cryptographic attacks; and (4) protocol attacks. Up to now, resistance to extensive geometric attacks is still a challenging issue. Geometric attacks introduce synchronization errors to disable watermark detection without needing to remove the hidden information.

In the literature, the watermarking methods resistant to geometric attacks can be divided into three categories. The first category is to embed the watermark into the geometric invariant domain. In,[6, 10] watermarking is conducted in the Fourier-Mellin domain to exploits its affine invariance. However, Fourier-Mellin domain is inherently vulnerable to cropping and other local geometric distortions.

The methods falling into the second category proposed to use template[11, 12] or insert periodic watermark pattern[5, 18] for the re-synchronization purpose. In,[11, 12] templates were embedded in DFT domain to generate a shape of local peaks, which can be easily retrieved in the detection process for recovering geometric parameters. On the other hand, the local peaks are also easily extracted by the pirates in order to remove the templates.[4] In,[5] the periodical structure of the watermark could be estimated from the autocorrelation function (ACF) to recover the imposed global transforms. However, the global watermark structure can be totally destroyed by means of the local geometric distortions. In,[18] the authors proposed to insert a periodic watermark pattern for the convenience of re-synchronization. The inserted periodic watermark was transformed as a lattice of peaks when ACF is applied in stego or geometrically attacked images. However, since the watermark is identical for every region, the collusion attack[7] can be used to efficiently estimate and remove the hidden watermarks. Although the synchronization problem is somewhat solved, the watermark information still cannot survive in collusion environments.

Further author information: (Send correspondence to Chun-Shien Lu)
Chun-Shien Lu: E-mail: lcs@iis.sinica.edu.tw, Telephone: +886 2 2788 3799 X 1513
Shih-Wei Sun: E-mail: swsun@iis.sinica.edu.tw, Telephone: +886 2 2788 3799 X 1552
Pao-Chi Chang: E-mail: pcchang@ce.ncu.edu.tw, Telephone: +886 3 422 7151 X 4466

Security, Steganography, and Watermarking of Multimedia Contents VII, edited by
Edward J. Delp III, Ping Wah Wong, Proc. of SPIE-IS&T Electronic Imaging,
SPIE Vol. 5681 © 2005 SPIE and IS&T · 0277-786X/05/$15

The third category is called a "feature-based watermarking scheme." The feature points detected in the original image are used to form local regions for embedding. At the detection end, the feature points are expected to be robustly detected. Among the ubiquitous feature point extraction methods, Harris detector[2] has been popularly used in the fields of pattern recognition and computer vision. However, we found Harris detector is still not robust enough to be used in digital watermarking.[1] This is because Harris detector is rotation and scaling-sensitive. In,[16] Mexican-Hat wavelet filtering was used for feature point extraction. The Mexican-Hat wavelet filtering was implemented in frequency domain using FFT. Although 1-D FFT is widely used in implementing 2-D FFT to improve the computation efficiency, this implementation may lead to another severe problem. That is, the input coefficient of 1-D FFT is quite different from the rotated version such that different 1-D FFT filters will lead to different filtering results. This is mainly due to that asynchronization effect is propagated to the final result of Mexican-Hat wavelet filtering. In,[15] the scale-space theory was applied for feature point extraction in that feature points were determined by automatic scale selection together with local extrema detection. Although the idea of scale-space feature point detection[9] is useful to deal with scaling attacks, this approach is exactly a kind of exhaustive search.

In this paper, a novel robust mesh-based content-dependent image watermarking method is proposed. Our method belongs to the third category of geometric distortion resilient watermarking technologies. This selection is based on our observations that the first category is restricted to be affine invariant, the periodic patterns are easily removed in the second category, and the third category seems to be the best choice for watermarking applications. However the stability of feature points plays a key role in the third category. In view of this fact, we propose to use the Gaussian kernel as the pre-processing filter to stabilize the feature points. The Gaussian kernel is a circular and symmetric filter in that all the neighboring information of a pixel can be equally contributed to filtering. A Gaussian kernel of large size, which is the marginal concept of scale-space theory, is used in our scheme. In order to resist watermark-estimation attacks, image hash[8] is further extracted and combined with the hidden watermarks to generate the Content-Dependent Watermark (CDW).[7] CDW is able to resist watermark estimation attack in that even though the pirates can estimate the watermarks from meshes, they still cannot be successfully colluded to generate an even more correct watermark to remove it. In addition to robustness, we also investigate the false positive issue in determining the proper threshold used to indicate the presence/ absence of a watermark. Experiment results obtained from standard benchmark verify that our scheme outperforms conventional feature-based watermarking methods.[1, 15, 16]

The remainder of this paper is organized as follows. In section 2, we describe two important issues, including robust feature extraction and content-dependent watermark that are fundamental in our method. In section 3, the proposed mesh-based content-dependent watermarking is described. Experimental results are demonstrated in section 4 to verify the performance of our scheme. Robustness comparisons with other methods are also discussed. Finally, conclusions are given in section 5.

2. ROBUST FEATURE EXTRACTION AND CONTENT-DEPENDENT WATERMARK

Two key issues of robust watermarking, including robust feature extraction and content-dependent watermark, will be described in this section.

2.1. Robust Feature Extraction

Since our watermarking method is mesh-based, feature point extraction needs to be robust enough to approximately tolerate common filtering, compression, and geometric attacks for robust mesh generation. In our method, Gaussian kernel filtering, local maximum determination, and scale determination are integrated for feature point extraction.

2.1.1. Gaussian Kernel Filtering

The Gaussian kernel filtering is a special case of scale-space filtering. In scale-space filtering, an image is filtered by more than one filters of different sizes to generate multiple frequency components. In some applications, filter size can be modified to adapt different affine transformation environments. But in digital watermarking, for the purpose of blind detection, we only select a specific filter size to generate one level scale-space, which

is convenient for watermark embedding and detection. Let $I(x, y)$ be a cover image and let Gaussian kernel be defined as

$$g(\sigma) = \frac{1}{2\pi\sigma^2} exp^{(-\frac{x^2+y^2}{2\sigma^2})}, \tag{1}$$

where σ is the standard deviation. The convolution of the Gaussian kernel and the cover image is defined as

$$L(x, y, \sigma) = g(\sigma) * I(x, y). \tag{2}$$

Because the Gaussian kernel is a circular shape, the resultant filtering response is rotation insensitive, which is adopted in our geometric-distortion resilient scheme.

Another important thing is that the selection of σ in the Gaussian kernel $g(\sigma)$. Suppose the Gaussian kernel can be represented with at least k times of standard deviation, and can be described as a 2-D filter of size $(2k * \sigma + 1) \times (2k * \sigma + 1)$. In Gaussian distribution, the probabilities within one, two, and three standard deviations of its mean are about 68%, 95%, and 99.7%, respectively. Therefore, 3 standard deviations of mean could sufficiently represent the energy of a Gaussian distribution and $k = 3$ is adopted here.

2.1.2. Local maximum determination

The local maximum determination process is operated in the Gaussian kernel filtered signal. At first, a maximum filter of size 3×3 is applied to $L(x, y, \sigma)$ and is expressed

$$MF(x, y) = \max_{(x_t, y_t) \in (N_8(L(x,y,\sigma)) \cup L(x,y,\sigma))} \{L(x_t, y_t, \sigma)\}, \tag{3}$$

where $N_8(L(x, y, \sigma))$ denotes the 8-neighborhood of $L(x, y, \sigma)$. Next, the set of feature points is decided as

$$P = \{(x, y) | MF(x, y) = L(x, y, \sigma)\}, \tag{4}$$

which means that a feature point satisfies that the filtering responses of $MF(x, y)$ and $L(x, y, \sigma)$ are the same. In addition, the detected feature points P are utilized to form a set of meshes by means of the Delaunay tessellation. Each mesh is a basic unit for watermark embedding.

2.1.3. How to Choose σ?

When the Gaussian kernel is utilized as the feature point detector, the Gaussian filtering responses play an important role. If a larger σ is used, lower frequency (corresponding to larger scale) information tends to be detected. On the contrary, high frequency (smaller scale) information can be detected when a smaller σ is used. Therefore, which σ should be used is an important issue. The selection of σ's is also relevant to the ability of dealing with scaling attacks because if σ's are not properly used to filter the scaled image, feature points will be wrongly detected.

These problems are dealt with by observing the number of feature points across different σ's (ranging from 2 to 5) for different image sizes, as shown in Table 1. Since at least 3 points are required to form a mesh, it is expected to choose σ's that can produce at least 3 feature points. Let σ_l be the largest value that can still generate at least 3 feature points. In addition, the number of feature points cannot be too large to yield small meshes such that a watermark cannot be completely embedded. According to Table 1, the value of σ_d that can be effective for watermark embedding is set to $\sigma_l - 2(\geq 1)$, which is defined as a detection scale. For example, for a 512×512 image, $\sigma_d = 5 - 2 = 3$ is adopted.

Table 1. Number of detected feature points at different scales for the image Lena.

image size	$\sigma = 2$	$\sigma = 3$	$\sigma = 4$	$\sigma = 5$
64×64	6	-	-	-
128×128	20	6	-	-
256×256	55	18	6	-
512×512	224	55	19	6

2.2. Content-Dependent Watermark

Some researches[1, 15, 16, 18] proposed to insert multiple redundant watermarks into an image with the hope that it suffices to maintain resistance to geometric distortions as long as at least one watermark exists. The common framework is that some kinds of image units such as blocks,[18] meshes,[1] or disks[15, 16] were extracted as carriers for embedding. With this unique characteristic, we propose to treat each image unit in an image like a frame in a video; in this way, collusion attacks can be equally applied to those image watermarking methods that employ a multiple redundant watermark embedding strategy. Therefore, we argue that once the hidden watermarks are successfully estimated by means of a collusion attack, the ability of resisting geometric distortions become fragile so that the false negative problem occurs. Of particular interest is the possible quality improvement of attacked media data by means of collusion attack. In addition, copy attack is also efficient in defeating a watermarking system by creating ambiguity problem. Since the common operation of realizing both the collusion and copy attacks is watermark estimation, they are called watermark-estimation attack (WEA).[7]

In order to withstand watermark-estimation attack, we propose to embed content-dependent watermark (CDW),[7] which is composed of a watermark and a hash. Since this paper investigates a mesh-based watermarking scheme, the mesh-based hash[8] is considered here. For each mesh, its robust hash is extracted in the block-DCT domain. First, each normalized mesh is flipped and padded with its flipped version to form a $L_B \times L_B$ block, which is then divided into subblocks of $L_{sub} \times L_{sub}$. For a pair of $L_{sub} \times L_{sub}$ blocks, a hash bit, defined as the magnitude

$$MH_i(b) = \begin{cases} 1, & \text{if } |f_k(p_1)| - |f_l(p_2)| \geq 0 \\ 0, & \text{otherwise,} \end{cases}$$

where $MH_i(\cdot)$ is a hash bit in a hash sequence MH_i, and $f_k(p_1)$ and $f_l(p_2)$ are two AC coefficients at positions p_1 and p_2 in $L_{sub} \times L_{sub}$ blocks k and l, respectively. Given a pair of hash MH_i and watermark W, a content-dependent watermark can be generated as

$$CDW_i = S(W, MH_i), \tag{5}$$

where $S(\cdot)$ is a shuffling function, which is basically application-dependent and will be used to control the combination of W and MH_i. Please refer[8] for more details about the verification of robustness. The sequence CDW_i is the watermark that we want to embed in each mesh.

3. PROPOSED WATERMARKING METHOD

Basically, the proposed method is similar to the mesh-based watermarking framework.[1] The major difference is that we have investigated some important issues (described in Section 2) to further improve the overall performance. Further, we found from Bas et Al.[1] that the watermark signal is warped from the normalized domain to the spatial domain for embedding, while the extraction process is operated in the normalized domain. This asymmetric embedding and extraction cannot achieve efficient. However, in our proposed scheme, the watermark embedding and extraction process are both performed in the normalized domain. Besides, the modified coefficients in the normalized domain are warped to the spatial domain to accomplish embedding. Therefore, the trade-off between transparency and robustness can be better achieved. In the following, proposed the watermark embedding and extraction processes will be described.

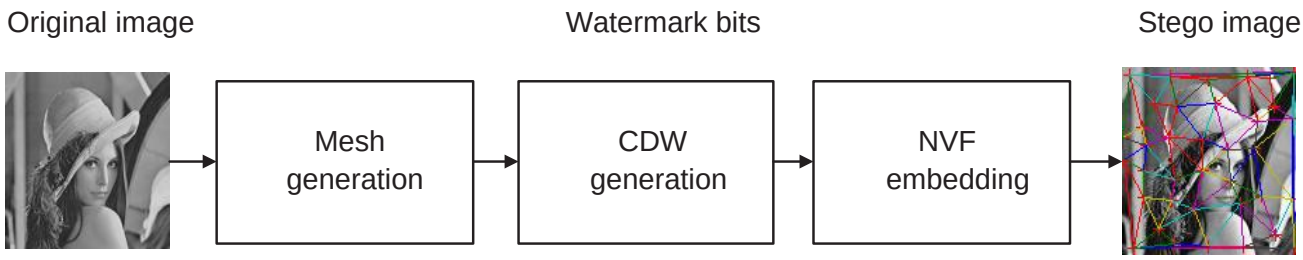

Figure 1. Block diagram of the embedding process.

3.1. Watermark Embedding

The watermark embedding process is outlined in Fig. 1 and described as follows.

3.1.1. Mesh Generation

The first step in mesh generation is to filter a cover image using the Gaussian filtering, as described in Sec. 2, to generate a set of feature points P. Next, the Delaunay tessellation is performed using P to generate a set of meshes, $M = \{M_i\}_{i=1,2\cdots L_M}$, where L_M denotes the number of meshes extracted from a cover image. Each M_i is a basic unit used for watermark embedding and extraction.

3.1.2. Content-Depentent Watermark Generation

The content-dependent watermark generation process including (i) mesh normalization, (ii) media hash extraction, (iii) hash-based content-dependent watermark, and (iv) watermark bit arrangement are described as follows.

Mesh Normalization The mesh normalization process is utilized to affine transform each extracted mesh M_i into a right-angled isosceles triangle, which is called a normalized mesh, NM_i. The goals are not only to extract a fixed-length hash, but also to reduce the effect of image content shifting, caused by the imperfect extraction of feature points. If the watermark signals are embedded in the spatial domain, the shifting problem even with slice loss or pixel loss may make the watermark extraction process fail. Therefore, the size of a normalized mesh needs to be properly determined. Our empirical study finds that a larger region is always warped into a small but fixed region, which means that the warping process is a multiple-to-one pixel mapping. In other words, one pixel in NM_i represents several pixels in M_i. Under this circumstance, fewer pixels in NM_i will be affected by slice missing or shifting. In this study, the size of a normalized mesh is empirically found to be 48×48 to achieve the trade-off between transparency and robustness.

Media Hash Extraction A mesh-based media hash, MH_i, is extracted from each normalized mesh NM_i, as described in Sec. 2.2. The length of a hash sequence is 64.

Hash-based Content-dependent Watermark In this paper, the watermark length (L_W) is set to 128 bits. Although the length of media hash (MH_i) is only *64* bits, by repeating it two times, a media hash of 128 bits can be generated. Then, each media hash MH_i and watermark W are combined to generate the content-dependent watermarks, i.e., $CDW = \{CDW_i\}_{i=1,2\cdots L_M}$. Although there is only one watermark W embedded for a cover image, the principle of CDW would lead to different signals embedded into different meshes.

Watermark Bit Arrangement Since the length of a content-dependent watermark is 128 and the size of a normalized mesh is $(48 \times 48)/2 = 1152$, we propose to repeatedly embed the watermark to enhance robustness, as shown in Fig. 2. It is not hard to calculate that the time of repetition is $\lfloor \frac{1152}{128} \rfloor = 9$. So far, it can be observed that the watermark's length, the hash's length, and the normalized mesh's size are all designed in a sophisticated way to fit the embedding purpose.

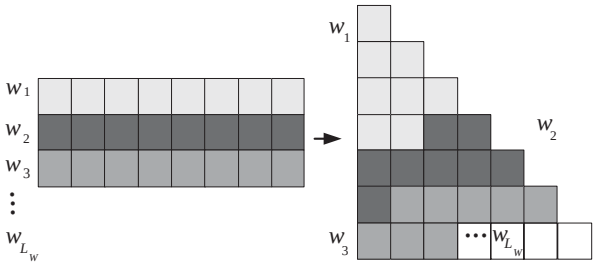

Figure 2. (left) The repeated watermark bits (each bit is repeated 8 times) are arranged and embedded into a normalized mesh (right).

3.1.3. Mesh-based Embedding

In order to maintain transparency after performing watermarking, Noise Visibility Function (NVF),[17] which is an image-dependent visual model, is adopted in this paper. According to,[17] the content adaptive watermark embedding is designed as

$$I^w(x,y) = I(x,y) + (1 - NVF(x,y)) \cdot w_j \cdot S + NVF(x,y) \cdot w_j \cdot S_1, \tag{6}$$

where S and S_1 denote the watermark strength, and w_j is an element of a bipolar watermark signal. Therefore, in our watermarking scheme, the watermark embedding process can be designed as

$$NM_i^w(x,y) = NM_i(x,y) + (1 - NVF(x,y)) \cdot cdw_{ij} \cdot S + NVF(x,y) \cdot cdw_{ij} \cdot S_1, \tag{7}$$

where cdw_{ij} denotes the jth watermark element of CDW_i, which is embedded in NM_i. Once the watermarked normalized mesh NM_i^w is obtained, the inverse normalization process is used to yield a watermarked mesh. Although the direct inverse normalization is intuitive, the transparency may be degraded, because blocking effects are caused by the one-to-multiple pixel mapping. To deal with this problem, the difference between NM_i and NM_i^w is represented as

$$NM_i^d(x,y) = (1 - NVF(x,y)) \cdot cdw_{ij} \cdot S + NVF(x,y) \cdot cdw_{ij} \cdot S_1, \tag{8}$$

which is inversely normalized to yield the difference (caused by watermarking in the normalized domain) in the spatial domain. Hence, the watermarked mesh in the spatial domain can be obtained as

$$M_i^w = M_i + M_i^d. \tag{9}$$

Based on Eqs.(8) and (9), the original high-frequency components of M_i can be preserved to maintain transparency. Finally, by integrating all watermarked meshes, the stego image can be obtained.

3.2. Watermark Extraction

The watermark extraction process described in this section is depicted in Fig. 3. Basically, the watermark extraction process is the inverse process of watermark embedding.

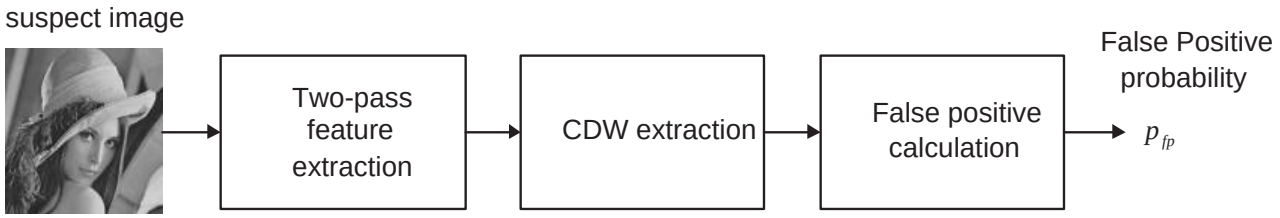

Figure 3. Block diagram of the extraction process.

3.2.1. Two-pass Feature Extraction

The two-pass feature extraction process needs to determine a set of filter sizes, σ_d and σ_f, where σ_d is exactly the same as determined in the embedding end and σ_f is additionally used to deal with cropping. More specifically, σ_d is used to adapt to scaling, while σ_f is used to deal with attacks that are not caused by scaling. Both σ_d and σ_f need to be used because if the suspect image's size is smaller than the cover image's size, the detector does not know whether the image is scaled or cropped. If the suspect image is scaled, σ_d can help to adaptively extract the feature points. On the other hand, if the suspect image is cropped instead of scaled, a fixed value, σ_f, as determined in the embedding end is desired. For images of varying sizes, the watermark embedding and extraction processes should be operated in a manner of tiling. The tile size selected in our proposed scheme is 512×512. As described in Sec. 2.1.3, σ_d is set to 3, for a 512×512 image. Under this circumstance, σ_f determined in the extraction process only falls within the range of $\{1, 2, 3\}$. Therefore, the exhaustive search for σ_f is greatly reduced. With this two-pass feature extraction process, two set of meshes, M and M_f, are generated with respect to σ_d and σ_f, respectively. Each mesh set is used for watermark extraction.

3.2.2. Content-Dependent Watermark Extraction

The proposed content-dependent watermark extraction process is depicted in Fig. 4. The normalization process is utilized to transform M and M_f to the normalized form NM and NM_f, respectively, from which media hash MH and MH_f are extracted.

In this paper, Wiener filtering, is used to blindly extract the hidden signal. Wiener filtering is considered to be an efficient way[3] because watermark is usually a high-frequency signal. Let CDW_i^e and $CDW_{f\,i}^e$ be, respectively, extracted from NM_i and NM_{f_i}. Since the watermark bits are redundantly embedded, a bit is finally decided according to a majority selection rule. In this paper, each bits is embedded into a mesh 9 times. For an embedded bit, if most of its corresponding extracted bits are $1(-1)$, the final bit is determined to be $1(-1)$. Let CDW_i^d and $CDW_{f\,i}^d$ be the extracted watermarks by the majority determination process with respect to their embedded counterparts CDW_i^e and $CDW_{f\,i}^e$, respectively.

Next, two extracted media hashes, MH and MH_f, corresponding to σ_d and σ_f, respectively, are merged with their corresponding watermarks CDW_i^d and $CDW_{f\,i}^d$:

$$W^d = \left\{W_i^d\right\}_{i=1,2\cdots L_M}, W_i^d = (MH_i \cdot CDW_i^d); \tag{10}$$

$$W_f^d = \left\{W_{f\,i}^d\right\}_{i=1,2\cdots L_M}, W_{f\,i}^d = (MH_{f_i} \cdot CDW_{f\,i}^d), \tag{11}$$

to obtain the extracted watermark signals W^d and W_f^d.

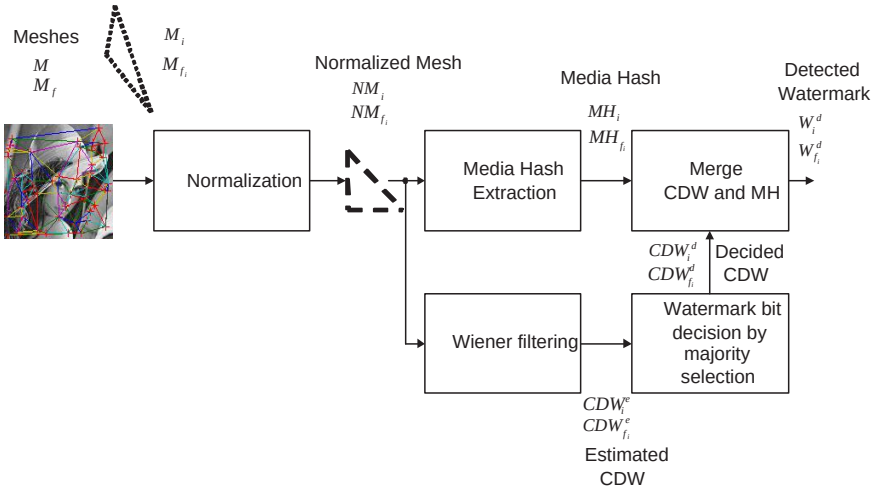

Figure 4. CDW Extraction block diagram.

3.3. Existence of a Watermark and False Positive Analysis

The bit-error rate (BER) between W and W^d (or W and W_f^d) is calculated for each NM_i (or NM_{f_i}). If BER is smaller than a threshold Th, it is said that a watermark exists. The threshold Th needs to be determined by considering false positive because it is meaningful to claim the robustness of a watermarking system only when the false positive probability is taken into consideration in measuring robustness. In this study, the bit detection process is treated as an independent random Bernoulli trail with the probability p_b, which is defined to be the probability that the bit b (-1 or 1) occurs, and is considered to be constant 0.5 here. Theoretically, the probability of detecting a watermark in a mesh can be represented as:

$$p_{M_s} = \sum_{i=(L_W - L_W \times Th)}^{L_W} C_i^{L_W} {p_b}^i (1-p_b)^{L_W - i}. \tag{12}$$

In order to reasonably determine Th, p_{M_s} of Eq. (12) should be consistent with practical results. Therefore, the BERs obtained from extensive sequence-pair comparisons are collected. First of all, every un-watermarked image chosen from the Corel image database is applied as the input to our watermark detection process, as described in Sec. 3.2. For each image, a set of BERs can be obtained. After testing all 20,000 images in the Corel image database, the BER distribution can be obtained. Under this circumstances, if Th is chosen to be 0.375, p_{M_s} of Eq. (12) is calculated to be 0.003 which is very close to the cumulative distribution function of the BER distribution measured from Corel image database, $cdf(BER \leq 0.375) = 0.0027$. Consequently, it can be concluded that $Th = 0.375$ is reasonable.

In the proposed method, there are three vertexes in each M_i. However, some geometric attacks may change the relationship between the three vertexes that is crucial for mesh normalization. In order to deal with this problem, we do not merely detect a watermark from one possible normalized mesh, instead 6(= 3!) possible normalized meshes are all fed into the watermark extraction process. Under this circumstance, the probability of detecting a watermark in a mesh, p_M, can be derived as

$$p_M = (p_{M_s})^1 \times (1 - p_{M_s})^5 \approx p_{M_s}, \tag{13}$$

which is still numerically close to p_M described in Eq. (12). On the other hand, the probability of failing to detect a watermark is derived as $p_{un-watermarked} = 1 - (p_{M_s})^1 \times (1 - p_{M_s})^5$.

Recall that L_M is the number of meshes in a suspect image. Let D_M be the number of meshes detected to have been watermarked. The probability of determining a suspect image to have been watermarked is derived as:

$$p_{fp} = \sum_{i=D_M}^{L_M} C_i^{L_M} {p_M}^i (1-p_M)^{L_M-i}, \tag{14}$$

which means the false positive probability of that at least D_M meshes is detected as watermarked in L_M meshes in one image. In addition, there are four p_{fp}'s in our method, because of the two-pass feature extraction. The smallest p_{fp} will be chosen as the final p_{fp}, because the smallest p_{fp} is caused by the most precise feature extraction result.

In order to claim the presence of a watermark with strong confidence, the p_{fp} calculated in Eq. (14) should be low enough. Here, a reasonable threshold is found from the Corel image database. Again, every un-watermarked image chosen from the Corel image database is applied as the input to our watermark detection process. For each image, one p_{fp} is obtained. Our results show that no watermark can be detected from near 90% of images. Further, the cumulative distribution function of the p_{fp}'s shows that $cdf(p_{fp} \leq (3.48e-004)) = 0$, and $cdf(p_{fp} \leq (4e-004)) = (6.28e-005)$. As a result, a threshold p_{fp}^T can be set to 1.00e-005 in the sense that if the detected p_{fp} is smaller than p_{fp}^T, it is confident that a watermark is detected. It should be noted that although mesh is adopted in this paper, similar results could be obtained from other types of image units such as blocks and disks.

3.3.1. Comparison with other methods[15, 16]

The false positive probability analysis are also discussed in.[15, 16] In their approaches, a watermark is embedded in several disks extracted from an original image. However, the existence of a watermark is not determined by taking the derived false positive probability into consideration.

In,[16] 16 watermark bits are embedded into two 32×32 blocks of a disk. The false positive probability derived from each disk is defined in Eq.(5) of[16] as:

$$P_{False-alarm\ on\ one\ disk} = \sum_{r_1=T_1, r_2=T_2, r_1+r_2 \geq T}^{r_1=n, r_2=n} (\frac{1}{2})^n \cdot (\frac{n!}{r_1!(n-r_1)!}) \cdot (\frac{1}{2})^n \cdot (\frac{n!}{r_2!(n-r_2)!}), \tag{15}$$

where $n = 16, T_1 = 10, T_2 = 10$, and $T = 24$. By substituting the parameters into Eq. (15), $P_{False-alarm\ on\ one\ disk} = 0.0034$ is obtained. On the other hand, the false positive probability derived from an image is defined in Eq.(6) of[16] as:

$$P_{False-alarm\ on\ one\ image} = \sum_{i=m}^{N} (P_{False-alarm\ on\ one\ disk})^i \cdot (1 - P_{False-alarm\ on\ one\ disk})^{N-i} \cdot (\tbinom{N}{i}), \tag{16}$$

where N is total number of disks in an image, and at least m disks are detected as "success." We will compare our method and[16] based on these two equations, Eq. (15) and Eq. (16).

In,[15] the false positive probability on one image is defined in Eq.(23) of[15] as:

$$P_{FA-image} = \sum_{i=\mu}^{N} (\tbinom{N}{i}) (P_{FA-disk})^i (1 - P_{FA-disk})^{N-i}, \tag{17}$$

where the watermark is detected from at least μ number of disks, and N is the number of disks in an image that are possible for watermarking, from N strongest feature points. In the method, the authors set $N = 100$. The experiments in[15] were conducted with the thresholds, $P_{FA-image} = 0.1918e-004, 0.1656e-004$, and $0.1547e-004$

for $\mu = 1, 2$, and 3, respectively. Therefore, we can inverse derivate that $P_{FA-disk}$ are $0.1918e-006, 0.5795e-004$, and $0.4625e-003$ for $\mu = 1, 2$, and 3, respectively. Because the simulation results[15] show the number of detected watermarked disks, μ, for different images, the $P_{FA-image}$ can be calculated precisely according to Eq. (17).

In the next section, the robustness between our method, Seo and Yoo,[15] and Tang and Hang[16] will be compared by taking the derived false positive probabilities into consideration. This means that it is more guaranteed to resist a certain attack if the obtained false positive probability is sufficiently low.

4. EXPERIMENTAL RESULTS

The robustness of the proposed scheme is verified using standard benchmark, Stirmark 3.1.[13,14] Three standard images, Baboon, Lena, and Pepper, are used as cover images, and the size of all of them is 512×512. After mesh-based watermark embedding, the PSNR values between the cover image and its stego image for Baboon, Lena, and Pepper are *37.56*dB, *39.87*dB, and *39.92*dB, respectively. No perceptual difference could be sensed. Although the PSNR of stego Baboon is smaller than *38*dB, it is still hard to find any quality degradation because the Baboon image is rather noisy. As described previously two threshold $Th = 0.375$ and p_{fp}^{T} =1.00e-005 are adopted in this paper.

In order to demonstrate the superiority of our method, we made comparisons with other feature-based watermarking methods.[15,16] Since Bas et al.'s scheme[1] was not evaluated using Stirmark, it is not considered for comparison here. It has been recognized that robustness is meaningful only if false positive is taken into consideration. Although false positive analyses were conducted in,[15,16] their detection results did not show the impact of this factor. Therefore, in this paper the false positive probability is derived as Eq. (16) for,[16] and Eq. (17) for.[15] Because of the limitation of space, the parameters that can produce better results in[15,16] are chosen for comparisons. In,[16] $n = 16$, $T_1 = 10$, $T_2 = 10$, and $T = 24$ are chosen, leading to $P_{False-alarm\ on\ one\ disk} = 0.0034$. In,[15] $\mu = 1$ is selected. Because the authors declared that when $\mu = 1$ and $P_{FA-image} = 0.1918e-004$, $P_{FA-disk} = 0.1918e-006$ is obtained. The false positive probabilities are substituted into Eq. (17), for calculating the false positive probability of each image. "D_M/T_M" in the following tables denotes "number of detected meshes(disks)/number of total meshes(disks)." Because N is set to 100,[15] the T_M's are all set to 100 in our comparisons.

4.1. Non-geometric Attacks

The watermark detection results with respect to non-geometric attacks are shown in Tables 2, 3, and 4. As a whole, our method when compared with[15,16] can survive most of the non-geometric attacks of Stirmark 3.1. In Table 2,[15] can only survive FMLR and Color reduce attacks, while our method and[16] can tolerate JPEG compression up to quality factor 40%. Besides, only our method can survive Sharpening attack. In Table 3, our method that can survive almost all attacks, except for JPEG10 and FMLR attacks, is more robust than[15] and.[16] The similar result can also be found in Table 4.

4.2. Geometric Attacks

As to comparisons of resistance to geometric distortions, the results are shown in Tables $5 \sim 7$. Basically, it can be observed that our method and[15] provide sufficiently lower p_{fp} than[16] for line removal, cropping attacks, and general linear transformations. Our method also provides much lower p_{fp}'s for shearing and random bending. For other attacks, our method was thoroughly evaluated and provides low p_{fp}'s, while others[15,16] were not. As a whole, it can be concluded that our results are consistently better than the other two by taking p_{fp} into consideration.

4.3. Watermark-Estimation Attacks[7]

The collusion attack and copy attack were used to verify the resistance our method to WEAs. Table 8 and Table 9 show the results of resisting collusion attack for CDW embedding and non-CDW embedding, respectively. After performing collusion attack, the number of detected meshes in Table 9 is smaller than that in Table 8, which implies our proposed scheme with CDW embedding efficiently defense the collusion attack. Table 10 and Table 11 show the results of resisting copy attack for CDW embedding and non-CDW embedding, respectively.

Table 2. Non-geometric attacks for Baboon.

attack	proposed method		[15]		[16]	
	D_M/T_M	p_{fp}	D_M	p_{fp}	D_M/T_M	p_{fp}
Median filter 2x2	5/104	1.75e-005	-	-	6/11	7.07e-013
Median filter 3x3	4/99	2.43e-004	-	-	2/11	6.24e-004
Median filter 4x4	2/98	3.54e-002	1	1.91e-005	-	-
Gaussian filter 3x3	10/97	5.86e-013	0	1.00e-000	8/11	2.94e-018
JPEG 90	10/107	1.59e-012	-	-	-	-
JPEG 80	9/107	5.42e-011	-	-	9/11	3.35e-021
JPEG 70	11/105	3.41e-014	1	1.91e-005	11/11	7.10e-028
JPEG 60	11/103	2.74e-014	1	1.91e-005	7/11	1.72e-015
JPEG 50	9/103	3.83e-011	1	1.91e-005	5/11	2.07e-010
JPEG 40	5/109	2.19e-005	1	1.91e-005	7/11	1.72e-015
JPEG 30	3/103	3.82e-003	0	1.00e-000	4/11	4.34e-008
JPEG 20	2/107	4.15e-002	-	-	-	-
JPEG 10	1/111	2.84e-001	-	-	-	-
FMLR	10/104	1.19e-012	4	5.30e-021	-	-
Color reduce	12/107	1.01e-015	2	1.82e-010	4/11	4.34e-008
Sharpening 3x3	6/98	6.06e-007	0	1.00e-000	2/11	6.24e-004

Table 3. Non-geometric attacks for Lena.

attack	proposed method		[15]		[16]	
	D_M/T_M	p_{fp}	D_M	p_{fp}	D_M/T_M	p_{fp}
Median filter 2x2	21/101	2.14e-032	-	-	1/8	2.69e-002
Median filter 3x3	24/103	4.01e-038	-	-	1/8	2.69e-002
Median filter 4x4	14/95	7.79e-020	5	1.95e-026	-	-
Gaussian filter 3x3	29/99	4.94e-049	3	1.14e-015	5/8	2.53e-011
JPEG 90	28/97	3.45e-047	-	-	-	-
JPEG 80	27/99	8.66e-045	-	-	6/8	4.32e-014
JPEG 70	18/101	1.14e-026	3	1.14e-015	7/8	4.22e-017
JPEG 60	17/101	8.14e-025	3	1.14e-015	6/8	4.32e-014
JPEG 50	18/99	7.74e-027	1	1.91e-005	5/8	2.53e-011
JPEG 40	16/99	3.85e-023	1	1.91e-005	3/8	2.18e-006
JPEG 30	11/111	6.37e-014	0	1.00e-000	2/8	3.19e-004
JPEG 20	6/103	8.13e-007	-	-	-	-
JPEG 10	5/113	2.61e-005	-	-	-	-
FMLR	4/91	1.76e-004	1	1.91e-005	-	-
Color reduce	35/97	1.23e-062	4	5.30e-021	7/8	4.22e-017
Sharpening 3x3	14/107	4.48e-019	1	1.91e-005	4/8	9.29e-009

After performing copy attack, the number of detected meshes in Table 11 is larger than that in Table 10, which implies our proposed scheme with CDW embedding efficiently defense the copy attack. However, the content-independent watermarking methods[1, 15, 16] cannot survive WEA.[7]

In sum, extensive experiment results verify that our method indeed outperforms all the other feature-based watermarking methods.

5. CONCLUSIONS

A mesh-based content-dependent image watermarking method that can resist extensive geometric attacks and watermark estimation attacks is proposed. The major contribution of our method is twofold. First, a robust

Table 4. Non-geometric attacks for Pepper.

attack	proposed method		[15]		[16]	
	D_M/T_M	p_{fp}	D_M	p_{fp}	D_M/T_M	p_{fp}
Median filter 2x2	35/106	5.30e-061	-	-	1/4	1.35e-002
Median filter 3x3	34/105	5.84e-059	-	-	1/4	1.35e-002
Median filter 4x4	26/107	1.05e-041	4	5.30e-021	-	-
Gaussian filter 3x3	45/109	2.33e-083	5	1.95e-026	1/4	1.35e-002
JPEG 90	49/112	3.25e-092	-	-	-	-
JPEG 80	46/112	4.81e-085	-	-	3/4	1.57e-007
JPEG 70	39/109	1.95e-069	6	5.93e-032	3/4	1.57e-007
JPEG 60	35/114	1.09e-059	6	5.93e-032	1/4	1.35e-002
JPEG 50	26/104	4.57e-042	4	5.30e-021	3/4	1.57e-007
JPEG 40	19/109	7.18e-028	4	5.30e-021	1/4	1.35e-002
JPEG 30	14/104	2.96e-019	4	5.30e-021	0/4	1.00e-000
JPEG 20	9/111	7.55e-011	-	-	-	-
JPEG 10	1/115	2.92e-001	-	-	-	-
FMLR	9/97	2.22e-011	0	1.00e-000	-	-
Color reduce	50/110	3.78e-095	2	1.82e-010	1/4	1.35e-002
Sharpening 3x3	15/114	2.26e-020	5	1.95e-026	4/4	1.34e-010

mesh extraction is proposed to enhance the feasibility of feature-based watermarking methods. Second, content-dependent watermark that is composed of a watermarking and a hash is proposed to resist watermarking-estimation attack. Standard benchmark has verified the robustness of the proposed scheme. The major weakness of our method is its high complexity, since most of time is spent in the mesh warping operation. As a result, our system at its current status is not suitable for real-time applications.

ACKNOWLEDGMENTS

This paper was supported, in part, by the National Science Council under NSC grant 92-2422-H-001-004.

REFERENCES

1. P. Bas, J. M. Chassery, and B. Macq, "Geometrically invariant watermarking using feature points," *IEEE Trans. Image Processing,* Vol. 11, No. 9, pp.1014-1028, September 2002.
2. C. Harris and M. Stephen, "A combined corner and edge detector," *Proc. 4th Alvey Vision Conf.,* pp.147-151, 1988.
3. J. R. Hernandez and F. Perez-Gonzalez, "Statistical analysis of watermarking schemes for copyright protection of images," *Proc. IEEE,* Vol. 87, pp. 1142-1143, July 1999.
4. A. Herrigel, S. Voloshynovskiy, Y. Rytsar, "The watermark template attack," *Proc. SPIE Security and Watermarking of Multimedia Contents III (Vol. 4314),* San Jose, January 2001.
5. M. Kutter, "Watermarking resisting to translation, rotation and scaling," *Proc. SPIE International Symposium on Voice, Video, and Data Communication,* Boston, November 1998.
6. C. Y. Lin, M. Wu, J. A. Bloom, I. J. Cox, M. L. Miller, and Y. M. Lui, "Rotation, scale and translation resilient watermarking for images," *IEEE Trans. Image Processing,* Vol. 10, No. 5, pp. 767-782, May 2001.
7. C. S. Lu and C.Y. Hsu, "Content-Dependent Anti-Disclosure Image Watermark," *Proc. 2nd Int. Workshop on Digital Watermarking,* LNCS 2939, pp. 61-76, Seoul, Korea, 2003.
8. C. S Lu, C. Y. Hsu, S. W. Sun, and P. C. Chang, "Robust Mesh-based Hashing for Copy Detection and Tracing of Images," *Proc. IEEE Int. Conf. on Multimedia and Expo: special session on Media Identification,* Taipei, Taiwan, 2004.
9. K. Mikolajczyk, "Detection of local features invariant to affine transformations," *Ph.D. thesis, INPG Grenoble,* July 2002.

10. J. O'Ruanaidh and T. Pun, "Rotation, scale and translation invariant spread spectrum digital image watermarking," *Signal Processing,* Vol.66, No. 3, pp. 303-317, May 1998.
11. S. Pereira, T. Pun, "Robust template matching for affine resistant image watermarks," *IEEE Trans. Image Processing,* Vol. 9, No. 6, pp. 1123-1129, June 2000.
12. S. Pereira, T. Pun, "An iterative template matching algorithm using the Chrip-Z transform for digital image watermarking," *Pattern Recognition (33),* pp. 173-175, 2000.
13. F. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Attacks on Copyright Marking Systems," *Proc. Int. Workshop on Information Hiding,* LNCS 1575, pp. 219-239, 1998.
14. F. Petitcolas, "Watermarking Schemes Evaluation," *IEEE Signal Processing Magazine,* Vol. 17, No. 5, pp. 58-64, 2000.
15. J. S. Seo and C. D. Yoo, "Localized image watermarking based on feature points of scale-space representation," *Pattern Recognition (37),* pp. 1365-1375, 2004.
16. C. W. Tang and H. M. Hang, "A Feature-Based Robust Digital Image Watermarking Scheme," *IEEE Trans. Signal Processing,* Vol. 51, No. 4, pp.950-958, April 2003.
17. S.Voloshynovskiy, A.Herrigel, N.Baumgartner and T.Pun, "A stochastic approach to content adaptive digital image watermarking," *Proc. Int. Workshop on Information Hiding,* LNCS 1768, pp. 211-236, 1999.
18. S. Voloshynovskiy, F. Deguillaume, and T. Pun, "Multibit digital watermarking robust against local nonlinear geometrical distortions," *Proc. IEEE Int. Conf. Image Processing,* Thessaloniki, pp. 999-1002, Oct. 2001.
19. S. Voloshynovskiy, S. Pereira, V. Iquise, and T. Pun, "Attack Modelling: Towards a Second Generation Watermarking Benchmark," *Signal Processing,* Vol. 81, pp. 1177-1214, 2001.

Table 5. Geometric attacks for Baboon.

attack	proposed method		[15]		[16]	
	D_M/T_M	p_{fp}	D_M	p_{fp}	D_M/T_M	p_{fp}
1 column, 1 row removed	6/107	1.02e-006	-	-	-	-
5 column, 1 row removed	7/103	3.37e-008	-	-	6/11	7.07e-013
1 column, 5 row removed	5/107	2.00e-005	-	-	-	-
17 column, 5 row removed	4/95	2.07e-004	1	1.91e-005	3/11	6.37e-006
5 column, 17 row removed	8/91	4.47e-010	-	-	-	-
Cropping 1% off	6/105	9.10e-007	-	-	-	-
Cropping 2% off	11/104	3.06e-014	-	-	-	-
Cropping 5% off	7/102	3.15e-008	-	-	2/11	6.24e-004
Cropping 10% off	8/90	4.09e-010	-	-	2/11	6.24e-004
Cropping 15% off	7/77	4.38e-009	4	5.30e-021	-	-
Cropping 20% off	7/75	3.63e-009	-	-	-	-
Cropping 25% off	5/61	1.26e-006	1	1.91e-005	-	-
Cropping 50% off	4/20	3.78e-007	-	-	-	-
Linear(1.007, 0.010, 0.010, 1.012)	9/102	3.51e-011	3	1.14e-015	4/11	4.34e-008
Linear(1.010, 0.013, 0.009, 1.011)	6/108	1.07e-006	1	1.91e-005	4/11	4.34e-008
Linear(1.013, 0.008, 0.011, 1.008)	9/104	4.18e-011	0	1.00e-000	5/11	2.07e-010
Aspect ratio change(0.80, 1.00)	2/77	2.27e-002	-	-	-	-
Aspect ratio change(0.90, 1.00)	8/92	4.88e-010	-	-	-	-
Aspect ratio change(1.00, 0.80)	3/81	1.93e-003	-	-	-	-
Aspect ratio change(1.00, 0.90)	7/89	1.22e-008	-	-	-	-
Aspect ratio change(1.00, 1.20)	5/117	3.08e-005	-	-	-	-
Aspect ratio change(1.00 1.10)	12/110	1.42e-015	-	-	-	-
Aspect ratio change(1.10, 1.00)	8/116	3.13e-009	-	-	-	-
Aspect ratio change(1.20, 1.00)	4/127	6.24e-004	-	-	-	-
Rotation 1.00	6/100	6.83e-007	-	-	3/11	6.37e-006
Rotation 2.00	7/101	2.94e-008	-	-	1/11	3.68e-002
Rotation 5.00	8/94	5.81e-010	-	-	0/11	1.00e-000
Rotation 10.00	9/92	1.37e-011	-	-	-	-
Rotation 15.00	6/76	1.33e-007	-	-	-	-
Rotation 30.00	7/363	1.34e-004	-	-	-	-
Rotation 45.00	3/43	3.05e-004	1	1.91e-005	-	-
Rotation 90.00	4/107	3.27e-004	-	-	-	-
Flipping	6/105	9.10e-007	-	-	-	-
Rotation Scale 1.00	9/104	4.18e-011	-	-	4/11	4.3451e-008
Rotation Scale 10.00	4/125	5.88e-004	-	-	-	-
Rotation Scale 15.00	3/125	6.53e-003	-	-	-	-
Rotation Scale 30.00	14/737	8.99e-008	-	-	-	-
Rotation Scale 45.00	11/699	1.22e-005	-	-	-	-
Rotation Scale 90.00	4/107	3.27e-004	-	-	-	-
Scaling 50%	6/103	8.13e-007	0	1.00e-000	-	-
Scaling 75%	4/323	1.70e-002	0	1.00e-000	-	-
Scaling 90%	7/542	1.43e-003	2	1.82e-010	-	-
Scaling 110%	3/121	5.97e-003	-	-	-	-
Scaling 150%	6/325	5.05e-004	-	-	-	-
Scaling 200%	8/107	1.64e-009	-	-	-	-
Shearing x-0% y-1%	8/103	1.21e-009	-	-	-	-
Shearing x-1% y-0%	12/107	1.01e-015	2	1.82e-010	-	-
Shearing x-1% y-1%	3/101	3.61e-003	-	-	4/11	4.34e-008
Shearing x-0% y-5%	5/107	2.00e-005	-	-	3/11	6.37e-006
Shearing x-5% y-0%	14/104	2.96e-019	-	-	-	-
Shearing x-5% y-5%	6/99	6.43e-007	0	1.00e-000	0/11	1.00e-000
Random Bending	8/109	1.90e-009	0	1.00e-000	-	-

Table 6. Geometric attacks for Lena.

attack	proposed method		[15]		[16]	
	D_M/T_M	p_{fp}	D_M	p_{fp}	D_M/T_M	p_{fp}
1 column, 1 row removed	29/97	2.47e-049	-	-	-	-
5 column, 1 row removed	28/97	3.45e-047	-	-	3/8	2.18e-006
1 column, 5 row removed	22/95	5.23e-035	-	-	-	-
17 column, 5 row removed	25/97	6.97e-041	5	1.95e-026	0/8	1.00e-000
5 column, 17 row removed	17/101	8.14e-025	-	-	-	-
Cropping 1% off	26/99	1.07e-042	-	-	-	-
Cropping 2% off	26/98	7.88e-043	-	-	-	-
Cropping 5% off	20/84	3.12e-032	-	-	2/8	3.19e-004
Cropping 10% off	12/68	3.31e-018	-	-	2/8	3.19e-004
Cropping 15% off	6/69	7.43e-008	6	5.93e-032	-	-
Cropping 20% off	8/63	2.19e-011	-	-	-	-
Cropping 25% off	5/50	4.60e-007	4	5.30e-021	-	-
Cropping 50% off	2/106	4.08e-002	-	-	-	-
Linear(1.007, 0.010, 0.010, 1.012)	22/99	1.44e-034	6	5.93e-032	5/8	2.53e-011
Linear(1.010, 0.013, 0.009, 1.011)	32/103	6.67e-055	7	1.52e-037	4/8	9.29e-009
Linear(1.013, 0.008, 0.011, 1.008)	25/97	6.97e-041	7	1.52e-037	4/8	9.29e-009
Aspect ratio change(0.80, 1.00)	12/84	4.92e-017	-	-	-	-
Aspect ratio change(0.90, 1.00)	17/89	7.98e-026	-	-	-	-
Aspect ratio change(1.00, 0.80)	2/84	2.67e-002	-	-	-	-
Aspect ratio change(1.00, 0.90)	17/91	1.20e-025	-	-	-	-
Aspect ratio change(1.00, 1.20)	9/118	1.31e-010	-	-	-	-
Aspect ratio change(1.00 1.10)	20/108	7.96e-030	-	-	-	-
Aspect ratio change(1.10, 1.00)	23/107	1.06e-035	-	-	-	-
Aspect ratio change(1.20, 1.00)	10/135	1.65e-011	-	-	-	-
Rotation 1.00	26/100	1.44e-042	-	-	3/8	2.18e-006
Rotation 2.00	20/86	5.28e-032	-	-	0/8	1.00e-000
Rotation 5.00	13/76	2.04e-019	-	-	0/8	1.00e-000
Rotation 10.00	15/76	3.43e-023	-	-	-	-
Rotation 15.00	13/63	1.45e-020	-	-	-	-
Rotation 30.00	5/57	8.94e-007	-	-	-	-
Rotation 45.00	4/36	4.42e-006	2	1.82e-010	-	-
Rotation 90.00	18/97	5.20e-027	-	-	-	-
Flipping	15/97	1.76e-021	-	-	-	-
Rotation Scale 1.00	29/100	6.93e-049	-	-	0/8	1.00e-000
Rotation Scale 10.00	4/102	2.72e-004	-	-	-	-
Rotation Scale 15.00	1/100	2.60e-001	-	-	-	-
Rotation Scale 30.00	3/113	4.94e-003	-	-	-	-
Rotation Scale 45.00	6/92	4.17e-007	-	-	-	-
Rotation Scale 90.00	18/97	5.20e-027	-	-	-	-
Scaling 50%	11/91	6.73e-015	2	1.82e-010	-	-
Scaling 75%	2/62	1.51e-002	3	1.14e-015	-	-
Scaling 90%	5/85	6.53e-006	4	5.30e-021	-	-
Scaling 110%	17/117	1.16e-023	-	-	-	-
Scaling 150%	3/63	9.37e-004	-	-	-	-
Scaling 200%	40/97	2.99e-074	-	-	-	-
Shearing x-0% y-1%	24/93	2.52e-039	-	-	-	-
Shearing x-1% y-0%	23/97	8.56e-037	5	1.95e-026	-	-
Shearing x-1% y-1%	21/95	5.17e-033	-	-	4/8	9.29e-009
Shearing x-0% y-5%	21/88	8.59e-034	-	-	2/8	3.19e-004
Shearing x-5% y-0%	21/91	1.89e-033	-	-	-	-
Shearing x-5% y-5%	19/75	2.84e-031	1	1.91e-005	1/8	2.69e-002
Random Bending	31/93	2.31e-054	4	5.30e-021	-	-

Table 7. Geometric attacks for Pepper.

attack	proposed method		[15]		[16]	
	D_M/T_M	p_{fp}	D_M	p_{fp}	D_M/T_M	p_{fp}
1 column, 1 row removed	52/112	1.59e-099	-	-	-	-
5 column, 1 row removed	45/110	3.93e-083	-	-	3/4	1.57e-007
1 column, 5 row removed	42/102	7.55e-078	-	-	-	-
17 column, 5 row removed	33/102	2.94e-057	5	1.95e-026	1/4	1.35e-002
5 column, 17 row removed	35/102	1.05e-061	-	-	-	-
Cropping 1% off	33/112	1.12e-055	-	-	-	-
Cropping 2% off	23/114	5.27e-035	-	-	-	-
Cropping 5% off	16/98	3.24e-023	-	-	2/4	6.91e-005
Cropping 10% off	19/92	2.16e-029	-	-	2/4	6.91e-005
Cropping 15% off	13/76	2.04e-019	2	1.82e-010	-	-
Cropping 20% off	13/61	9.16e-021	-	-	-	-
Cropping 25% off	12/60	6.51e-019	2	1.82e-010	-	-
Cropping 50% off	5/18	2.02e-009	-	-	-	-
Linear(1.007, 0.010, 0.010, 1.012)	46/105	1.06e-086	5	1.95e-026	1/4	1.35e-002
Linear(1.010, 0.013, 0.009, 1.011)	48/108	8.57e-091	7	1.52e-037	1/4	1.35e-002
Linear(1.013, 0.008, 0.011, 1.008)	38/104	4.03e-068	5	1.95e-026	4/8	9.29e-009
Aspect ratio change(0.80, 1.00)	16/88	5.08e-024	-	-	-	-
Aspect ratio change(0.90, 1.00)	25/91	1.12e-041	-	-	-	-
Aspect ratio change(1.00, 0.80)	4/79	1.02e-004	-	-	-	-
Aspect ratio change(1.00, 0.90)	24/96	5.99e-039	-	-	-	-
Aspect ratio change(1.00, 1.20)	17/120	1.83e-023	-	-	-	-
Aspect ratio change(1.00 1.10)	32/118	1.06e-052	-	-	-	-
Aspect ratio change(1.10, 1.00)	37/110	9.39e-065	-	-	-	-
Aspect ratio change(1.20, 1.00)	27/114	6.57e-043	-	-	-	-
Rotation 1.00	37/104	7.59e-066	-	-	2/4	6.91e-005
Rotation 2.00	33/101	2.00e-057	-	-	1/4	1.35e-002
Rotation 5.00	25/86	2.19e-042	-	-	0/4	1.00e-000
Rotation 10.00	19/74	2.13e-031	-	-	-	-
Rotation 15.00	12/57	3.32e-019	-	-	-	-
Rotation 30.00	7/55	3.91e-010	-	-	-	-
Rotation 45.00	4/48	1.42e-005	1	1.91e-005	-	-
Rotation 90.00	29/112	3.05e-047	-	-	-	-
Flipping	24/112	3.76e-037	-	-	-	-
Rotation Scale 1.00	34/108	1.82e-058	-	-	2/4	6.91e-005
Rotation Scale 10.00	8/90	4.09e-010	-	-	-	-
Rotation Scale 15.00	5/81	5.15e-006	-	-	-	-
Rotation Scale 30.00	3/93	2.86e-003	-	-	-	-
Rotation Scale 45.00	3/96	3.13e-003	-	-	-	-
Rotation Scale 90.00	29/112	3.05e-047	-	-	-	-
Scaling 50%	16/99	3.85e-023	2	1.82e-010	-	-
Scaling 75%	2/66	1.70e-002	6	5.93e-032	-	-
Scaling 90%	18/90	1.20e-027	6	5.93e-032	-	-
Scaling 110%	32/116	5.64e-053	-	-	-	-
Scaling 150%	4/65	4.74e-005	-	-	-	-
Scaling 200%	52/103	4.40e-102	-	-	-	-
Shearing x-0% y-1%	46/108	5.65e-086	-	-	-	-
Shearing x-1% y-0%	43/110	1.94e-078	4	5.30e-021	-	-
Shearing x-1% y-1%	36/113	4.95e-062	-	-	1/4	1.35e-002
Shearing x-0% y-5%	41/91	4.19e-078	-	-	1/4	1.35e-002
Shearing x-5% y-0%	39/97	6.86e-072	-	-	-	-
Shearing x-5% y-5%	23/95	4.99e-037	0	1.00e-000	0/4	1.00e-000
Random Bending	31/113	2.78e-051	3	1.14e-015	-	-

Table 8. Collusion attack on CDW embedding.

image	CDW stego image D_M/T_M	p_{fp}	PSNR (dB)	colluded image D_M/T_M	p_{fp}	PSNR (dB)
Baboon	16/105	1.04e-022	37.56	6/102	7.67e-007	34.39
Lena	36/95	2.52e-065	39.87	19/101	1.49e-028	36.71
Pepper	54/110	4.77e-105	39.92	30/111	1.86e-049	36.82

Table 9. Collusion attack on non-CDW embedding

image	Non-CDW stego image D_M/T_M	p_{fp}	PSNR (dB)	colluded image D_M/T_M	p_{fp}	PSNR (dB)
Baboon	10/101	8.85e-013	38.15	1/130	3.23e-001	36.29
Lena	44/96	3.90e-084	39.83	2/96	3.42e-002	38.16
Pepper	71/103	3.02e-153	39.88	6/118	1.80e-006	40.37

Table 10. Copy attack on CDW embedding.

image	CDW stego image D_M/T_M	p_{fp}	PSNR (dB)	copy attacked image D_M/T_M	p_{fp}	PSNR (dB)
Baboon	16/105	1.04e-022	37.56	1/109	2.79e-001	37.57
Lena	36/95	2.52e-065	39.87	2/100	3.67e-002	39.88
Pepper	54/110	4.77e-105	39.92	2/109	4.29e-002	39.88

Table 11. Copy attack on non-CDW embedding

image	Non-CDW stego image D_M/T_M	p_{fp}	PSNR (dB)	copy attacked image D_M/T_M	p_{fp}	PSNR (dB)
Baboon	10/101	8.85e-013	38.15	10/130	1.13e-011	37.57
Lena	44/96	3.90e-084	39.83	45/96	1.35e-086	39.82
Pepper	71/103	3.02e-153	39.88	51/118	1.46e-095	39.88