中央研究院
資訊科學研究所
Institute of Information Science, Academia Sinica • Taipei, Taiwan, ROC

TR-IIS-05-002

Media Hash-dependent Image Watermarking Resilient Against Both Geometric Attacks and Estimation Attacks Based on False Positive-Oriented Detection

Chun-Shien Lu, Shih-Wei Sun, Chao-Yong Hsu, and Pao-Chi Chang

January 2005 || Technical Report No. TR-IIS-05-002

http://www.iis.sinica.edu.tw/LIB/TechRept.htm

Media Hash-dependent Image Watermarking Resilient Against Both Geometric Attacks and Estimation Attacks Based on False Positive-Oriented Detection

Chun-Shien Lu[1], Shih-Wei Sun[1,2], Chao-Yong Hsu[1], and Pao-Chi Chang[2]

[1]Institute of Information Science, Academia Sinica, Taipei, Taiwan 115, ROC

[2]Dept. of Electrical Engineering, National Central University, Chung-Li, Taiwan 320, ROC

Abstract

The major disadvantage of existing watermarking methods is their limited resistance to extensive geometric attacks. In addition, we have found that the weakness of multiple watermark embedding methods that were initially designed to resist geometric attacks is their inability to withstand the watermark-estimation attacks (WEAs), leading to reduce resistance to geometric attacks. In view of these facts, this paper proposes a robust image watermarking scheme that can withstand geometric distortions and WEAs simultaneously. Our scheme is mainly composed of two components: (i) mesh generation and embedding to resist geometric distortions; and (ii) construction of media hash-based content-dependent watermark (CDW) to resist WEAs. Furthermore, we propose a false positive-oriented watermark detection mechanism, which can be used to determine the existence of a watermark so as to achieve a trade-off between correct detection and false detection. Extensive experimental results obtained using the standard benchmark and WEAs, and comparisons with relevant watermarking methods confirm the excellent performance of our method in improving robustness. To our knowledge, such a thorough evaluation has not been reported in the literature before.

keywords: Attack, Copyright protection, Embedding, Mesh, Media Hash, Robustness, Watermark

Corresponding author: Dr. C. S. Lu (lcs@iis.sinica.edu.tw). This paper was, in part, published in Proc. SPIE: Security, Steganography, and Watermarking of Multimedia Contents VII (EI120), San Jose, California, USA, January 2005.

I. INTRODUCTION

Digital watermarking has been recognized as a helpful technology for copyright protection, traitor tracing, and authentication. No matter which kinds of applications are considered, robustness is a critical issue affecting the practicability of the watermarking system. For example, we have found that content providers care very much about whether their works can be protected in a robust way prior to distribution. In data hiding, robustness refers to the capability of resistance to attacks that are used to destroy or remove hidden watermarks. In [26], attacks were divided into four categories: (1) removal attacks; (2) geometric attacks; (3) cryptographic attacks; and (4) protocol attacks. Among them, geometric attacks introduce synchronization errors in order to disable watermark detection without having to remove hidden information or degrade the quality of the watermarked contents. More importantly, geometric distortions are easy to realize without much effort. Therefore, motivated by the needs of content providers and the lack of sufficient robustness, this study focused on the challenging issue of resisting (extensive) geometric attacks. In this paper, signal processing related attacks, including removal attacks, geometric attacks, and the copy attack (a kind of protocol attacks), are major concerns while security related attacks, including cryptographic attacks and the ambiguity attack (a kind of protocol attacks) are left untouched since they represent another important topic in digital watermarking.

The existing watermarking methods that are resistant to geometric attacks can be divided into three categories. The first category includes those which embed a watermark into the geometric invariant domain. In [10], [16], watermarking was conducted in the magnitude part of the Fourier-Mellin domain to exploit its affine invariance. However, the Fourier-Mellin domain is inherently vulnerable to cropping and other local geometric distortions (e.g., changes of the aspect ratio). In addition, resistance to removal attacks is limited because most of the FMT information is contained in the phase instead of the magnitude part of the Fourier transformed domain. On the other hand, the concept of moment normalization, which is conventionally used in computer vision and pattern recognition, was employed [1], [13] to achieve geometric invariance. In [1], a watermark was claimed to exist if the perturbation of the moment invariants was within a small tolerance. The major disadvantages include: (i) an inability to preserve fidelity, i.e., the watermarked image will create contrast variations; and (ii) an inability to tolerate any change of the aspect ratio or cropping. Similarly, our recent efforts [13] improved the transparency of moment-based watermarking but still failed to resist attacks related to cropping because the lost contents lead to changes of moments.

The methods belonging to the second category uses a template [17], [18] or insert a periodic watermark pattern [9], [25] for the purpose of re-synchronization. This kind of prior information is also known as the pilot signal

[15]. In [17], [18], templates were embedded in the DFT domain to generate the shape of local peaks, which can be easily retrieved in the detection process to recover geometric parameters. On the other hand, the local peaks can also be easily extracted by pirates in order to remove templates [7]. In [9], Kutter was first to propose a watermarking scheme that can provide resistance to global geometrical distortions. The key step in this method is the embedding of a self-reference watermark, which is prepared in advance as a specific structural pattern, for the purpose of calibration. Kutter's reference watermark is composed of nine peaks that are extracted by means of an autocorrelation function and used to estimate the effects of geometrical attacks. Because the geometrical transformations are inverted, the hidden watermark can be recovered. The main drawbacks are that the other eight non-central peaks are inherently less robust to attacks, and the global watermark structure can be totally destroyed by means of local geometric distortions. A more powerful approach [25] extends Kutter's scheme through block-based periodical placement of self-reference watermarks so that the Fourier magnitude spectrum of periodical watermarks is composed of regular peaks distributed all over the image. This particular feature, i.e., a lattice of peaks, provides the capability of recovering global/local geometrical distortions. Again, because the positioned periodical block-based pilot signals inherently reveal peaks in the transformed domain, hints remain that a watermark estimation attack (e.g., the collusion attack) can be used to efficiently destroy them [11].

The third category includes methods which employ "feature-based watermarking." 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 existing feature point extraction methods, the Harris detector [4] is widely used in various applications. However, we have found that the Harris detector is still not robust enough to be used in digital watermarking [2]. This is because the Harris detector is rotation and scaling-sensitive. In [23], Mexican-Hat wavelet filtering was used for feature point extraction. Mexican-Hat wavelet filtering was implemented in the frequency domain using FFT. Although 1-D FFT is widely used to implement 2-D FFT in order to improve computational efficiency, this implementation may lead to another severe problem; i.e., 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 the fact that the asynchronization effect is propagated and coupled with the result of Mexican-Hat wavelet filtering. In [22], the scale-space theory was applied for feature point extraction. Feature points were determined through automatic scale selection and local extreme detection. For a chosen feature point, a circular disk is formed and used for embedding in the Fourier domain. However, there are two major drawbacks in [22]: (i) the embedding unit is a circular disk, which inherently limits the achievable robustness against geometric attacks that preserve the aspect ratio (this was also noted by the authors); (ii) since embedding is conducted in the

magnitude component of the Fourier domain, as noted in the above discussions of the first category of methods, resistance to removal attacks is limited (this will be seen later in the comparison of experimental results).

After surveying the existing watermarking methods that provide a certain degree of robustness against geometric distortions, we have observed that: (i) the methods in the first category are restricted to be affine invariant; (ii) the pilot signals that are employed in the methods in the second category for the recovery of geometric parameters are easily removed; and (iii) robust extraction of feature points plays a key role in the methods in the third category. In particular, we find that Voloshynovskiy *et al.*'s scheme [25] was thoroughly verified by means of the standard benchmark, Stirmark [19], [20], and possesses strong robustness. Thus, we can treat Voloshynovskiy *et al.*'s scheme as a state-of-the-art, robust watermarking technology. However, as described previously, this method is vulnerable to collusion, so initially embedded watermarks can be removed and the ability to resist extensive geometric attacks can be lost. Furthermore, we are aware of a recent paper [15] in which Manuel *et al.* exhaustively analyzed pilot-based synchronization algorithms and confirmed that pilot signals are easy to destroy. As a consequence, we do not adopt the paradigm of pilot-based watermarking even though it exhibits promising robustness against geometric attacks. Since the purpose of this paper is to propose an image watermarking scheme that can resist extensive geometric attacks and the watermark estimation attacks [11] simultaneously, we adopt feature-based watermarking based on the prerequisite that the robustness of feature point extraction can be enhanced. This selection is believed to be more helpful for satisfying our goal since our mesh-based image hashing scheme [8] has been confirmed to be quite resistant to two versions of Stirmark. Moreover, in our companion paper [11], we proposed a block-based content-dependent watermarking scheme that combines our content-dependent watermark with the approach in [25] to tolerate the watermark estimation attacks. However, the preset periodical regularity of a watermark pattern is destroyed, thus, resistance to geometric distortions is lost because the content-dependent watermarks resulting from all the image blocks are dissimilar. In order to further address this issue, we investigate mesh-based instead of block-based watermarking in this paper.

In this paper, 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 used to filtering, leading to geometric-invariant filtering. In order to resist watermark-estimation attacks, image hashing [8] is further extracted and combined with hidden watermarks to generate the media hash-based Content-Dependent Watermark (CDW) [11]. CDW is able to resist the watermark estimation attacks because even though pirates can estimate watermarks from meshes, they still cannot be successfully colluded to generate an even more correct watermark that is to be removed. We also study how mesh-based watermarking can be achieved

without causing perceptual quality degradation. In addition to robustness, due to the unique characteristic of multiple mesh-based watermark embedding, we propose a false positive-oriented watermark detection mechanism to indicate the presence/absence of a watermark. We investigate how to determine the existence of a watermark in a mesh and in an image, respectively. In order to demonstrate the performance of our method in improving robustness, the standard benchmark, Stirmark, and watermark estimation attacks (including the collusion and copy attacks) were used to perform a thorough evaluation.

The remainder of this paper is organized as follows. In Sec. II, we describe two important issues, including robust feature extraction and media hash-based content-dependent watermark, that are fundamental to our method. In Sec. III, the proposed media hash-based content-dependent watermarking is described. We describe in detail how a trade-off between transparency and robustness can be achieved. In Sec. IV, based on the characteristics of feature-based watermarking methods, a false positive-oriented watermark detection mechanism is described to achieve a trade-off between correct detection and false detection. Extensive experimental results together with robustness comparisons with other feature-based methods are given in Sec. V to verify the performance of our scheme. Finally, conclusions are drawn in Sec. VI.

II. ROBUST FEATURE EXTRACTION AND MEDIA HASH-BASED CONTENT-DEPENDENT WATERMARK

Two issues concerning the proposed watermarking method, robust feature extraction and media hash-based content-dependent watermark, will be discussed in this section. They play key roles in achieving the desired goal.

A. 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.

1) Gaussian Kernel Filtering: Gaussian kernel filtering is a special case of scale-space filtering. In scale-space filtering, an image is filtered by several filters of different sizes to generate multiple frequency responses. In some applications, the filter size can be adaptive to different affine transformation environments. But in digital watermarking, we only select a fixed filter size to generate one level scale-space for watermark embedding. This benefits our watermark detection scheme in that only a small set of filters is required to achieve blind detection (as will be described in Sec. III-B). Let $I(x, y)$ be a cover image, and let the 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 circular in shape, the resultant filtering response is rotation insensitive, which is beneficial for obtaining geometric-invariant feature points.

2) Local maximum determination: The local maximum determination process is operated in the Gaussian kernel filtered signal for feature point extraction. First, a maximum filter of size 3×3 is applied to $L(x, y, \sigma)$ and is expressed as

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

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

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

which means that a feature point at (x, y) satisfies that the filtering responses, $MF(x, y)$ and $L(x, y, \sigma)$, are equal. In addition, the set of feature points, P, is used to form a set of meshes by means of the Delaunay tessellation. In this paper, each mesh is a basic unit used for watermark embedding and extraction.

3) How Can We Choose σ?: When the Gaussian kernel is used as the feature point detector, it is important to determine how many σ's have to be used. If a larger σ is used, lower frequency (corresponding to larger scale) information tends to be revealed. On the other hand, 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 related to the ability to deal with geometric attacks because if the σ's do not properly match the characteristics of geometrically attacked images, then the feature points will not be correctly detected.

These problems can be dealt with by observing the number of feature points across different σ's (ranging from 2 to 5) for different image sizes (up to 512×512), as shown in Table I. Since at least 3 points are required to form a mesh, we need to choose σ's that can produce at least 3 feature points. Let σ_s be the largest value that cannot generate at least 3 feature points. In addition, the number of feature points cannot be so large as to yield small meshes such that a watermark cannot be completely embedded. According to Table I, the value of σ_d that can be

effective for watermark embedding is set to $\sigma_s - 3$ (≥ 1), which is defined as a detection scale. For example, for a 512×512 image, $\sigma_d = 6 - 3 = 3$ is adopted.

TABLE I

NUMBER OF DETECTED FEATURE POINTS AT DIFFERENT SCALES (σ'S) AND THE DETERMINED σ_s'S FOR THE IMAGE LENA OF DIFFERENT SIZES.

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

B. Content-Dependent Watermark

Some researchers [2], [22], [23], [25] have proposed inserting multiple redundant watermarks into an image in the hope that this will suffice to maintain resistance to geometric distortions as long as at least one watermark exists. The common framework is that certain types of image units, such as blocks [25], meshes [2], or disks [22], [23], are 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 to resist geometric distortions become weaker such that the false negative problem occurs. Of particular interest is the possible quality improvement of attacked media data that can be achieved by means of collusion attack. In addition, copy attack can also efficiently defeat a watermarking system by creating ambiguity problems. Since the common operation involving in both collusion and copy attacks is watermark estimation, they are called watermark-estimation attacks (WEAs) [11].

To withstand watermark-estimation attack, the key is to make the embedded watermarks different so that the hidden watermark cannot be approximately estimated by means of collusion. To this end, we propose to embed a media hash-based content-dependent watermark (CDW), which is composed of a watermark and a media hash. Our analyses [11] show that CDW is able to resist both copy and collusion attacks. Here, the block-based content-dependent watermark [11] is introduced. Each block of size $L_B \times L_B$ is divided into sub-blocks of size $L_{sub} \times L_{sub}$, and a block-pair relation is created by means of a secret key (the key is the same as that used to generate the watermark). For a pair of $L_{sub} \times L_{sub}$ blocks, a hash bit, defined as the magnitude relationship between two AC

coefficients, is represented as

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

where $MH(\cdot)$ is a hash bit in a hash sequence MH, 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 consisting of hash MH_i and watermark W, a media hash-based content-dependent watermark can be generated as

$$CDW = S(W, MH), \tag{5}$$

where $S(\cdot)$ is a shuffling function, which is basically application-dependent and will be used to control the combination of W and MH. Since media hashing is not the main theme of this paper, please refer [8] for more details about robustness verification of our image hashing scheme. The signal CDW is the watermark that we want to embed into a local region.

III. Proposed Watermarking Method

Basically, the proposed method is similar to the framework of mesh-based watermarking [2] proposed by Bas *et al.*. However, there are many differences between these two methods. First, we have investigated some important issues (described in Section 2) to improve the robustness. Second, we find from [2] 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 paradigm cannot efficiently achieve robustness. However, in our proposed scheme, the watermark embedding and extraction processes are both performed in the normalized domain. In addition, the modified coefficients in the normalized domain are warped to the spatial domain to accomplish embedding. Therefore, a trade-off between transparency and robustness can be better achieved. Third, we also propose a false positive-oriented watermark detection mechanism so that the trade-off between correct detection and false detection can be more successfully guaranteed (see Sec. IV). In the following, the proposed watermark embedding and extraction processes will be described.

A. Watermark Embedding

The watermark embedding process is outlined in Fig. 1. In this paper, the hidden watermark W is generated with a secret key and is a bipolar sequence of length L_W, i.e., $W = \{W_j\}_{j=1,2,\ldots,L_W}$ with each $W_j \in \{-1, +1\}$.

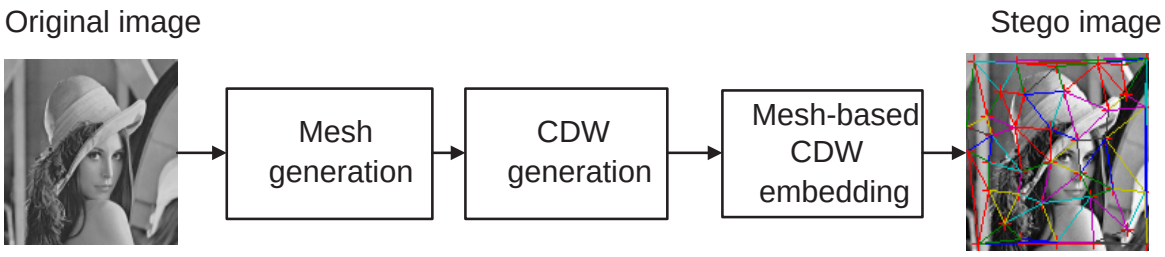


Fig. 1. **Block diagram of the embedding process.**

1) Mesh Generation: The first step in mesh generation is to filter a cover image using Gaussian filtering, as described in Sec. 2, so that a set of feature points P can be obtained. 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.

2) Content-Depentent Watermark Generation: The content-dependent watermark generation process, including (i) mesh normalization, (ii) media hash extraction, and (iii) hash-based content-dependent watermark, will be described in the following.

a) Mesh Normalization: Before embedding is performed, each triangle mesh has to be normalized to obtain a canonical form. Here, a mesh normalization process is performed to affine transform each extracted mesh M_i to obtain 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 cause the watermark extraction process to fail. Therefore, the size of a normalized mesh needs to be properly determined. Our empirical research has shown that if a larger region is warped into a small region, which means that the warping process is a multiple-to-one pixel mapping, then 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, which implies that a small normalized mesh of small size is beneficial for achieving robustness. In this study, the size of a normalized mesh is empirically found to be 48×48 for achieving a trade-off between transparency and robustness (this choice will become clear in the next two paragraphs). Let $NM = \{NM_i\}_{i=1,2,\cdots,L_M}$ denote the set of normalized meshes.

b) Mesh-based Hash Extraction: A mesh-based media hash, MH_i, is extracted from each normalized mesh NM_i, as described in Sec. II-B. Since this paper investigates a mesh-based watermarking scheme, each normalized mesh prior to hash extraction needs to be transformed into a block. Here, each normalized mesh is flipped and then

the flipped mesh is padded with the original version to form a block. If we set $L_B = 48$ and $L_{sub} = 6$, then the length of a hash sequence is 64.

c) Media Hash-based Content-dependent Watermark: In this paper, the watermark length (L_W) is set to be 128 bits. Although the length of the media hash (MH_i) is *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 (Eq. (5)) to generate the content-dependent watermarks, i.e., $CDW = \{CDW_i\}_{i=1,2\cdots L_M}$. Although only one watermark W is embedded for a cover image, the principle behind CDW leads to different signals embedded in different meshes.

3) Arrangement of Watermark Bits for Embedding: 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 see that the time of repetition is $\left\lfloor \frac{1152}{128} \right\rfloor = 9$. Let $R9CDW = \{R9CDW_i\}_{i=1,2\cdots L_M}$, where each element of CDW_i is repeated 9 times to form $R9CDW_i$. This repeated embedding is very important for achieving better robustness, in particular when the mesh is (slightly) perturbed because its constituent feature points are not exactly the same as the ones detected in the embedding process. In other words, the feature extraction error and other numerical errors such as interpolation errors and rounding errors will affect the watermark detection performance. In order to deal efficiently with these problems, the repeated embedding of a watermark bit is performed [5], [11], [12], [21]. Recall that in [22] the authors proposed to deal with this problem through locally searching (75 times) for the possibly correct feature point in the neighborhood of the detected point.

In summary, 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 satisfy the embedding purpose so that robustness can be better achieved.

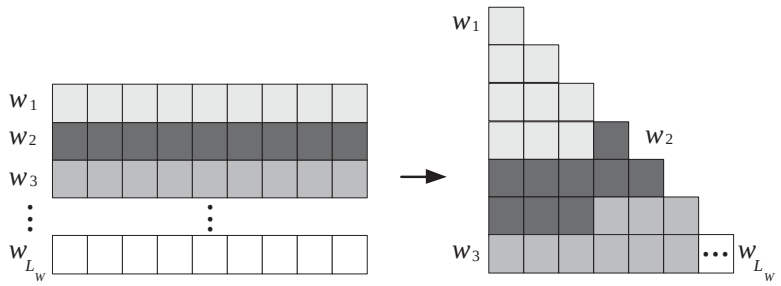


Fig. 2. **(left) The repeated watermark bits (each bit is repeated 9 times) are arranged and embedded in a normalized mesh (right).**

4) Mesh-based Embedding: In order to maintain transparency after performing watermarking, we adopt the Noise Visibility Function (NVF) [24], which is an image-dependent visual model. Content adaptive watermark embedding is designed to insert watermarks into the cover image $I(,)$ to form a stego image $I^w(,)$ as follows:

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

where S and S_1 denote the watermark strength, and w_j is an element of a bipolar watermark signal. In [24], the authors proposed to set S_1 to 3 for most real world and computer generated images. As for S, it can be adjusted to keep the PSNR higher than a certain value. In our method, $S_1 = 3$ is adopted, and S is adjusted to keep the PSNRs all at about 38 dB. Therefore, in our watermarking scheme, the watermark embedding process can be designed as

$$NM_i^w(x,y) = NM_i(x,y) + (1 - NVF(x,y)) \cdot r9cdw_{ij} \cdot S + NVF(x,y) \cdot r9cdw_{ij} \cdot S_1, \tag{7}$$

where $r9cdw_{ij}$ denotes the jth watermark element of $R9CDW_i$, 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 "direct inverse normalization" is intuitive, 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, i.e., the second term on the right-hand side of Eq. (7), which is caused by watermarking in the normalized domain, is inversely normalized to yield the difference M_i^{diff} in the spatial domain. Hence, the watermarked mesh in the spatial domain can be obtained as

$$M_i^w = M_i + M_i^{diff}. \tag{8}$$

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

In order to illustrate the advantage of our embedding method (Eq. (8)) over inverse normalization (Eq. (7)), an example is shown in Fig. 3 for visual comparison. Fig. 3(a) shows a stego Lena image that is generated through inverse normalization of watermarked meshes. Many interpolation errors and blocky effects can be observed. On the other hand, if the embedded signal in the normalized domain is transformed back to the spatial domain and then added to the original image, then as Fig. 3(b) shows, the visual quality is not perceptually degraded.

B. Watermark Extraction

The process of determining the existence of a watermark is depicted in Fig. 4. Basically, the watermark extraction process is the inverse process of watermark embedding.

Fig. 3. **Transparency comparison for watermarked Lena images based on (a) direct inverse normalization of watermarked meshes (Eq. (7)), PSNR=**28.99 **dB; (b) inverse normalization of the embedded signal (Eq. (8)) plus the original image, PSNR=**39.87 **dB.**

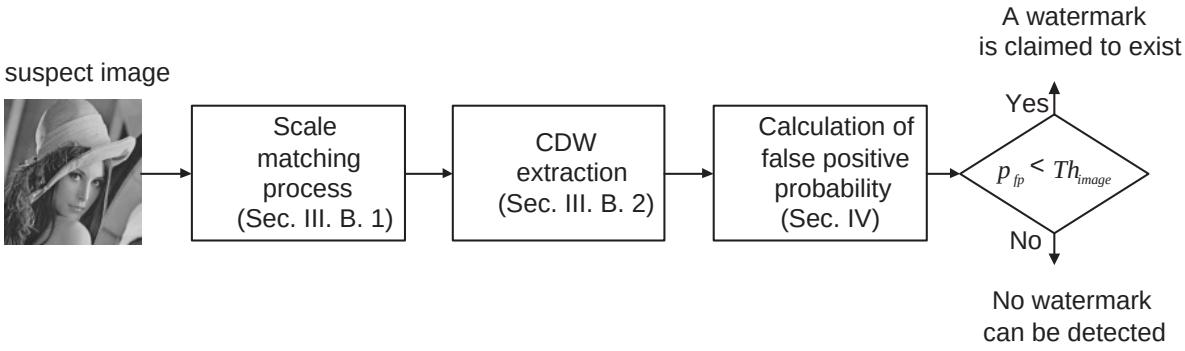


Fig. 4. **Block diagram of the process of determining the existence of a watermark.**

1) Scale Matching Process: In the watermark extraction end, the first step is to determine σ's that will be used for filtering (Eq. (2)). Initially, σ_d as determined in the embedding end can be used; however, due to possible modifications of the stego image, a single value, σ_d, cannot be guaranteed to match the characteristics of the encountered attacked images. In order to tolerate varied attacks, in addition to σ_d, other σ's may be needed. Some scenarios that will change the size of an image are described in the following to prove the need for several σ's. If the size of a stego image is changed due to cropping (e.g., rotation+cropping), then σ_d will fail to capture the characteristics of the cropped images because it cannot distinguish between scaling and cropping that lead to changes of the images' sizes. On the other hand, for a huge image, the watermark embedding and extraction

processes should be operated in a tiling manner. The tile size selected in our proposed scheme is 512×512, which always sets σ_d to a fixed value 3, as described in Sec. II-A.3. If the huge image is scaled down or up, then σ_d will be useless for capturing this change. Therefore, a scale matching process is proposed here to help us determine proper σ's for filtering.

First of all, we have to know the possible range of change of an image's size. Let us take the standard benchmark, Stirmark [19], [20], as an example. For all non-geometric attacks, scaling, and other attacks that cause slight changes of an image's size, σ_d as determined in the embedded process can be used. For those attacks that have cropping effects (in Stirmark, rotation of 45° and cropping of more than 50% cause severer cropping effects), the size of an image could be quartered. Under these circumstances, $\sigma_d + 1$ instead of σ_d needs to be used. Here, let $\sigma_d + 1$ be written as σ_{d+1}.

On the other hand, in the case of a huge image, it is not known whether the contents contained within a tile have been attacked or not. When we consider the modifications caused by scaling with factors ranging from $50\% \sim 200\%$ (as provided in Stirmark), it is not hard to see that $\sigma_{d-1} = \sigma_d - 1$, σ_d, and σ_{d+1} are necessary to adapt to various tile sizes.

In summary, three filtering parameters, σ_{d-1}, σ_d, and σ_{d+1}, are required for filtering to extract the desired feature points under the constraint that Stirmark is considered for possible attacks. Of course, more filtering parameters can be used at the cost of more time spent to deal with attacks that cause severer effects. Here, let M_{d-1}, M, and M_{d+1}, respectively, denote the sets of meshes extracted using σ_{d-1}, σ_d, and σ_{d+1}.

2) Media Hash-based Content-Dependent Watermark Extraction: The proposed content-dependent watermark extraction process is depicted in Fig. 5. The normalization process is used to, respectively, transform the three sets of meshes, M, M_{d+1}, and M_{d-1}, into corresponding sets of normalized meshes, NM, NM_{d+1}, and NM_{d-1}, from which three sets of media hashes, MH_d, MH_{d+1}, and MH_{d-1}, can be extracted.

In this paper, Wiener filtering is used to blindly extract the hidden signal. Wiener filtering is considered to be an efficient method [6], [11], [25] because the watermark is usually a high-frequency signal. Let $R9CDW^e_{d\,i}$, $R9CDW^e_{d+1_i}$, and $R9CDW^e_{d-1_i}$ be, respectively, extracted from NM_{d_i}, NM_{d+1_i}, and NM_{d-1_i}. Since the watermark bits are redundantly embedded, a bit is finally determined based on a majority selection rule. In this paper, each bit is repeatedly embedded into a mesh 9 times. For an embedded bit, if most of its corresponding extracted bits are 1(-1), then the extracted bit is finally determined to be 1(-1). Let $CDW^e_{d\,i}$, $CDW^e_{d+1_i}$, and $CDW^e_{d-1_i}$ be the extracted watermarks after the majority determination process is completely.

Next, three sets of extracted media hashes, MH_d, MH_{d+1}, and MH_{d-1}, corresponding to σ_d, σ_{d+1}, and σ_{d-1},

respectively, are separated from their corresponding watermarks, CDW^e_{di}, CDW^e_{d+1i}, and CDW^e_{d-1i}, as follows:

$$W^e_d = \{W^e_{di}\}_{i=1,2\cdots L_M}, W^e_{di} = (CDW^e_{di}/MH_{di}), \tag{9}$$

$$W^e_{d+1} = \left\{W^e_{d+1i}\right\}_{i=1,2\cdots L_M}, W^e_{d+1i} = (CDW^e_{d+1i}/MH_{d+1i}), \tag{10}$$

$$W^e_{d-1} = \left\{W^e_{d-1i}\right\}_{i=1,2\cdots L_M}, W^e_{d-1i} = (CDW^e_{d-1i}/MH_{d-1i}). \tag{11}$$

Thus, we obtain the extracted watermark signals W^e_d, W^e_{d+1}, and W^e_{d-1}.

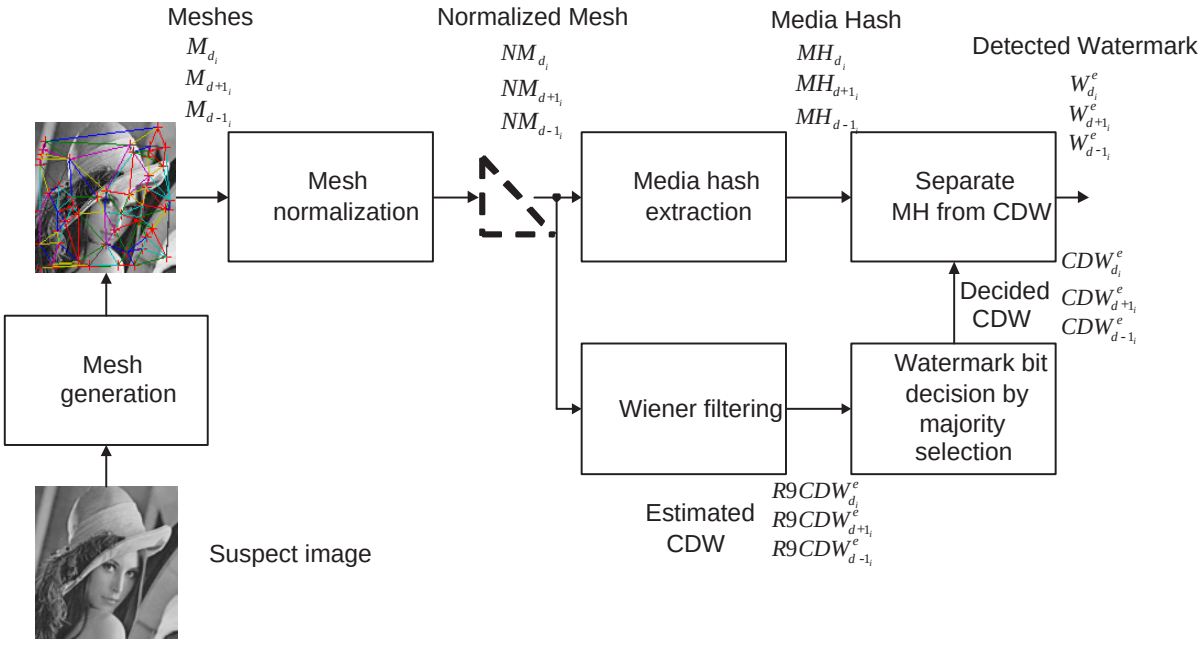


Fig. 5. **Block diagram of the CDW extraction process.**

IV. False Positive-Oriented Determination of the Existence of a Watermark

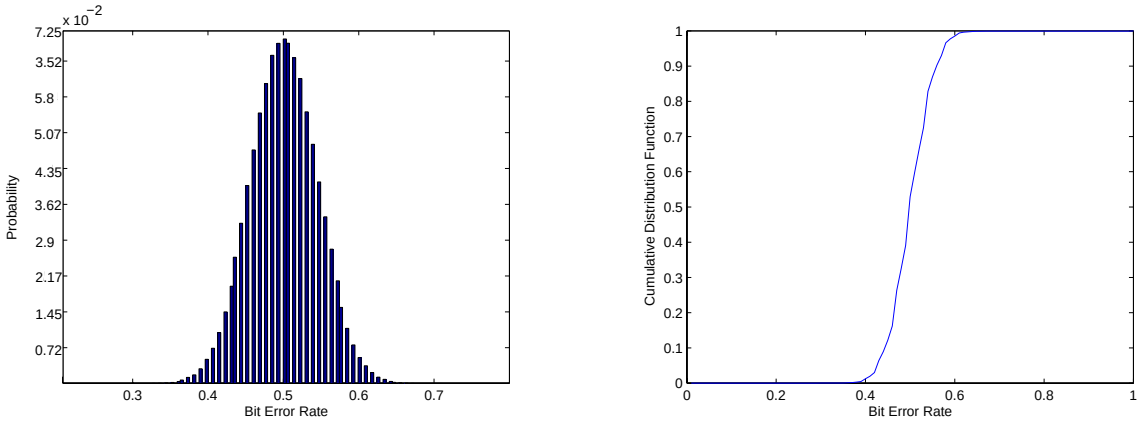
In order to indicate the presence/absence of a watermark in an image, the first step is to determine whether a watermark exists in a mesh. For each NM_{di} (or NM_{d+1i}, NM_{d-1i}), the bit-error rate (BER) between W and W^e_{di} (or W and W^e_{d+1i}, W and W^e_{d-1i}) is calculated. If the BER is smaller than a threshold Th_{mesh}, it is said that a watermark exists in a mesh. The threshold Th_{mesh} needs to be determined by considering the false positive factor because to claim the robustness of a watermarking system is meaningful 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 probability p_b, which is the probability that the bit b (-1 or 1) will occur, and is considered to always be 0.5 here. Theoretically, the probability of truly detecting a watermark in a mesh when BER$\leq Th_{mesh}$ holds can be represented as

$$p_{Ms} = \sum_{j=(L_W - L_W \times Th_{mesh})}^{L_W} (_j^{L_W}) {p_b}^j (1-p_b)^{L_W - j}. \tag{12}$$

Eq. (12) also specifies the probability that a watermark can be found in a mesh that has not, in fact, been watermarked. As a result, determining the threshold Th_{mesh} is important.

In order to reasonably determine Th_{mesh}, p_{Ms} in Eq. (12) should be consistent with practical results. To this end, the BERs obtained from extensive "sequence-pair" comparisons were collected. A sequence-pair is composed of the watermark known by the owner and a signal that is extracted from one of the meshes in a random image. First of all, every un-watermarked image chosen from the Corel image database was applied as the input to our watermark detection process, as described in Sec. III-B. For each image, a set of BERs could be obtained after sequence-pair comparisons were performed. After testing all $20,000$ images in the Corel image database, we obtained the BER distribution and its cumulative distribution shown in Fig. 6. Based on this information, if Th_{mesh} is chosen to be 0.375, then p_{Ms} in Eq. (12) is calculated to be 0.003, which is very close to the cumulative distribution function (*cdf*), $cdf(BER \leq 0.375) = 0.0027$, of the BER distribution measured using the Corel image database. Consequently, it can be concluded that $Th_{mesh} = 0.375$ is a reasonable choice.



(a) (b)

Fig. 6. **Sequence-pair comparisons (one is the watermark and the other one is one of the signals extracted from the Corel image database: (a) distribution of the BERs; (b) cumulative distribution of (a).**

On the other hand, there are three vertexes in each M_i. However, some geometric attacks may change the relationship between the three vertexes, which 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. Thus, the probability of detecting a watermark in a mesh, p_M, can be derived as

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

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

So far, we have discussed how one can determine the existence of a watermark in a mesh. Now, we will proceed to explain how one can determine the existence of a watermark in an image by incorporating the mesh-based detection results. Recall that L_M is the number of meshes in an image (no matter whether it is attacked or not). Let D_M be the number of meshes found to have been watermarked, as described in the above paragraphs. The probability of determining that a suspect image was watermarked before is derived as

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

based on the constraint that D_M out of a total L_M of meshes are regarded to having been watermarked. In fact, Eq. (14) also reveals the probability that a random image will be "wrongly" determined as having been watermarked. Furthermore, this also implies that different attacks lead to different p_{fp}'s; i.e., a more challenging attack will generate a higher false positive probability.

In order to claim the presence of a watermark with strong confidence (without causing a non-negligible false positive), p_{fp} should be low. On the other hand, p_{fp} should be large to achieve robustness. Here, a reasonable threshold, Th_{image}, is required to satisfy the trade-off between robustness and false positive. Again, the Corel image database was adopted here to derive Th_{image}. Every un-watermarked image chosen from the Corel image database was applied as the input to our watermark detection process. For each image, one p_{fp} was obtained based on Eq. (14). By integrating all the p_{fp}'s, the cumulative distribution function showed that $cdf(p_{fp} \leq (3.50e-004)) = 0$ and $cdf(p_{fp} \leq (4.00e-004)) = (6.28e-005)$. Thus as a guideline, it is helpful to set the threshold Th_{image} to $3.50e-004$ according to the information obtained from the Corel image database. It should be noted that although meshes are adopted in this paper, similar results can be obtained using other types of image units, such as blocks or disks.

It should be noted that since three σ's are employed for watermark detection, three p_{fp}'s are generated. The smallest value will be chosen as the final p_{fp} (corresponding to the largest D_M).

A. Comparison with other methods

In this section, some recent papers that have proposed feature-based watermarking methods will be discussed. False positive probability analysis was also conducted in [22], [23], which proposed to embed watermarks into disks that are extracted from an image. However, the existence of a watermark was not finally determined by taking the derived false positive probability into consideration. On the contrary, these authors only indicate the number of disks (out of the number of total disks) that can be found to contain the hidden watermark.

In [23], the false positive probability derived from each disk was defined in Eq.(5) of their paper as follows:

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

where $n = 16$, $T_1 = 10$, $T_2 = 10$, and $T = 24$. When the parameters are substituted 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 [23] as

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

where N is total number of disks in an image, and at least m disks are detected as "successful."

In [22], the false positive probability for one image was defined in Eq.(23) of their paper as follows:

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

where the watermark is detected from at least μ disks and N is the number of disks in an image that are available for watermarking. In their method, $N = 100$.

In this study, robustness comparisons among our method, Seo and Yoo's method [22], and Tang and Hang's method [23] were conducted by taking the derived false positive probabilities into consideration. The results will be reported in the next section.

V. EXPERIMENTAL RESULTS

In order to thoroughly verify the robustness of the proposed scheme, the standard benchmark, Stirmark 3.1 [19], [20], and the watermark-estimation attacks (WEAs) [11] were adopted. In the literature, a more thorough verification could only be found in [25]. Three standard images, Baboon, Lena, and Pepper, were used here as cover images, and the size of each one was 512×512. After mesh-based watermark embedding was performed, the PSNR values between the cover image and its stego image for Baboon, Lena, and Pepper were *36.06*dB, *38.44*dB, and *38.32*dB, respectively. No perceptual differences could be observed. Although the PSNR of stego Baboon was smaller than *38*dB, it was still hard to find any quality degradation because the Baboon image was rather noisy. As described previously, two thresholds, $Th_{mesh} = 0.375$ and $Th_{image} = 3.50e-004$, were adopted in this study.

In order to demonstrate the superiority of our method, we compared it with other feature-based watermarking methods [22], [23]. Since Bas *et al.*'s scheme [2] was not evaluated using Stirmark, it was not considered for comparison here. In digital watermarking, it has been recognized that robustness is meaningful only if false positives are taken into consideration. Although false positive analyses were conducted in [22], [23], the detection results did not show the impact of this factor, so the reported results are not fully convincing. Therefore, in this study the false positive probability was derived using Eq. (16) for the method in [23], and Eq. (17) for the method in [22]. To avoid tedious comparisons, the parameters that could produce better results in [22], [23] were adopted here. In [23], $n = 16$, $T_1 = 10$, $T_2 = 10$, and $T = 24$ were used, leading to $P_{False-alarm\ on\ one\ disk} = 0.0034$. The number of disks, m, found to contain watermarks and the number of total disks, N, in Eq. (16) are denoted in the following tables as D_M and T_M, respectively. In [22], the authors declared that when $\mu = 1$ and $P_{FA-image} = 0.1918e-004$ are used, $P_{FA-disk} = 0.1918e-006$ is obtained according to Eq. (17). The number of disks, μ, detected to contain a watermark and the number of total disks, N, in Eq. (17) are denoted in the following tables as D_M and T_M, respectively. $N = 100$ was adopted in [22]. "D_M/T_M" in the following tables denotes "the number of detected watermarked meshes(disks)/the number of total meshes(disks)." In this paper, experimental results will be demonstrated with respect to resistance to removal (non-geometric) attacks, resistance to geometric attacks, and resistance to watermark-estimation attack in the following subsections. Finally, the reasons that may lead to the obtained results will be identified.

A. Resistance to Non-geometric Attacks

The watermark detection results with respect to non-geometric attacks are shown in Tables II, III, and IV for the three standard images, respectively. In Table II, the method in [22] can only survive FMLR and Color reduce attacks,

TABLE II

NON-GEOMETRIC ATTACKS FOR BABOON.

attack	proposed method		[22]		[23]	
	D_M/T_M	p_{fp}	D_M	p_{fp}	D_M/T_M	p_{fp}
Median filter 2x2	8/106	1.52e-009	-	-	6/11	7.07e-013
Median filter 3x3	6/111	1.26e-006	-	-	2/11	6.24e-004
Median filter 4x4	6/109	1.13e-006	1	1.91e-005	-	-
Gaussian filter 3x3	8/108	1.77e-009	0	1.00e-000	8/11	2.94e-018
JPEG 90	5/111	2.39e-005	-	-	-	-
JPEG 80	7/115	7.22e-008	-	-	9/11	3.35e-021
JPEG 70	4/115	4.30e-004	1	1.91e-005	11/11	7.10e-028
JPEG 60	6/103	8.13e-007	1	1.91e-005	7/11	1.72e-015
JPEG 50	5/110	2.29e-005	1	1.91e-005	5/11	2.07e-010
JPEG 40	4/113	4.02e-004	1	1.91e-005	7/11	1.72e-015
JPEG 30	5/114	2.72e-005	0	1.00e-000	4/11	4.34e-008
JPEG 20	4/106	3.15e-004	-	-	-	-
JPEG 10	1/124	3.11e-001	-	-	-	-
FMLR	6/106	9.62e-007	4	5.30e-021	-	-
Color reduce	8/109	1.90e-009	2	1.82e-010	4/11	4.34e-008
Sharpening 3x3	4/120	5.04e-004	0	1.00e-000	2/11	6.24e-004

while our method and that in [23] can tolerate JPEG compression up to a quality factor of 40%. Furthermore, only our method can survive the Sharpening attack. As shown in Table III, our method can survive almost all attacks except for JPEG10 and FMLR attacks, so it is more robust than the other two. A similar result can also be found in Table IV. On a whole, our method when compared with those in [22], [23], can survive most of the non-geometric attacks of Stirmark 3.1. We also note that it is challenging to extract robust feature points from complex images such as Baboon. Thus, the overall performance with respect to Baboon is not as robust as that for other smoothing images. This phenomenon was observed in [2], [22], [23] as well as in our study.

B. Resistance to Geometric Attacks

The results of comparisons of resistance to geometric distortions are shown in Tables V $\sim$ VII. Basically, it can be observed that our method and that in [22] provide p_{fp} that is sufficiently lower than that in [23] for line removal, cropping attacks, and general linear transformations. Our method also consistently provides much lower p_{fp}'s for shearing and random bending attacks. For other attacks, our method was thoroughly evaluated and found

TABLE III

NON-GEOMETRIC ATTACKS FOR LENA.

attack	proposed method		[22]		[23]	
	D_M/T_M	p_{fp}	D_M	p_{fp}	D_M/T_M	p_{fp}
Median filter 2x2	28/110	1.98e-045	-	-	1/8	2.69e-002
Median filter 3x3	16/111	2.68e-022	-	-	1/8	2.69e-002
Median filter 4x4	16/102	6.40e-023	5	1.95e-026	-	-
Gaussian filter 3x3	23/103	4.00e-036	3	1.14e-015	5/8	2.53e-011
JPEG 90	32/106	1.97e-054	-	-	-	-
JPEG 80	35/104	2.38e-061	-	-	6/8	4.32e-014
JPEG 70	31/104	1.40e-052	3	1.14e-015	7/8	4.22e-017
JPEG 60	26/111	3.08e-041	3	1.14e-015	6/8	4.32e-014
JPEG 50	15/111	1.49e-020	1	1.91e-005	5/8	2.53e-011
JPEG 40	21/117	6.22e-031	1	1.91e-005	3/8	2.18e-006
JPEG 30	18/116	1.65e-025	0	1.00e-000	2/8	3.19e-004
JPEG 20	7/100	2.74e-008	-	-	-	-
JPEG 10	1/114	2.90e-001	-	-	-	-
FMLR	3/97	3.23e-003	1	1.91e-005	-	-
Color reduce	29/104	2.59e-048	4	5.30e-021	7/8	4.22e-017
Sharpening 3x3	18/115	1.40e-025	1	1.91e-005	4/8	9.29e-009

to provide low p_{fp}'s, while [22], [23] did not. This is particularly obvious for resistance to change of the aspect ratio because the circular disk adopted in [22], [23] could not accommodate such an attack. Since our method adopts a triangular mesh for watermarking, it is better able to adapt to varied attacks.

C. Resistance to Watermark-Estimation Attacks (WEAs)

The collusion attack and copy attack were used to verify the resistance achieved by our method to WEAs [11]. Table VIII and Table IX show the results of resisting collusion attack for CDW embedding and non-CDW embedding, respectively. After a collusion attack was performed, the number of detected meshes as shown in Table IX was smaller than that shown in Table VIII, which implies that our proposed scheme with CDW embedding efficiently defends against the collusion attack. It should also be noted that mesh-based collusion does not increase the PSNRs of colluded images as block-based collusion does [11]. This may be due to the fact that the interpolation errors involving in mesh warping neutralize the expected PSNR improvement of collusion. Table X and Table XI show the results of resisting copy attack for CDW embedding and non-CDW embedding, respectively. After a copy attack

TABLE IV

NON-GEOMETRIC ATTACKS FOR PEPPER.

attack	proposed method		[22]		[23]	
	D_M/T_M	p_{fp}	D_M	p_{fp}	D_M/T_M	p_{fp}
Median filter 2x2	38/108	2.33e-067	-	-	1/4	1.35e-002
Median filter 3x3	40/107	4.12e-072	-	-	1/4	1.35e-002
Median filter 4x4	24/109	1.83e-037	4	5.30e-021	-	-
Gaussian filter 3x3	36/108	7.07e-063	5	1.95e-026	1/4	1.35e-002
JPEG 90	39/111	4.63e-069	-	-	-	-
JPEG 80	44/109	5.36e-081	-	-	3/4	1.57e-007
JPEG 70	44/107	1.90e-081	6	5.93e-032	3/4	1.57e-007
JPEG 60	33/106	1.33e-056	6	5.93e-032	1/4	1.35e-002
JPEG 50	30/108	7.24e-050	4	5.30e-021	3/4	1.57e-007
JPEG 40	27/111	2.92e-043	4	5.30e-021	1/4	1.35e-002
JPEG 30	20/112	1.75e-029	4	5.30e-021	0/4	1.00e-000
JPEG 20	9/118	1.31e-010	-	-	-	-
JPEG 10	2/115	4.72e-002	-	-	-	-
FMLR	11/101	2.20e-014	0	1.00e-000	-	-
Color reduce	54/109	2.44e-105	2	1.82e-010	1/4	1.35e-002
Sharpening 3x3	21/117	6.22e-031	5	1.95e-026	4/4	1.34e-010

was performed, the number of detected meshes as shown in Table XI was larger than that shown in Table X, which implies our proposed scheme with CDW embedding efficiently defends against the copy attack. However, the content-independent watermarking methods [2], [22], [23] cannot survive WEAs [11].

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

D. Discussions

In this section, we shall discuss the impact of each step in our method on the detection results and identify which step mostly affects the overall performance. As described previously in Sec. III, in addition to media hashing, feature point extraction and denoising-based blind detection are recognized as two main factors that may affect the performance of our method. Since the robustness of our media hashing has been verified in [8], it is not discussed here again. According to the experimental results shown in the above tables, it is important to know how many meshes of a stego image, under the absence of attacks, can be detected to contain watermarks. Two experiments

TABLE V

GEOMETRIC ATTACKS FOR BABOON.

attack	proposed method		[22]		[23]	
	D_M/T_M	p_{fp}	D_M	p_{fp}	D_M/T_M	p_{fp}
1 column, 1 row removed	5/111	2.39e-005	-	-	-	-
5 column, 1 row removed	11/115	9.47e-014	-	-	6/11	7.07e-013
1 column, 5 row removed	6/111	1.26e-006	-	-	-	-
17 column, 5 row removed	3/106	4.14e-003	1	1.91e-005	3/11	6.37e-006
5 column, 17 row removed	8/100	9.55e-010	-	-	-	-
Cropping 1% off	10/110	2.11e-012	-	-	-	-
Cropping 2% off	4/112	3.89e-004	-	-	-	-
Cropping 5% off	5/110	2.29e-005	-	-	2/11	6.24e-004
Cropping 10% off	6/96	5.36e-007	-	-	2/11	6.24e-004
Cropping 15% off	7/526	1.21e-003	4	5.30e-021	-	-
Cropping 20% off	5/87	7.32e-006	-	-	-	-
Cropping 25% off	5/411	8.52e-003	1	1.91e-005	-	-
Cropping 50% off	13/648	1.38e-007	-	-	-	-
Linear(1.007, 0.010, 0.010, 1.012)	6/115	1.55e-006	3	1.14e-015	4/11	4.34e-008
Linear(1.010, 0.013, 0.009, 1.011)	5/109	2.19e-005	1	1.91e-005	4/11	4.34e-008
Linear(1.013, 0.008, 0.011, 1.008)	6/111	1.26e-006	0	1.00e-000	5/11	2.07e-010
Aspect ratio change(0.80, 1.00)	7/84	8.09e-009	-	-	-	-
Aspect ratio change(0.90, 1.00)	8/93	5.33e-010	-	-	-	-
Aspect ratio change(1.00, 0.80)	1/90	2.37e-001	-	-	-	-
Aspect ratio change(1.00, 0.90)	4/96	2.16e-004	-	-	-	-
Aspect ratio change(1.00, 1.20)	7/121	1.02e-007	-	-	-	-
Aspect ratio change(1.00 1.10)	9/115	1.04e-010	-	-	-	-
Aspect ratio change(1.10, 1.00)	8/127	6.40e-009	-	-	-	-
Aspect ratio change(1.20, 1.00)	6/131	3.31e-006	-	-	-	-
Rotation 1.00	11/113	7.78e-014	-	-	3/11	6.37e-006
Rotation 2.00	6/107	1.02e-006	-	-	1/11	3.68e-002
Rotation 5.00	3/103	3.82e-003	-	-	0/11	1.00e-000
Rotation 10.00	9/99	2.67e-011	-	-	-	-
Rotation 15.00	4/84	1.29e-004	-	-	-	-
Rotation 30.00	4/61	3.69e-005	-	-	-	-
Rotation 45.00	8/359	1.64e-005	1	1.91e-005	-	-
Rotation 90.00	5/111	2.39e-005	-	-	-	-
Flipping	1/111	2.84e-001	-	-	-	-

TABLE V

-Continued.

attack	proposed method		[22]		[23]	
	D_M/T_M	p_{fp}	D_M	p_{fp}	D_M/T_M	p_{fp}
Rotation Scale 1.00	6/113	1.40e-006	-	-	4/11	4.3451e-008
Rotation Scale 10.00	7/122	1.08e-007	-	-	-	-
Rotation Scale 15.00	1/29	8.34e-002	-	-	-	-
Rotation Scale 30.00	8/748	2.19e-003	-	-	-	-
Rotation Scale 45.00	10/740	1.05e-004	-	-	-	-
Rotation Scale 90.00	5/111	2.39e-005	-	-	-	-
Scaling 50%	2/104	3.94e-002	0	1.00e-000	-	-
Scaling 75%	6/323	4.89e-004	0	1.00e-000	-	-
Scaling 90%	5/77	4.01e-006	2	1.82e-010	-	-
Scaling 110%	5/132	5.48e-005	-	-	-	-
Scaling 150%	4/328	1.78e-002	-	-	-	-
Scaling 200%	7/119	9.13e-008	-	-	-	-
Shearing x-0% y-1%	8/111	2.20e-009	-	-	-	-
Shearing x-1% y-0%	9/110	6.96e-011	2	1.82e-010	-	-
Shearing x-1% y-1%	5/114	2.72e-005	-	-	4/11	4.34e-008
Shearing x-0% y-5%	10/109	1.92e-012	-	-	3/11	6.37e-006
Shearing x-5% y-0%	6/106	9.62e-007	-	-	-	-
Shearing x-5% y-5%	6/103	8.13e-007	0	1.00e-000	0/11	1.00e-000
Random Bending	6/116	1.63e-006	0	1.00e-000	-	-

were performed based on the conditions that (i) the feature points and media hashes extracted from the original image are directly applied to the stego image, which means that feature point extraction is perfect and we are only interested in understanding the effect of Wiener filtering; and (ii) all the processes are the same as those described in Sec. III, which means that by comparing the results obtained from conditions (i) and (ii) we can understand the effect of feature point extraction (and media hashing). The results of these two experiments are depicted in Table XII.

As we can see from Table XII that when condition (i) is considered, denoising-based blind detection slightly affects the detection results. For example, the number, T_M, of total meshes in Baboon is 103 and the number of meshes, D_M, detected to contain watermarks is 67. The similar results can also be found in Lena and Pepper. However, when condition (ii) is considered, D_M for each stego image, when compared with the results obtained in

TABLE VI

GEOMETRIC ATTACKS FOR LENA.

attack	proposed method		[22]		[23]	
	D_M/T_M	p_{fp}	D_M	p_{fp}	D_M/T_M	p_{fp}
1 column, 1 row removed	34/100	7.99e-060	-	-	-	-
5 column, 1 row removed	35/104	2.38e-061	-	-	3/8	2.18e-006
1 column, 5 row removed	29/104	2.59e-048	-	-	-	-
17 column, 5 row removed	24/110	2.33e-037	5	1.95e-026	0/8	1.00e-000
5 column, 17 row removed	10/100	8.00e-013	-	-	-	-
Cropping 1% off	27/106	7.14e-044	-	-	-	-
Cropping 2% off	22/103	3.78e-034	-	-	-	-
Cropping 5% off	17/86	4.23e-026	-	-	2/8	3.19e-004
Cropping 10% off	16/77	4.95e-025	-	-	2/8	3.19e-004
Cropping 15% off	15/65	2.58e-024	6	5.93e-032	-	-
Cropping 20% off	12/68	3.31e-018	-	-	-	-
Cropping 25% off	12/53	1.27e-019	4	5.30e-021	-	-
Cropping 50% off	5/21	4.75e-009	-	-	-	-
Linear(1.007, 0.010, 0.010, 1.012)	32/104	9.60e-055	6	5.93e-032	5/8	2.53e-011
Linear(1.010, 0.013, 0.009, 1.011)	39/104	2.05e-070	7	1.52e-037	4/8	9.29e-009
Linear(1.013, 0.008, 0.011, 1.008)	28/100	9.28e-047	7	1.52e-037	4/8	9.29e-009
Aspect ratio change(0.80, 1.00)	6/87	2.99e-007	-	-	-	-
Aspect ratio change(0.90, 1.00)	15/94	1.07e-021	-	-	-	-
Aspect ratio change(1.00, 0.80)	3/97	3.23e-003	-	-	-	-
Aspect ratio change(1.00, 0.90)	7/104	3.60e-008	-	-	-	-
Aspect ratio change(1.00, 1.20)	18/121	3.69e-025	-	-	-	-
Aspect ratio change(1.00 1.10)	31/104	1.40e-052	-	-	-	-
Aspect ratio change(1.10, 1.00)	19/122	7.10e-027	-	-	-	-
Aspect ratio change(1.20, 1.00)	13/132	3.69e-016	-	-	-	-
Rotation 1.00	21/109	1.24e-031	-	-	3/8	2.18e-006
Rotation 2.00	21/93	3.15e-033	-	-	0/8	1.00e-000
Rotation 5.00	18/78	6.94e-029	-	-	0/8	1.00e-000
Rotation 10.00	15/77	4.25e-023	-	-	-	-
Rotation 15.00	12/73	8.25e-018	-	-	-	-
Rotation 30.00	9/57	1.56e-013	-	-	-	-
Rotation 45.00	6/38	1.85e-009	2	1.82e-010	-	-
Rotation 90.00	23/108	1.34e-035	-	-	-	-
Flipping	19/108	5.95e-028	-	-	-	-

TABLE VI

-Continued.

attack	proposed method		[22]		[23]	
	D_M/T_M	p_{fp}	D_M	p_{fp}	D_M/T_M	p_{fp}
Rotation Scale 1.00	24/105	6.72e-038	-	-	0/8	1.00e-000
Rotation Scale 10.00	7/98	2.38e-008	-	-	-	-
Rotation Scale 15.00	5/89	8.18e-006	-	-	-	-
Rotation Scale 30.00	1/115	2.92e-001	-	-	-	-
Rotation Scale 45.00	0/96	1.00e+000	-	-	-	-
Rotation Scale 90.00	23/108	1.34e-035	-	-	-	-
Scaling 50%	10/110	2.11e-012	2	1.82e-010	-	-
Scaling 75%	3/58	7.36e-004	3	1.14e-015	-	-
Scaling 90%	4/94	1.99e-004	4	5.30e-021	-	-
Scaling 110%	19/120	5.08e-027	-	-	-	-
Scaling 150%	3/57	7.00e-004	-	-	-	-
Scaling 200%	32/102	4.61e-055	-	-	-	-
Shearing x-0% y-1%	23/100	1.88e-036	-	-	-	-
Shearing x-1% y-0%	33/102	2.94e-057	5	1.95e-026	-	-
Shearing x-1% y-1%	23/100	1.88e-036	-	-	4/8	9.29e-009
Shearing x-0% y-5%	15/92	7.57e-022	-	-	2/8	3.19e-004
Shearing x-5% y-0%	20/94	3.81e-031	-	-	-	-
Shearing x-5% y-5%	12/78	1.92e-017	1	1.91e-005	1/8	2.69e-002
Random Bending	17/110	3.83e-024	4	5.30e-021	-	-

condition (i), is dramatically reduced. This obviously implies that the correctness of extracted points plays a major role in the performance of our watermarking method. More specifically, it can be observed from Table XII that the average displacements (in pixels) of feature points illustrate the obtained detection results. As a consequence, we can conclude that the stability of feature point extraction mainly affects the overall performance of our watermarking method. This conclusion is also consistent with the robustness verifications described in the above subsections that resistance to attacked Baboon images is apparently inferior to resistance to other smoother images.

VI. CONCLUSIONS

Although multiple watermarks can be embedded into an image to provide resistance to geometric distortions, we found in our companion study [11] that they are, unfortunately, vulnerable to watermark estimation attacks (including collusion and copy attacks) such that the desired geometric invariance is lost. In view of this fact, a mesh-based

TABLE VII

GEOMETRIC ATTACKS FOR PEPPER.

attack	proposed method		[22]		[23]	
	D_M/T_M	p_{fp}	D_M	p_{fp}	D_M/T_M	p_{fp}
1 column, 1 row removed	45/111	6.59e-083	-	-	-	-
5 column, 1 row removed	42/108	1.56e-076	-	-	3/4	1.57e-007
1 column, 5 row removed	39/105	3.25e-070	-	-	-	-
17 column, 5 row removed	34/104	3.96e-059	5	1.95e-026	1/4	1.35e-002
5 column, 17 row removed	34/104	3.96e-059	-	-	-	-
Cropping 1% off	34/110	3.81e-058	-	-	-	-
Cropping 2% off	24/110	2.33e-037	-	-	-	-
Cropping 5% off	17/94	2.19e-025	-	-	2/4	6.91e-005
Cropping 10% off	17/88	6.47e-026	-	-	2/4	6.91e-005
Cropping 15% off	14/74	1.84e-021	2	1.82e-010	-	-
Cropping 20% off	14/59	5.61e-023	-	-	-	-
Cropping 25% off	6/60	3.18e-008	2	1.82e-010	-	-
Cropping 50% off	4/19	3.03e-007	-	-	-	-
Linear(1.007, 0.010, 0.010, 1.012)	41/111	1.31e-073	5	1.95e-026	1/4	1.35e-002
Linear(1.010, 0.013, 0.009, 1.011)	46/108	5.65e-086	7	1.52e-037	1/4	1.35e-002
Linear(1.013, 0.008, 0.011, 1.008)	45/110	3.93e-083	5	1.95e-026	4/8	9.29e-009
Aspect ratio change(0.80, 1.00)	17/94	2.19e-025	-	-	-	-
Aspect ratio change(0.90, 1.00)	31/97	1.10e-053	-	-	-	-
Aspect ratio change(1.00, 0.80)	9/89	1.01e-011	-	-	-	-
Aspect ratio change(1.00, 0.90)	27/100	1.18e-044	-	-	-	-
Aspect ratio change(1.00, 1.20)	18/128	1.07e-024	-	-	-	-
Aspect ratio change(1.00 1.10)	30/130	4.02e-047	-	-	-	-
Aspect ratio change(1.10, 1.00)	38/110	5.43e-067	-	-	-	-
Aspect ratio change(1.20, 1.00)	20/138	1.51e-027	-	-	-	-
Rotation 1.00	33/106	1.33e-056	-	-	2/4	6.91e-005
Rotation 2.00	20/101	1.85e-030	-	-	1/4	1.35e-002
Rotation 5.00	13/90	2.11e-018	-	-	0/4	1.00e-000
Rotation 10.00	12/74	9.82e-018	-	-	-	-
Rotation 15.00	13/61	9.16e-021	-	-	-	-
Rotation 30.00	12/55	2.07e-019	-	-	-	-
Rotation 45.00	5/46	3.01e-007	1	1.91e-005	-	-
Rotation 90.00	25/111	3.10e-039	-	-	-	-
Flipping	25/109	1.87e-039	-	-	-	-

TABLE VII

-Continued.

attack	proposed method		[22]		[23]	
	D_M/T_M	p_{fp}	D_M	p_{fp}	D_M/T_M	p_{fp}
Rotation Scale 1.00	29/106	4.90e-048	-	-	2/4	6.91e-005
Rotation Scale 10.00	7/102	3.15e-008	-	-	-	-
Rotation Scale 15.00	4/84	1.29e-004	-	-	-	-
Rotation Scale 30.00	4/85	1.35e-004	-	-	-	-
Rotation Scale 45.00	3/91	2.69e-003	-	-	-	-
Rotation Scale 90.00	25/111	3.10e-039	-	-	-	-
Scaling 50%	13/101	1.02e-017	2	1.82e-010	-	-
Scaling 75%	4/66	5.03e-005	6	5.93e-032	-	-
Scaling 90%	22/94	4.03e-035	6	5.93e-032	-	-
Scaling 110%	22/136	2.89e-031	-	-	-	-
Scaling 150%	5/65	1.73e-006	-	-	-	-
Scaling 200%	48/105	1.45e-091	-	-	-	-
Shearing x-0% y-1%	43/110	1.94e-078	-	-	-	-
Shearing x-1% y-0%	37/110	9.39e-065	4	5.30e-021	-	-
Shearing x-1% y-1%	32/111	1.10e-053	-	-	1/4	1.35e-002
Shearing x-0% y-5%	30/95	8.05e-052	-	-	1/4	1.35e-002
Shearing x-5% y-0%	30/98	2.42e-051	-	-	-	-
Shearing x-5% y-5%	16/94	1.58e-023	0	1.00e-000	0/4	1.00e-000
Random Bending	26/109	1.81e-041	3	1.14e-015	-	-

TABLE VIII

COLLUSION ATTACK ON CDW EMBEDDING.

image	CDW stego image		PSNR	colluded image		PSNR
	D_M/T_M	p_{fp}	(dB)	D_M/T_M	p_{fp}	(dB)
Baboon	7/111	5.65e-008	36.06	5/107	2.00e-005	33.07
Lena	31/108	5.48e-052	38.44	17/97	3.89e-025	35.35
Pepper	57/109	5.95e-113	38.32	29/109	1.24e-047	35.21

TABLE IX

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	20/103	2.84e-030	36.06	0/99	1.00e+000	34.64
Lena	56/105	1.13e-111	38.43	7/111	5.65e-008	38.17
Pepper	65/119	2.57e-130	38.31	12/109	1.27e-015	38.42

TABLE X

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	7/111	5.65e-008	36.06	3/105	4.03e-003	36.02
Lena	31/108	5.48e-052	38.44	1/103	2.66e-001	38.39
Pepper	57/109	5.95e-113	38.32	1/115	2.92e-001	38.27

TABLE XI

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	20/103	2.84e-030	36.06	24/105	6.72e-038	36.02
Lena	56/105	1.13e-111	38.43	53/99	6.71e-106	38.38
Pepper	65/119	2.57e-130	38.31	59/105	1.75e-119	38.27

TABLE XII

IMPACT OF FEATURE POINT EXTRACTION AND DENOISING-BASED BLIND DETECTION ON THE PERFORMANCE OUR WATERMARKING METHOD.

image	Condition (i) D_M/T_M	p_{fp}	Condition (ii) D_M/T_M	p_{fp}	Average displacement (in pixels) of feature points
Baboon	67/103	6.08e-142	7/113	6.40e-008	4.13
Lena	88/100	9.83e-208	32/106	1.97e-054	2.59
Pepper	95/107	5.07e-225	55/109	7.33e-108	1.60

content-dependent image watermarking method that can resist extensive geometric attacks and watermark estimation attacks simultaneously has been proposed here. There are three major contributions of our method. First, robust mesh extraction is adopted to enhance the feasibility of feature-based watermarking methods. Second, a media hash-based content-dependent watermark that is composed of a watermark and a hash is used to resist watermarking-estimation attack. Third, a false positive-oriented watermark detection mechanism is applied to determine the existence of a watermark so as to achieve a trade-off between correct detection and false detection. The performance of our scheme in enhancing robustness has been thoroughly verified using the standard benchmark, Stirmark, and watermark estimation attacks.

However, the major weakness of our method is its high complexity since most of the time is spent on mesh warping, which makes the method in its current state unsuitable for real-time applications. By keeping the achievable robustness, reducing the complexity of our method deserves further researching. In addition, as described in Sec. V-D, enhancing the stability of feature point extraction can further improve the overall performance of the proposed method. Finally, the important issue of security against protocol attacks based on the proposed method was also investigated. Due to limits of space, the results were reported elsewhere [14].

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

REFERENCES

[1] M. Alghoniemy and A. H. Tewfik, “Image Watermarking by Moment Invariants,” *Proc. IEEE Int. Conf. Image Processing*, Vancouver, Canada, Vol. II, pp. 73-76, 2000.

[2] 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.

[3] S. Craver, N. Memon, BL Yeo, and M. M. Yeung, “Resolving rightful ownerships with invisible watermarking techniques: Limitations, attacks, and implications,” *IEEE Journal on Selected Areas in Communications*, Vol. 16, No. 4, pp. 573–586, 1998.

[4] C. Harris and M. Stephen, “A combined corner and edge detector,” *Proc. 4th Alvey Vision Conf.*, pp.147-151, 1988.

[5] F. Hartung and B. Girod, “Watermarking of Uncompressed and Compressed Video,” *Signal Processing*, Vol. 66, No. 3, pp. 283-302, 1998.

[6] 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.

[7] 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.

[8] C. Y. Hsu and C. S. Lu, “A Geometric-Resilient Image Hashing System and Its Application Scalability,” *Proc. ACM Multimedia and Security Workshop*, pp. 81-92, Magdeburg, Germany, 2004. (the extended journal version is under review)

[9] M. Kutter, “Watermarking resisting to translation, rotation and scaling,” *Proc. SPIE International Symposium on Voice, Video, and Data Communication*, Boston, November 1998.

[10] 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.

[11] 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. (the extended journal version is under review)

[12] C. S. Lu, S. W. Sun, and P. C. Chang, “Robust Mesh-based Content-dependent Image Watermarking with Resistance to Both Geometric Attack and Watermark-Estimation Attack,” *Proc. SPIE: Security, Steganography, and Watermarking of Multimedia Contents VII (EI120)*, San Jose, California, USA, 2005.

[13] C. S. Lu, “Towards Robust Image Watermarking: Combining Content-Dependent Watermark, Moment Normalization, and Side-Informed Embedding,” *Signal Processing: Image Communication*, Vol. 20, No. 2, pp. 129-150, 2005.

[14] C. S. Lu, C. M. Yiu, S. W. Sun, and P. C. Chang “On the Security of Mesh-based Media Hash-dependent Watermarking Against Protocol Attacks,” submitted to *IEEE Int. Conf. on Multimedia and Expo*, 2005.

[15] A. R. Manuel and P. G. Fernando, “Analysis of Pilot-based Synchronization Algorithms for Watermarking of Still Images,” *Signal Processing: Image Communication*, Vol. 17, pp. 611-633, 2002.

[16] 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.

[17] 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.

[18] S. Pereira, T. Pun, “An iterative template matching algorithm using the Chirp-Z transform for digital image watermarking,” *Pattern Recognition (33)*, pp. 173-175, 2000.

[19] 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.

[20] F. Petitcolas, “Watermarking Schemes Evaluation,” *IEEE Signal Processing Magazine*, Vol. 17, No. 5, pp. 58-64, 2000.

[21] M. Ramkumar and A. N. Akansu, “A Robust Scheme for Oblivious Detection of Watermarks/Data Hiding in Still Images,” *Proc. SPIE Multimedia Systems and Applications*, Vol. 3528, pp. 474-481, 1998.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.