A Wavelet-based Watermarking Algorithm for Ownership Verification of Digital Images

Yiwei Wang*, *Student Member, IEEE*, John F. Doherty⊤§, *Senior Member, IEEE*, and Robert E. Van Dyck†, *Member, IEEE*

* Chrontel Inc. San Jose, CA 95132

E-mail: ywang@chrontel.com

⊤ Department of Electrical Engineering

The Pennsylvania State University

University Park, PA 16802

E-mail: jfdoherty@psu.edu

† National Institute of Standards and Technology, Gaithersburg, MD 20899

E-mail: vandyck@antd.nist.gov

§ Person of Contact: Phone: 814-863-8102, fax: 814-863-5341

EDICS: 5-AUTH

Abstract

In recent years, access to multimedia data has become much easier due to the rapid growth of the Internet. While this is usually considered an improvement of everyday life, it also makes unauthorized copying and distributing of multimedia data much easier, therefore presenting a challenge in the field of copyright protection. Digital watermarking, which is inserting copyright information into the data, has been proposed to solve the problem. In this paper, we first discuss the features that a practical digital watermarking system for ownership verification requires. Besides perceptual invisibility and robustness, we claim that the private control of the watermark is also very important. Second, we present a novel wavelet-based watermarking algorithm. Experimental results and analyses are then given to demonstrate that the proposed algorithm is effective and can be used in a practical system.

Keywords

Digital watermark, copyright protection, ownership verification, wavelet, filter banks.

This paper is partially sponsored by AFRL under contract number F30602-98-0061
Parts of this work were presented at CISS 2000
This work is done when the authors were with the Pennsylvania State University

I. Introduction

The rapid growth of the Internet increases access to multimedia data tremendously. This fact, combined with more powerful image processing software and faster personal computers, presents a challenge to copyright protection of multimedia data as unauthorized copying and distributing of digital images, video, etc. also become easier. During recent years, digital watermarking has drawn a lot of attention as a solution of this problem [1]-[14]. In general, a digital watermarking algorithm tries to adhere some copyright information to the original data. Although watermarks can be visible, invisible watermarks are usually preferred. Hence, this paper, like many others, will focus on invisible watermarking schemes. Another requirement for digital watermarking algorithms is robustness. In other words, the watermark should survive the common signal processing operations and counterfeit attempts.

We shall focus on digital image watermarking in this paper. Quite a few digital image watermarking schemes have been proposed in the 1990s [1]-[6]. Although the early algorithms usually require the subtraction of the original image from the test image to detect the watermark [1]-[4], recent work showed that for ownership verification, the above subtraction would create severe problems [5]. Basically, an "original" image can either be *the* original image or obtained by subtracting the counterfeiter's watermark from *the* original image, and there is no way to distinguish one from the other. Therefore, a watermarking algorithm for ownership verification should avoid performing such subtraction in the detection process.

Most digital watermarking papers in the literature are proposing new algorithms. Some of these algorithms, although very clever, cannot be used in a practical system. Therefore, before proposing our algorithm, we want to discuss the requirements of a practical image watermarking system for ownership verification of digital images. The rest of this paper is organized as follows: Section II discusses the necessary features that a digital watermarking system for ownership verification needs in the real world. Section III reviews some necessary background on wavelets and filter banks for our algorithm. We present our algorithm in section IV. Experimental results and analyses are given in section V to demonstrate the performance of the proposed algorithm. Conclusions and comments are given in section VI.

II. Requirements for Practical Watermarking Systems

In a practical digital image watermarking system for ownership verification, there should be at least two parties: the owner of the images and a legal authority. We want to start this section by discussing the division of responsibilities between these two parties.

The first question is what each of the two parties should store. We begin with the storage requirement for the legal authority. The need to register the watermark is obvious. Otherwise, if one can claim anything to be his watermark, he can claim the ownership of any image. However, we do not think that it is practical to require the registration of each image. A system only makes sense when it serves a fairly large number of owners. Each of these owners possesses many images and keeps producing new images everyday. So, the total number of images will be huge, and the storage of all the images is too expensive. Besides the storage problem, when a new image comes in, the authority also needs to check all the registered images to make sure that it has not been registered. This is expensive as well. Therefore, even if the registration of each image is possible, it will increase the cost of the registration. Some of the owners may give up on the registration process and some others will only register some of their images because of the expense. In general, a copyright protection system works better when everybody complies because there is less ambiguity. On the other hand, if only the watermark requires registration, an owner can register one or several watermarks at the beginning and only pay a very small annual fee. The burden on the authority also decreases. The storage requirement is much lower, and the authority only needs to make sure the watermark of a new user does not coincide with the old ones.

Then, what should the owner store? It seems that the owner only needs to store the images. However, we do not think that it is so simple. If we model the watermarking insertion procedure by equation $X' = X + W$, where X' is the watermarked image, X is the original image, and W is the added watermark, we claim that W should not be the same in each image. Otherwise, one can easily show that a counterfeiter can estimate the owner's watermark if he adds a number of the watermarked images from the owner. Although the number of the images required is very large, there will always be a distant danger. When a counterfeiter knows W, he can easily remove the watermark from all the images. Another problem for inserting the same watermark in every image is that it is obvious when the system is created, the database at the legal authority will be a target of hackers. In the case of a security breach, a counterfeiter will know the watermark

of the owner. We shall argue later in this section that the watermark algorithm might need to be standardized. In that case, a counterfeiter will know both the watermark and the algorithm, hence he can remove the watermark. To avoid these problems, the owner should insert different formats of the watermark into images. In other words, he will have private keys. If these keys do not require registration at the authority, even if the database in the legal authority been hacked into, no one can remove the watermarks. Basically, even the authority itself cannot remove the watermarks without the private keys. Furthermore, the owner may want to create a set of private keys for each image. Most digital watermarking papers in the literature assume the counterfeiter does not know the watermark and/or the algorithm. However, we know that a lot of crimes are committed with inside help. In our case, if the owner creates a set of keys for each image, then even if a counterfeiter knows the watermark, the algorithm, and some sets of keys, the damage is limited to the corresponding images instead of the whole image database of the owner. This may be extremely desirable when the owner is a large company. So, the effect of one misbehaved operator will be limited. If such keys exist, it is quite natural to require they have a reasonable length to limit the storage overhead.

The second question is who should perform the watermark detection procedure. There is no doubt that the authority should perform the detection procedure and provide testimony when a copyright dispute is taken to the court. However, it is not realistic to ask the authority to detect all the copyright violations in the first place, because it would require that the legal authority process all images and try to detect the watermarks of all owners in an image. This is very resource-consuming and will increase the operation cost dramatically. Particularly, if the system provides owners private control of the watermark, this may become impossible. A more realistic way is that the owner should first try to find any counterfeiters of his images. Because the owner is more familiar with his images, it is usually easy for him to recognize the images even if they have been manipulated. In addition, the number of images and the number of watermarks of one owner are much smaller than those of all owners. When the owner is suspicious of an image, he will perform the watermark detection procedure on that image. If he detects his watermark in the image, he will take the case to the court. The court will ask the authority to verify the owner's claim, and the owner will provide the private key for the particular image to enable the authority to perform the detection procedure. The cost of such verification can be easily added to the punishment of the counterfeiter if the court decides that the copyright violation really

takes place. Therefore, the operating cost of the system will stay low.

Now, we discuss what kind of algorithm the system should use. As mentioned above, we think that the algorithm(s) used by the system should be standardized. Most of the algorithms proposed in the literature use some sort of pattern recognition algorithm in the detection procedure. For any algorithm, there will be a probability of false alarm. In other words, given an arbitrary unwatermarked image, there is a chance that the detection procedure will claim that there is a watermark in it. To make a watermarking algorithm work, such false alarm probability should be very small. If the owner can use any algorithm, when a case is taken to the court, a lot of resources will be spent to verify if the false alarm probability of the owner's algorithm is small enough. To avoid doing this on a case by case basis, the algorithm(s) should be standardized. The probability of false alarm is known and consented to by all parties.

Then, what should a standardized algorithm satisfy? Besides the common requirements of invisible watermarks and robustness, we already showed that the algorithm should not depend on the subtraction of the original image in the detection procedure. We also showed that the algorithm should have the ability to create a private key for each image and insert the watermark according to the key. However, this also adds another requirement for the algorithm. Given an arbitrary unwatermarked image, the owner must not be able to create a key so that he can detect his watermark in the image. In other words, there should not exist a reverse engineering algorithm to create the key based on the image.

In summary, we list all the requirements below:

- The legal authority
  - stores the watermarks
  - does not store all the images
  - performs the final watermark detection
  - does not search for copyright violations in the first place
  - cannot remove the watermark from the image
- The owner
  - stores images and private keys
  - searches for counterfeiters of his images
- The Algorithm
  - should be standardized

- achieves watermark invisibility
- achieves robustness
- does not require subtraction of the original image in watermark detection
- gives owner private control of the watermark
- prevents reverse engineering of the private keys

III. Background on Wavelets and Filter Banks

Before we introduce our algorithm, we review some necessary background on wavelets and filter banks.

The study of the wavelet transform has thrived in the past two decades. Now, the wavelet transform is considered to be a fairly simple mathematical tool, and it has many applications in various fields. The continuous and discrete wavelet transforms are given in equations (1) and (2), respectively [15].

$$(T^{wav}f)(a,b) = |a|^{-1/2} \int f(t)\psi(\frac{t-b}{a})dt. \tag{1}$$

$$T^{wav}_{m,n}(f) = a_0^{-m/2} \int f(t)\psi(a_0^{-m}t - nb_0)dt. \tag{2}$$

The wavelet transform also finds its way into the field of signal analysis. Compared with the traditional transforms, the Fourier Transform for instance, the wavelet transform has an advantage of achieving both spatial and frequency localization. In digital signal and image processing, the discrete wavelet is closely related to filter banks. A typical 2-channel decomposition and reconstruction structure is given in Fig. 1.

It is well-known that the filter banks will provide perfect reconstruction (i.e., $x = \widehat{x}$ in Fig. 1) if they satisfy equations (3) and (4) [16].

$$H_0(z)G_0(z) + H_1(z)G_1(z) = 2; \tag{3}$$

$$H_0(-z)G_0(z) + H_1(-z)G_1(z) = 0. \tag{4}$$

A very important class of filter banks are orthonormal filter banks. For two-channel, orthonormal, FIR, real-coefficient filter banks (T.O.F.R.FB.), equations (3) and (4) are equivalent to equations (5), (6) and (7)[16].

$$G_0(z)G_0(z^{-1}) + G_0(-z)G_0(-z^{-1}) = 2; \tag{5}$$

$$G_1(z) = -z^{-2k+1}G_0(-z^{-1}), k \in \mathbf{Z}; \tag{6}$$

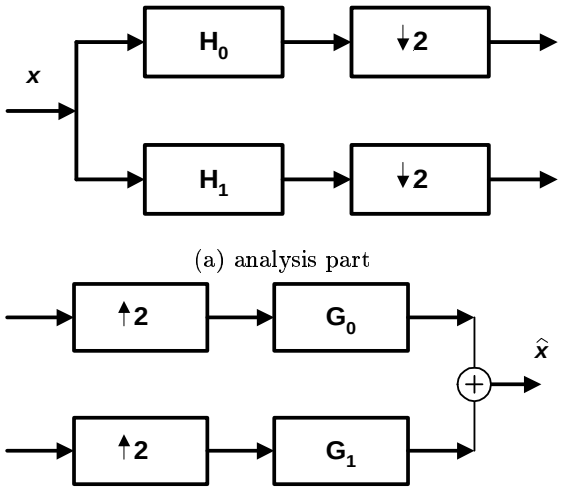


Fig. 1. 2-channel decomposition and reconstruction structure

$$H_i(z) = G_i(z^{-1}), i \in \{0, 1\}. \tag{7}$$

Furthermore, it is easy to show that if we define $P(z) = G_0(z)G_0(z^{-1})$, then

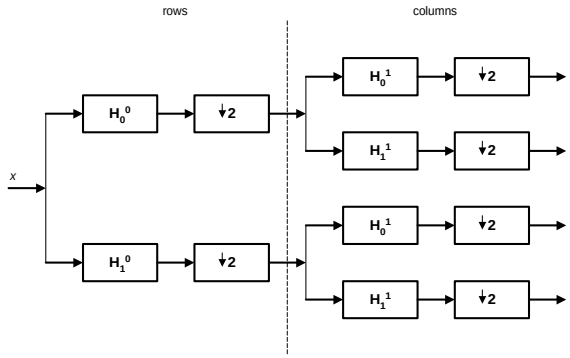
$$P(z) = 1 + \sum_{k \ odd} a_k z^{-k}, a_k = a_{-k} \text{ [16]}. \tag{8}$$

To extend the above analysis to images, we can consider digital images as 2-D signals and apply the 1-D wavelet transform to the horizontal and vertical directions separately. The structure is given in Fig. 2 [17]. in Fig. 1 or Fig. 2 depending on the dimension of the signal.

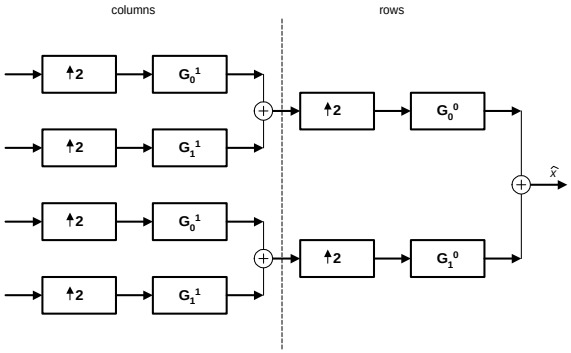
IV. Watermarking Algorithm

In this section, we present our algorithm for watermarking digital images. As the authors in [18] stated, the human visual system has different sensitivities for different frequencies. The low frequency noise is usually more noticeable [3]. Another known fact is that lossy compression schemes often eliminate high frequency components. Therefore, our algorithm, like many existing algorithms, inserts the watermark into the middle-frequency range to achieve both perceptual invisibility and robustness to compression.

In Section II, we claim that the owner should have private control over the watermark. In our algorithm, we use randomly generated orthonormal filter banks as a major part of the private key. In equation (8), we showed a requirement for orthonormal filter banks. On the other hand,



(a) analysis part



(b) synthesis part

Fig. 2. Image decomposition and reconstruction structure

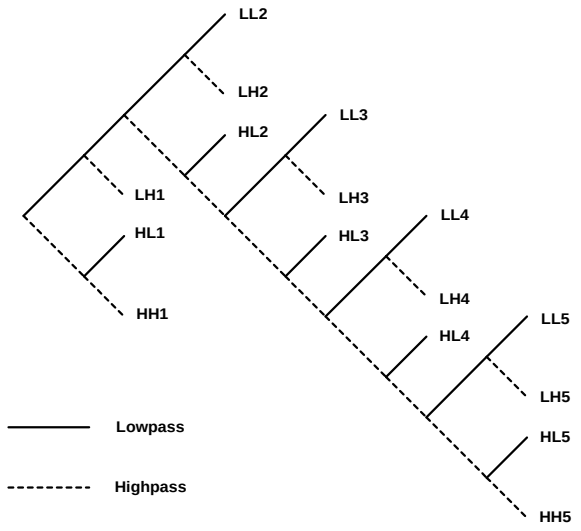
suppose we have a polynomial $P'(z)$ that satisfies:

$$P'(z) = 1 + \sum_{k \ odd} a'_k z^{-k}, a'_k = a'_{-k}; \tag{9}$$

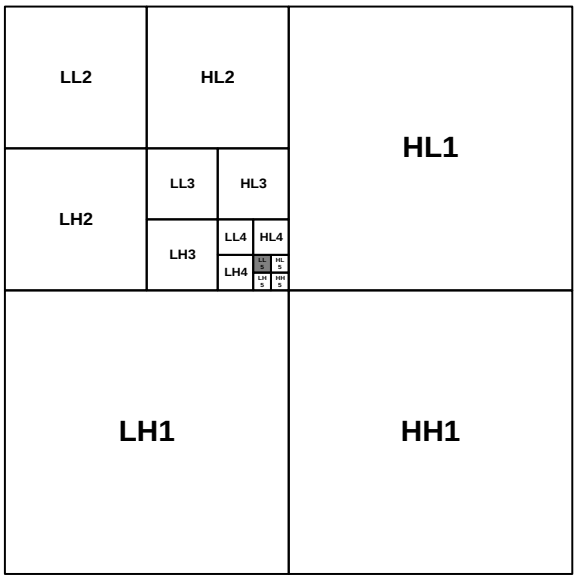
where $\sum a'_k = 1$, $a'_k \geq 0$; because $P'(z) \geq 0, \forall z \in \{z : |z| = 1\}$, $P'(z)$ is spectral factorizable, i.e., $\exists\ G'_0(z)$, s.t. $P'(z) = G'_0(z)G'_0(z^{-1})$ and $G'_0(z)$ is the lowpass synthesis filter of a T.O.F.R.FB. Therefore, we can randomly generate orthonormal filter banks by randomly generating $P'(z)$'s. In addition, we can control the behavior of the filter bank by adding certain constraints to the a'_k's.

Besides generating random filter banks, by choosing which middle-frequency band that the watermark will be embedded into, the owner can have more private control over the watermark. The stuctures in Fig. 2 can be cascaded to create a wavelet pyramid. Several middle-frequency bands will be valid candidates. Fig. 3 shows a possible decomposition structure and a middle-frequency band to embed the watermark.

The watermarks used in our algorithm are binary images with values ± 1. These binary images



(a) decomposition structure



(b) wavelet pyramid (shaded LL5 chosen to embed watermark)

Fig. 3. An example of eligible middle-frequency band to embed the watermark (note: the LL, LH, HL, and HH notations are different from those in standard compression papers)

do not require much storage space and can be designed as the owner's seal. However, to make the watermark hard for counterfeiters to detect, the watermark that we want to embed should be an image of real numbers because the wavelet coefficients are real numbers. Therefore, we develop the following method to transfer a binary image into a real-numbered image. First, we scan the binary image row by row. If the size of the image is n by n (n is always even in our algorithm), we will have a vector of n^2 by 1. We define it as $\overrightarrow{v_0}$ and compute

$$\overrightarrow{v_1} = R_1 R_2 \overrightarrow{v_0}; \tag{10}$$

where

$$R_1 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & & & \\ \sin\theta_1 & \cos\theta_1 & & & \\ & & \ddots & & \\ & & & \cos\theta_{n^2-1} & -\sin\theta_{n^2-1} \\ & & & \sin\theta_{n^2-1} & \cos\theta_{n^2-1} \end{bmatrix};$$

and

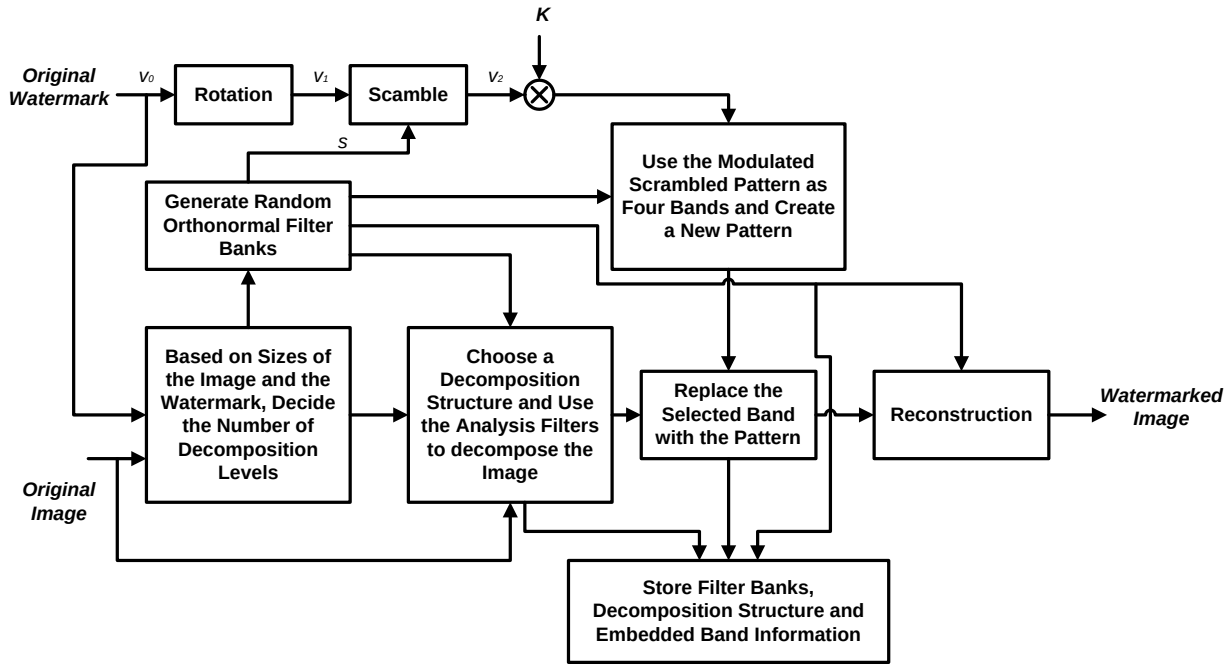
$$R_2 = \begin{bmatrix} 1 & & & & & \\ & \cos\theta_2 & -\sin\theta_2 & & & \\ & \sin\theta_2 & \cos\theta_2 & & & \\ & & & \ddots & & \\ & & & \cos\theta_{n^2-2} & -\sin\theta_{n^2-2} & \\ & & & \sin\theta_{n^2-2} & \cos\theta_{n^2-2} & \\ & & & & & 1 \end{bmatrix}.$$

By rescanning $\overrightarrow{v_1}$ back to a matrix, we will have a real-numbered images. It is obvious that $R_1 R_2$ is equivalent to a series of rotations. So, the energy of the new image is the same as the original binary image. The legal authority can generate a set of θ_i, $i \in \{1, 2, \ldots, n^2 - 1\}$ (note: all θ's in R_1 are odd indexed and θ's in R_2 are even indexed), and every registered watermark can be transformed into a real-numbered image using the set of θ's. We define it as w_t. Another obvious fact is that if two original watermarks are different, their transformed images will be different as well.

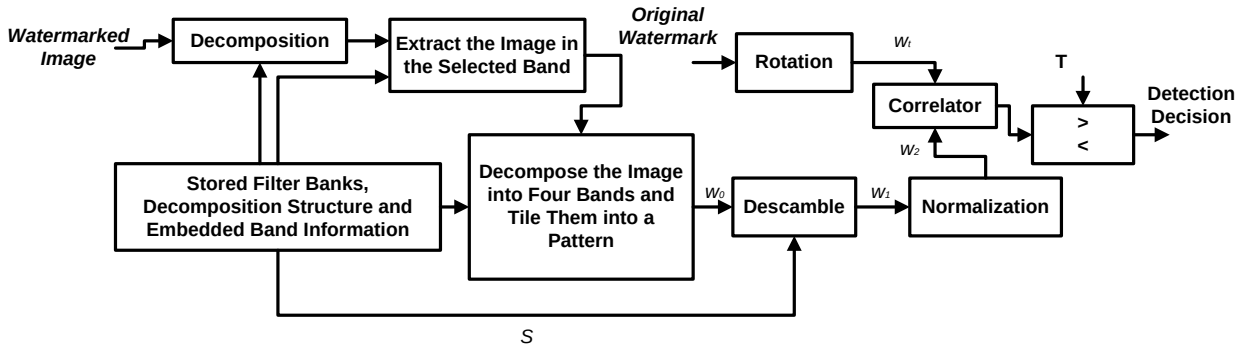
Although the above procedure transforms the original binary watermarks into real-numbered watermarks, the watermark that will be embedded by the same owner is still the same for every image. Here is an algorithm to scramble the watermark to make it different from image to image.

First, we convert the image back to vector $\overrightarrow{v_1}$. Then, given an arbitrary vector $\overrightarrow{s}$ whose elements are non-negative integers, we can create vector $\overrightarrow{v_2}$ using $\overrightarrow{s}$ by doing the following. Starting from the first element in $\overrightarrow{v_1}$, we count $s(1)$ elements and assign the resulting element as the first element of $\overrightarrow{v_2}$, i.e., $v_2(1) = v_1(s(1) + 1)$. Then, we remove the element that we just processed from $\overrightarrow{v_1}$ and count $s(2)$ elements from that point, then assign this element as the second element of $\overrightarrow{v_2}$. If we consider $\overrightarrow{v_1}$ and $\overrightarrow{s}$ to be circular vectors and repeat the above procedure, then after n^2 steps, $\overrightarrow{v_2}$ will be a rearranged vector of $\overrightarrow{v_1}$. The rearranging process is reversible assuming we always have access to $\overrightarrow{s}$. Again, we can easily convert $\overrightarrow{v_2}$ into an n by n image.

In this paragraph, we present the detailed steps of our watermark embedding algorithm. First, based on the size of the image and the size of the watermark pattern, the number of decomposition levels is determined. Second, if the number of levels determined is L, 2L+2 sets of T.O.F.R.FB.'s are randomly generated. These filter banks are very important and have to be saved for the watermark embedding and detection processes. The wavelet decomposition pyramid and the middle-frequency band to insert the watermark are also chosen. Then, the analysis filters are used to decompose the image. For each 1-D decomposition, a different pair of analysis filters are used. Hence, 2L pairs of filters are used for L levels. The middle-frequency band that we chose should be one of the four bands of the highest level, and the next step is to replace the coefficients in that band by an image achieved using the following method. The first digits after the decimal point of the first coefficient of each analysis filter are put together to create a vector $\overrightarrow{s}$ of length 2L+2, then using the methods described in the previous paragraphs, we can get a scrambled real-valued version of the owner's watermark. This watermark is multiplied by a proper factor to adjust the energy of the embedded watermark, and then the magnified watermark is simply divided into four square parts equally. Using these parts as coefficients of LL, LH, HL, and HH bands and the synthesis filters of the two sets of filter banks that have not been used, we can create another image of the size of n by n using 1-level wavelet reconstruction structure for images. The common belief is that a watermark is hard to attack if the counterfeiter cannot find where it is. We use this reconstructed image to replace the coefficients in the pre-selected band. Since the above procedure decreases the differences between the embedded watermark image and the coefficients in the pre-selected band, this will make it hard for the counterfeiter to decide which frequency band is used to embed the watermark as demonstrated later in Section V. The corresponding synthesis filters are then used to reconstruct the watermarked image. In addition,



(a) embedding process



(b) detection process

Fig. 4. Block diagram

we need to round off the pixel values of the watermarked image to make sure that it has the same gray scale range as the original image. The block diagram of the embedding process is given in Fig. 4.

The detection process is basically the inversion of the embedding process. Using the stored information of the filter banks, decomposition structure and middle-frequency band, we can obtain the coefficients of that band of the suspect image. Then using the corresponding analysis filters, we decompose the image in that band into four bands and tile them together to create

a new image w_0. Furthermore, we recreate the scramble vector $\vec{s}$ and descramble w_0 into w_1. Then, we normalize the energy of $w1$ to be the same as our watermark and define the normalized image to be w_2. Next, we compute the correlation between w_2 and w_t. If it is greater than a preset threshold, we claim that the suspect image contains the watermark. The block diagram of the detection process is also given in Fig. 4.

The watermark detection process in our algorithm, like in many existing algorithms, is similar to determining the existence or absence of a signal in a noisy environment; this fact implies we have to analyze the probability of false alarm. Assuming the watermark size is n by n, because we normalize the energy before we compute the correlation, all possible patterns are lying on a sphere of dimension n^2 with radius one. We define $m = n^2$. The surface area of a m-dimensional sphere of radius ρ is

$$S = mV_m\rho^{m-1}, \tag{11}$$

where $V_m = \frac{\pi^{m/2}}{(m/2)!}$ [19]. We assume that all possible patterns are of equal probability. In other words, they are uniformly distributed on the sphere. Then, the false alarm probability P_f, which is the probability that an arbitrary pattern has a correlation with the watermark pattern larger than the threshold T, equals to the fraction of two areas $\frac{A_1}{A}$. A is the area of the whole sphere, while A_1 contains all points on the sphere whose inner product with the point corresponding to the rotated watermark pattern w_t is larger than T. Without loss of generality, we can rotate the coordinate axes to make the rotated watermark pattern correspond to the point $[1, 0, 0, \ldots, 0]^T$. Then using equation (11), it is easy to show that

$$A_1 = \int_T^1 (m-1)V_{m-1}(\sqrt{1-x^2})^{m-2}\frac{dx}{\sqrt{1-x^2}} \tag{12}$$

and $A = mV_m$. Therefore, P_f can be calculated by equation (13).

$$P_f = \frac{\int_T^1 (m-1)V_{m-1}(\sqrt{1-x^2})^{m-3}dx}{mV_m} \tag{13}$$

The threshold T should be chosen to give a relatively small false alarm probability. For instance, if the watermark is 16 by 16, we list the false alarm probability for various thresholds in Table I. We can see that any reasonable threshold should be greater than 0.35.

We claimed in Section II that a practical algorithm should embed different versions of the owner's watermark for different images. In our algorithm, because we create different filter banks for each image, create a scramble vector based on the filter banks, and might use different

TABLE I

THE FALSE ALARM PROBABILITY FOR DIFFERENT THRESHOLDS

Threshold(T)	False alarm probability(P_f)
0	0.5
0.1	5.487×10^{-2}
0.2	6.337×10^{-4}
0.3	4.811×10^{-7}
0.35	4.035×10^{-9}
0.36	1.394×10^{-9}
0.37	4.642×10^{-10}
0.38	1.488×10^{-10}
0.39	4.583×10^{-11}
0.4	1.356×10^{-11}
0.41	3.850×10^{-12}
0.42	1.048×10^{-12}
0.43	2.729×10^{-13}
0.44	6.795×10^{-14}
0.45	1.615×10^{-14}
0.46	3.661×10^{-15}
0.47	7.897×10^{-16}
0.48	1.619×10^{-16}
0.49	3.150×10^{-17}
0.5	5.803×10^{-18}
0.6	8.018×10^{-27}
0.7	1.843×10^{-39}
0.8	8.351×10^{-59}
0.9	3.046×10^{-94}
1	0

decomposition structures and middle-frequency bands, the watermarks embedded are indeed different from image to image. In addition, given an arbitrary image, there does not exist a practical algorithm to generate a series of filter banks and use them to decompose the image so that one band will have high correlation with a given pattern. The dependence of the scramble vector on the filter banks adds more difficulty. Hence, the reverse engineering of the private keys is essentially impossible.

Another thing that we need to address is the storage requirement. The legal authority only needs to store the binary watermarks because the rotation matrices are always the same. If the watermark size is n by n, each watermark only requires n^2 bits. For instance, a 16 by 16 watermark only requires 32 bytes. Therefore, the storage requirement for the legal authority is reasonable. The owner needs to store the filter banks for each image. We have shown that the filter banks can be generated from $P'(z)$'s and the coefficients of the $P'(z)$'s are symmetric. Hence, if the length of the filters is 2K (the length of any orthonormal filter is even), only K floating point numbers need to be stored for each set of filters. For example, if the decomposition level L is 5 and the filter length is 6, $2L + 2 = 12$ sets of filter banks are needed. Therefore, the overhead storage for these filter banks is storing $12 \times 6/2 = 36$ floating point numbers. The storage requirement for band information is 2 bits per level and because the scramble vector depends on the filter banks, it does not require any extra storage. Therefore, the total storage overhead for each image is low as well.

The last topic that we want to address in this section is what kind of filters we should choose. To make the algorithm more robust to compression and counterfeit attempts, we want the filters to have large sidelobes to put some of the watermark energy into lower frequencies. For instance, if we pick the filter length to be six, then $P'(z) = 1 + a'_1(z + z^{-1}) + a'_3(z^3 + z^{-3}) + a'_5(z^5 + z^{-5})$. If we force $a'_1 \leq 0.2$, then the filters will have large sidelobes. For example, if $P'(z) = 1 + 0.1869(z + z^{-1}) + 0.1201(z^3 + z^{-3}) + 0.1930(z^5 + z^{-5})$, then the lowpass analysis filter is $H_0(z) = 0.2149 - 0.0731z^{-1} + 0.1868z^{-2} - 0.1178z^{-3} + 0.3054z^{-4} + 0.8980z^{-5}$ and the highpass analysis filter is $H_1(z) = -0.8980 + 0.3054z^{-1} + 0.1178z^{-2} + 0.1868z^{-3} + 0.0731z^{-4} + 0.2149z^{-5}$. As shown in Fig. 5, they will have large sidelobes.

V. Experimental Results

In this section, we demonstrate the performance of our algorithm. We use test images of size 512 by 512 and choose the number of decomposition levels to be five. The band to embed the

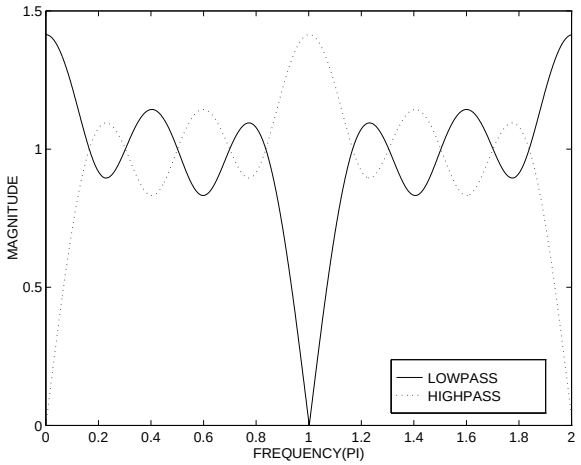


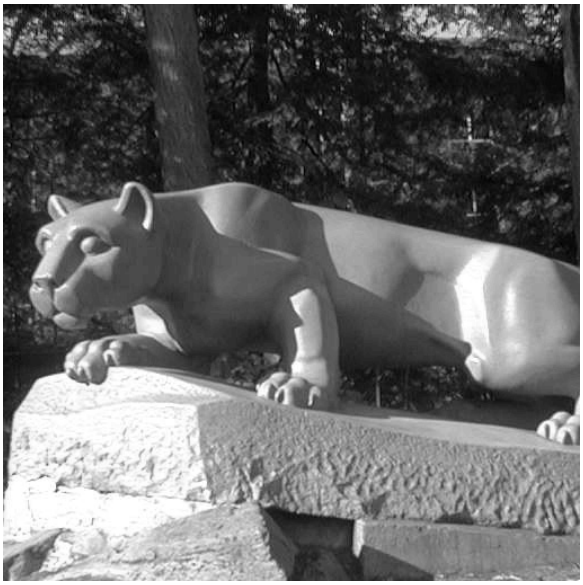
Fig. 5. An example of filters

watermark is the LL5 band shown in Fig. 3. The filter length is six, and we force $a_1' \leq 0.2$ for all $P'(z)$'s. Our binary watermark is a 16 by 16 "PSU" pattern as shown in Fig. 6. We create our rotation matrices by generating $\theta_i, i \in \{1, 2, \ldots, 255\}$ that are uniformly distributed over the interval $[0, 2\pi)$. The rotated watermark is also shown in Fig. 6. The multiplying factor, denoted by K in Fig. 4, is decided by the empirical equation $K = (M_c/M_w + \sqrt{E_c/E_w})/2$, where M_c is the largest value of the magnitude of the wavelet coefficients in the selected band, M_w is the largest value of the magnitude of the rotated watermark, E_c is the total energy of the coefficients in the selected band, and E_w is the energy of the watermark. We choose the threshold to be 0.4, which produces a false alarm probability on the order of 10^{-11}. Because the gray levels of the original images are from 0 to 255, we truncate and round up the gray levels of the watermarked images into the same range. We show the original "lion" image and its watermarked version in Fig. 7. Because of the nature of this paper, we feel obligated to avoid reprinting other's images. However, we list the numerical results of some standard images with our image in Table II. We use different sets of filter banks to embed the watermark into different images. Because of the truncation and the rounding up, the correlations obtained by the detection process will not be 1, but close to 1 instead. From Table II, one can see our algorithm achieves perceptual invisibility.

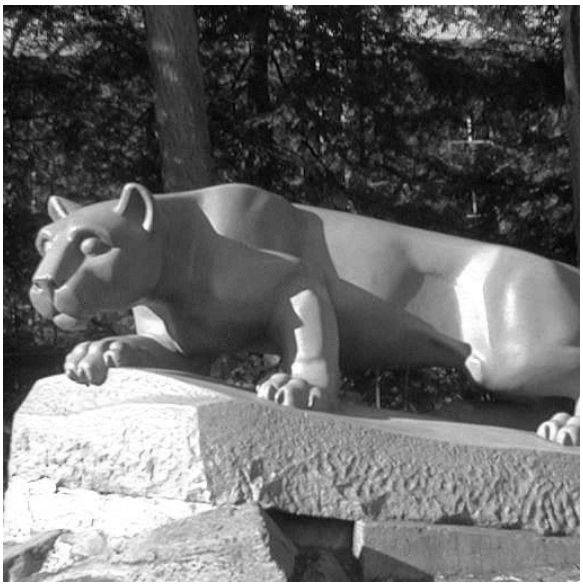
Another point we want to address is that the embedded band should be hard for the counterfeiters to find. Since the counterfeiters do not know our filter banks, the best thing they can do is to generate their own filter banks using the same procedure and then decompose the wa-

(a) original watermark (b) rotated watermark

Fig. 6. Watermarks



(a) original image



(b) watermarked image

Fig. 7. Original and watermarked images

TABLE II

THE PSNR AND CORRELATION OF WATERMARKED IMAGES

Image	PSNR(dB)	Correlation
Lion	42.8	0.970
Lena	42.5	0.982
Barbara	42.2	0.987
Baboon	41.9	0.988
Goldhill	42.3	0.989
Peppers	41.8	0.986

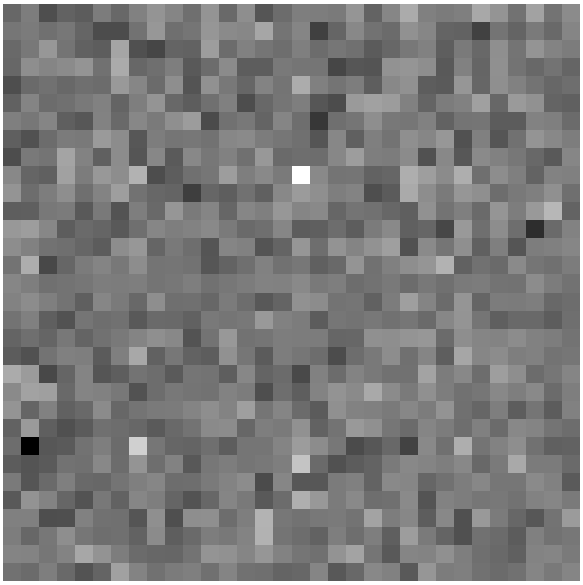


Fig. 8. Bands LL5, LH5, HL5, and HH5 from the watermarked "Lion" image generated by random filter banks

termarked image. We show the normalized image of bands LL5, LH5, HL5, and HH5, achieved by this method from the watermarked "lion" image, tiled together in Fig. 8. One can see that LL5 does not show extreme differences from the other three bands. Other images provide similar results.

Robustness is very important to watermarking algorithms. The first test on our algorithm is the robustness to noise. Two kinds of noise are tested. One is a zero mean Gaussian noise with

TABLE III

THE ROBUSTNESS TO NOISE

Image	Correlation (Gaussian noise)	Correlation (S.&P. noise)
Lion	0.950	0.938
Lena	0.971	0.951
Barbara	0.971	0.955
Baboon	0.974	0.961
Goldhill	0.980	0.957
Peppers	0.975	0.960

TABLE IV

THE ROBUSTNESS TO HISTOGRAM EQUALIZATION

Image	Correlation
Lion	0.930
Lena	0.970
Barbara	0.972
Baboon	0.967
Goldhill	0.951
Peppers	0.957

variance 100 and the other is a 1% salt-and-pepper noise. The results are given in Table III and demonstrate that our algorithm is robust to noise.

The watermarking algorithm should also be robust to image processing techniques. One common branch of image processing is histogram manipulation. The most popular method in this branch is histogram equalization. In Table IV, we list the correlations computed from histogram equalized watermarked images. From the results, one can see that our algorithm is robust to histogram equalization.

Another popular image processing tool is the median filter, which can be considered as a case

TABLE V

THE ROBUSTNESS TO MEDIAN FILTER

Image	Correlation
Lion	0.425
Lena	0.384
Barbara	0.501
Baboon	0.367
Goldhill	0.488
Peppers	0.413

of pixel permutation. The robustness to median filters is seldom addressed in watermarking papers. However, the authors of [14] mentioned in their paper that pixel permutation will cause serious problems for many watermarking algorithms because it destroys the synchronization. We apply 3 by 3 median filters to our test images, and the corresponding correlations are given in Table V. We can see that the correlations are much smaller than those in the previous cases. However, most of them are above the 0.4 threshold. We also find out in our experiments that the correlation values depend on the filter banks. For those images with smaller than threshold correlation values, by changing the filter banks, the correlation may be larger than the threshold. We think that this is enough to discourage the counterfeiters. Because they do not know the filter banks, they cannot predict whether the median filter will bring the correlation value below the threshold. If they just apply it blindly, there is a very large chance that they get caught. To illustrate this point, we run 200 tests on the image "baboon", which has the smallest correlation in Table V. Each time, we randomly generate the filter banks, insert the watermark, perform 3 by 3 median filter on the watermarked image, and then perform the detection procedure on the median filtered image. The range of the correlation values is from 0.352 to 0.818. However, only six samples are below the threshold 0.4. The histogram of the test results is given in Fig. 9. Hence, in a sense, our algorithm is robust to median filters as well.

Another important issue mentioned in [14] is the robustness to geometric transforms, as a lot of existing algorithms may not survive it. However, we think that in the context of digital watermarking for ownership verification, it is reasonable to use the original image to align the

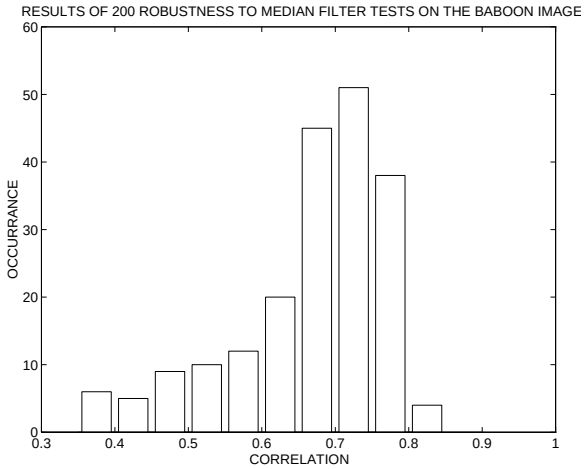
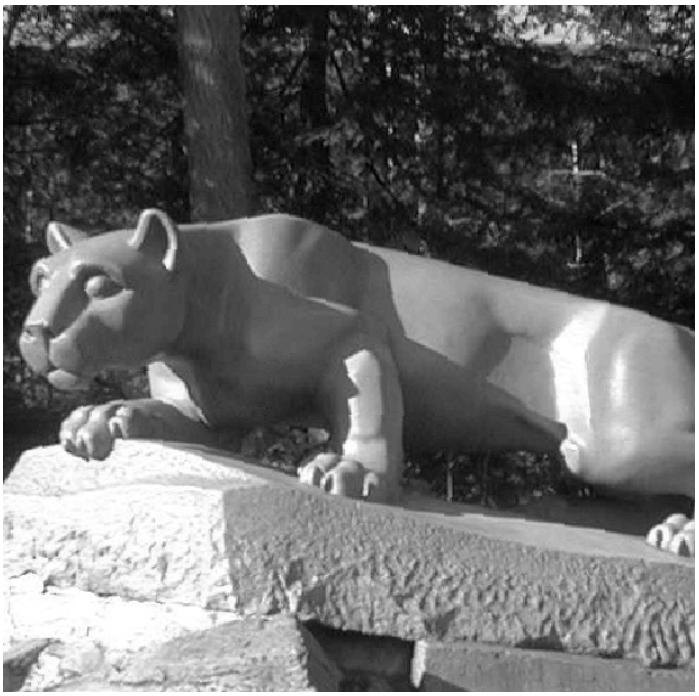


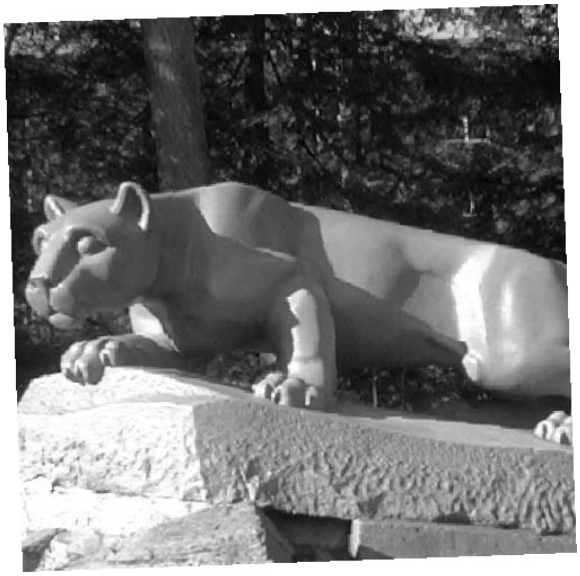
Fig. 9. Histogram of 200 median filter tests on the "baboon" image

suspect image. Unlike subtracting the original image from the suspect image, this alignment will not introduce any ambiguity. After the alignment, the suspect image will go through a geometric transform, which will not introduce any new signal. A counterfeiter cannot argue that because we rotate or shift his image, our watermark will appear in it. A very good method is presented in [20] for geometric alignment of images. The watermarking algorithm is also required to be robust to resampling and cropping. In the real world, if a owner suspects that somebody steals his image and manipulates the image using a geometric transform, resampling and cropping, he can align the images and try to detect his watermark. If he detects his watermark, he can provide the suspect image, his private key for that image and the aligning parameters to the legal authority for verification. In our experiments, we rotate the images 2 degrees clockwise, up-sample them to 640 by 640 and then crop them to get off-center 610 by 615 images. The final result of the "lion" image is given in Fig. 10. Then we use the method in [20] to align the images. The aligned "lion" image is also given in Fig. 10. The corresponding correlation results are given in Table VI. We realign each image individually, and the parameters returned from the aligning algorithm are slightly different for each image. However, we can see that for all images, the correlation results are much larger than the threshold.

Another desirable property of a watermark algorithm is robustness to image compression. We test two popular compression methods on our algorithm. The first one is JPEG baseline. The second one is JPEG2000. The bitrate, the PSNR with respect to the watermarked image, and the correlation values are listed in Table VII. If the bitrate increases, the correlation values



(a) rotated, resampled, and cropped image



(b) realigned image

Fig. 10. Transformed and realigned images

TABLE VI

THE ROBUSTNESS TO GEOMETRIC TRANSFORMS, RESAMPLING AND CROPPING

Image	Correlation
Lion	0.703
Lena	0.714
Barbara	0.776
Baboon	0.695
Goldhill	0.751
Peppers	0.767

will also increase. Fig. 11 shows both compressed "lion" images have noticeable degradation in fine details. Other images show similar results. When an owner suspects that a compressed image is an illegal copy of one of his images, he can perform the watermark detection process on the image. If the compression ratio for the image is not very high, as shown by Table VII, the correlation values will be larger than the threshold. If the compression ratio is high enough to reduce the correlation values to be less than the threshold, the image details will have noticeable degradation even if the most advanced compression methods, such as JPEG2000, are used. It is easy to convince a jury that such degraded images cannot be *the* original images. The court then can request the counterfeiter to submit a non-compressed version of his image. If the counterfeiter cannot provide such image, then the case is solved. If the counterfeiter gives a less compressed image that does not have noticeable degradation in fine details, then the correlation value produced by the watermark detection algorithm from that image will be larger than the threshold. Therefore, our algorithm is robust to compression.

A good watermarking algorithm should also have some resilience to counterfeit attempts. If a counterfeiter somehow gains the knowledge of our decomposition structure and the band to embed the watermark, one possible counterfeit attack is to set all coefficients of that band to zero. However, it is unlikely that the counterfeiter will know the filter banks that we use to embed the watermark. The only thing he can do is to use the same algorithm and randomly generate his own filter banks. Then he uses those filter banks to decompose the image and set all coefficients to zero in the corresponding band. However, because our filters have large sidelobes

TABLE VII

THE ROBUSTNESS TO COMPRESSION

JPEG Baseline			
Image	Bitrate(bpp)	PSNR(dB)	Correlation
Lion	0.32	28.8	0.502
Lena	0.26	33.9	0.404
Barbara	0.35	26.8	0.671
Baboon	0.50	24.0	0.661
Goldhill	0.28	29.0	0.599
Peppers	0.25	30.8	0.478
JPEG 2000			
Image	Bitrate(bpp)	PSNR(dB)	Correlation
Lion	0.25	29.6	0.422
Lena	0.25	36.5	0.519
Barbara	0.25	28.6	0.517
Baboon	0.25	23.0	0.506
Goldhill	0.25	30.3	0.549
Peppers	0.25	33.0	0.453

(a) original image (b) JPEG baseline image (c) JPEG 2000 image

Fig. 11. Fine detail comparison

TABLE VIII

THE RESILIENCE TO COUNTERFEIT ATTEMPTS

Image	Correlation
Lion	0.933
Lena	0.945
Barbara	0.882
Baboon	0.926
Goldhill	0.974
Peppers	0.971

as shown in Fig. 5, the watermark signal is distributed across all frequencies. Therefore, as Table VIII shows, we can still detect the existence of our watermark using the original filter banks.

As shown from the above discussions, the precise knowledge of the coefficients of the original filter banks, used for embedding the watermark, is very important. Hence arises the natural question of whether it is possible to approximate the filter coefficients by exhaustive search. In our example, for a counterfeiter to remove the watermark, he needs to know the first ten sets of filters. (The last two sets are just used to make the watermark similar with adjacent bands.) The easiest and most efficient way to approximate the filters is to approximate the $P'(z)$'s. There are ten $P'(z)$'s in our case. For each $P'(z)$, a'_1 and a'_3 are chosen freely while $a'_5 = 0.5 - a'_1 - a'_3$. One of the coarsest approximations is to approximate the a's using a single digit. Since we force $a'_1 \leq 0.2$, the total possible number of polynomials in this case is

$$(\sum_{i=0}^{2}\sum_{j=0}^{5-i} 1)^{10} = 15^{10} = 5.7665 \times 10^{11}. \tag{14}$$

The number is very large. Therefore, the exhaustive search will be very computationally expensive, if not impossible.

VI. CONCLUSIONS AND COMMENTS

In this paper, we first discuss the requirement for a practical digital watermarking system for ownership verification. Then, we present an algorithm that meets the requirements. As our experimental results have shown, the proposed algorithm achieves invisibility, robustness,

resilience to counterfeit attacks, and provides private control of the watermark. The detection process does not subtract the original image, hence avoiding ambiguity. The storage requirement for the watermark and the size of overhead storage for the private keys are reasonable as well.

We claim that there are enough choices for watermark patterns. For example, if the watermark is 16 by 16, the total possible patterns are $2^{16\times16} = 1.1579 \times 10^{77}$. The legal authority can even add some restrictions to the watermark patterns to make sure that there do not exist two very similar patterns. In our experiments, we used a watermark whose size is the same as the selected band. If the size of the watermark is smaller than the size of the selected band, we can always increase the size of the watermark by padding minus ones. Also, by choosing the number of decomposition levels wisely, we can make sure that the size of the watermark is never larger than the selected band.

Another point that we want to mention is that our algorithm is designed for "natural" images. Instead of pixel representation, the vector representation can be used for images of a graphic nature (e.g. cartoons, charts, etc.). This makes it possible for the images to go through totally different processing and compression procedures, which is beyond of the scope of this paper.

The proposed algorithm is relatively simple. However, similar to most algorithms, it requires its user to have some knowledge of the basic ideas of the algorithm and the relative technical background.

Although for simplicity, we choose orthonormal filter banks, biorthogonal filter banks may be used based on the same idea. The authors of [21] and [22] have proposed algorithms to construct 1-D and 2-D biorthogonal filter banks from a given filter. The issue of what type of filters are used in the algorithm is not important. To make the algorithm more robust to counterfeit attempts, we need to create as many ways as possible to randomly generate perfect reconstruction filter banks, which will increase the difficulty for counterfeiters to gain the exact knowledge of the filters. Another possibility is to add a PN sequence to the coefficients instead of replacing a certain band by a watermark pattern. In our future study, we will investigate these possible extensions of our algorithm.

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful suggestions. The first author would also like to take this opportunity to thank the other two authors for their kind guidance during his study at the Pennsylvania State University. In addition, the first author

would like to thank Dr. Nirmal K. Bose, Dr. David J. Miller, and Dr. Jesse L. Barlow, all from the Pennsylvania State University, for the fruitful discussions.

References

[1] I. Cox, J. Kilian, F. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Proc.*, vol. 6, no. 12, pp. 1673-1687, Dec. 1997.

[2] I. Pitas, "A method for watermark casting on digital image" *IEEE Trans. Circ. Syst. for Video Techn.*, vol. 8, no. 6, pp. 775-780, Oct. 1998.

[3] C. Hsu, and J. Wu, "Hidden digital watermarks in images," *IEEE Trans. Image Proc.*, vol .8, no. 1, pp. 58-68, Jan. 1999.

[4] C. Podilchuk, and W. Zeng, "Image-adaptive watermarking using visual models," *IEEE J. Sel. Areas Comm.*, vol. 16, no. 4, pp. 525-539, May 1998.

[5] W. Zeng, and B. Liu, "A statistical watermark detection technique without using original images for resolving rightful ownerships of digital images," *IEEE Trans. Image Proc.*, vol . 8, no. 11, pp. 1534-1548, Nov. 1999.

[6] J. Hernández, M. Amado, and F. Pérez-González, "DCT-domain watermarking techniques for still images, detector performance analysis and a new structure," *IEEE Trans. Image Proc.*, vol. 9, no. 1, pp. 55-68, Jan. 2000.

[7] Y. Wang, J. Doherty, and R. Van Dyck, "A novel wavelet-based algorithm for watermarking digital images," *Proc. CISS 2000*, vol. 1, pp. TA5.7-12, 2000.

[8] P. Bassia, and I. Pitas, "Robust audio watermarking in the time domain," *EUSIPCO 98*, Rhodes, Greece, Sept. 1998.

[9] M. Swanson, B. Zhu, and A. Tewfik, " Current state of the art, challenges and future directions for audio watermarking, " *IEEE Int. Conf. Multimedia Comp. Syst.*, vol. 1, pp, 19-24, 1999.

[10] C. Hsu, and J. Wu, "DCT based watermarking for video," *IEEE Trans. Cons. Elec.*, vol. 44, no. 1, pp. 206-216, Feb. 1998.

[11] M. Swanson, M. Kobayashi, and A. Tewfik, "Multimedia data-embedding and watermarking technologies," *Proc. of IEEE*, vol. 86, no. 6, pp. 1064-1087, June 1998.

[12] R. Wolfgang, C. Podilchuk, and E. Delp, "Perceptual watermarks for digital images and video," *Proc. of IEEE*, vol. 87, no. 7, pp. 1108-1126, July 1999.

[13] J. Bloom, I. Cox, T. Kalker, J. Linnartz, M. Miller, and C. Traw, "Copy protection for DVD video," *Proc. of IEEE*, vol. 87, no. 7, pp. 1267-1276, July 1999.

[14] F. Hartung, and M. Kutter, "Multimedia watermarking techniques," *Proc. of IEEE*, vol. 87, no. 7, pp. 1079-1107, July 1999.

[15] I. Daubechies, *Ten Lectures on Wavelets*, SIAM, 1992.

[16] M. Vetterli, and J. Kovačević, *Wavelets and Subband Coding*, Prentice-Hall PTR, 1995.

[17] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image Coding Using Wavelet Transform," *IEEE Trans. on Image Proc.*, vol. 1 no. 2, pp. 205-220, Apr. 1992.

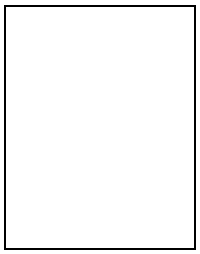
[18] N. Jayant, J. Johnston, and R. Safranek, "Signal compression based on models of human perception," *Proc. of IEEE*, vol. 81, no. 10, pp. 1385-1422, Oct. 1993.

[19] J. Conway, and N. Sloane, *Sphere Packings, Lattices and Groups* Springer-Verlag New York, Inc., 1999.

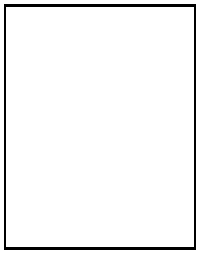
[20] S. Mann, R. W. Picard, "Video Orbits of The Projective Group: A Simple approach To Featureless Estimation Of Parameters," *IEEE Trans. Image Proc.*, vol. 6 no. 9, pp. 1281-1295, Sept. 1997.

[21] S. Basu, and H. Choi, "Hermite reduction methods for generation of a complete class of linear-phase perfect reconstruction filter banks-Part I: Theory," *IEEE Trans. Circ. Syst. Anal. Dig. Sig. Proc.*, vol. 46, no. 4, pp. 434-447, Apr. 1999.

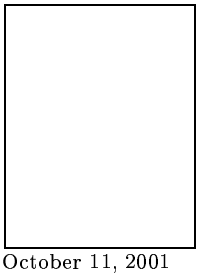
[22] C. Charoenlarpnopparut, and N. Bose, "Multidimensional FIR filter bank design using Grobner bases," *IEEE Trans. Circ. Syst. Anal. Dig. Sig. Proc.*, vol. 46, no. 12, pp. 1475-1486, Dec. 1999.



Yiwei Wang (S'98) received his B.E. from Tsinghua University, Beijing, P. R. China in 1996, his M. S. E. from Wright State University, Dayton, Ohio in 1998, and his Ph. D. in Electrical Engineering from the Pennsylvania State University, University Park, PA in 2001. Since July 2001, he has been with Chrontel Inc. His research interests include digital image processing, digital watermarking, digital signal processing, and other related areas.



John F. Doherty received the B.S. (Honors) degree in engineering science from The College of Staten Island of The City University of New York in 1982, the M.Eng. degree in electrical engineering from Steven Institute of Technology in 1985, and the Ph.D. degree in electrical engineering from Rutgers University in 1990. He worked as an integrated circuit reliability engineer at IBM from 1982 to 1984. From 1985 to 1988, he was a Member of Technical Staff at AT&T Bell Laboratories working in sonar signal processing. In 1990, he joined the Electrical and Computer Engineering Department at Iowa State University, Ames, IA, as Assistant Professor and Harpole-Pentair Fellow. Currently, he is Associate Professor and Charles H. Fetter Fellow of Electrical Engineering at The Pennsylvania State University, University Park, PA. His current research activities include interference rejection in wireless communication systems, spatial-division multiple access techniques, and robust video transmission over wireless channels. Dr. Doherty is a Senior Member of the IEEE and he is a former AFOSR Summer Faculty Research Fellow at Rome Laboratory and an Army Research Office Young Investigator.



Robert E. Van Dyck (S'92-M'92) received the B.E and M.E.E degrees from Stevens Institute of Technology, Hoboken, NJ, in 1985 and 1986, respectively, and the Ph.D. degree in electrical engineering from the North Carolina State University at Raleigh in 1992. Since June 2000, he has been a member of the Advanced Network Technologies Division of the National Institute of Standards and Technology, Gaithersburg, MD. Prior to that, he was an Assistant Professor

in the Department of Electrical Engineering, the Pennsylvania State University, University Park, PA. During 1999, he was a Summer Faculty Research Fellow at Rome Laboratory. His other previous affiliations include GEC-Marconi Electronic Systems, Wayne, NJ (1995-1996), the Center for Computer Aids for Industrial Productivity, Rutgers University, Piscataway, NJ (1992-1995), the Computer Science Corporation, Research Triangle Park NC (1989), and the Communications Laboratory, Raytheon Co., Marlborough, MA (1985-1988). His research interests are in multimedia communications and networking, video signal processing, and source and channel coding for wireless communications.