Semi-Fragile Watermarking for Authenticating JPEG Visual Content

Ching-Yung Lin and Shih-Fu Chang

Department of Electrical Engineering
Columbia University
New York, NY 10027, USA

ABSTRACT

In this paper, we propose a semi-fragile watermarking technique that accepts JPEG lossy compression on the watermarked image to a pre-determined quality factor, and rejects malicious attacks. The authenticator can identify the positions of corrupted blocks, and recover them with approximation of the original ones. In addition to JPEG compression, adjustments of the brightness of the image within reasonable ranges, are also acceptable using the proposed authenticator. The security of the proposed method is achieved by using the secret block mapping function which controls the signature generating/embedding processes. Our authenticator is based on two invariant properties of DCT coefficients before and after JPEG compressions. They are deterministic so that no probabilistic decision is needed in the system. The first property shows that if we modify a DCT coefficient to an integral multiple of a quantization step, which is larger than the steps used in later JPEG compressions, then this coefficient can be exactly reconstructed after later acceptable JPEG compression. The second one is the invariant relationships between two coefficients in a block pair before and after JPEG compression. Therefore, we can use the second property to generate authentication signature, and use the first property to embed it as watermarks. There is no perceptible degradation between the watermarked image and the origianl. In additional to authentication signatures, we can also embed the recovery bits for recovering approximate pixel values in corrupted areas. Our authenticator utilizes the compressed bitstream, and thus avoids rounding errors in reconstructing DCT coefficients. Experimental results showed the effectiveness of this system. The system also guarantees no false alarms, *i.e.*, no acceptable JPEG compression is rejected.

Keywords: image authentication, content verification, tampering detection, JPEG, watermark.

1 Introduction

Multimedia authentication techniques are used to verify the information integrity, the alleged source of data, and the reality of data. This topic distinguishes itself from other generic message authentication in it unique requirements of *integrity*. Multimedia data are generally compressed using standards such as JPEG, MPEG or H.26+. In many applications, compressed multimedia data may be accepted as authentic. Therefore, we consider that robustness to lossy compression is an essential requirement for multimedia authentication techniques.

Multimedia authentication techniques can be classified into three categories: *complete authentication*, *robust authentication*, and *content authentication*. *Complete authentication* refers to techniques that consider the whole piece of multimedia data and do not allow any manipulation[1][2]. Because the non-manipulable data are like generic messages, many existing message authentication techniques can be directly applied. For instance, digital signatures can be placed in the LSB of uncompressed data, or the header of compressed data. Then, manipulations will be detected because the hash values of the altered content bits may not match the information in the altered digital signature. In practice, fragile watermarks may be used for complete authentication.

We define *robust authentication* as a technique that treats altered multimedia data as authentic if manipulation is imperceptible. For example, authentication techniques, that accept lossy compression up to an allowable level of quality loss and reject other manipulations, belong to this category.

Content authentication techniques are designed to authenticate multimedia content in a semantic level even though manipulations may be perceptible. Such manipulations may include filtering, color manipulation, geometric distortion, *etc.* We distinguish these manipulations from lossy compression because these perceptible changes may be considered as acceptable to some observers but may be unacceptable to others.

A common objective for authentication is to reject the crop-and-replacement process that may change the meaning of data. Many robust watermarking techniques in the literature are designed to be robust to all manipulations for copyright protection purpose. They usually fail to reject the crop-and-replacement process so that they are not suitable for robust authentication considered here. In practice, we need to design semi-fragile watermarks for robust authentication and content authentication.

An authentication system can be considered as effective if it satisfies the following requirements:

- Sensitivity: The authenticator is sensitive to malicious manipulations such as crop-and-replacement.
- Robustness: The authenticator is robust to acceptable manipulations such as lossy compression, or other content-preserving manipulations.
- Security: The embedded information bits cannot be forged or manipulated. For instance, if the embedded watermarks are independent of the content, then an attacker can copy watermarks from one multimedia data to another.
- Portability: Watermarks have better portability than digital signatures because authentication can be conducted directly from the received content.
- Identification of manipulated area: Users may need partial information. The authenticator should be able to detect location of altered areas, and verify other areas as authentic.
- Recovery capability: Users may hope to know the content of original data in the manipulated area. The authenticator may need the ability to recover the lost content in the manipulated areas (at least approximately).

Previous Techniques for Robust Authentication and Content Authentication

Content authentication techniques are based on either digital signature or watermark. A detailed list of multimedia authentication research papers can be found in [3]. Using digital signature, Schneider and Chang first proposed the concept of salient feature extraction and similarity measure for image content authentication[4]. It also discussed issues of embedding such signatures into the image. However, it lacks comprehensive analysis of adequate features and embedding schemes. Bhattacha and Kutter proposed a method which extracts "salient" image feature points by using a scale interaction model and Mexican-Hat wavelets [5]. They generate digital signature based on the location of these feature points. The advantage of this technique is at the efficiency in its signature length. But, it lacks a rigorous mechanism to select visually interesting points. It may not be able to detect crop-and-replace manipulations inside the objects. Its robustness to accept lossy compression is questionable. Queluz proposed techniques to generate digital signature based on moments and edges of an image[6]. Using moments as features ignore the spatial information. Images can be manipulated without changing their moments. Edge-based features may be a good choice for image authentication because the contour of objects should be consistent during acceptable manipulations. However, it still has several open issues such as the excessive signature length, the consistency of edge detection, and the robustness to color manipulation.

Previously, we have developed authentication signatures that can distinguish JPEG/MPEG compression from malicious manipulations[7][8]. Our authentication signature is an encrypted feature vector generated from the invariant relationship between DCT coefficients in separate blocks of an image. We proved this relationship is preserved when the DCT coefficients are quantized or requantized in the JPEG/MPEG processes. Because the feature codes are generated based on the inherent characteristics of JPEG/MPEG processes, they can effectively distinguish such compressions from unacceptable manipulations, especially the crop-and-replacement attacks. The probability of falsely reporting JPEG/MPEG compression as attacks is negligible. Other acceptable attacks, *e.g.*, brightness and contrast enhancement, scaling, noise addition, can also be accepted by relaxing a tolerance threshold in the authenticator.

Using watermarking, Zhu *et. al.* proposed a method by measuring the error between the watermarked image and the manipulated image[9]. They estimate a masking function from the image, and use it to measure distortion. Their method adds imperceptible changes to the image. But, it is not clear that whether this estimated masking function will be the same in the watermarked image and in the images with acceptable manipulation. Further, it may not provide the information of error measurement, because the masking function will change if the image is manipulated by pixel replacement. Wolfgang and Delp developed an authentication method that embeds bipolar m-sequence into blocks[10]. This method can localize manipulation, and showed moderate robustness. But, its watermarks are generated from the checksum of pixel values excluding LSB. Because acceptable compression may result in the change in the LSB as well as other bits, a larger probability of false alarm may appear in the system. Fridrich proposed a robust watermarking technique for authentication [11][12]. He divided images to 64 pixel × 64 pixel blocks. For each block, quasi-VQ codes are embedded using the spread spectrum method. This technique is robust to manipulations. But, comparing his experiments in [11] and in [12], we saw that JPEG compressions result in more error than pixel replacement. It is unclear whether this method can detect small area modification or distinguishes JPEG compression from malicious manipulations.

Proposed Approaches

In this paper, we present a watermarking technique for embedding our previously proposed authentication signatures into images. Such signature-based image watermarks need to satisfy the following criteria. (1) The watermark extracted from the watermarked image should match the authentication signature of the watermarked image. This may be different from the original signature extracted from the un-watarmarked image. To achieve this, some iterations may be needed in implementation. (2) The signature and the watermark consist two layers of protection. Malicious attacks will destroy either layer or both layers. Acceptable manipulations should preserve both layers. The performance of an authentication system depends on these two layers.

We propose a semi-fragile watermarking technique that accepts some acceptable manipulations such as JPEG lossy compression and reasonable brightness adjustment on the watermarked image to a pre-determined quality factor, and rejects crop-and-replacement process. Images with excessive compression rate are considered un-authentic due to poor quality. The authenticator can identify the position of corrupted blocks, and even recover them with approximation of the original. Security of the method is achieved by using a secret block mapping function which indicates the formation of block pairs and signature/watermarking groups.

Our authenticator is based on the invariant properties of DCT coefficients before and after the JPEG compression. These properties are guaranteed so that no probabilistic decision is needed. The first property shows if we quantize a DCT coefficient to a reference value, then this pre-quantized coefficient can be *exactly* reconstructed after subsequent JPEG compression, if the original quantized step is *larger* than the one used in the JPEG compression. We utilize this property to embed signature as watermarks. The second one is the invariant *relationship* of two coefficients in a block pair. We use this property to form the authentication bits of signature. In addition to these properties, two methods are applied in practical system design: (1) the authentication process utilizes the compressed bitstream to reconstruct the quantized DCT coefficients without going back to the pixel domain, and (2) the embedding process recursively apply integral DCT and Inverse DCT until the designated DCT values can be directly obtained from integer pixel values. These methods help avoid computation errors and false alarm in practical implementations.

This paper is organized as follows. In section 2, we show two important properties mentioned above. In Section 3, we describe details of our authentication system. The performance of this authentication system is analyzed in Section 4. In Section 5, we show some testing results. Conclusion and discussion of future directions are shown in Section 6.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

(a)

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

(b)

Table 1: The quantization tables, $\mathbf{Q_{50}}$, of JPEG compression with *Quality Factor(QF) = 50* : (a) luminance,(b) chrominance[13]. The quantization tables, $\mathbf{Q}_{QF}$ of other Quality Factor are *Integer Round*$(\mathbf{Q}_{50} \cdot q)$, where $q = 2 - 0.02 \cdot QF$, if $QF \geq 50$, and $q = \frac{50}{QF}$, if $QF < 50$. In the baseline JPEG, $\mathbf{Q}_{QF}$ will be truncated to be within 1 to 255.

2 Two Invariant Properties in JPEG compression

In this section, we describe and prove two invariant properties during JPEG compression. The first one is used for embedding watermark, and the second one is proposed in [7] and is used for generating authentication signature.

Theorem 1 *Assume* $\mathbf{F_p}$ *is a DCT coefficient vector of an arbitrary* 8×8 *non-overlapping blocks of image X, and* $\mathbf{Q_m}$ *is a pre-selected quantization table for JPEG lossy compression. For any* $\nu \in \{1, .., 64\}$ *and* $p \in \{1, .., \wp\}$, *where* $\wp$ *is the total number of blocks, if* $\mathbf{F_p}(\nu)$ *is modified to* $\grave{\mathbf{F}}_\mathbf{p}(\nu)$ s.t. $\frac{\grave{\mathbf{F}}_\mathbf{p}(\nu)}{\mathbf{Q'_m}(\nu)} \in Z$ *where* $\mathbf{Q'_m}(\nu) > \mathbf{Q_m}(\nu)$, *and define* $\tilde{\mathbf{F}}_\mathbf{p}(\nu) \equiv$ *Integer Round*$(\frac{\grave{\mathbf{F}}_\mathbf{p}(\nu)}{\mathbf{Q}(\nu)}) \cdot \mathbf{Q}(\nu)$ *for any* $\mathbf{Q}(\nu) \leq \mathbf{Q_m}(\nu)$, *the following property holds:*

$$\textit{Integer Round}(\frac{\tilde{\mathbf{F}}_\mathbf{p}(\nu)}{\mathbf{Q'_m}(\nu)}) \cdot \mathbf{Q'_m}(\nu) = \grave{\mathbf{F}}_\mathbf{p}(\nu) \quad (1)$$

□

Proof of Theorem 1: See the Appendix.

Theorem 1 shows that if a DCT coefficient is modified to an integral multiple of a pre-selected quantization step, $\mathbf{Q'_m}$, which is larger than all possible quantization steps in subsequent acceptable JPEG compression, then this modified coefficient can be *exactly reconstructed* after future acceptable JPEG compression. It is reconstructed by quantizing the subsequent coefficient again using the same quantization step, $\mathbf{Q'_m}$. We call such exactly reconstructible coefficients, $\grave{\mathbf{F}}_\mathbf{p}$, "reference coefficients."

We first define the meaning of acceptable JPEG compression. Table 1 shows that quantization tables of JPEG compression for all quality factors. From Table 1, we know that

$$\mathbf{Q_{QF}}(\nu) \leq \mathbf{Q_m}(\nu), \quad \forall \nu \in \{1, ..64\} \textit{ and } QF \geq m. \quad (2)$$

In other words, the higher QF (quality factor) is, the smaller the quantization step is. In Eq. 2, the equality will still hold even if $QF > m$, because of integer rounding (shown in the description of Table 1). In general, JPEG recommends a quality factor of 75-95 for visually indistinguishable quality difference, and a quality factor of 50-75 for merely acceptable quality[14]. If we adopt this recommendation and set the quantization table, $\mathbf{Q_{50}}$, as a quality threshold for acceptable JPEG compression, *i.e.*, $\mathbf{Q_m} = \mathbf{Q_{50}}$, then all future quantization table, $\mathbf{Q_{QF}}$, $\forall QF \geq 50$, will be smaller than or equal to $\mathbf{Q_{50}}$.

For watermarking, we quantize original DCT coefficients using a pre-determined quantization step, $\mathbf{Q'_m}(\nu)$, which is larger than $\mathbf{Q_m}(\nu)$ (note the greater than but not equal sign in Theorem 1). For instance, $\mathbf{Q'_m}(\nu) = \mathbf{Q_m}(\nu) + 1$. If $\mathbf{F_p}(\nu)$ is modified to $\grave{\mathbf{F}}_\mathbf{p}(\nu)$, the reference coefficient, *s.t.* $\frac{\grave{\mathbf{F}}_\mathbf{p}(\nu)}{\mathbf{Q'_m}(\nu)} \in Z$, then this reference coefficient could be exactly reconstructed after future acceptable JPEG compressions according to Theorem 1. Given the reconstructible coefficients, we have many choices to embed watermarks into the image. For instance, in the authentication system, we can use the LSB of the quantized reference value to represent the watermark bit. In this way, hiding a bit in the image needs to modify only one DCT coefficient (with a distortion within $\mathbf{Q'_m}(\nu)$) and leave other DCT coefficients intact.

It should be noted that Theorem 1 can be applied to a broader area than just JPEG compression. It holds whenever new distortion is smaller than $\frac{1}{2}\mathbf{Q'_m}(\nu)$.

Theorem 2 *(as in [7])* *Assume $\mathbf{F_p}$ and $\mathbf{F_q}$ are DCT coefficient vectors of two arbitrary 8×8 non-overlapping blocks of image X, and $\mathbf{Q}$ is a quantization table of JPEG lossy compression. $\forall \nu \in \{1,..,64\}$ and $p, q \in \{1,..,\wp\}$, where $\wp$ is the total number of blocks, define $\mathbf{\Delta F_{p,q}} \equiv \mathbf{F_p} - \mathbf{F_q}$ and $\mathbf{\Delta \tilde{F}_{p,q}} \equiv \mathbf{\tilde{F}_p} - \mathbf{\tilde{F}_q}$ where $\mathbf{\tilde{F}_p}$ is defined as $\mathbf{\tilde{F}_p}(\nu) \equiv$ Integer Round$(\frac{\mathbf{F_p}(\nu)}{\mathbf{Q}(\nu)}) \cdot \mathbf{Q}(\nu)$. Assume a fixed threshold $k \in \Re$. $\forall \nu$, define $\tilde{k}_\nu \equiv$ Integer Round $(\frac{k}{\mathbf{Q}(\nu)})$. Then,*
if $\mathbf{\Delta F_{p,q}}(\nu) > k$,

$$\mathbf{\Delta \tilde{F}_{p,q}}(\nu) \geq \begin{cases} \tilde{k}_\nu \cdot \mathbf{Q}(\nu), & \frac{k}{\mathbf{Q}(\nu)} \in Z, \\ (\tilde{k}_\nu - 1) \cdot \mathbf{Q}(\nu), & elsewhere, \end{cases} \tag{3}$$

else if $\mathbf{\Delta F_{p,q}}(\nu) < k$,

$$\mathbf{\Delta \tilde{F}_{p,q}}(\nu) \leq \begin{cases} \tilde{k}_\nu \cdot \mathbf{Q}(\nu), & \frac{k}{\mathbf{Q}(\nu)} \in Z, \\ (\tilde{k}_\nu + 1) \cdot \mathbf{Q}(\nu), & elsewhere, \end{cases} \tag{4}$$

else $\mathbf{\Delta F_{p,q}}(\nu) = k$,

$$\mathbf{\Delta \tilde{F}_{p,q}}(\nu) = \begin{cases} \tilde{k}_\nu \cdot \mathbf{Q}(\nu), & \frac{k}{\mathbf{Q}(\nu)} \in Z, \\ (\tilde{k}_\nu \ or \ \tilde{k}_\nu \pm 1) \cdot \mathbf{Q}(\nu), & elsewhere. \end{cases} \tag{5}$$

□

Proof of Theorem 2: See [7].

In a special case when $k = 0$, Theorem 2 describes the invariance property of the sign of $\mathbf{\Delta F_{p,q}}$. Because all DCT coefficients matrices are divided by the same quantization table in the JPEG compression process, the relationship between two DCT coefficients of the same coordinate position from two blocks will not be changed after the quantization process. The only exception is that "*greater than*" or "*less than*" may become "*equal*" due to quantization. These properties hold for any times of recompression and/or any quantization table utilizing JPEG. By applying Theorem 2, we can generate authentication bits of an image from the relationship between two DCT coefficients of the same position in two separate 8×8 blocks, *i.e.*, a block pair. These authentication bits, or their encrypted version, are then embedded as a watermark. For the authentication process, the authenticator compares the extracted authentication bits and the relationship of the corresponding DCT coefficients of the block pairs from the received image. Authenticity of a block pair is verified if their DCT coefficient relationships match the criteria predicted by Theorem 2 using the extracted authentication bits.

3 System Description

We generate and embed two kinds of signature bits: authentication bits, $\mathbf{\Phi}$, and recovery bits, $\mathbf{\Psi}$. Users can choose to embed either one or both of them . If only the authentication bits are embedded, then the authenticator can detect malicious manipulations, but can not recover approximate values of the original. Similarly, if users only embed the recovery bits, then the authenticator can retrieve an approximate image and leaves the users to judge the authenticity by themselves. The embedding process of these two kinds of bits are independent, because they are placed in different positions of DCT coefficients. One important issue is determination of the embedding positions for authentication and recovery bits. We will address this issue in the following.

3.1 Generating and Embedding Authentication Bits

For a watermark-based authentication system, the whole space of coefficients is divided into three subspaces: *signature generating*, *watermarking*, and *ignorable* zones. Zones can be overlapped or non-overlapped. Usually, if the first two zones are overlapped, then some iteration procedures are needed to guarantee the extracted signature matches the signature generated from the watermarked image. It should be noted that these conceptual zones exist in all watermark-based authentication methods. Coefficients in the signature generating zone are used to generate authentication bits. The watermarking zone is used for embedding signature back to image as watermark. The last zone is negligible. Manipulations of coefficients in this zone do not affect the processes of signature generation and verification. In our system, we use non-overlapping zones to generate and embed authentication bits. For security, the division method of zones should kept secret or be indicated by a secret (time-dependent and/or location-dependent) mapping method using a seed.

We use a signature generation method we proposed in [7]. Similar to the JPEG process, images are first divided to 8×8 blocks. Then, blocks are formed into block pairs using a pre-determined secret mapping function, T_b. For instance, for a block p, we use T_b to choose a counterpart block to form a block pair, such that $q = T_b(p)$. For each block pair, we pre-select β_a out of 64 positions in the zigzag order, and denote these positions as a set, $\mathbf{B}_p$, which represents the signature generating zone of the block pair (p, q). Then, we generate their authentication bits, $\mathbf{\Phi}_p$, such that

$$\mathbf{\Phi}_p(\nu) = \begin{cases} 1, & \mathbf{\Delta F}_{p,q}(\nu) \geq 0 \\ 0, & \mathbf{\Delta F}_{p,q}(\nu) < 0, \end{cases} \tag{6}$$

where $\nu \in \mathbf{B}_p$.

To embed the authentication bits, the system has to set a threshold for acceptable JPEG quality factor, $\mathbf{m}$, a mapping function, T_a, and sets $\mathbf{E}_p$ that indicate the watermarking zone of each block. Each $\mathbf{E}_p$ includes $\frac{1}{2}\beta_a$ positions (since there are two blocks in a pair for embedding). For instance, if $\beta_a = 6$, then each block has to embed 3 authentication bits. The mapping function T_a is used to indicate where the embedding authentication bits are generated. These parameters,$\mathbf{m}$, T_a, and $\mathbf{E}_p$, are image independent secret information and can be set to default values for each digital camera. They are applied to all images captured from the same device. If a more secure mechanism is needed, they can be designed by using time-dependent seeds that change these parameters over time, and then embedding these seeds as watermarks into the image using methods like global spread spectrum method.

To embed an authentication bit $\mathbf{\Phi}_{T_a(p)}(\nu)$, to a specific DCT coefficient, $\mathbf{F}_p(\nu)$, we have to calculate $\mathbf{f'}_p(\nu) = \mathit{Integer\ Round}(\frac{\mathbf{F}_p(\nu)}{\mathbf{Q'_m}(\nu)})$, where $\mathbf{Q'_m}(\nu) = \mathbf{Q_m}(\nu)+1$. Then we embed the authentication bits by modifying $\mathbf{F}_p(\nu)$ to $\grave{\mathbf{F}}_p(\nu)$ as follows

$$\grave{\mathbf{F}}_p(\nu) = \begin{cases} \mathbf{f'}_p(\nu) \cdot \mathbf{Q'_m}(\nu), & LSB(\mathbf{f'}_p(\nu)) = \mathbf{\Phi}_{T_a(p)}(\nu) \\ (\mathbf{f'}_p(\nu) + sgn(\frac{\mathbf{F}_p(\nu)}{\mathbf{Q'_m}(\nu)} - \mathbf{f'}_p(\nu)) \cdot \mathbf{Q'_m}(\nu), & LSB(\mathbf{f'}_p(\nu)) \neq \mathbf{\Phi}_{T_a(p)}(\nu), \end{cases} \tag{7}$$

where $sgn(x) = 1$, if $x \geq 0$, and $sgn(x) = 0$, if $x < 0$. Note the above quantization and embedding operations are applied to selected coefficients (for embedding) only, not the whole block. Different coefficients in the block can be used to embed recovery bits, using different quantization steps.

In practical systems, converting the modified DCT coefficient back to the integer pixel domain and then converting them again to the DCT domain may not get the same result. Therefore, an iteration procedure, which examines the DCT of modified integral pixel values, is needed to guarantee the watermark bits be exactly extracted from the watermarked image. (But theoretical convergence of such iterative process remains to be proved.) In our experiments, this iteration is needed for about 10% of the blocks, and most of them need no more than 2 iterations.

In the image blocks with "flat" areas, using the second equation in Eq. 7 to modify AC values may introduce visible distortion, if the acceptable quality factor is not very high (*e.g.*, QF $\leq$ 75). To address this problem, we can use the following alternative. If we want to embed β_a bits to two blocks (p, q), instead of embedding two bits in $\mathbf{F}_p(\nu)$ and $\mathbf{F}_q(\nu)$, we can embed only one bit. We use the *XOR* function, denoted as

x_ν, of $LSB(\mathbf{f'}_p(\nu))$ and $LSB(\mathbf{f'}_q(\nu))$ to represent a bit, b_ν. If $x_\nu = b_\nu$, then $\grave{\mathbf{F}}_p(\nu) = \mathbf{f'}_p(\nu) \cdot \mathbf{Q'_m}(\nu)$ and $\grave{\mathbf{F}}_q(\nu) = \mathbf{f'}_q(\nu) \cdot \mathbf{Q'_m}(\nu)$. If $x_\nu \neq b_\nu$ and either $\mathbf{f'}_p(\nu)$ or $\mathbf{f'}_q(\nu)$ equals 0, then

$$(\grave{\mathbf{F}}_p(\nu), \grave{\mathbf{F}}_p(\nu)) = \begin{cases} (\mathbf{f'}_p(\nu),\ \mathbf{f'}_q(\nu) + sgn(\frac{\mathbf{F}_q(\nu)}{\mathbf{Q'_m}(\nu)} - \mathbf{f'}_q(\nu)) \cdot \mathbf{Q'_m}(\nu), & if\ \mathbf{f'}_p(\nu) = 0,\ \mathbf{f'}_p(\nu) \neq 0, \\ (\mathbf{f'}_p(\nu) + sgn(\frac{\mathbf{F}_p(\nu)}{\mathbf{Q'_m}(\nu)} - \mathbf{f'}_p(\nu)),\ \mathbf{f'}_q(\nu)) \cdot \mathbf{Q'_m}(\nu), & if\ \mathbf{f'}_p(\nu) \neq 0,\ \mathbf{f'}_p(\nu) = 0. \end{cases} \tag{8}$$

If $x_\nu \neq b_\nu$ and both of $\mathbf{f'}_p(\nu)$ and $\mathbf{f'}_q(\nu)$ are 0 or *non-zero*, then

$$(\grave{\mathbf{F}}_p(\nu), \grave{\mathbf{F}}_p(\nu)) = \begin{cases} (\mathbf{f'}_p(\nu),\ \mathbf{f'}_q(\nu) + sgn(\frac{\mathbf{F}_q(\nu)}{\mathbf{Q'_m}(\nu)} - \mathbf{f'}_q(\nu)) \cdot \mathbf{Q'_m}(\nu), & if\ |\frac{\mathbf{F}_p(\nu)}{\mathbf{Q'_m}(\nu)} - \mathbf{f'}_p(\nu)| \geq |\frac{\mathbf{F}_q(\nu)}{\mathbf{Q'_m}(\nu)} - \mathbf{f'}_q(\nu)|, \\ (\mathbf{f'}_p(\nu) + sgn(\frac{\mathbf{F}_p(\nu)}{\mathbf{Q'_m}(\nu)} - \mathbf{f'}_p(\nu)),\ \mathbf{f'}_q(\nu)) \cdot \mathbf{Q'_m}(\nu), & if\ |\frac{\mathbf{F}_p(\nu)}{\mathbf{Q'_m}(\nu)} - \mathbf{f'}_p(\nu)| < |\frac{\mathbf{F}_q(\nu)}{\mathbf{Q'_m}(\nu)} - \mathbf{f'}_q(\nu)|. \end{cases} \tag{9}$$

Eq.8 is used to avoid large modifications on the AC coefficients in the "flat" blocks. Eq. 9 is applied to choose a smaller modification when the two coefficients are all zero or all non-zero. In practical system, we choose the block pair (p, q) to be such that one is in the corner and the other near the center of image. We found that, applying this method, the distortion introduced by watermarking will become invisible in most images if the acceptable JPEG quality factor is set to be 50. For a more secure system, the XOR method can be substituted by a position dependent look-up table.

3.2 Generating and Embedding Recovery Bits

The recovery bits are used to reconstruct an approximation of the original block. These recovery bits have to cover the whole image, because each block is possibly manipulated. They can be generated using a procedure similar to low-bit rate lossy compression. We use JPEG compression with a low quality factor, because most digital cameras or image editing tools already have components of JPEG compression, and therefore, the incurred implementation cost is low.

To generate the recovery bits, $\mathbf{\Psi}$, we first scale down the image by 2 along each axis, and divide the image into 8×8 blocks. Then, we use a JPEG quantization table with low QF (*e.g.*, 25) to quantize DCT coefficients, and apply Huffman coding on the quantized coefficients. These quantization and Huffman coding procedures are the same as those in standard JPEG compression. Because images are scaled-down by 2, we need to embed the encoded bits of each scaled block into 4 original blocks.

The embedding process of recovery bits is similar to that of authentication bits. We also need to set a threshold for acceptable JPEG quality factor, $\mathbf{m_r}$, which can be different from the one used in embedding authentication bits. Selected coefficients are pre-quantized based on $\mathbf{Q'_{m_r}}(\nu) = \mathbf{Q_{m_r}}(\nu) + 1$ to get reference values. A mapping function, T_r, is used for selecting 4 blocks (denoted as $p1$, .. ,$p4$) in the original image to embed recovery bits of a block in the down-scaled image. We use $\mathbf{E'}_p$ to indicate the second watermarking zone for embedding recovery bits. Each $\mathbf{E'}_p$ includes β_r positions in a block. These parameters are image independent. Then, recovery bits are embedded in a similar way as in Eq. 7 (or Eq. 8 and Eq. 9). They are embedded in these four blocks in a round robin fashion. Because the coded bit length of a block in the scaled-down image is variable, if the coded bit length of an block is larger than $4\beta_r$, then those bits exceeding the capacity will be discarded.

3.3 Authentication Process

In the authentication process, the system extracts the authentication bits from the watermarking zone of received image, and uses them to verify whether the DCT coefficient relationships in the signature generation zone match the criteria predicted by Theorem 2. If they match, the image is said to be authentic. Otherwise, the changed blocks are identified and recovered by using the recovery bits if they are available.

When a new DCT coefficient relationship does not match the prediction of the authentication bit reconstructed from the watermark, we know this image has been manipulated. Note there could be as many as four blocks involved here. The examined DCT coefficients are in the signature zone of a block pair (say blocks p_1 and p_2). The authentication bit is recovered from the watermarking zones of two blocks (say blocks p_3 and p_4). When

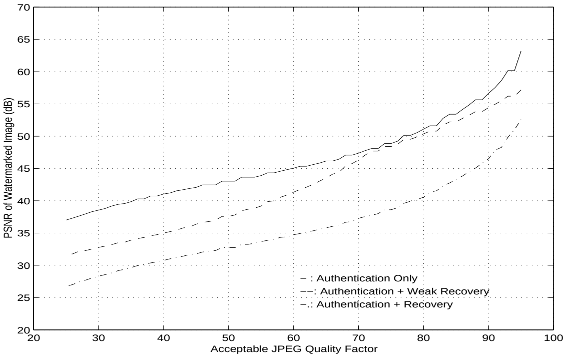


Figure 1: Expected value of PSNR of the watermarked image v.s. Acceptable JPEG Quality Factor. The embedded bits are: (1) Authentication Only: 3 bits/block, (2) Authentication + Weak Recovery: 9 bits/block , and (3) Authentication + Recovery: 9 bits/block.

the above comparison process reports a mismatch from an authentication bit in p_3, there are three possible areas that could have been changed: the signature generation zone of p_1, the signature generation zone of p_2, and the watermarking zone of p_3. Assume only one block has been changed. A problem of the authentication process is how to identify the manipulated block. To test whether p_1 has been manipulated, we can test the watermarking zone of p_1 to see whether it can successfully verify the authenticity of its referred block pair, because, in general, all zones in a block may have been altered after manipulation. Similar tests can be applied to p_2 and p_3. It should be noted that these solutions are based on the assumption that manipulations are localized. If they are uniformly existed in the whole image, then our authenticator may report some false alarm blocks. In the previous example, if p_2 is the only manipulated block in these four blocks but the referred block pair of p_1 has been manipulated, then we may report both p_1 and p_2 as manipulated.

4 Performance Evaluation of Authentication System

We use three measures to evaluate an authentication system: *the probability of false alarm* (of acceptable manipulations), P_{FA}, *the probability of miss* (on detecting malicious manipulations), P_M, and *the probability of successful attack*, P_S [7]. The first two are from the viewpoints of signature generator. The last one is from the viewpoints of attacker. The last two are distinguished based on different information known to the signature generator and the attacker. These probabilities usually depend on each individual image and the length of the signature. Usually, the longer the signature length is, the better the system performance is. However, for a watermarking system, the longer the embedded signature is, the worse the watermarked image quality will be. There is a tradeoff between system performance and image quality. The analysis in this section is based on the simplest implementation described in Eq. 6 and Eq. 7.

Quality of Watermarked Image

In our system, if we use PSNR to measure the degradation of image quality caused by watermarking, the expectation value of PSNR will be image independent. We first show that the expectation value of error power of an individual DCT coefficient is,

$$E[\sigma_{\mathbf{w}}{}^2(\nu)] = \frac{1}{2} \cdot \int_0^{\mathbf{Q}'_{\mathbf{m}}(\nu)} x^2 f(x)dx + \frac{1}{2} \cdot \int_0^{\mathbf{Q}'_{\mathbf{m}}(\nu)} (\mathbf{Q}'_{\mathbf{m}}(\nu) - x)^2 f(x)dx = \frac{1}{3}\mathbf{Q}'_{\mathbf{m}}{}^2(\nu), \tag{10}$$

where we assume x to be a random variable which is uniformly distributed between 0 and $\mathbf{Q}'_{\mathbf{m}}(\nu)$, *i.e.*, $f(x) = \frac{1}{\mathbf{Q}'_{\mathbf{m}}(\nu)}$ which is the probability density function of x. The first and second terms are the cases that x is quantized to 0 and $\mathbf{Q}'_{\mathbf{m}}(\nu)$, respectively. Then the expectation value of PSNR of a watermarked image is,

$$E[PSNR] = 10log_{10}\frac{64 \cdot 255^2}{\sum_{\nu_i \in \mathbf{E}} E[\sigma_{\mathbf{w}}{}^2(\nu_i)]}. \tag{11}$$

Applying Table 1, we can obtain the expected PSNR values of watermarked images after setting maximally acceptable JPEG compression and pre-determined embedding positions $\mathbf{E}$. A figure of these values is shown in Figure 1. In Figure 1, authentication bits are assumed to be embedded in $\nu \in \{6,7,8\}$, and recovery bits are in $\nu \in \{9,..,14\}$. In this way, each block pair is protected by 6 bits, and each recovery block is composed of 24 bits. We can see that if the acceptable quality factor is 50, then the PSNR of the watermarked image compared to the original is 43.03 dB for authentication bits only, and 32.75 dB for embedding authentication bits and recovery bits. This PSNR value is 37.80 dB for embedding authentication bits and weak recovery bits. The notion of "weak recovery" is used to explore the tradeoff between the image quality and the authentication strength. As discussed earlier, we can set the pre-quantization levels of authentication and recovery independently. In practice, we can set a different pre-quantization level for recovery from the that for authentication. Thus the received image is authenticated to some quality factor but it can only be recovered up to some higher quality factor. In Figure 1, we set the quality factor for weak recovery to be 25 larger than that for authentication.

Probability of False Alarm

Usually, an authentication system is designed based on a pre-determined acceptable level of probability of false alarm. In a watermark-based system, P_{FA} is composed of two probabilities: the probability of reconstructing false authentication bits, $P_{FA,\mathbf{E}}$, and the probability of false DCT relationships that violate Theorem 2, $P_{FA,\mathbf{B}}$. According to Theorem 1 and Theorem 2, the probability of false alarm,

$$P_{FA} = 0, \tag{12}$$

if the image goes through by the JPEG lossy compression. In practical systems, Eq. 12 is true if the authenticator directly reconstruct DCT coefficients from the compressed bitstream, and utilizes integral DCT and Inverse DCT, that use integer values in both the spatial domain and the DCT domain, for authentication bits generation and signature embedding.

If the image is distorted by i.i.d. zero mean Gaussian noises with variance σ_N^2 instead of JPEG compression, then the probability of false alarm in a block pair,

$$P_{FA} = 1 - (1 - P_{FA,\mathbf{E}})(1 - P_{FA,\mathbf{B}}) \approx P_{FA,\mathbf{E}} + P_{FA,\mathbf{B}} \tag{13}$$

where

$$P_{FA,\mathbf{E}} = 1 - \prod_{\nu\in\mathbf{E}} [1 - \sum_{i=0}^{\infty} [erfc(\frac{(\frac{1}{2}+2i)\mathbf{Q'm}(\nu)}{\sqrt{2}\sigma_N}) - erfc(\frac{(\frac{3}{2}+2i)\mathbf{Q'm}(\nu)}{\sqrt{2}\sigma_N})]]. \tag{14}$$

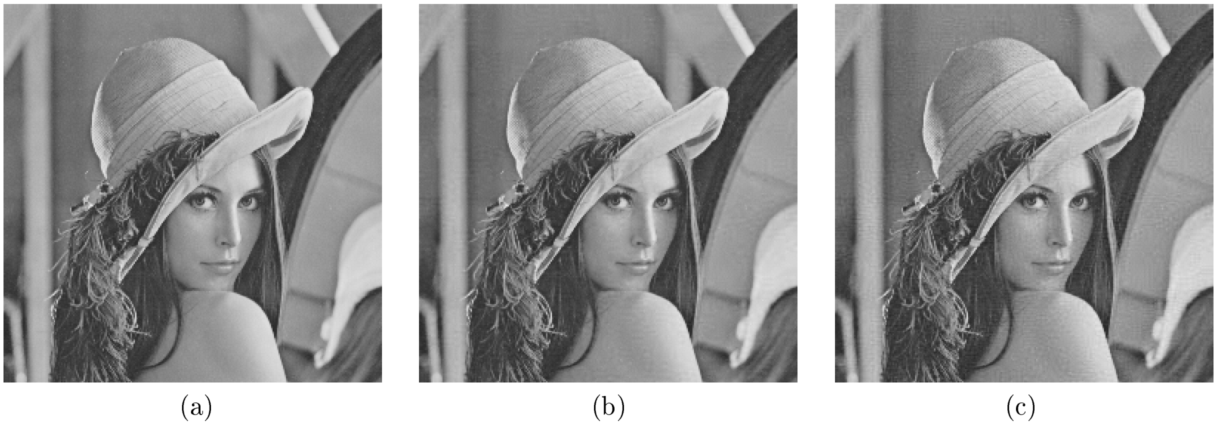
where $erfc()$ is the complementary error function[15]. And

$$P_{FA,\mathbf{B}} = 1 - \prod_{\nu\in\mathbf{E}} [1 - \frac{1}{2} erfc(\frac{|\mathbf{\Delta F}_{p,q}(\nu)|}{2\sigma_N})]. \tag{15}$$

We can see that $P_{FA,\mathbf{E}}$ is image independent, but $P_{FA,\mathbf{B}}$ is image dependent. For instance, if we set $\mathbf{Q'm} = \mathbf{Q_{50}}$ and use $\nu \in \{6,7,8\}$ to embed authentication bits, then $P_{FA,\mathbf{E}} = 0.0017$ for $\sigma_N = 2$ (*i.e.*, PSNR = 42 dB). In a 256×256 "lenna" image, if we use the adjacent blocks as block pairs and extract 6 bits for each block pair, then the median value of $P_{FA,\mathbf{B}} = 0.12$ for $\sigma_N = 2$. These high values are from the high possibility that small $\mathbf{\Delta F}_{p,q}(\nu)$ may change sign in the present of noise. However, if we use the tolerance bound in authentication[7] and set the bound equal to $\mathbf{Q'm}$, then $P_{FA,\mathbf{B}} = 9 \times 10^{-9}$ which decreases significantly.

Probability of Miss and Probability of Successful Attack

The probability of Miss, P_M, and the probability of Successful Attack, P_S, are measures of the capability of an authentication to detect unacceptable manipulations. They are calculated from different given information. If a block p is manipulated, then $P_{M,p}$ is the probability that, after manipulation, the relationships of the DCT coefficients $\in \mathbf{B}_p$ of the block pair (p,q) do not violate Theorem 2, given the original $\mathbf{F}_p$, $\mathbf{F}_q$, and $\mathbf{B}_p$. This is a measure from the signature authenticator. In other words, that is a probability that the content distributor knows



(a) (b) (c)

Figure 2: (a) The original image, (b) the watermarked image after embedding authentication bits (PSNR = 40.7 dB), (c) the watermarked image after embedding authentication bits and weak recovery bits (PSNR = 37.0 dB).

how each specific watermarked image may miss a manipulation. $P_{S,p}$ is the probability that, after manipulation, the relationships of the DCT coefficients $\in \mathbf{B}_p$ of the block pair (p,q) do not violate Theorem 2 given different scenario. These scenario include: attacks with visual meaning changes, attacks based on the DCT values of the replaced block, attacks based on known mapping functions, attacks based on know signature generation positions, *etc.* This probability is used to indicate the chance of successful manipulation. Detailed discussion and derivation of these two probabilities are in [7].

In this paper, we only show a simpler measure of P_S in the case that attacks are based on pixel replacement (for changing visual meaning of content). This is the case that no information of the authenticator is known to the attacker. Here,

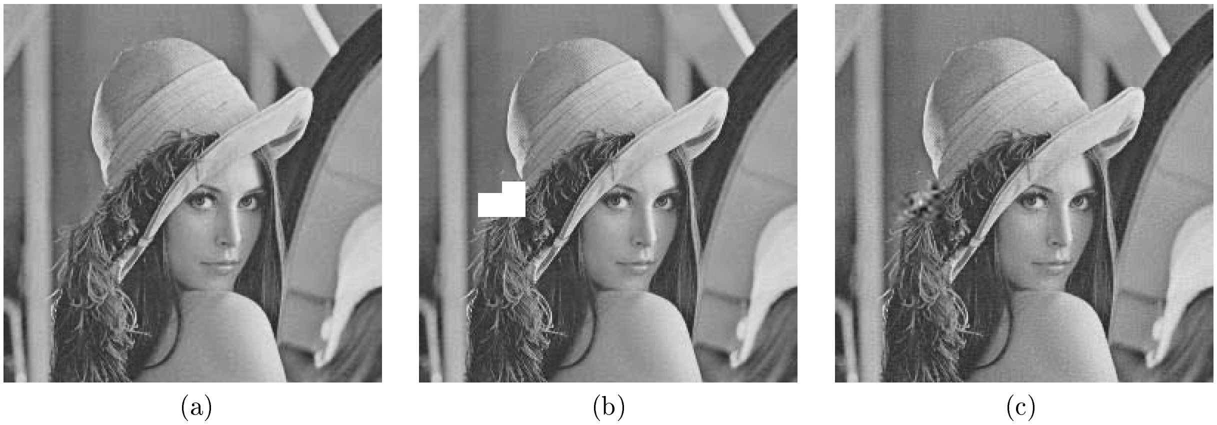
$$P_S \approx 2^{-1.5 \cdot \beta_a \cdot N} \tag{16}$$

where N is the number of 8×8 blocks that are affected by the attack. For instance, if each block is protected by $\beta_a = 6$, and $\frac{1}{2}\beta_a = 3$ authentication bits are embedded in this block, then the P_S that an attacker replace a block is approximately $\approx 2^{-9}$. In practical, because manipulation may cross the boundary, if an attacker replace an area of 20×20, which may affect 16 blocks, then $P_S \approx 2^{-9 \times 16} \approx 10^{-43}$. Practically, Eq. 16 is an optimistic estimation of the probability of successful attack, because the probability that a manipulated coefficient pair pass the authenticator may be larger than $\frac{1}{2}$, and is image dependent[7].

5 Experimental Results

We use the 256×256 gray-level "lenna" image to test our system. The original and watermarked images are shown in Figure 2. We use $\beta_a = 6$ authentication bits for each block pair, and set the acceptable JPEG quality factor to be 50. And, we use enhanced embedding method as in Eq. 8 and Eq. 9. In Figure 2(b), we can see that, the watermarked image looks the same as the original after embedding authentication bits. The PSNR of this image is lower than the expected value in Figure 1, because this enhanced method modifies more bits in each block. In Figure 2(c), we show the watermarked image with both authentication bits and weak recovery bits. For each block, in addition to the authentication bits, $\beta_r = 6$ recovery bits are embedded. These recovery bits survives JPEG compression to QF =75. There is visible distortion after embedding, but overall, the degradation is not obvious, and could be considered as acceptable. An experiment of embedding recovery bits that can survive QF=50 shows a noticeable quality degradation, with PSNR = 32.95 dB. Its quality degradation may be too much to be considered as acceptable.

We saved the watermarked image in the raw format, and then use XV on workstation and Adobe Photoshop on PC to compress them. These two commercial software use different methods to generate quantization tables in JPEG. XV uses the same quality factors suggested by JPEG. We found that our watermarked images can



(a) (b) (c)

Figure 3: (a) Manipulation on the watermarked image in Figure 2(b), (b) the authentication result of (a), (c) the authentication and recovery result from the manipulated image of Figure 2(c).

survive all the JPEG lossy compressions with $QF \geq 50$. Adobe Photoshop uses different scales of low, medium, high, and maximum to determine the quantization table. We found that our watermarked images can survive the last three levels, but introduce some false alarm after compression using the first level. The reason is that its quantization steps are larger than $\mathbf{Q_{50}}$. In practice, if we hope to survive all JPEG compression in Photoshop, we can use these quantization steps from Photoshop instead of $\mathbf{Q_{50}}$.

We manipulate the watermarked images using Photoshop. Two images are manipulated in a similar way by deleting the pin of lenna's hat. The image manipulated from Figure 2(b) is shown in Figure 3(a). After manipulation, the watermarked image are saved as JPEG files with the medium quality. Using the authenticator, we get the authentication results in Figure 3(b) and (c). We see that the manipulated area can be clearly identified in (b). And the manipulated areas can even be recovered approximately (shown in Figure 2(c)) if recovery bits are used.

6 Conclusion and Future Direction

In this paper, we present a novel semi-fragile watermarking technique that accepts JPEG lossy compression on the watermarked image to a pre-determined quality factor, and rejects unacceptable manipulations such as crop-and-replacement process. The embedded information includes authentication bits, which are used to identify the position of malicious attacks, and recovery bits, which are used to recover the corrupted blocks approximately. We base our techniques on two unique invariant properties of JPEG compression. Our techniques guarantee zero false alarm probability and achieve excellent performance in terms of miss probability. The experiment results verify the effectiveness of the proposed system. We will address the detailed performance analysis of this system and its robustness toward multiple recompression processes in our forthcoming report. Our future directions include: (1) including more general acceptable manipulations, (2) developing semi-fragile watermarks suitable for JPEG 2000/MPEG, and (3) using the proposed watermarking technique for general information hiding.

Appendix

Proof 1 *First, for any real coefficient $\mathbf{F}_p(\nu)$, if it is quantized with a quantization step $\mathbf{Q}(\nu)$, and the result after quantization is denoted as $\tilde{\mathbf{F}}_p(\nu) \equiv$ Integer Round$(\frac{\mathbf{F}_p(\nu)}{\mathbf{Q}(\nu)}) \cdot \mathbf{Q}(\nu)$, then the quantized coefficient will be in the following range,*

$$\mathbf{F}_p(\nu) - \frac{1}{2}\mathbf{Q}(\nu) \leq \tilde{\mathbf{F}}_\mathbf{p}(\nu) \leq \mathbf{F}_p(\nu) + \frac{1}{2}\mathbf{Q}(\nu). \tag{17}$$

Assume a real coefficient $\grave{\mathbf{F}}_p(\nu) = c \cdot \mathbf{Q}'_\mathbf{m}(\nu)$ where c is an integer and $\mathbf{Q}'_\mathbf{m}(\nu) > \mathbf{Q}(\nu)$. If the coefficient, $\grave{\mathbf{F}}_p(\nu)$, is further quantized (by JPEG compression) using a quantization step $\mathbf{Q}(\nu)$, then, from Eq. 17, the quantization *result, $\tilde{\mathbf{F}}'_p(\nu)$, will be,*

$$\grave{\mathbf{F}}_p(\nu) - \frac{1}{2}\mathbf{Q}(\nu) \le \tilde{\mathbf{F}}'_p(\nu) \le \grave{\mathbf{F}}_p(\nu) + \frac{1}{2}\mathbf{Q}(\nu). \tag{18}$$

Using the properties that $\mathbf{Q}'_{\mathbf{m}}(\nu) > \mathbf{Q}(\nu)$ and $\grave{\mathbf{F}}_p(\nu) = c \cdot \mathbf{Q}'_{\mathbf{m}}(\nu)$,

$$c \cdot \mathbf{Q}'_{\mathbf{m}}(\nu) - \frac{1}{2}\mathbf{Q}'_{\mathbf{m}}(\nu) < \tilde{\mathbf{F}}'_{\mathbf{p}}(\nu) < c \cdot \mathbf{Q}'_{\mathbf{m}}(\nu) + \frac{1}{2}\mathbf{Q}'_{\mathbf{m}}(\nu). \tag{19}$$

If we quantize $\tilde{\mathbf{F}}'_{\mathbf{p}}(\nu)$ again using $\mathbf{Q}'_{\mathbf{m}}(\nu)$, (ı.e., dividing all coefficients in Eq. 19 by $\mathbf{Q}'_{\mathbf{m}}(\nu)$ and then round them to integers), because all real coefficients in the range of ($c \cdot \mathbf{Q}'_{\mathbf{m}}(\nu) - \frac{1}{2}\mathbf{Q}'_{\mathbf{m}}(\nu)$, $c \cdot \mathbf{Q}'_{\mathbf{m}}(\nu) + \frac{1}{2}\mathbf{Q}'_{\mathbf{m}}(\nu)$) will be quantized to $c \cdot \mathbf{Q}'_{\mathbf{m}}(\nu)$, we can get

$$\textit{Integer Round}(\frac{\tilde{\mathbf{F}}'_p(\nu)}{\mathbf{Q}'_{\mathbf{m}}(\nu)}) \cdot \mathbf{Q}'_{\mathbf{m}}(\nu) = c \cdot \mathbf{Q}'_{\mathbf{m}}(\nu) = \grave{\mathbf{F}}_p(\nu), \tag{20}$$

which proves Theorem 1.

□

References

[1] M. Yeung and F. Mintzer, "An Invisible Watermarking Technique for Image Verification," *IEEE Proc. of ICIP*, Santa Barbara, Oct 1997.

[2] M. Wu and B. Liu, "Watermarking for Image Authentication," *IEEE Proc. of ICIP*, Chicago, Oct 1998.

[3] C.-Y. Lin, "Bibliography of Multimedia Authentication Research Papers," web page at *http : //www.ctr.columbia.edu/ ∼ cylin/auth/bibauth.html.*

[4] M. Schneider and S.-F. Chang, "A Robust Content Based Digital Signature for Image Authentication," *IEEE Proc. of ICIP*, Laussane, Switzerland, Oct 1996.

[5] S. Bhattacharjee and M. Kutter, "Compression Tolerant Image Authentication," *IEEE Proc. of ICIP*, Chicago, October 1998.

[6] M. P. Queluz, "Content-based Integrity Protection of Digital Images," *SPIE Conf. on Security and Watermarking of Multimedia Contents*, Vol. 3657, pp. 85-93, San Jose, January 1999.

[7] C.-Y. Lin and S.-F. Chang, "A Robust Image Authentication Method Distinguishing JPEG Compression from Malicious Manipulation," *CU/CTR Technical Report 486-97-19*, Dec 1997; *SPIE Storage and Retrieval of Image/Video Databases*, San Jose, Jan 1998.

[8] C.-Y. Lin and S.-F. Chang, "Issues and Solutions for Authenticating MPEG Video," *Proc. of SPIE Security and Watermarking of Multimedia Contents*, EI '99, San Jose, CA, Jan. 1999.

[9] B. Zhu, M. D. Swanson, and A. H. Tewfik, "Transparent Robust Authentication and Distortion Measurement Technique for Images," *The 7th IEEE Digital Signal Processing Workshop*, pp. 45-48, Sep 1996.

[10] R. B. Wolfgang and E. J. Delp, "A Watermark for Digital Images", *IEEE Proc. of ICIP*, Laussane, Switzerland, Oct 1996.

[11] J. Fridrich, "Image Watermarking for Tamper Detection," *IEEE International Conf. on Image Processing*, Chicago, October 1998.

[12] J. Fridrich, "Methods for Detecting Changes in Digital Images," *IEEE Workshop on Intelligent Signal Processing and Communication Systems*, Melbourne, Australia, November 1998.

[13] Independent JPEG Group's free JPEG software. http://www.ijg.org.

[14] http://www.faqs.org/faqs/jpeg-faq

[15] N. S. Jayant and P. Noll, "Digital Coding of Waveforms," *Prentice-Hall*, 1984.