# Watermarking and Digital Signature Techniques for Multimedia Authentication and Copyright Protection

Ching-Yung Lin

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

in the Graduate School of Arts and Sciences

Columbia University

2000

ABSTRACT

# Watermarking and Digital Signature Techniques for Multimedia Authentication and Copyright Protection

Ching-Yung Lin

Multimedia authentication techniques are required in order to ensure trustworthiness of multimedia data. They are usually designed based on two kinds of tools: digital signature or watermarking. Digital signature is a non-repudiatible, encrypted version of the message digest extracted from the data. Watermarking techniques consider multimedia data as a communication channel transmitting owner identification or content integrity information. Given an objective for multimedia authentication to reject the crop-and-replacement process and accept content-preserving manipulations, traditional digital signature or watermarking methods cannot be directly applied. In this thesis, we first propose robust digital signature methods that have proved to be useful for such types of content authentication. Also, we have developed a novel semi-fragile watermarking technique to embed the proposed robust digital signatures. We have implemented a unique Self-Authentication-and-Recovery Images (SARI) system, which can accept quantization-based lossy compression to a determined degree without any false alarms and can sensitively detect and locate malicious manipulations. Furthermore, the corrupted areas can be approximately recovered by the information hidden in the other part of the content. The amount of information embedded in our SARI system has nearly reached the theoretical maximum zero-error information hiding capacity of digital images.

Watermarking is a promising solution that can protect the copyright of multimedia data through transcoding. A reasonable expectation of applying watermarking

techniques for copyright protection is to consider specific application scenarios, because the distortion behavior involved in these cases (geometric distortion and pixel value distortion) could be reasonably predictable. We propose a practical public watermarking algorithm that is robust to rotation, scaling, and/or translation (RST) distortion. This proposed algorithm plays an important role in our design of the public watermarking technique which survives the image print-and-scan process.

In addition, we present our original work in analyzing the theoretical watermarking capacity bounds for digital images, based on the information theory and the characteristics of the human vision system. We investigate watermarking capacity in three directions: the zero-error capacity for public watermarking in magnitude-bounded noisy environments, the watermarking capacity based on domain-specific masking effects, and the watermarking capacity issues based on sophisticated Human Vision System models.

# Contents

# List of Figures

# List of Tables

## Acknowledgements

I would like to acknowledge the help and support of everyone in the ADVENT Lab who made my life more pleasant and my work more fun during my doctoral tenure at Columbia University. The tireless support and the invaluable intellectual inspiration provided by Prof. Shih-Fu Chang, my thesis advisor, cannot be emphasized enough. I particularly appreciate his willingness to share his seemingly endless supply of knowledge and his endeavor to improve every single word in our papers. He is my mentor in various aspects, from my research and career, to my daily life. I cannot express more my appreciation of all the things he has done for me.

I would also like to thank Prof. Dimitris Anastassiou for helping me and guiding me during the initial stages of my Ph.D. study at Columbia and for chairing my thesis defense committee. I particularly thank Prof. Alexandros Eleftheriadis and Prof. John Pavlik for guiding me in both my oral qualify committee and thesis defense committee. With their insights and suggestions, they made this research become more interesting and more fun. I also would like to thank all the other committee members for their time, efforts, signatures that entitle me to call myself a doctor! Special thanks are extended to Dr. Ingemar Cox, my advisor during my summer stay at NEC Research Institute. I have personally benefited immensely from his wide expertise.

I also take pleasure in thanking Dr. Jeffrey Bloom, Dr. Matt Miller and Yui Man Lui, for their support and brainstorming on our project in NEC. I am also grateful for the invaluable discussion with my friends, Dr. Heather Yu at Panasonic and Dr. Jana Dittmann at GMD. I would like to thank Prof. Nasir Memon at Polytech Univ. for his encouragement which made realization of our SARI system. Also, I personally appreciate my master thesis advisor, Prof. Soo-Chang Pei, for his willingness to support and guide me even after my departure.

During the development of this thesis I also had the benefit of interacting with a number of colleagues and friends. Special thanks go to my officemates Huitao Luo and Mei Shi with whom we shared intellectual ideas and enjoyed numerous pleasant moments. I would like to thank Lexing Xie and Dr. Qibin Sun for their help in benchmarking our authentication system that made my graduation faster. I specially thank Sara Brook for proofreading most of our papers. And I also thank to Nicole Luke and Hari Sundaram for their valuable comments on my thesis and presentations. In particular, I thank Daby Sow, with whom I had invaluable discussions during our drive to IBM. In my tenure here at Columbia, I would specially thank Wei Shi, Mei Shi, and their family for their warm hearts in providing me with much-needed relief and family-like life in New York. Also, I would like to thank Kuang-Yu Chin for her support for years.

Last but not least, there is no way I could acknowledge enough the support from my family. I especially thank my parents for everything. They are and will always be the driving force that helps me pursuing this long term dream and all the future ones. My sisters' support also helped me tremendously. Thank you very much. Thank you all!!

To my parents

# Chapter 1

# Introduction

This thesis addresses two closely related problems – multimedia authentication and copyright protection. We also examine the important issue regarding the maximum amount of watermark information without causing noticeable perceptual degradation.

The well-known adage that "seeing is believing" is no longer true due to the pervasive and powerful multimedia manipulation tools. Such development has decreased the credibility that multimedia data such as photos, video or audio clips, printed documents, *etc.* used to command. To ensure trustworthiness, multimedia authentication techniques are being developed to protect multimedia data by verifying the information integrity, the alleged source of data, and the reality of data. This distinguishes from other generic message authentication in its 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 are usually designed based on two kinds of tools: *digital signature* or *watermarking*. Digital signature is a non-repudiatible, encrypted version of the message digest extracted from the data. It is usually stored

as a separate file, which can be attached to the data to prove integrity and originality. Watermarking techniques consider multimedia data as a communication channel. The embedded watermark, usually imperceptible, may contain either a specific producer ID or some content-related codes that are used for authentication. Given the objective for multimedia authentication to reject the crop-and-replacement process and accept content-preserving or imperceptible manipulations, traditional digital signature or watermarking method cannot be directly applied to authentication. Traditional digital signature does not allow even a single bit change in the data. On the other hand, traditional watermarking techniques are designed for surviving all kinds of manipulations that may miss a lot of content-altering manipulations. Therefore, there is a need for designing novel robust digital signature or semi-fragile watermarks for multimedia authentication.

**Watermarking** has been considered to be a promising solution that can protect the copyright of multimedia data through transcoding, because the embedded message is always included in the data. However, today, there is no evidence that watermarking techniques can achieve the ultimate goal to retrieve the right owner information from the received data after all kinds of content-preserving manipulations. Because of the fidelity constraint, watermarks can only be embedded in a limited space in the multimedia data. There is always a biased advantage for the attacker whose target is only to get rid of the watermarks by exploiting various manipulations in the finite watermarking embedding space. A more reasonable expectation of applying watermarking techniques for copyright protection may be to consider specific application scenarios. For instance, the print-and-scan (PS) process is commonly used for image reproduction and distribution. It is popular to transform images between the electronic digital format and the printed format. The rescanned image may look similar to the original, but may have been distorted

during the process. For copyright protection applications, users should be able to detect the embedded watermark even if it is printed-and-scanned. Since the distortion behavior involved in this case (geometric distortion and pixel value distortion) is reasonably predictable, we can design useful watermarking techniques which survive such processes.

In addition, we study the theoretic issue with regard to watermarking embedding space existing in multimedia data. This space should depend on the properties of human audio-visual system. It is a complex scientific question that we may not be able to find a thorough answer in this thesis. Our objective is to study existing human vision system models, achieve better understanding of various watermarking space, and then develop information-theoretic estimation of information capacity via watermark. Better understanding of capacity and embedding space will contribute to future development of watermarking techniques.

This chapter is organized as follows. First, we will discuss the meaning, classification, and requirements of multimedia authentication, as well as priori works in this field. Then, we will introduce issues related to geometric distortion resilient public watermarking. After that, we will discuss theoretical watermarking capacity issues. Finally, we will present the structure of this thesis.

## 1.1 Multimedia Authentication

Authenticity, by definition, means something "as being in accordance with fact, as being true in substance", or "as being what it professes in origin or authorship, as being genuine [101]." A third definition of authenticity is to prove that something is "actually coming from the alleged source or origin [136]." For instance, in the courtroom, insurance company, hospital, newspaper, magazine, or television news, when we watch/hear a clip of multimedia data, we hope to know whether

the image/video/audio is *authentic*. For electronic commerce, once a buyer purchases multimedia data from the Internet, she needs to know whether it comes *from the alleged producer* and she must be assured that *no one has tampered with the content*. The credibility of multimedia data is expected for the purpose of being *electronic evidence* or a *certified product*. In practice, different requirements affect the methodologies and designs of possible solutions [10].

In contrast with traditional sources whose authenticity can be established from many physical clues, multimedia data in electronic forms (digital or analog) can only be authenticated by non-physical clues. One approach, called *blind authentication*, is to examine the characteristics of content for inferencing authorship and the continuity of content for detecting forgery. This method is widely used in traditional authentication techniques, but it is still under development for multimedia application. Another practical solution is the *digital signature* method introduced by Diffie and Hellman in 1976 [36]. The digital signature shall depend on the content and some secret information only known to the signer [139]. Therefore, the digital signature cannot be forged, and the authenticator can verify multimedia data by examining whether its content matches the information contained in the digital signature. In other words, we trust the signer as well as her digital signature to verify the data integrity.

Machines can be given the role of signer. This approach has been used byt Friedman in his work on the "trustworthy" camera in 1993 [45]. By embedding an encryption chip in the camera, the camera endorses its captured pictures and generates content-dependent digital signatures.

We should note that no matter how the authentication algorithm is designed, trustworthiness of the signer will be always of concern. In the traditional research of message authentication, the signer is usually the one who generates and distributes

the message. However, multimedia data are usually distributed and re-interpreted by many interim entities (*e.g.*, editors, agents). Because of this, it becomes important to guarantee end-to-end trustworthiness between the origin source and the final recipient. That can be achieved by the robust digital signature method that we have proposed [112, 73, 75].

Although the word "authentication" has three broad meanings: the integrity of data, the alleged source of data, and the reality of data, we primarily refer to the first in this thesis. We use the word "copyright protection" to indicate the second meaning: alleged source. The third meaning, the reality of data, may be addressed by using a mechanism linking the information of alleged source to real-world capturing apparatus such as a digital camera.

### 1.1.1 Multimedia Authentication Objectives: Complete Authentication v.s. Content Authentication

Based on the objectives of authentication, multimedia authentication techniques can be classified into two categories: *complete authentication* and *content authentication*. *Complete authentication* refers to techniques that consider the whole piece of multimedia data and do not allow any manipulations or transformation [143, 145]. Early works of multimedia authentication were mostly in this category. Because the non-manipulable data are like 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 message bits will not match the information in the digital signature. In practice, fragile watermarks or traditional digital signatures may be used for complete authentication.

Content Authentication refers to a different objective that is unique for multime-

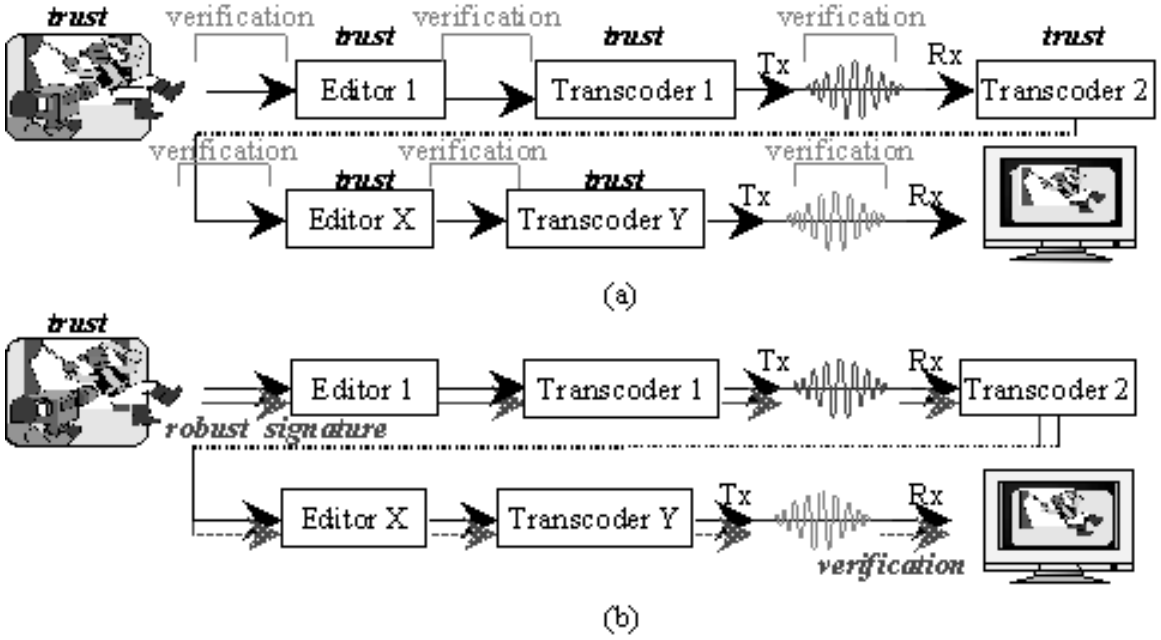



Figure 1-1: (a) Complete Authentication: multimedia data have to be examined in each transmission, and each intermediate stage must be trustworthy; (b) Content Authentication: multimedia data are endorsed by the producer and verified only in the last stage.

dia data. The meaning of multimedia data is based on their content instead of the bitstreams. In some applications, manipulations on the bitstreams without changing the meaning of content are considered as acceptable. Compression is an example. Today, most digital multimedia data are stored or distributed in compressed forms. To satisfy various needs of broadcasting, storage and transmission, transcoding of compressed digital videos may also be required. For instance, digital video clips are usually shot and stored in the compressed format with a pre-determined bitrate, but distributed with a different bitrate in transmission. Transcoding processes change the pixel values of the digital video but not its content. Therefore, videos that are obtained by transcoding the original one should be considered as authentic.

Figure 1-1 shows the benefit of the Multimedia Content Authentication (MCA). It represents the complete process of multimedia data, from being produced to being consumed. With complete verification, we have to verify the data at every transmission stage and trust all the interim entities. However, with content verification, we can transmit the robust signature with the data and only verify it at the last stage. Therefore, we do not need to verify the data at each stage and question the trustworthiness of the intermediate people. This enhances the authenticity of the data. As in Figure 1-1, if the producer is a trustworthy camera, it can somehow provide credibility of reality to the data, *i.e.*, proving that the multimedia data are "real." This is especially useful for those multimedia data that are needed as electronic evidence.

A broad meaning of *content authentication* is to authenticate multimedia content on 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 unaccept-

able to others. A content authentication technique may choose only to authenticate the altered multimedia data if manipulation is imperceptible. One example is the authentication techniques that accept lossy compression up to an allowable level of quality loss and reject other manipulations. We designed our system to "perfectly" accept lossy compression and "statistically" accept other manipulations.

### 1.1.2 Multimedia Authentication Sources: Raw Data v.s. Compressed Data

Multimedia compression standards have been designed and widely adopted by various applications: JPEG in the WWW, MPEG-1 in VCD, MPEG-2 format in DVD, and H.261 and H.263 in video conferencing. The source of a multimedia authentication system may be raw data or compressed data. In practical applications, the raw format of multimedia data may not be available. For instance, a scanner generates temporary raw images but only saves them in their compressed format; a digital camera which captures image/video produces compressed files only, without generating any raw data. Therefore, an authentication system which can only authenticate raw data may have limited uses in practice. However, exceptions exist in (1) non-standard data such as 3D objects, and (2) medical images which usually do not tolerate lossy compression.

### 1.1.3 Multimedia Authentication Methods: Watermarking v.s. Digital Signature

Since the meaning of multimedia data is based on its content, we can modify the multimedia bitstream to embed some codes, *i.e.*, watermarks, without changing the meaning of the content. The embedded watermark may represent either a specific digital producer identification label (PIL) or some content-based codes generated

| *Researcher* | *Method* | | *Objective* | | *Source* | |
|---|---|---|---|---|---|---|
| | DS | WMK | CPA | CTA | RD | CD |
| Friedman[45] | X | | X | | X | X |
| Van Schyndel *et. al.*[127] | | X | X | | X | |
| Walton[130] | | X | X | | X | |
| Wolfgang and Delp[140] | | X | X | X | X | |
| Zhu *et. al.*[149] | | X | | X | X | |
| Schneider and Chang[112] | X | | | X | X | X |
| Yeung and Mintzer[145] | | X | X | | X | |
| Lin and Chang[73, 72] | X | | | X | X | X |

Table 1.1: Previous Research Work (DS: Digital Signature, WMK: Watermark, CPA: Complete Authentication, CTA: Content Authentication, RD: Raw Data, CD: Compressed Data)

by applying a specific rule. In the authenticator, the watermarks are examined to verify the integrity of the data.

For complete authentication of uncompressed raw multimedia data, watermarking may work better than digital signature methods because:

- the watermarks are always associated with the data and can be conveniently examined, and
- there are many spaces in the multimedia data in which to embed the watermarks with negligible quality degradation (known as invisible watermarks).

Previous works in [127, 130, 145] have shown effective watermarking methods for these applications.

However, there is no advantage to using the watermarking method in compressed multimedia data for complete verification. Compression standards, *e.g.*, MPEG or JPEG, have user-defined sections where a digital signature can be placed. Because multimedia data are stored or distributed in specific file format instead of pixel values, the digital signature can be considered as being "embedded" in the data. Once the multimedia data is modified, the user-defined section of the original data is usually discarded by the editing software. Even if the digital signature can be

preserved by the software, we can easily detect the modification, since the hash values of the modified data will not be the same as the original. Moreover, compressed multimedia data offer less space for hiding watermarks. Visual quality of the data may be compromised in order to ensure that enough watermarking bits for adequately protecting the data.

For content authentication, compression should be distinguished from other manipulations. Previous watermarks are either too fragile for compression or too flexible to detect malicious manipulations. The performance of an authenticator should be simultaneously evaluated by two parameters: the probability of false alarm and the probability of missing manipulations. Fragile watermarks, which have low probability of miss, usually fail to survive compressions such that their probability of false alarm is very high. Previous researchers have attempted to modify the fragile watermark to make it more robust with compression [149, 140]. However, such modifications failed to distinguish compression and tampering. When they lower the probability of false alarm, the probability of miss in their systems increases significantly. On the other hand, robust watermarks are robust to most manipulations, but are usually too robust to detect malicious manipulations. Their probability of miss is usually too high. These drawbacks motivated our design of novel semi-fragile watermarks we proposed in [84], which will be shown in Chapter 3. Our proposed Self-Authentication-and-Recovery Images (SARI) system can distinguish quantization-based lossy compressions from malicious manipulations.

Digital signatures can be stored in two different ways. If the header of the compressed source data remains intact through all processing stages, then the digital signature can be saved in the header. Otherwise, it can be stored as an independent file. Anyone who needs to authenticate the received multimedia data has to request the source to provide the signature. This may be inconvenient in some cases and

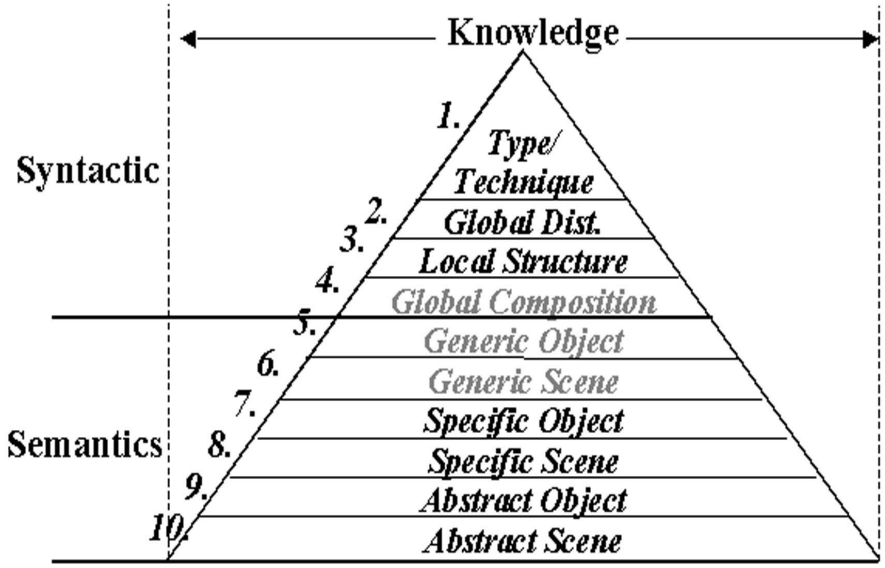



Figure 1-2: A multi-level conceptual framework for indexing multimedia information (A. Jaimes and S.-F. Chang [56])

considered as a drawback. But, since the digital signatures remain unchanged when the pixel values of the images/videos are changed, they provide a better prospect for achieving robustness. Currently, the robust digital signature methods we proposed in [72] (shown in Chapter 2) have proved to be useful for content authentication. Because our techniques were based on the characteristics of DCT-based compression standards, they can distinguish JPEG/MPEG compression from other manipulations.

There are still no reliable content authentication techniques. The reason may be the ambiguity of content meaning. "Content" can indicate several different meanings of multimedia data. Figure 1-2 represents several layers of content description [56]. Among them, only the first three layers in the syntax level may be automatically described by machines. The remaining layers need more human knowledge and manual efforts. Even when using only the top three syntactic layers, it is still an open issue in extracting reliable invariant properties from multimedia data. All

of these demonstrate the difficulty in finding a content authentication technique. Therefore, it is not our goal in this thesis in developing such techniques.

In Table 1.1, we compare different previous authentication techniques based on the type of methods used, the objective, and the source data being authenticated.

### 1.1.4 Requirements of Multimedia Authentication System

An authentication system should be evaluated based on 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: Authentication had better be conducted directly from the received content. Watermarks have better portability than digital signatures.
- Location of manipulated area: The authenticator should be able to detect location of altered areas, and verify other areas as authentic.
- Recovery capability: The authenticator may need the ability to recover the lost content in the manipulated areas (at least approximately).

These are the essential requirements of an "ideal" authenticator. Our proposed semi-fragile watermarking system, *i.e.* SARI, in Chapter 3, has achieved most of these six requirements. The only exception is that the system can totally accept

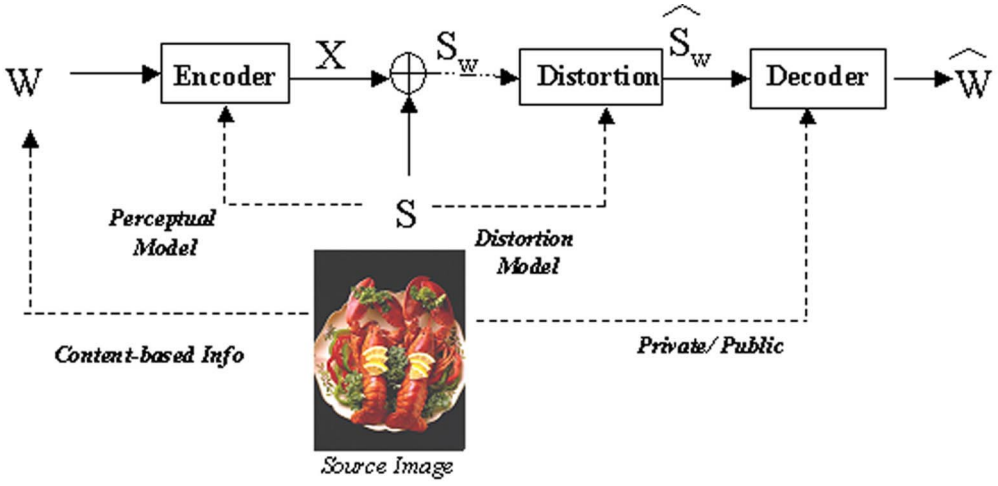



Figure 1-3: Watermarking: multimedia data as a communication channel

DCT-based JPEG compression without error, but only statistically accept other content-preserving manipulations.

## 1.2 Models of Watermarking

In this section, we discuss a general watermarking model (shown in Figure 1-3) as a reference for later discussions. Watermarking techniques consider multimedia data as a communication channel. Users decode messages based on the received data, which may have been distorted. Here, a message, $W$, is encoded to $X$ which is added to the source multimedia data, $S$. The encoding process may apply some perceptual model of $S$ to control the formation of the watermark codeword $X$. No matter what kind of method is used in encoding, the resulted watermarked image, $S_W$, can always be considered as a summation of the source image and a watermark $X$. At the receiver end, this watermarked image may have suffered from some distortions, *e.g.*, additive noise, geometric distortion, nonlinear magnitude distortion, *etc.* The received watermarked image, $\hat{S}_W$, is then served as the input of the decoder to get the reconstructed message, $\hat{W}$. In general, we call the watermarking method

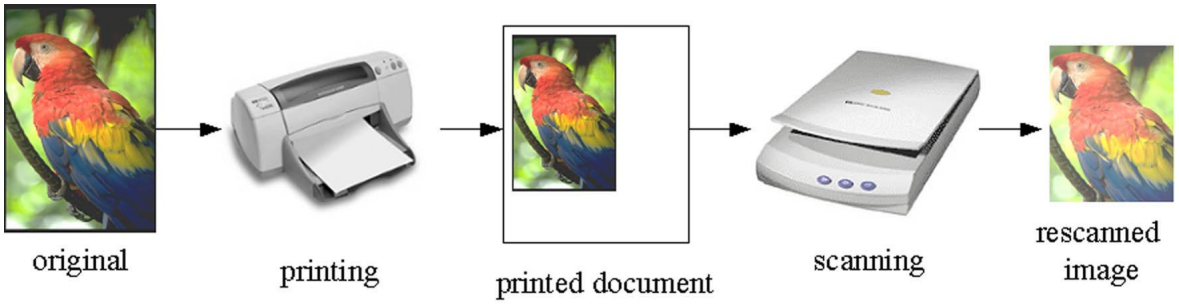



Figure 1-4: Digital image print-and-scan process. Distortion may include geometric distortion and pixel value distortion.

"private" if the decoder needs the original source image $S$, and "public" or "blind" if $S$ is not required in the decoding process. We should note that the message $W$ may be content-dependent over some applications such as authentication and recovery. The distortion models may be also application-dependent. In the literature, sometimes, there is confusion in the definition of the term "watermark". While "watermarking" indicates a secure information hiding technique, "watermark" may indicate the message $W$ in some papers and the codeword $X$ in others. In this thesis, we use the latter definition because we usually refer to a watermark as invisible, which is a property of $X$. Watermarking capacity refers to the amount of message bits in $W$ that can be transmitted in such processes.

## 1.3 Geometric Distortion Resilient Public Watermarking for Copyright Protection

Figure 1-4 represents an example of digital image print-and-scan process. Distortion occurs in both the pixel values and the geometric boundary of the rescanned image. The distortion of pixel values is caused by (1) the luminance, contrast, gamma correction and chromnance variations, and (2) the blurring of adjacent pixels. These are typical effects of the printer and scanner, and cause perceptible visual

quality changes to a rescanned image.

To design a watermarking scheme that can survive geometric distortion as well as pixel value distortion is important. There has been much emphasis on the robustness of watermarks to pixel value distortions such as compression and signal filtering. However, recently it has become clear that even very small geometric distortions may break the watermarking method [105, 126]. This problem is most severe when the original unwatermarked image is unavailable to the detector. Conversely, if the original image is available to the detector, then the watermarked image can often be registered to the original and the geometric distortion thereby inverted.[1] However, public watermarking requires that detection of the watermark be performed without access to the original unwatermarked image. As such, it is not possible to invert the geometric distortion based on registration of the watermarked and original images.

Before proceeding further, it is important to define what we mean by geometric distortions (*e.g.*, rotation, scale and translation). Specifically, we are interested in the situation in which a watermarked image undergoes an *unknown* rotation, scale and/or translation prior to the detection of the watermark. The detector should detect the watermark if it is present. This definition is somewhat obvious, so it may be more useful to describe what we are not interested in. In particular, some watermark algorithms claim robustness to scale changes by first embedding a watermark at a canonical scale, then changing the size of the image and finally, at the detector, scaling the image back to the canonical size prior to correlation. In our opinion, the detector does not see a scale change. Rather, the process is more closely approximated by a low pass filtering operation that occurs when the image is reduced in size. Similarly, tests that rotate an image by some number of degrees

[1] Although the original Cox *et al* [23] algorithm did not include this step, subsequent commercial implementations did so. More recently, Johnson *et al* [59] observed that it is not necessary to retain the entire image, a sufficiently small set of key points will suffice.

and subsequently rotate the image by the same amount in the opposite direction are not adequate tests of robustness to rotation. The same is true for translation.

The common situation we are concerned with occurs when a watermarked image is printed and then cropped or padded and scanned back into the digital domain. In these circumstances, the image dimensions have changed both because of cropping and possibly scaling. There is also likely to be an associated translational shift. In this example, scaling to a canonical size does not undo the scaling. Rather, if the cropping is not symmetric in both the rows and columns, then scaling to a canonical size will result in a change in the image's aspect ratio. We have discussed this and proposed a solution in [80], which will not be addressed in this thesis. We discussed applying the proposed watermarking method of the print and scan process in [81], which will be shown in Chapter 4.

One strategy for detecting watermarks after geometric distortion is to try to identify what the distortions were and invert them before applying the watermark detector. This can be accomplished by embedding a registration pattern along with the watermark [103, 30].

One problem with this solution is that it requires the insertion and detection of two watermarks, one for registration and one to carry the data payload. This approach is more likely to reduce the image fidelity. A second problem arises because all images watermarked with this method will share a common registration watermark. This fact may improve collusion attempts to discern the registration pattern and, once found, the registration pattern could be removed from all watermarked images thus restricting the invertibility of any geometric distortions.

Another way to implement the above strategy is to give the watermark a recognizable structure. For example, as suggested in [68], the watermark might be embedded multiple times in the image at different spatial locations. The autocorre-

lation function of a watermarked image yields a pattern of peaks corresponding to the embedded locations. Changes in this pattern of peaks can be used to describe any affine distortions to which the watermarked image has been subject. This method has significant potential, but, similar to the above methods, has two failure modes. For successful detection both the identification of the geometric distortion and the detection of the watermark after inversion of that distortion must be successful. Both of these processes must be robust and resistant to tampering. Second, spurious autocorrelation peaks may occur as a result of JPEG/MPEG compression artifacts introduced after embedding but prior to detection. Further experiments are needed to validate this approach.

It becomes clear that a public watermarking scheme that can survive geometric distortion as well as pixel value distortion by the watermark itself is an advantage in real applications. We will propose such a novel technique in Chapter 4.

## 1.4 Theoretical Watermarking Capacity Issues

Regardless of security issues, watermarking capacity is determined by invisibility and robustness requirements. A conceptual description is shown in Figure 1-5. There are three dimensions in this figure. If one parameter is determined, the other two parameters are inverse-proportional. For instance, a specific application may determine how many message bits are needed, copyright protection may need to embed about 10 bytes and authentication may need anywhere from 100-1000 bytes for a $256 \times 256$ image. After the embedded amount is decided, there always exists a trade-off between visual quality and robustness. Robustness refers to the extraction of embedded bits with an error probability equal to or approaching zero. Visual quality represents the quality of watermarked image. In general, if we want to make our message bits more robust against attacks, then a longer codeword or larger

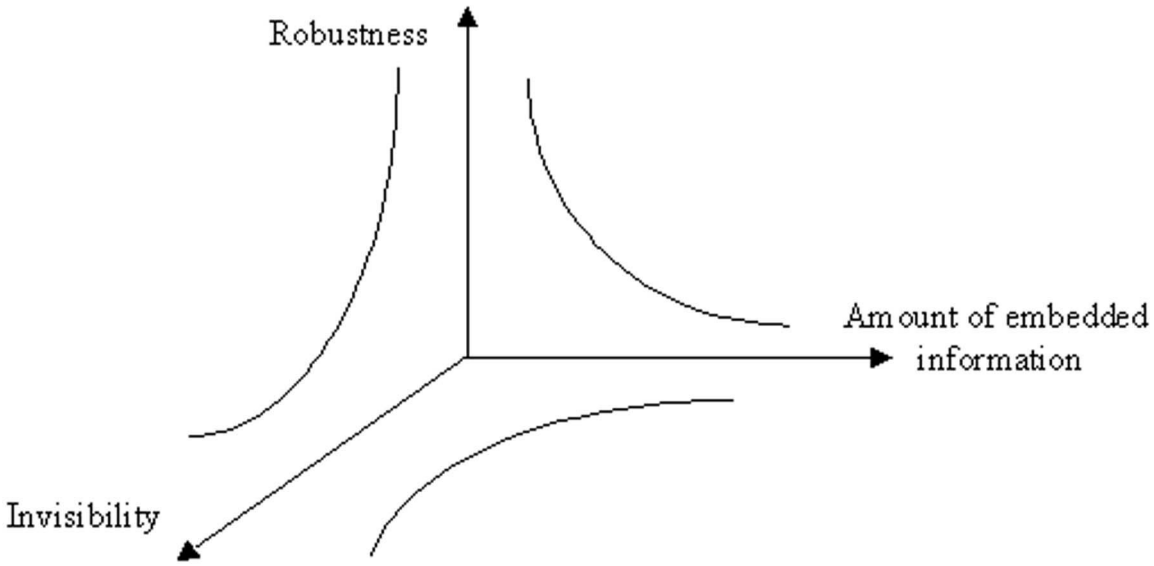



Figure 1-5: Three parameters of watermarking: information quantity of embedded message, invisibility, and robustness

codeword amplitudes will be necessary to provide better error-resistence. However, visual quality degradation can be expected. Another scenario may be that with a default visual quality, there exists a trade-off between the information quantity of embedded message and robustness. For instance, the fewer the message bits are embedded, the more redundant the codeword can be. Therefore, the codeword has better error correction capability against noises. What we show in Figure 1-5 is only a concept. It is our objective to theoretically draw the curves.

Theoretical capacity issues of digital watermarking have not been fully understood. Most of the previous works on watermarking capacity [6, 109, 114] directly apply information-theoretic channel capacity bounds without considering the properties of multimedia data. Shannon's well-known channel capacity bound,

$$C = \frac{1}{2} log_2(1 + \frac{P}{N}), \tag{1.1}$$

is a theoretic capacity bound of an analog-value time-discrete communication chan-

nel in a static transmission environment, *i.e.*, where the (codeword) signal power constraint, $P$, and the noise power constraint, $N$, are constants [116]. Transmitting message rate at this bound, the probability of decoding error can approach zero if the length of codeword approaches infinite, which implies that infinite transmission samples are expected.

Considering multimedia data, we found there are difficulties if we directly apply Eq. (1.1). The first is the number of channels. If the whole image is a channel, then this is not a static transmission environment because the signal power constraints are not uniform throughout the pixels, based on the human vision properties. If the image is a composition of parallel channels, then this capacity is meaningless because there is only one or few sample(s) in each channel. The second difficulty is the issue of digitized values in the multimedia data. Contrary to floating point values which have infinite states, integer value has only finite states. This makes a difference in both the applicable embedding watermark values and the effect of noises. The third obstacle is that we will not know how large the watermark signals can be without an extensive study of human vision system models, which is usually ignored in most previous watermarking researches, perhaps because of its difficulties and complexity. The fourth hurdle is that of noise modeling. Despite the existence of various distortion/attack, we think that additive noises might be the easiest modeling case. Other distortions may be modeled as additive noises if the distorted image can be synchronized/registered. There are other issues such as private or public watermarking and questions as to whether noise magnitudes are bounded. For instance, Eq. (1.1) is a capacity bound derived for Gaussian noises and is an upper bound for all kinds of additive noises. However, in an environment with finite states and bounded-magnitude noises, transmission error can actually be zero, instead of approaching zero as in Eq. (1.1). This motivated a research of zero-error

capacity initialed by Shannon in 1956 [117]. Quantization, if an upper bound on the quantization step exists, is an example of such a noise. We can find the zero-error capacity of a digital image if quantization is the only source of distortion such as in JPEG. These difficulties motivated our work in Chapter 5. Avoiding to directly apply Shannon's channel capacity theory, we try to find the theoretical watermarking capacity based on the properties of digital images discussed above.

## 1.5 Structure of Thesis

In Chapter 2, we present an effective technique for image authentication which can prevent malicious manipulations but allow JPEG lossy compression. The authentication signature is based on the invariance of the relationships between DCT coefficients at the same position in separate blocks of an image. These relationships are preserved when DCT coefficients are quantized in JPEG compression. Our proposed method can distinguish malicious manipulations from JPEG lossy compression regardless of the compression ratio or the number of compression iterations. We describe adaptive methods with probabilistic guarantee to handle distortions introduced by various acceptable manipulations such as integer rounding, image filtering, image enhancement, or scaling-rescaling. We also present theoretical and experimental results to demonstrate the effectiveness of the technique. The invariant properties proposed in this chapter can achieve negligible probability of false alarm as well as a very low probability of miss, and may be the best solution in extracting invariants which survive lossy compression.

In addition, we will describe extension of such techniques to authenticate MPEG compressed video. We first discuss issues of authenticating MPEG videos under various transcoding situations, including dynamic rate shaping, requantization, frame type conversion, and re-encoding. In the second part of this chapter, we propose

a robust video authentication system which accepts some MPEG transcoding processes but is able to detect malicious manipulations. It is based on unique invariant properties of the transcoding processes. Digital signature techniques as well as public key methods are used in our robust video authentication system.

In Chapter 3, we propose a Self-Authentication-and-Recovery Images (SARI) system which utilizes our novel semi-fragile watermarking technique for authentication. This technique can accept 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 an approximation of the original ones. SARI has achieved the six multimedia authentication system requirements illustrated in Section 1.1.4. 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 or equal to the steps used in later JPEG compressions, then this coefficient can be exactly reconstructed after later acceptable JPEG compression. We later proved in Chapter 5 that this property has almost explored the maximum zero-error capacity of digital image. The second property 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 original. In addition to authentication signatures, we can embed the recovery bits for recovering approximate pixel values in corrupted ar-

eas. 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 retrieved no false alarms, *i.e.*, no acceptable JPEG compression was rejected, and demonstrated great sensitivity in rejecting crop-and-replacement manipulations.

In Chapter 4, we propose a public watermarking algorithm that is robust to rotation, scaling, and/or translation (RST) distortion.[2] The watermark is embedded into a 1-dimensional signal obtained by first taking the Fourier transform of the image, resampling the Fourier magnitudes into log-polar coordinates, and then summing a function of those magnitudes along the log-radius axis. If the image is rotated, the resulting signal is cyclically shifted. If it is scaled, the signal is multiplied by some value. And if the image is translated, the signal is unaffected. We can therefore compensate for rotation with a simple search, and compensate for scaling by using the correlation coefficient as the detection metric. False positive results on a database of 10,000 images are reported. Robustness results on a database of 2,000 images are described. It is shown that the watermark is robust to rotation, scale and translation. In addition, we describe tests examining the watermark's resistance to cropping and JPEG compression.

After an image is printed-and-scanned, it is usually filtered, rotated, scaled, cropped, and contrast-and-luminance adjusted, as well as distorted by noises. Chapter 4 also presents models for the print-and-scan process, considering both pixel value distortion and geometric distortion. We show properties of the discretized, rescanned image in both the spatial and frequency domains, then further analyze the changes in the Discrete Fourier Transform (DFT) coefficients. Based on these

[2]Part of this chapter represents joint work with J. Bloom, M. Miller, I. Cox, M. Wu and Y. Lui [83].

properties, we show several techniques for extracting invariants from the original and rescanned image, with potential applications in image watermarking and authentication. These invariants helped design the watermarking algorithm proposed in this chapter to survive print-and-scan process. Preliminary experiments show the validity of the proposed model and the robustness of the watermark.

In Chapter 5, we address the following important question: how much information can be reliably transmitted as watermarks without causing noticeable quality losses, while being robust to some distortions on the watermarked images? Our objective is to find theoretical watermarking capacity bounds of digital images based on the information theory and the characteristics of the human vision system. We investigate watermarking capacity in three directions. First, the zero-error capacity for public watermarking in magnitude-bounded noisy environment. In an environment with finite states and bounded-magnitude noises, transmission error can actually be zero, instead of stochastically approaching zero as usually addressed in Shannon's capacity theory. We find the zero-error capacity of a digital image if quantization is the only source of distortion such as in JPEG. We consider that signal, watermark and noise are all discrete values, as in real representation of digital images. Second, we study the watermarking capacity based on domain-specific masking effects. We show the capacity of private watermarking in which the power constraints are not uniform in different samples. Then, we apply domain-specific HVS models to estimate the constraints in images and show the theoretical watermarking capacity of an image in the general noisy environments. Third, we study the watermarking capacity issues based on Human Vision System models. We study in details the Human Vision System developed by Daly and Lubin, and then show the difficulties and the necessity of future work in order to use them to estimate watermarking capacity.

In Chapter 6, we present the conclusion of our thesis and describe several future research issues in this field.

### 1.5.1 Original Contributions of Thesis

The original contributions of this thesis include the following:

- Invariant feature codes of image or video that can distinguish DCT-based lossy compression from malicious crop-and-replacement manipulations. These codes can achieve no false alarm in DCT-based lossy compressions and have great sensitivity in detecting malicious manipulations. (Chapter 2)

- Robust digital signatures for authenticating MPEG video through various transcoding and editing. (Chapter 2)

- Unique Self-Authentication-and-Recovery Images (SARI) system based on semi-fragile watermarking (http://www.ctr.columbia.edu/sari). SARI authenticator can authenticate the watermarked images after DCT-based lossy compressions, detect and locate malicious manipulations, and recover an approximation of the original on the corrupted areas. (Chapter 3)

- Novel public watermarking that is robust to affine transformation and pixel value distortion. We model the image print-and-scan process and develop a public watermarking scheme which survives this process (Chapter 4).

- Zero-Error watermarking capacity of digital images in a magnitude-constrained noisy environment (Chapter 5).

- Private watermarking capacity for digital images in a power-constrained noisy environment based on the characteristics of the Human Vision System (Chapter 5).

# Chapter 2

# Image/Video Authentication Using Robust Digital Signature

## 2.1 Introduction

Development of robust image authentication techniques becomes an important issue. If we consider a digital image to be merely an ordinary bitstream on which no modification is allowed, then there is not much difference between image authentication and other message authentication problems. Two methods have been suggested for achieving the authenticity of digital images: having a digital camera sign the image using a digital signature [45], or embedding a secret code in the image [130]. The first method uses an encrypted digital "signature" which is generated in the capturing devices. A digital signature is based on the method of Public Key Encryption citepublic, marc. A private key is used to encrypt a hashed version of the image. This encrypted message is called the "signature" of the image, and it provides a way to ensure that this signature cannot be forged. This signature then travels with the image. The authentication process of this image needs an associated public key to decrypt the signature. The image received for authentication is hashed and compared to the codes of the signature. If they match, then the

received image is authenticated. The second method embeds a "watermark" in an image [74, 130, 145]. The fragile watermark usually will be destroyed after manipulation. Authenticity is determined by examining the watermark extracted from the received image. Both the above methods have clear drawbacks. Authenticity will not be preserved unless every pixel of the images is unchanged. However, since lossy compression such as JPEG is often acceptable - or even desired - in practical applications, an authentication method needs to be able to distinguish lossy compression from malicious manipulations.

Manipulations on images can be considered in two ways: *method* and *purpose*. Manipulation methods include compression, format transformation, shifting, scaling, cropping, quantization, filtering, replacement, *etc*. The purpose of manipulations may be *transformation* or *attack*. The former are usually acceptable, and the latter, unacceptable. We list two kinds of transformation of representation below:

1. Format transformation and lossless compression. Disregarding the noise caused by the precision limitation during computation, pixel values are not changed after these manipulations. Therefore, we exclude these manipulations in the discussion in this paper.

2. Application-specific transformations. Some applications may require lossy compression in order to satisfy the resource constraints on bandwidth or storage. Some applications may also need to enhance the image quality, crop the image, change the size, or perform some other operations. A common aspect of these manipulations is that they change the pixel values, which results in different levels of visual distortion in the image. Usually, most of these operations try to minimize the visual distortion.

Attacks, or malicious manipulations, change the image to a new one which carries a different visual meaning to the observer. One typical example is replacing

some parts of the image with different content.

It is difficult for an authenticator to know the purpose of manipulation. A practical approach is to design an authenticator based on the manipulation method. In this paper, we design an authenticator which accepts format transformation, lossless compression, and the popular JPEG lossy compression. The authenticator rejects replacement manipulations because they are frequently used for attacks. Our authenticator does not aim to reject or accept, in absolute terms, other manipulation methods because the problem of whether they are acceptable depends on applications. But, if necessary, some manipulations can be clearly specified by users, such as shifting, cropping, or constant intensity enhancement. We will discuss this more rigorously later. The proposed authentication techniques have been extended and applied to MPEG video authentication as in [76] (also shown in Section 2.6 - Section 2.9).

For an image, there are some invariance properties which can be preserved during JPEG lossy compression. Let us consider the relationship between two DCT coefficients of the same position in two separate $8 \times 8$ blocks of an image. This relationship will hold even if these coefficients are quantized by an arbitrary quantization table in a JPEG compression process. In this paper, we will use this invariance property and propose a robust authentication method which can distinguish malicious manipulations from JPEG lossy compression.

A comprehensible list of multimedia authentication research papers can be found in [79]. Bhattacha and Kutter proposed an authentication method which extracts "salient" image feature points by using a scale interaction model and Mexican-Hat wavelets [11]. They generate a digital signature based on the locations of these feature points. The advantage of this technique is its compact signature length.

But, the selection process and relevance of the selected points are not clear. This technique may not be adequate for detecting some crop-and-replace manipulations inside the objects. Its robustness to lossy compression is also unclear. Queluz proposed techniques to generate digital signatures based on moments and edges [108]. Moment features ignore the spatial distribution of pixels. Images can be easily manipulated without changing their moments. Edge-based features may be a good choice for image authentication because the contour of objects should keep consistent for acceptable manipulations. However, several issues have to be further solved such as the reduction of signature length, the consistency of edge detector, and the robustness to color manipulations. Fridrich proposed a robust watermarking technique for authentication [42][43]. He divided images to $64 \times 64$ blocks. For each block, quasi-VQ codes are embedded by the spread spectrum method[23]. This technique is robust to manipulations. But, it cannot detect small area modification. The error between the extracted watermark and the reconstructed quasi-VQ codes is too large after JPEG compression[43]. Therefore, this technique would have difficulty distinguishing malicious manipulations from JPEG compressions.

This chapter is organized as follows. We briefly review the JPEG system in Section 2.2. In Section 2.3, a general system for authentication will be proposed. Also, we will describe how to control parameters for different practical uses. A simple example is shown in this section. We will present rigorous performance analysis in Section 2.4. Experimental results will be shown in Section 2.5. From Section 2.6 to Section 2.9, we will show how to design robust digital signature for video authentication. In Section 2.10, we will present conclusions and discuss future work.

## 2.2 Review of JPEG Lossy Compression

In this section, we briefly review the JPEG lossy compression standard. At the input to the JPEG[129] encoder, the source image, X, is grouped into $\wp$ nonoverlapping $8 \times 8$ blocks, $X = \bigcup_{p=1}^{\wp} \mathbf{X_p}$. Each block is sent sequentially to the Discrete Cosine Transform (DCT). Instead of representing each block, $\mathbf{X_p}$, as a $8 \times 8$ matrix, we can rewrite it as a $64 \times 1$ vector following the "zigzag" order[129]. Therefore, the DCT coefficients, $\mathbf{F_p}$, of the vector, $\mathbf{X_p}$, can be considered as a linear transformation of $\mathbf{X_p}$ with a $64 \times 64$ transformation matrix $\mathbf{D}$, *s.t.*,

$$\mathbf{F_p} = \mathbf{D}\mathbf{X_p}. \tag{2.1}$$

Each of the 64 DCT coefficients is uniformly quantized with a 64-element quantization table $\mathbf{Q}$. In JPEG, the same table is used on all blocks of an image. (For color images, there could be three quantization tables for YUV domains, respectively.) Quantization is defined as the division of each DCT coefficient by its corresponding quantizer step size, and rounding to the nearest integer:

$$\tilde{\mathbf{f}}_\mathbf{p}(\nu) \equiv \ Integer\ Round(\frac{\mathbf{F_p}(\nu)}{\mathbf{Q}(\nu)}), \tag{2.2}$$

where $\nu = 1...64$. In eq.(2.2), $\tilde{\mathbf{f}}_\mathbf{p}$ is the output of the quantizer. We define $\tilde{\mathbf{F}}_\mathbf{p}$, a quantized approximation of $\mathbf{F_p}$, as

$$\tilde{\mathbf{F}}_\mathbf{p}(\nu) \equiv \tilde{\mathbf{f}}_\mathbf{p}(\nu) \cdot \mathbf{Q}(\nu). \tag{2.3}$$

In addition to quantization, JPEG also includes scan order conversion, DC differential encoding, and entropy coding. Inverse DCT (IDCT) is used to convert $\tilde{\mathbf{F}}_\mathbf{p}$ to

the spatial-domain image block $\tilde{\mathbf{X}}_\mathbf{p}$.

$$\tilde{\mathbf{X}}_\mathbf{p} = \mathbf{D}^{-1}\tilde{\mathbf{F}}_\mathbf{p}. \tag{2.4}$$

All blocks are then tiled to form a decoded image frame.

Theoretically, the results of IDCT are real numbers. However, the brightness of an image is usually represented by an 8-bit integer from 0 to 255 and thus a rounding process mapping those real numbers to integers is necessary. We found that popular JPEG softwares such as PhotoShop, xv, *etc.* use the integer rounding functions in several steps of their DCT and IDCT operators in order to save computation or memory. The input and output of their DCT and IDCT operators are all integers. This approximation may not introduce too much visual distortion but may affect the authentication system performance that we will discuss in more detail in Section 2.4.

## 2.3 Authentication System

The proposed authentication method is shown in Figure 2-1. Our method uses a concept similar to that of the digital signature method proposed by Friedman[45], but their technique doesn't survive lossy compression. A signature and an image are generated at the same time. The signature is an encrypted form of the feature codes or hashes of the image. When a user needs to authenticate the image he receives, he should decrypt this signature and compare the feature codes (or hashed values) of this image to their corresponding values in the original signature. If they match, this image is said to be "authenticated." The most important difference between our method and Friedman's "trustworthy camera" is that we use invariance properties in JPEG lossy compression as robust feature codes instead of using hashes of raw

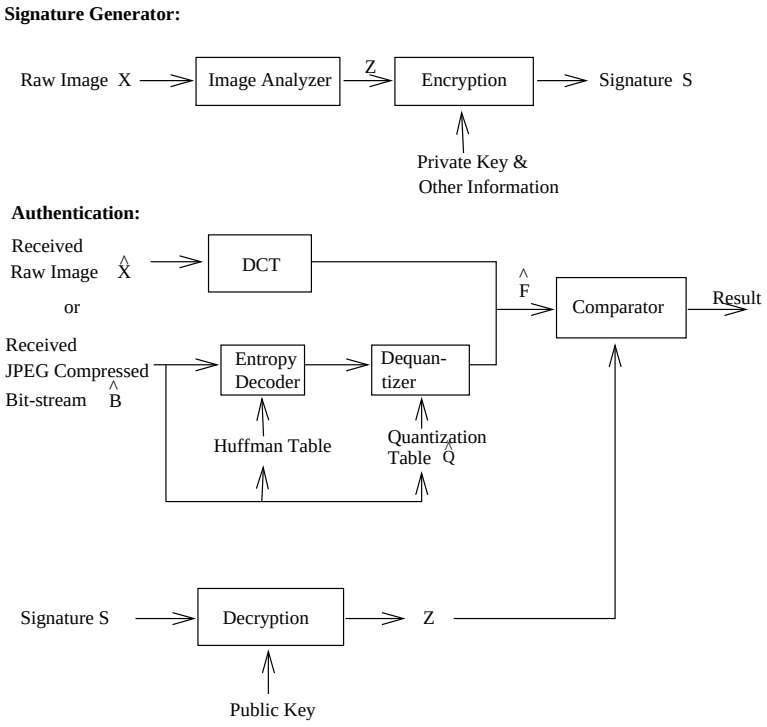



Figure 2-1: Signature Generator & Authentication Process

images.

### 2.3.1 Invariants of an image before and after JPEG compression

From the compression process of JPEG, we have found that some quantitative invariants and predictable properties can be extracted. Two steps in the JPEG compression process reduce the required bits representing an image: 1.) *quantization and rounding of the DCT coefficients*, and 2.) *entropy coding*. The second step is a lossless operation. The first step is a lossy operation which alters pixel values but keeps important visual characteristics of the image. Therefore, if robust feature codes are expected for authentication, they must survive this step. The following theorems provide a technical basis for generating such robust feature codes. Proofs of these theorems are included in the Section 2.11.

**Theorem 1** *Assume* $\mathbf{F_p}$ *and* $\mathbf{F_q}$ *are DCT coefficient vectors of two arbitrary* $8 \times 8$ *non-overlapping blocks of image X, and* $\mathbf{Q}$ *is the quantization table of JPEG lossy compression.* $\forall \nu \in [1, .., 64]$ *and* $\mathbf{p}, \mathbf{q} \in [1, .., \wp]$*, 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*$(\frac{\mathbf{F_p}(\nu)}{\mathbf{Q}(\nu)}) \cdot \mathbf{Q}(\nu)$*. Then, the following properties must be true:*

- *if* $\mathbf{\Delta F_{p,q}}(\nu) > 0$*, then* $\mathbf{\Delta \tilde{F}_{p,q}}(\nu) \geq 0$*,*
- *else if* $\mathbf{\Delta F_{p,q}}(\nu) < 0$*, then* $\mathbf{\Delta \tilde{F}_{p,q}}(\nu) \leq 0$*,*
- *else* $\mathbf{\Delta F_{p,q}}(\nu) = 0$*, then* $\mathbf{\Delta \tilde{F}_{p,q}}(\nu) = 0$*.*

□

In summary, because all DCT coefficient matrices are divided by the same quantization table in the JPEG compression process, the relationship between two DCT coefficients of the same coordinate position will not change after quantization. The only exception is that "*greater than*" or "*less than*" may become "*equal*" due to the rounding effect of quantization. The above theorem assumes that the same quantization table is used for the whole image. Theorem 1 is valid no matter how many recompression iterations and what the quantization tables are used .

For practical implementations, the quantization table can be extracted from the compressed file or estimated from the DCT coefficients of decompressed file. Note that Theorem 1 only preserve the sign of coefficient differences. The following theorem extends it to preserve the difference values, with various resolutions.

**Theorem 2** *Use the parameters defined in Theorem 1. Assume a fixed threshold* $k \in \Re$*.* $\forall \nu$*, define* $\tilde{k}_\nu \equiv$ *Integer Round* $(\frac{k}{\mathbf{Q}(\nu)})$*. Then,*

*if* $\mathbf{\Delta F_{p,q}}(\nu) > k$,

$$\mathbf{\Delta \tilde{F}_{p,q}}(\nu) \geq \begin{cases} \tilde{k}_\nu \cdot \mathbf{Q}(\nu), & \frac{k}{\mathbf{Q}(\nu)} \in Z, \\ (\tilde{k}_\nu - 1) \cdot \mathbf{Q}(\nu), & elsewhere, \end{cases} \tag{2.5}$$

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

$$\mathbf{\Delta \tilde{F}_{p,q}}(\nu) \leq \begin{cases} \tilde{k}_\nu \cdot \mathbf{Q}(\nu), & \frac{k}{\mathbf{Q}(\nu)} \in Z, \\ (\tilde{k}_\nu + 1) \cdot \mathbf{Q}(\nu), & elsewhere, \end{cases} \tag{2.6}$$

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

$$\mathbf{\Delta \tilde{F}_{p,q}}(\nu) = \begin{cases} \tilde{k}_\nu \cdot \mathbf{Q}(\nu), & \frac{k}{\mathbf{Q}(\nu)} \in Z, \\ (\tilde{k}_\nu \ or \ \tilde{k}_\nu \pm 1) \cdot \mathbf{Q}(\nu), & elsewhere. \end{cases} \tag{2.7}$$

□

In Theorem 2, $k$ is a designated threshold value used to bound the difference of two DCT coefficients of the same position in two separate blocks of an image. In contrast, Theorem 1 only describes the invariance property of the sign of $\mathbf{\Delta F_{p,q}}$. We can consider Theorem 1 as a special case of Theorem 2 (with $k$ set to be 0). Several different $k$'s (*e.g.*, a series of binary division of a fixed dynamic range) can be used for a single authentication system of different levels of strength. Based on Theorem 2, we can predict the difference relationships between coefficients after compression. Extension of the invariance property to the case of variable quantization table is included in Section 2.11.2.

As shown in Figure 2-1, by applying Theorem 1 and Theorem 2, we can extract feature codes $Z$ of an image from the relationships between two DCT coefficients of the same position in two separate blocks. These feature codes are then encrypted

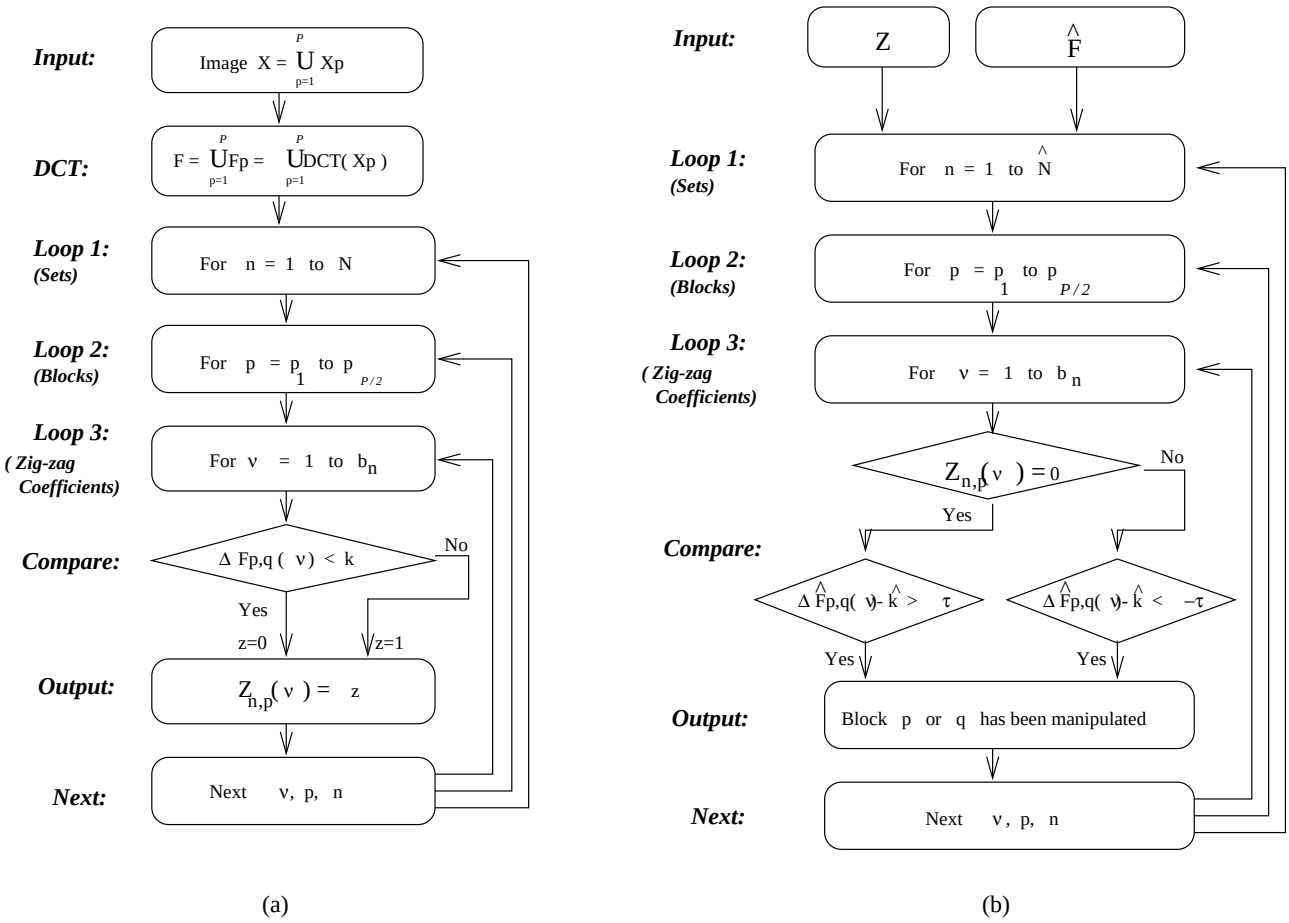



Figure 2-2: (a) Feature Extraction, (b) Authentication: Comparator

as a signature. For the authentication process, a user has to calculate the DCT coefficients of the image, and compare them to the features decrypted from the digital signature $S$. This image is said to be authenticated if all the DCT coefficient relationships satisfy the criteria predicted by the features of the original image.

### 2.3.2 Image Analyzer: Feature Extraction

Figure 2-2(a) is the flow chart of the feature extraction process. First, a digital image $X$ is sent into the image analyzer. Each $8 \times 8$ block of this image is then transformed to the DCT coefficients.

There are three loops for generating feature codes:

- Loop 1: Generate N sets of feature codes, $Z_{n,p}$, $n = 1$ to $N$. Each set uses different $k$ and $b_n$, where $k$ is defined in Theorem 2, $b_n$ is the number of DCT coefficients compared in each block pair.

- Loop 2: Iterate over all possible block pairs, $p = p_1$ to $p_{\frac{\wp}{2}}$, where $\wp$ is the total number of blocks in the image.

- Loop 3: Iterate over each of the $b_n$ selected coefficient pairs.

In Loop 1, $N$ sets of feature codes are generated. For each set, parameter $b_n$ represents how many bits are generated in each block. Parameter $k$ represents the precision threshold used in Theorem 2. The first set, $k = 0$, protects the sign of $\Delta F_{p,q}$. From the second set to the last set, $k$'s are set to protect the magnitude of $\Delta F_{p,q}$ with increasing accuracy. We will discuss how to define the thresholds later in this section.

In Loop 2, we need to form DCT blocks into pairs. As defined in Theorem 2, the DCT coefficient difference between block $p$ and block $q$ is computed. Let us denote one set of blocks $P_p = \{p_1, p_2, ..., p_{\frac{\wp}{2}}\}$ and another set of blocks $P_q = \{q_1, q_2, ..., q_{\frac{\wp}{2}}\}$. For example, $P_p$ can be all the even blocks, $\{0, 2, 4, ..., \wp - 1\}$, and $P_q$ can be all the odd blocks, $\{1, 3, 5, ..., \wp - 2\}$. The formation of all blocks in an image into pairs can be based on an arbitrary mapping function, $W$, as long as the following conditions are kept.

$$P_q = W(P_p), \tag{2.8}$$

and

$$P_p \cap P_q = \emptyset, \quad P_p \cup P_q = P. \tag{2.9}$$

If redundancy is allowed, $P_p$ and $P_q$ each may contain more blocks than $\frac{\wp}{2}$. The

choice of the mapping function $W$ can serve as a secret parameter, used to enhance the security of authenticator. For example, the image analyzer uses a seed to generate the mapping function $W$ and provides the seed with the feature codes to the authenticator. Each authenticator can transform this seed into the mapping function. This transformation method will not be made public. Therefore, each manufacturer of the image authenticator can implement his/her own transformation method.

In Loop 3, for each block, we compare the $b_n$ selected values (indexed in the zigzag order) in the DCT domain. Both DC and AC values in the DCT domain are used. At first, the difference of DC values in block $p$ and $q$, $\mathbf{\Delta F}_{p,q}(1)$, is used for comparison. If this value is smaller than $k$, then a feature code bit $z = 0$ is added to the end of the previous feature code. Otherwise, if this value is greater or equal to $k$, we will assign $z = 1$. (We classify two cases, "greater" and "equal", to the same type because the probability of $\mathbf{\Delta F_{p,q}}(\nu) = 0$ is quite small. If they are classified into three different types, i.e., "greater", "equal" and "less than", two bits should be used in this case. This will result in the increased length of feature codes.) Thereafter, the differences in selected AC values are compared with $k$. Only $b_n - 1$ AC differences, are used in this process. After Loops 1, 2 and 3 are completed, the feature codes, $Z$, of this image are generated. Usually, the $b_n$ selected positions are located in the low and middle frequency bands for the following two reasons: 1.) they are usually larger than the high-band coefficients because of energy concentration, and 2.) their values are usually conserved after JPEG compression because the values in the quantization table, $\mathbf{Q}$, in these bands are small.

#### 2.3.2.1 Precision Thresholds and Other Considerations

Theoretically, threshold values, $k$, can be determined arbitrarily, and they may vary for different $n$ and $\nu$. In our system, for the first set, all $k_{1,p}(\nu)$ are set to zeros. We use a binary division method to set thresholds for other sets. Assume the dynamic range of $\mathbf{\Delta F_{p,q}}(\nu)$ is from $-\zeta$ to $\zeta$. If we know that $\mathbf{\Delta F_{p,q}}(\nu) < 0$ in the first set, then we can set the threshold in the second set as $-\zeta/2$. Furthermore, if we know that this value $\mathbf{\Delta F_{p,q}}(\nu) > -\zeta/2$ in the second set, the threshold in the third set can be set as $-\zeta/4$. These thresholds result in dynamic binary decision ranges. This method protects the magnitude of $\mathbf{\Delta F_{p,q}}(\nu)$ with an increasing accuracy as more sets are being used. The larger $N$ is, the more precisely will the coefficient differences be limited.

Define a constant $\zeta$ which is a power of 2, and the threshold used in the n-th set of block $p$ at the position $\nu$ is $k_{n,p}(\nu)$. A closed form of $k_{n,p}(\nu)$ is

$$k_{n,p}(\nu) = \zeta \sum_{i=1}^{n-1} (\frac{1}{2})^i (-1)^{Z_{i,p}(\nu)+1}, \quad n > 1. \tag{2.10}$$

To simplify the notation in later discussions, we use $k = k_{n,p}(\nu)$ instead.

In addition to the parameters used in the three loops, some extra information about the image is necessary for defeating attacks. In our authentication system, a possible attack is to make a constant change to DCT coefficients at the same location in all blocks. This will not change the difference values between pairs of DCT coefficients from two different blocks. For instance, raising the image intensity uniformly changes the DC parameter in all blocks and defeats the previous approach. To defeat this attack, we record the mean value of DCT coefficients in each (selected) position for all blocks in the signature. These additional feature codes need no more than 64 bytes. When the DCT coefficients are changed by constant values, they will

be easily detected by the deviation of their mean values.

### 2.3.3 Authentication Process

Figure 2-1 includes the authentication process. It is composed of three parts. First, the received image, $\hat{X}$ or $\hat{B}$, has to be transformed to the DCT domain, $\hat{F}$. This involves the DCT transform block by block if a raw image, $\hat{X}$, is used. If a JPEG compressed image, $\hat{B}$, is used, a parser has to be used for reconstructing the Huffman Table and Quantization Table, $\hat{\mathbf{Q}}$. The signature, $S$, has to be decrypted to reconstruct feature codes, $Z$. After $\hat{F}$ and $Z$ are available, they will be sent to the authentication comparator in order to determine whether this image has been manipulated.

The Authentication Comparator is shown in Figure 2-2(b). Similar to the three loops in the image analyzer, there are also three corresponding loops here. In Loop 1, the number of loops, $n$, can be different from the one used in the Image Analyzer. Fewer loops may be used. Loop 2 and Loop 3 are the same as those used in the Image Analyzer. Inside these loops, we have to compare each of the DCT coefficient relationships obtained from the original image and that of the image received.

From Theorem 2, we can define

$$\hat{k} = \begin{cases} \tilde{k}_\nu \cdot \mathbf{Q}(\nu), & \frac{k}{\mathbf{Q}(\nu)} \text{ is an integer,} \\ (\tilde{k}_\nu + 1) \cdot \mathbf{Q}(\nu), & \frac{k}{\mathbf{Q}(\nu)} \text{ is not an integer } \textit{and } Z_n(\nu) = 0, \\ (\tilde{k}_\nu - 1) \cdot \mathbf{Q}(\nu), & \frac{k}{\mathbf{Q}(\nu)} \text{ is not an integer } \textit{and } Z_n(\nu) = 1. \end{cases} \tag{2.11}$$

(*Note that $\hat{k}$ is a function of $\nu$, $p$, and $n$.*) Observe from Figure 2-2(b), if $Z_n(\nu) = 0$, that is, $\mathbf{\Delta F_{p,q}}(\nu) < k$, then $\mathbf{\Delta \hat{F}_{p,q}}(\nu) - \hat{k} \leq 0$ must be satisfied. Therefore, if $\mathbf{\Delta \hat{F}_{p,q}}(\nu) - \hat{k} > 0$, we know that some parameters of block $p$ or $q$ must have been modified. Similar results can be obtained in the case of $\mathbf{\Delta F_{p,q}}(\nu) \geq k$.

However, some integer rounding noise may be introduced if the following cases occur: (1) the image is converted back to integral pixel values during the decode-reencode process, (2) the compressor and the signature generator use different chromatic decimation algorithms for color images, or (3) the JPEG encoder calculates imprecise DCT. Therefore, we must introduce a tolerance bound $\tau$ in the authenticator. We augment the comparing process with the following position.

**Proposition 1:** *Block p or q can be said to be manipulated*

*if*

$$\Delta\hat{\mathbf{F}}_{\mathbf{p},\mathbf{q}}(\nu) - \hat{k} > \tau, \tag{2.12}$$

*for the case of* $\Delta\mathbf{F}_{\mathbf{p},\mathbf{q}}(\nu) - k < 0$, *(or equivalently* $Z_n(\nu) = 0$*),*

*or if*

$$\Delta\hat{\mathbf{F}}_{\mathbf{p},\mathbf{q}}(\nu) - \hat{k} < -\tau, \tag{2.13}$$

*for the case of* $\Delta\mathbf{F}_{\mathbf{p},\mathbf{q}}(\nu) - k \geq 0$, *(or equivalently* $Z_n(\nu) = 1$*.)*

The tolerance, $\tau$, is determined by the level of integer rounding errors. Optimal levels of the rounding tolerance will be discussed in Section 2.4.1.

Note that the result of authentication can be a binary indicator, *true or false*, for the whole image, or it may indicate the authenticity or forgery of specific parts in an image.

#### 2.3.3.1 Other Considerations

Manipulation in specific block pairs can be located by the proposed technique. However, the authenticator using non-overlapping sets in Eq.(2.9) will not be able to identify which block in the pair has been modified. If identification of specific blocks is needed, we can use overlapping sets in Eq.(2.9). Identifying local changes is very useful to some applications in which both global and local contents are important.

For instance, in a picture of ten people, even if a man's face has been substituted by that of another person or has been removed, another parts of the image can still be verified to authenticate the appearance of the other nine people. Another advantage is that the system can verify authenticity in a selected area (*e.g.*, some news agency may cut out boundary areas of file photos).

Boundary cropping and/or position shifting are often performed on images to suit application needs. The proposed authentication signature is sensitive to cropping and shifting. However, for cropping, image block pairs that are not affected by cropping may still be authenticated. If cropping is allowed in some situations, we can design a robust digital signature with carefully selected mapping function, *e.g.*, selecting pairs from adjacent blocks. For shifting, if no DCT quantization is done on the shifted image (*e.g.*, shifting in the pixel domain only), the shifted image can be adjusted to the right position that results in the matched DCT block structure. Then, the DCT domain signature can be verified.

Constant intensity change in the image is sometimes expected, especially when the image is too dark or too bright. Our proposed authenticator solves this problem by relaxing the change threshold $\tau_s(1)$ of the mean value of DC coefficients.

Scaling is a common operation in many situations. For instance, a user may scan a picture with high resolution, and then down-sample it to an appropriate size. In this case, the signature generator has to record the original size of the image. Then, the authenticator can resize the image to its original size before the authentication process. Because the distribution of these sampling/interpolation noises can be modeled by a Gaussian function whose variance is not too large [57], there will be no large changes in the DCT coefficients. Similar to the recompression distortions, these changes can be accepted by setting adequate tolerance values.

Other lossy compression techniques such as wavelet-based methods or color space

decimation methods can be also considered as noise-adding processes. Similarly, we can use larger tolerances for these cases. Filtering, such as low-pass filtering and edge enhancement, may cause more visual changes and may cause challenges to the proposed technique. However, if the change in pixel values is not too large, we can consider them as some kind of noise and use adequate tolerance values. This strategy of adjusting tolerance levels can also be applied to other operations as well.

The authenticator is sometimes expected to pass only those images that are compressed by JPEG up to a certain compression ratio or quality factor. For example, if the image is JPEG compressed below the 20:1 ratio, the image is acceptable. Otherwise, if it is compressed more, it will fail the test. The argument for failing highly compressed images is that such images usually have poor quality and should not be considered as authentic. To satisfy this need, we can apply one of the following methods. The first one is to calculate the compression ratio from the raw image size and the compressed file size. If it is too high, the authenticator can reject it before any authenticating process. The second method is to calculate the increase of the number of the "equal" signature bits after compression. The number of "equal" signature bits increases if the image is compressed more. We can set a threshold on this change to reject those images that have too many "equal" coefficients in the block pairs.

### 2.3.4 Encryption, Decryption and Signature Length

The feature codes are encrypted by a secret private key of the Public Key method. As described in Section 3.2, the length, $l_f$, of feature codes is determined by the comparison bits $\frac{\varrho}{2} \cdot (\sum_{n=1}^{N} b_n)$, the seeds of the block pair mapping function and selected DCT positions, and the DCT mean values (see Section 3.2.1). For instance, assume the image size is $320 \times 240 = 76800(bytes)$. In a scenario that 10

bits of feature codes are used for each block pair, *i.e.*, $N = 1$ and $b = 10$. Assume both the seeds are 2 bytes long, and 6 DCT coefficient averages are recorded, then the length of feature codes, $l_f$, will be $\frac{40\times30}{2} \cdot 10 \cdot \frac{1}{8} + 2 + 2 + 6 = 760(bytes)$. The signature length can be further reduced with the reduction of the authenticator's effectiveness. We will analyze this trade-off in detail in Section 4.

The Public Key algorithm is used so that any user can easily access a public key to decrypt the signature. The most famous public key algorithm is RSA (Rivest, Shamir, and Adleman)[64][113]. The key length of RSA is variable but the most commonly used length is 512 bits [113], while the message block size must be smaller than the key length. If we choose to divide the feature codes into $B$-bit blocks, it needs $\lceil l_f \cdot 8 \cdot \frac{1}{B} \rceil$ RSA calculations (where $\lceil x \rceil$ denotes the integer ceiling function). Assume the output length of each RSA is $l_r$, then the signature length will be $\lceil l_f \cdot 8 \cdot \frac{1}{B} \rceil \cdot l_r$ bits. For instance, in previous example, if $B = 510$ and $l_r = 511$ are used, then the RSA algorithm has to be run 12 times and the signature length will be 767 bytes. It is about $\frac{1}{100}$ of the original image size.

A problem with Public Key algorithms is the speed. In hardware, the RSA Public Key algorithm is 1000 times slower than the DES Secret Key algorithm. The difference is about 100 times in software [113]. Therefore, if efficiency is critical, we can choose the Secret Key algorithm instead of the Public Key algorithm. The drawback is that users have to keep their secret keys safe, and the image can be authenticated by only the few people who own the secret key.

Implementation of a Public Key Infrastructure (PKI) is necessary for practical application. A PKI is the set of security services that enable the use and management of public-key cryptography and certificates, including key, certificate, and policy management [54]. If the image is generated from a hardware device, such as digital camera or scanner, then the manufacturer can serve as a Certification

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 486 | 91 | -66 | -91 | -17 | -1 | 14 | -0 | 727 | -188 | -3 | -28 | -16 | -4 | -6 | -1 |
| 140 | 41 | 44 | 35 | -8 | -12 | -6 | -4 | 51 | -77 | 22 | 45 | 11 | 1 | 2 | 3 |
| 43 | 108 | -54 | 5 | 16 | 13 | -9 | -0 | 31 | -52 | -73 | -8 | 5 | 5 | 10 | 7 |
| -143 | -21 | 84 | 34 | 22 | -0 | -12 | 6 | 73 | 40 | -21 | -7 | 1 | -13 | -2 | -2 |
| 9 | -18 | -2 | -32 | 8 | 5 | 5 | 12 | 19 | 12 | -21 | -17 | 4 | 2 | 2 | -1 |
| -23 | -9 | 1 | -1 | -8 | 1 | 2 | -0 | 20 | 15 | -2 | -17 | -5 | 2 | -0 | -1 |
| 3 | 10 | -14 | 4 | 6 | -1 | -1 | -6 | 16 | 16 | 13 | 1 | 2 | 6 | -2 | 0 |
| -8 | -10 | 14 | 3 | -1 | -2 | -2 | -3 | -1 | -3 | -6 | -12 | -6 | -1 | 1 | 3 |
| | | | | (a) | | | | | | | | (b) | | | |

Table 2.1: Two DCT coefficient blocks for a $16 \times 8$ area cut from the image "Lenna" (right eye region).

Authority (CA) issuing all user private-public key pairs. Private keys can be either embedded in the hardware device or issued to the driver software while customers register their information. Any end entity can examine the authenticity of images by requesting the public key and the authentication software from the manufacturer. Similarly, if the image is generated by a Content Holder, he/she can ask a private-public key pairs from any CA, which provides both key pairs and authentication software. The details of the PKI and the related standard $X.509$ can be found in [52].

### 2.3.5 Example: A small $16 \times 8$ image

We will use a small $16 \times 8$ image, $X$, as an example to illustrate the proposed authentication technique. This image is divided into two $8 \times 8$ blocks, from which DCT coefficients are computed. Therefore, $\wp = 2$. Its DCT coefficients are shown in Table 2.1. For simplicity, only integral values of them are shown in the table.

First, let us consider the case of $N = 1$, *i.e.*, only one threshold value $k = 0$ is used for feature code generation. Assume the first 10 coefficients in the zigzag order of the two blocks are compared. In this case, the length of the feature codes, $Z$, will be 10 bits ($b_1 = 10$) $\mathbf{\Delta F_{1,2}}(1) = -241 < 0$. Therefore, the first bit of the feature codes, $Z$, is 0. The second coefficients in the zigzag order are: $\mathbf{F_1}(2) = 91$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 480 | 96 | -64 | -96 | -16 | 0 | 16 | 0 |
| 144 | 48 | 48 | 32 | -16 | -16 | 0 | 0 |
| 48 | 112 | -48 | 0 | 16 | 16 | -16 | 0 |
| -144 | -16 | 80 | 32 | 16 | 0 | -16 | 0 |
| 16 | -16 | 0 | -32 | 16 | 0 | 0 | 16 |
| -16 | -16 | 0 | 0 | -16 | 0 | 0 | 0 |
| 0 | 16 | -16 | 0 | 0 | 0 | 0 | 0 |
| -16 | -16 | 16 | 0 | 0 | 0 | 0 | 0 |

(a)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 720 | -192 | 0 | -32 | -16 | 0 | 0 | 0 |
| 48 | -80 | 16 | 48 | 16 | 0 | 0 | 0 |
| 32 | -48 | -80 | -16 | 0 | 0 | 16 | 0 |
| 80 | 48 | -16 | 0 | 0 | -16 | 0 | 0 |
| 16 | 16 | -16 | -16 | 0 | 0 | 0 | 0 |
| 16 | 16 | 0 | -16 | 0 | 0 | 0 | 0 |
| 16 | 16 | 16 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | -16 | 0 | 0 | 0 | 0 |

(b)

Table 2.2: DCT coefficients in Table 1 quantized by a uniform matrix.

and $\mathbf{F_2}(2) = -188$, respectively. Since $\mathbf{\Delta F_{1,2}}(2) = 279 > 0$, the second bit of the feature codes is 1. After 10 iterations, the feature codes, $Z$, are: 0111100110.

Consider longer feature codes, we set $N = 4$, $b_1 = 10$, $b_2 = 6$, $b_3 = 3$ and $b_4 = 1$. The reason for a decreasing value of $b_n$ is that the lower frequency coefficients need more protection than the higher frequency ones. The threshold values, $k$'s, are 0, 128, 64 and 32 (in the absolute form). The first 10 bits of $Z$ are the same as the previous case. For the next 6 bits, the first six coefficients are compared again using $|k| = 128$. For example, since $\mathbf{\Delta F_{1,2}}(1) = -241 < -128$, the 11th bit = 0. $\mathbf{\Delta F_{1,2}}(2) = 279 > 128$, so the 12th bit = 1. The final feature codes are: 01111001100100010110. The length of $Z$ is $\sum_{n=1}^{4} b_n = 20$.

Table 2.2 shows the DCT coefficients after quantization (*i.e.*, $\tilde{\mathbf{F}}_\mathbf{1}$ and $\tilde{\mathbf{F}}_\mathbf{2}$) with a uniform matrix of 16. This is to simulate the quantization process in JPEG. Using Figure 2-2(b), we authenticate the compressed image by comparing $\mathbf{\Delta \tilde{F}_{1,2}}$ to the feature codes $Z$. For instance, $\mathbf{\Delta \tilde{F}_{1,2}}(1) = -240 < 0$ and $Z_1(1) = 0$, this value is authenticated to be true. Similar process continues until all feature codes are used. Note if the quantization table is not known to the authenticator, the first set of codes (with $k = 0$) can still be verified.

Consider an example of manipulation. Assume $X(0,2)$ and $X(0,3)$ are modified from 72 and 26 to 172 and 126. ($X$ can be obtained from the IDCT of Table 2.1.) Assume we use the same quantization matrix. Repeating the above process, the

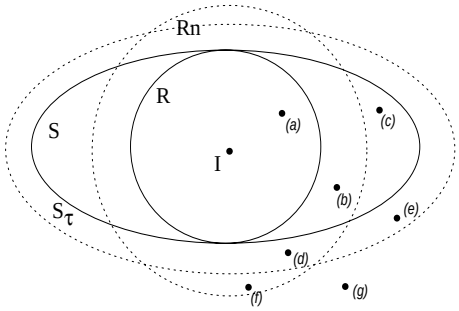



Figure 2-3: Conceptual illustration of 'miss', 'false alarm' and other scenarios.

authenticator will detect the manipulation due to the mismatch of the 4th bit of the feature codes.

### 2.3.6 Color Images

In the JPEG standard, color images are considered to be in the $YC_bC_r$ format. Chromatic components $(C_b, C_r)$ are usually down-sampled at the rate of 2:1 (horizontal direction only) or 4:1 (one-half in both horizontal and vertical directions). To authenticate a color image, we first down-sample the chromatic components with the sampling rate $4:1$. Then, we generate the feature codes of $Y$, $C_b$, $C_r$ in the same way as described earlier. In the authenticator, if the chromatic components are sampled by $2:1$, they are subsampled again in the other direction in order to obtain $4:1$ subsampled color components.

## 2.4 Performance Analysis

The image authenticator is a manipulation detector with two types of error involved: *miss* and *false alarm*[111]. 'Miss' refers to the situation in which an image is manipulated by unacceptable manipulations but the system reports the image as authentic. 'Miss' is also called *Type II error* in Hypotheses Testing. 'False alarm' means that the system reports the existence of manipulation while the image is, in

| | Image | Mapping Function & Number of Bits in Sets | Manipulation | Rounding Noise |
|---|---|---|---|---|
| Signature Generator | fixed | selected | random | random |
| Authenticator | fixed | fixed | random | random |
| Attacker | fixed | random | fixed | random |
| System Evaluation | random | random/fixed | random/fixed | random |

Table 2.3: Properties of different system variables from viewpoints of different parties

fact, not modified by unacceptable manipulations. It is also called a *Type I error*. In our authentication system, the test is based on block pairs. For each block pair, we perform the following test: $H_0$: *the pixels in the image block pair are not modified, or modified to new values that can be obtained by the JPEG compression processes.*, versus $H_1$: *the pixels in the image block pair are modified to new values that cannot be obtained by any JPEG process.* The test function is defined in Proposition 1. Conceptual illustration of 'Miss', 'False alarm' and other scenarios are shown in Figure 2-3. $I$ represents the original image. $R$ is the set of images obtained by JPEG compression of $I$. $R_n$ is $R$ augmented with rounding errors in JPEG compression. $S$ is the set of images passing authentication. $S_\tau$ is the set of images passing authentication allowing tolerance bounds. (a), (b) and (d) are correct authentication. (c) is a miss. (d) is a false alarm by $S$ but correct authenticated by $S_\tau$. (e) is a correct authentication by $S$ but missed by $S_\tau$. (f) is a false alarm. (g) is a successful detection of manipulation.

The Probability of Miss, $P_m$, and the Probability of False Alarm, $P_f$, are estimated by the signature generator and are useful to users of the authenticator. An additional evaluation metric, the Probability of Success, $P_s$, can also be used from the attacker's viewpoint. The attacker may try to manipulate the image based on his best knowledge of the authentication technique. Detailed discussion using these metrics will be shown in this section.

Several variables are needed to estimate these probabilities. We can classify

variables to three types: *pre-determined values*, *selectable variables*, and *stochastic variables*. The signature generator estimates a list of $P_f$ and $P_m$ based on different quantization tables and tolerances. Based on the quantization table used in the compressed image, the user may choose tolerances, $\tau$, to satisfy constraints on $P_f$ and $P_m$. Various properties of system variable from viewpoints of different parties are shown in Table 2.3.

### 2.4.1 Noise from the Compression Process and the Probability of False Alarm

Rounding noise may be added during the JPEG compression process and they may cause false alarm. In practice, computer software and hardware calculate the DCT with finite precision. For some cases, not only the input and the output of DCT operations are integers, but also some of the intermediate values. This will add rounding noise to the DCT values. In addition, some applications may drop small values in the high frequency positions. Combining these considerations, we can modify Eq.(2.2) to

$$\tilde{\mathbf{f}}_{\mathbf{p}}(\nu) = \ Integer\ Round(\frac{\mathbf{F_p}(\nu) + N_d}{\mathbf{Q}(\nu)}) + N_r, \tag{2.14}$$

where $N_d$ is the noise of DCT operation and $N_r$ is the noise of integer rounding. Both are random variables. $N_d$ usually depends on specific implementations and the number of recompression processes. Also, in most systems, the rounding rules are consistent over different positions and thus the effect of $N_r$ can be ignored in computing DCT coefficient differences.

The Probability of False Alarm of a block pair, $P_f$, represents the probability that at least one DCT difference value in the block pair triggers the detector in

Proposition 1, because of the effect of rounding noise. We can write $P_f$ as

$$P_f = 1 - \prod_{n=1}^{N} \prod_{\nu=b_1}^{b_n} (1 - \alpha_{n,\nu}) \approx \sum_{n=1}^{N} \sum_{\nu=b_1}^{b_n} \alpha_{n,\nu}, \tag{2.15}$$

where $\alpha_{n,\nu}$ is the probability that a DCT difference value $\mathbf{\Delta\tilde{F}_{p,q}}(\nu)$ triggers the false alarm. That is,

$$\alpha_{n,\nu} = \begin{cases} P[\mathbf{\Delta\tilde{F}_{p,q}}(\nu) - \hat{k} < -\tau], & given\ \mathbf{\Delta F_{p,q}}(\nu) \geq k, \\ P[\mathbf{\Delta\tilde{F}_{p,q}}(\nu) - \hat{k} > \tau], & given\ \mathbf{\Delta F_{p,q}}(\nu) < k. \end{cases} \tag{2.16}$$

Because of symmetry, these two probabilities are the same. To calculate $\alpha_{n,\nu}$, we first define a discrete random variable $N'_{d,p}$, $s.t.$, $\lfloor f_p + \frac{1}{2} \rfloor + N'_{d,p} \equiv Integer\ Round(\frac{\mathbf{F_p}(\nu) + N_{d,p}}{\mathbf{Q}(\nu)}) = \lfloor \frac{\mathbf{F_p}(\nu) + N_{d,p}}{\mathbf{Q}(\nu)} + \frac{1}{2} \rfloor$ where $f_p = \frac{\mathbf{F_p}(\nu)}{\mathbf{Q}(\nu)}$ and $\lfloor \cdot \rfloor$ represents the 'floor' function. $N'_{d,p}$ is the noise effect in the quantized coefficient, and can be derived from the continuous noise $N_d$. Its probability density function is

$$P[N'_{d,p} = n_d] = P[\ (n_d + \lfloor f_p + \frac{1}{2} \rfloor - f_p + \frac{1}{2}) \cdot \mathbf{Q}(\nu) > N_{d,p} \geq (n_d + \lfloor f_p + \frac{1}{2} \rfloor - f_p - \frac{1}{2}) \cdot \mathbf{Q}(\nu)\ ]. \tag{2.17}$$

The probability density function of $N'_{d,q}$ can be obtained in a similar way. After some transformations from Eq. (2.16),

$$\alpha_{n,\nu} = P[N'_{d,p} - N'_{d,q} < \hat{k}' - \tau' - \lfloor f_p + \frac{1}{2} \rfloor + \lfloor f_q + \frac{1}{2} \rfloor], \tag{2.18}$$

where $\hat{k}' = \frac{\hat{k}}{\mathbf{Q}(\nu)}$ and $\tau' = \frac{\tau}{\mathbf{Q}(\nu)}$. Then, we can obtain $\alpha_{n,\nu}$ by using the pdf in Eq. (2.17).

Applying Eq. (2.15), the user of image authenticator can set suitable tolerance value $\tau$ depending on the Quantization Table reconstructed from the bitstream, the

estimated variances of noise, and the thresholds. In practical applications, the user has to assume models for the pdf of $N_r$ *a priori*, for instance, Gaussian distribution. If the model of $N_r$ is not available, a practical rule is to set $\tau$ to *zero* or $\mathbf{Q}(\nu)$. The former is suitable for authenticating one-time compressed images while the latter is better for images that may be recompressed several times.

### 2.4.2 Manipulation and the Probability of Miss

The Probability of Miss represents the *reliability* of the authenticator. To obtain the Probability of Miss of a manipulated block pair, we may assume the block $p$ of the image is manipulated and its corresponding block $q$ is untouched. From the viewpoint of the signature generator, any manipulation on the block $p$ of image can be modeled as an additive random variable matrix $\mathbf{M_p}$, *s.t.*,

$$\hat{\mathbf{f}}_{\mathbf{p}}(\nu) = \ Integer\ Round(\frac{\mathbf{F_p}(\nu) + \mathbf{M_p}(\nu) + N_d}{\mathbf{Q}(\nu)}) + N_r. \tag{2.19}$$

where $N_d$ and $N_r$ are computation noises described above. In general, $M_p$ is much larger than $N_d$ and $N_r$. Therefore, the difference value of the DCT block pair is

$$\mathbf{\Delta}\hat{\mathbf{F}}_{\mathbf{p,q}}(\nu) = [\ Integer\ Round(\frac{\mathbf{F_p}(\nu) + \mathbf{M_p}(\nu)}{\mathbf{Q}(\nu)}) - Integer\ Round(\frac{\mathbf{F_q}(\nu)}{\mathbf{Q}(\nu)})\ ] \cdot \mathbf{Q}(\nu). \tag{2.20}$$

From Section 2.3.2, we know that the range of $\mathbf{\Delta F_{p,q}}(\nu)$ is bounded by the thresholds used in different sets of the authentication signature. Assume, in the position $\nu$, the range of DCT coefficients is divided into $K$ ranges by the thresholds of the authentication signature. The upper bound and the lower bound of a range are $k_l$ and $k_u$. For instance, if there is only one threshold, $k$, then $[k_{l,1}, k_{u,1}) = [-\infty, k)$ in the first range and $[k_{l,2}, k_{u,2}) = [k, \infty)$ in the second range. Assume a coefficient

$\mathbf{\Delta F_{p,q}}(\nu)$ is in this range,

$$\mathbf{\Delta F_{p,q}}(\nu) \in [\ k_{l,\nu}, k_{u,\nu}\ ). \tag{2.21}$$

After JPEG compression, the range of $\mathbf{\Delta \tilde{F}_{p,q}}(\nu)$ should be bounded by $[\hat{k}_{l,\nu}, \hat{k}_{u,\nu}]$ within a tolerance level, $\tau$. Therefore, the probability that the authenticator fails to detect a manipulation on position $\nu$ of the block pair $(p, q)$ is

$$\beta_\nu = P[\ \mathbf{\Delta \hat{F}_{p,q}}(\nu) \in [\hat{k}_{l,\nu} - \tau, \hat{k}_{u,\nu} + \tau]\ ], \quad given\ \mathbf{\Delta F_{p,q}}(\nu) \in [\ k_{l,\nu}, k_{u,\nu}\ ). \tag{2.22}$$

If we consider $\mathbf{M_p}$ as a random variable and apply Eq. (2.20), Eq. (2.22) becomes

$$\beta_\nu = P[\ m_{l,\nu} \leq \mathbf{M_p}(\nu) \leq m_{u,\nu}\ ], \tag{2.23}$$

where

$$\begin{cases} m_{l,\nu} &= \hat{k}_{l,\nu} - \tau + (\lfloor \frac{\mathbf{Fq}(\nu)}{\mathbf{Q}(\nu)} + \frac{1}{2} \rfloor - \frac{1}{2}) \cdot \mathbf{Q}(\nu) - \mathbf{F_p}(\nu), \\ m_{u,\nu} &= \hat{k}_{u,\nu} + \tau + (\lfloor \frac{\mathbf{Fq}(\nu)}{\mathbf{Q}(\nu)} + \frac{1}{2} \rfloor + \frac{1}{2}) \cdot \mathbf{Q}(\nu) - \mathbf{F_p}(\nu). \end{cases} \tag{2.24}$$

Assume a $b_n \times 1$ vector $\mathbf{\hat{M}_p}$, which is a subset of $\mathbf{M_p}$ representing the selected $b_n$ elements of $\mathbf{M_p}$, has a probability density function (pdf) $f(\mathbf{\hat{M}_p})$. Also, assume a range set $\mathbf{RB}$, $\mathbf{RB} = \{\mathbf{\hat{M}_p}(\nu) : m_{l,\nu} \leq \mathbf{\hat{M}_p}(\nu) \leq m_{u,\nu}\}, \forall \nu$, to specify the accepted range of manipulation. Then, the Probability of Miss, $P_m$, of a specific image block pair is

$$P_m = \int_{\mathbf{RB}} f(\mathbf{\hat{M}_p})\ d\mathbf{\hat{M}_p}. \tag{2.25}$$

To derive $P_m$, we need to know the pdf of manipulation, *i.e.*, $f(\mathbf{\hat{M}_p})$. We first consider manipulations in the spatial domain. Since the possible manipulation to an image block is arbitrary, from the signature generator's viewpoint, there is no exact distribution function. However, we can assume that the manipulated image

| Image | Replacement | Blur | Sharpen | Histogram Equalization |
|---|---|---|---|---|
| Lenna | 25.8 – 55.0 | 8.7 – 10.7 | 9.43 – 12.7 | 23.1 |

Table 2.4: Standard deviation of different operations (results of experiments using Photoshop 3.0 to manipulate image in the pixel domain)

block will be similar to its adjacent blocks, otherwise this manipulated image block will cause a noticeable artificial effect, which is easily detectable by people. Thus, we may use a multivariate zero-mean Gaussian distribution, $\mathbf{\Delta X_p}: \ N[\mathbf{0}, \sigma^2\mathbf{R}]$, to model the probability of additive intensity change of each pixel in the block. The variance parameter $\sigma^2$ depends on what kind of manipulation is expected. Some experimental values are shown in Table 2.4. $\mathbf{R}$ is the covariance matrix of the pixels in a block. In the DCT domain, we can get the probability distribution of $\mathbf{M_p}$ as follows,

$$\mathbf{M_p}: \ N[\ \mathbf{0}, \sigma^2\mathbf{DRD}^t\ ], \tag{2.26}$$

where $\mathbf{D}$ is the DCT transform matrix defined in Eq. (2.1).

To evaluate an authentication system, we can calculate the Probability of Miss based on the two extreme cases of $\mathbf{\Delta X_p}$, uncorrelated and fully correlated. In the uncorrelated case, i.e. $\mathbf{R} = \mathbf{I}$, manipulations on each pixels are totally uncorrelated. They are similar to Additive White Gaussian Noise. Therefore, $\mathbf{M_p}: \ N[\ \mathbf{0}, \sigma^2\mathbf{I}\ ]$ because $\mathbf{DD}^t = \mathbf{I}$ in DCT. In this case, the probability of miss $P_m$ will be

$$P_m = \prod_{\nu=b_1}^{b_n} \beta_\nu = \prod_{\nu=b_1}^{b_n} [\ \Phi(\frac{m_{u,\nu}}{\sigma}) - \Phi(\frac{m_{l,\nu}}{\sigma})\ ], \tag{2.27}$$

where $\Phi(.)$ is the *standard normal distribution function*, $\Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{\frac{-u^2}{2}} du$. In the fully correlated case, assume there is no weighting on specific positions in the pixel domain. The intensity change of each pixel is the same, i.e., $\mathbf{R} = [r_{ij}: \ i, j =$

1..64] where $r_{ij} = 1$. Then, $\mathbf{M_p} : N[\ \mathbf{0},\ \sigma^2\hat{\mathbf{R}}]$ where $\hat{\mathbf{R}} = [r_{ij} :\ i,j = 1..64]$ with $r_{1,1} = 64$ and $r_{i,j} = 0, elsewhere$. In this case, $P_m$ will be

$$P_m = \Phi(\frac{m_{u,1}}{8\sigma}) - \Phi(\frac{m_{l,1}}{8\sigma}). \tag{2.28}$$

Given a specific image block pair with the Quantization Table, the tolerance values, and the thresholds, we can use Eq. (2.27), (2.28), and Table 2.4 to estimate the range of Probability of Miss in an image block pair.

The above computation estimates the Probability of Miss for a single block pair. In some applications, the authenticator does not need to localize the manipulation. In these cases, the miss probability for the whole image is the product of the miss probabilities of all manipulated block pairs.

### 2.4.3 The Probability of Attack Success

From the attackers' point of view, they want to know the chance of success in attacking the authentication system. There are two kinds of attack. First, attackers may manipulate the image to generate a new image with different visual meaning. In this case, the attacker may use replacement, deletion, or cloning to change the pixel values. This kind of manipulation strategy attempts to blend the replaced pixels smoothly with adjacent areas. Second, attackers may manipulate the image (or synthesize an image) based on their knowledge about the authentication algorithm and secret information in the signature. This strategy of is to generate a different image to fool the authenticator. Note the image content may be clearly altered or distorted. In particular, if the attack is done in the DCT domain, noticeable distortion usually can be found in the pixel domain. In the following, we analyze the probabilities of success for these two types of attacks.

#### 2.4.3.1 Attacks with Visual Meaning Changes

Changing visual meaning of an image is a common attack. Based on the changes and estimation of authentication parameters, an attacker can estimate his chance of success. For instance, DCT values of the changed blocks are known to the attacker. If the image will not be further recompressed, the quantization table is also known. Otherwise, he can estimate the range of success probability based on different quantization tables. The threshold values can be estimated by looking at the DCT values and the signature. If this is not available, the first threshold value can be assumed to be zero. The tolerance values used in the authenticator are unknown. But he can assume some reasonable values such as zero or $\mathbf{Q}(\nu)$, and observe their effects on authentication. The only random part for estimating the Probability of Success would be the values of the DCT coefficients in another block of the pair.

Therefore, the Probability of Success, $P_s$, of a manipulated block can be modeled as

$$P_s = \prod_{\nu=1}^{b_n} \gamma_\nu, \tag{2.29}$$

where $\gamma_\nu$ is the probability of success for each DCT coefficient. We can compute $\gamma_\nu$ as follows,

$$\begin{aligned}
\gamma_\nu &= \textstyle\sum_K \{P[\boldsymbol{\Delta}\hat{\mathbf{F}}_{\mathbf{p},\mathbf{q}}(\nu) - \hat{k}_l \geq -\tau,\ \boldsymbol{\Delta}\mathbf{F}_{\mathbf{p},\mathbf{q}}(\nu) \geq k_l] + \\
&\qquad P[\boldsymbol{\Delta}\hat{\mathbf{F}}_{\mathbf{p},\mathbf{q}}(\nu) - \hat{k}_u \leq \tau,\ \boldsymbol{\Delta}\mathbf{F}_{\mathbf{p},\mathbf{q}}(\nu) < k_u]\} \\
&= \textstyle\sum_K \{P[\ \mathbf{F}_{\mathbf{q}}(\nu) \leq (\hat{\mathbf{f}}_{\mathbf{m}}(\nu) + \frac{1}{2}\ )\mathbf{Q}(\nu) - \hat{k}_l + \tau\ ,\ \mathbf{F}_{\mathbf{q}}(\nu) \leq \mathbf{F}_{\mathbf{p}}(\nu) - k_l] + \\
&\qquad P[\ \mathbf{F}_{\mathbf{q}}(\nu) \geq (\hat{\mathbf{f}}_{\mathbf{m}}(\nu) - \tfrac{1}{2}\ )\mathbf{Q}(\nu) - \hat{k}_u - \tau\ ,\ \mathbf{F}_{\mathbf{q}}(\nu) > \mathbf{F}_{\mathbf{p}}(\nu) - k_u]\} \\
&\equiv \textstyle\sum_K \gamma_{\kappa,\nu}
\end{aligned} \tag{2.30}$$

where

$$\hat{\mathbf{f}}_{\mathbf{m}}(\nu) = \lfloor \frac{\mathbf{F}_{\mathbf{p}}(\nu) + \mathbf{M}_{\mathbf{p}}(\nu)}{\mathbf{Q}(\nu)} + \frac{1}{2} \rfloor. \tag{2.31}$$

To estimate $P_s$, the attacker can assume $\mathbf{F_q}(\nu)$ to be a random variable with a Gaussian distribution with a zero mean and a variance of $\sigma_\nu^2$. Therefore, $P[\mathbf{F_q}(\nu) \leq \mathbf{F_p}(\nu) - k_l]$ can be written as $\Phi(\frac{\mathbf{F_p}(\nu)-k_l}{\sigma_\nu})$. Other probabilities can be approximated in a similar way. We can obtain the success probability of each coefficient, $\gamma_{\kappa,\nu}$, as

$$\gamma_{\kappa,\nu} = \begin{cases} \min\,[\,0,\ \Phi(\frac{(\hat{\mathbf{f}}_{\mathbf{m}}(\nu)+\frac{1}{2})\mathbf{Q}(\nu)-\hat{k}_l+\tau}{\sigma_\nu}) - \Phi(\frac{\mathbf{F_p}(\nu)-k_u}{\sigma_\nu})\,] \ , & \hat{\mathbf{f}}_{\mathbf{m}}(\nu) < \frac{\mathbf{F_p}(\nu)-\tau+\hat{k}_l-k_l}{\mathbf{Q}(\nu)} - \frac{1}{2} \\ \Phi(\frac{\mathbf{F_p}(\nu)-k_l}{\sigma_\nu}) - \Phi(\frac{\mathbf{F_p}(\nu)-k_u}{\sigma_\nu}) \ , & elsewhere, \\ \min\,[\,0, \Phi(\frac{\mathbf{F_p}(\nu)-k_l}{\sigma_\nu}) - \Phi(\frac{(\hat{\mathbf{f}}_{\mathbf{m}}(\nu)-\frac{1}{2})\mathbf{Q}(\nu)-\hat{k}_u-\tau}{\sigma_\nu})\,] \ , & \hat{\mathbf{f}}_{\mathbf{m}}(\nu) > \frac{\mathbf{F_p}(\nu)+\tau+\hat{k}_u-k_u}{\mathbf{Q}(\nu)} + \frac{1}{2}. \end{cases} \tag{2.32}$$

It should be noticed that the attacker has to calculate the DCT values of the manipulated blocks and estimate $\sigma_\nu^2$ before applying Eq. (2.29)-(2.32).

From Eq.(2.32), we can observe that $\forall\nu$, if $\hat{\mathbf{f}}_{\mathbf{m}}(\nu) \in [\ \frac{\mathbf{F_p}(\nu)-\tau+\hat{k}_l-k_l}{\mathbf{Q}(\nu)} - \frac{1}{2},\ \frac{\mathbf{F_p}(\nu)+\tau+\hat{k}_u-k_u}{\mathbf{Q}(\nu)} + \frac{1}{2}\ ]$ in all ranges, then the probability of success $P_s$ will be equal to 1. Using transformations similar to those in Eq.(2.23), we can represent this range in the DCT domain,

$$\begin{aligned} \{\lfloor\tfrac{\mathbf{F_p}(\nu)-k_l}{\mathbf{Q}(\nu)} + \tfrac{1}{2}\rfloor - \tfrac{1}{2} - \tfrac{\mathbf{F_p}(\nu)-\hat{k}_l}{\mathbf{Q}(\nu)}\} \cdot \mathbf{Q}(\nu) - \tau \ \leq\ \mathbf{M_p}(\nu) < \\ \{\lfloor\tfrac{\mathbf{F_p}(\nu)-k_u}{\mathbf{Q}(\nu)} + \tfrac{1}{2}\rfloor + \tfrac{1}{2} - \tfrac{\mathbf{F_p}(\nu)-\hat{k}_u}{\mathbf{Q}(\nu)}\} \cdot \mathbf{Q}(\nu) + \tau. \end{aligned} \tag{2.33}$$

Eq.(2.33) specifies the range in which an attacker can change the coefficients without triggering the authentication alarm. Note the size of this undetected manipulation range is equal to $\mathbf{Q}(\nu)$.

We can rewrite the above range as

$$\mathbf{M_p}(\nu) \in [a - \mathbf{Q}(\nu), a], \tag{2.34}$$

where $a$ is a coefficient dependent variable within the range of $[\tau - 1.5\mathbf{Q}(\nu), \tau + 2.5\mathbf{Q}(\nu)]$. Given that $\tau$ and $\mathbf{Q}(\nu)$ are unknown, the attacker cannot determine a fixed bound for undetected manipulations. Therefore, an attacker has no way to

maliciously manipulate an image without taking the risk of triggering the authentication alarm.

#### 2.4.3.2 Attacks with Knowledge of Authentication Rules

Some attackers may try to manipulate an image based on their knowledge about the authentication techniques, but regardless of the visual meaning of the manipulated image. Attackers may want to manipulate or even synthesize an image that can fool the system without triggering the alarm. In our authentication system, the security mechanism is based on: 1.) the private key used for the signature encryption, which ensures the signature cannot be forged; 2.) the secret transformation mechanism and a seed to generate the mapping function for selecting the block pairs; and 3.) the secret method and another seed used to select DCT coefficient positions in block pairs for comparison. In the following paragraphs, we will discuss four possible situations, with different extent of knowledge possessed by the attacker.

**Security Level I: All information in the signature is secret**

If all information in the signature is kept secret from the attacker, the performance of the proposed authenticator is the highest, as analyzed in previous sections. The only possible attack is to make a constant change to DCT coefficients at the same location in all blocks. We have proposed a way to solve this problem by recording the mean values of DCT coefficients as discussed in Section 2.3.2.1 and Section 2.3.3.

**Security Level II: The selected DCT coefficient positions are known**

The locations of the selected block pairs and the DCT coefficients are determined by some secret algorithms, which are in turn driven by random seed numbers. The

secret algorithms are usually pre-designed by the manufacturer of the authentication system. They can be stored as secret bytecodes embedded in the system. Therefore, even though the random seeds can be known by the attacker, the real selected positions are still unknown to the attacker.

In a pessimistic scenario, the attacker knows the secret algorithms and seeds for the selected DCT coefficients. Once he knows the real selected positions, he can arbitrarily change the coefficients that are not compared in the authentication process without triggering the authentication alarm. To avoid this problem, the authenticator can change the rule of selected positions, block by block, in a more complicated method. Furthermore, if this threat persists, the signature generator can eventually use all the 64 DCT coefficients in each block.

**Security Level III: The mapping function of block pairs is known**

Once the mapping function is known, the attacker also knows the DCT differences for each pair of blocks. For example, if only the sign of the DCT differences are used for authentication, and the attacker knows $\mathbf{\Delta}\hat{\mathbf{F}}_{\mathbf{p},\mathbf{q}}(\nu) = 10$ in the original compressed image, he can manipulate this value to $\mathbf{\Delta}\hat{\mathbf{F}}_{\mathbf{p},\mathbf{q}}(\nu) = 60$, which will not be detected by the authenticator. In this case, multiple threshold sets, $\mathbf{k}$, should be used because they can protect each coefficient with a higher accuracy. Although the DCT differences are known to the attacker, he still cannot manipulate those DCT coefficients too much, because the allowed degree of manipulation is reduced as more bits (*i.e.*, smaller $k$ values) are used.

**Security Level IV: The private key used for signature encryption is known**

The use of private key ensures that only the right source can generate the authentication signature. In the extreme hypothetical case, the private key used

by the original source may be known to the attacker. This is a general problem for any secure communication and is out of the scope of this paper. However, one possible way to solve this problem is to ask the original source to register and store its signature in a trustable institution. The institution stamps a digital postmark on the signature to prove its receiving time and its originality. Therefore, the forged signature will be considered invalid because its originality cannot be proven.

It is also worth noting that subjective inspection may provide another means of protecting the image authenticity. The attacker may try to develop special manipulations in the DCT domain in order to break the proposed scheme. But at the same time, it is difficult for the attacker to control the resulting artifacts in the pixel domain. These artifacts may be very obvious to humans, even as they are able to circumvent the authentication process.

## 2.5 Experimental Results

### 2.5.1 Experiments

In evaluating the proposed image authenticator, we test different manipulations on the well-known 'Lenna' image. The original image is shown in Figure 2-4(a). In our experiment, the authentication results together with the DCT coefficients $\hat{F}$ are sent to an IDCT to convert those coefficients to the pixel domain. Those blocks detected as manipulated will be highlighted, with the highlight intensity proportional to the number of manipulated coefficients in that block. Therefore, the more coefficients modified, the brighter this block will be. 10 bits per block pair are used in generating the signature codes.

(a) (b) (c)

(d) (e) (f)

Figure 2-4: (a) The original image, (b) JPEG compressed image (compression ratio 9:1), (c) middle of hat brim cloned, (d) authentication result of (c), (e) mouth manipulated, (f) authentication result of (e).

### Experiment 1: Lossy Compression

The 'Lenna' image is compressed with compression a ratio 9 : 1. The authentication signature is generated based on the original 'Lenna' image. The compressed bitstream is sent to the system for authentication. The tolerance bound of the authenticator is set to $\tau = 0$, since no integral rounding is involved. As previously predicted, the authenticator will verify the compressed image as authentic and decompress this image perfectly. The authentication result is shown in Figure 2-4(b).

### Experiment 2: Recompression and Integer Rounding

The original image is compressed with a compression ratio 6 : 1. Then, this image is decompressed by Photoshop, rounded to integral values, and recompressed into an image with compression ratio 9 : 1. In this case, if we use $\tau = \mathbf{Q}(\nu)$, the recompression process (9:1) will not trigger the manipulation detector and the final compressed image is still verified as authentic. The final decoded image is similar to Figure 2-4(b).

### Experiment 3: Detection of Manipulation

The third experiment is made by manipulating the image by deleting the feather fringe hanging over the hat brim, just above Lenna's right eye. This feather area ($16 \times 16$ pixels) is removed and cloned by its neighboring pixels. This image is shown in Figure 2-4(c). The authentication result is shown in Figure 2-4(d). It is clearly shown that the manipulated part has been detected as fake; it is highlighted by the authenticator. The other example is shown in Figure 2-4(e). In this image, Lenna's mouth was flipped in the vertical direction. Its authentication result is shown in Figure 2-4(f).

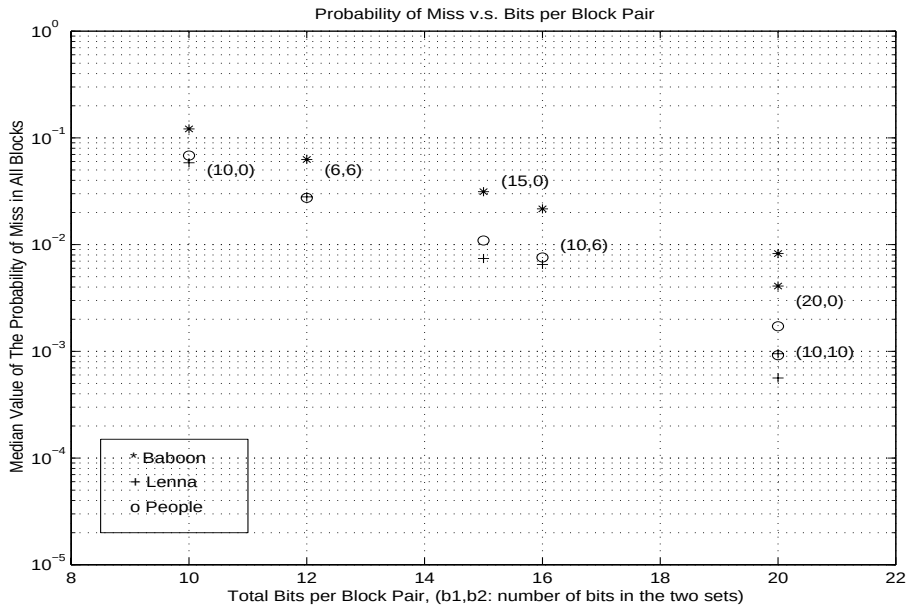



Figure 2-5: The Probability of Miss with different images.

### 2.5.2 Probability of Miss and Attack Success

From Figure 2-5 to Figure 2-8, we evaluate practical system performance by analyzing the Probability of Miss and the Probability of Success in different cases. Figure 2-5 shows the median values of the Probability of Miss in several images. The tolerance value, $\tau = 0$, the threshold values, $k = 0, \pm 16$, and the standard deviation of manipulations, 35, are used in this figure. (If not specified, these settings are kept the same for other figures.) In these figures, a $(b_1, b_2)$ symbol means $b_1$ bits are used in the first set of the feature codes, and $b_2$ are used in the second set. For instance, 10 bits used per block pair are denoted by a $(10, 0)$ symbol.

Figure 2-6 shows the Probability of Miss with different standard deviations of manipulations. These values are derived from Eqs. (2.27) and (2.28). Refer to Table 2.4, the standard deviation of a replacement manipulation is between 25.8 and 55.0. Through our experiments of 10 images, the medain value of this change is between 35 and 40. If we use 40 as the possible standard deviation of malicious

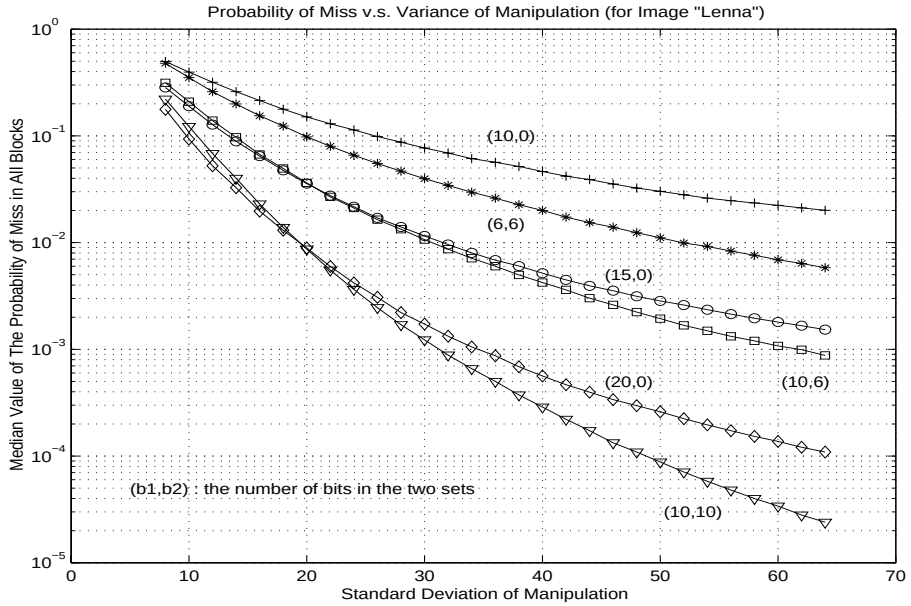



Figure 2-6: The Probability of Miss with different signature lengths.

manipulation, the estimated value of $P_m$ will be 0.04 for a $(10, 0)$ signature or 0.0004 for a $(20, 0)$ signature. The JPEG quality factor is 75 in this case.

Although the same authentication system is valid regardless of the image compression rate, the Probability of Miss is variable because the allowable modification range is increased as the quality factor decreases. This is shown in Figure 2-7. We use Eqs. (2.27) and (2.28) to compute these values. The standard deviation of manipulation is set to 35. We can see that $P_m$ increases when the image is compressed more.

Because the Probability of Success is case dependent, it can only be estimated when the attacker's actual manipulations are given. It is impractical to compute a single universal $P_s$ for an image. However, as an example, we change all DCT coefficients in a block with a constant and compute the $P_s$. Then we vary to location of the changed block in the image. For each block, we obtain a $P_s$. Finally, the median value of all probabilities versus the change magnitudes is shown in Figure 2-8. For instance, if we use the $(15, 0)$ signature and increase each DCT coefficient in

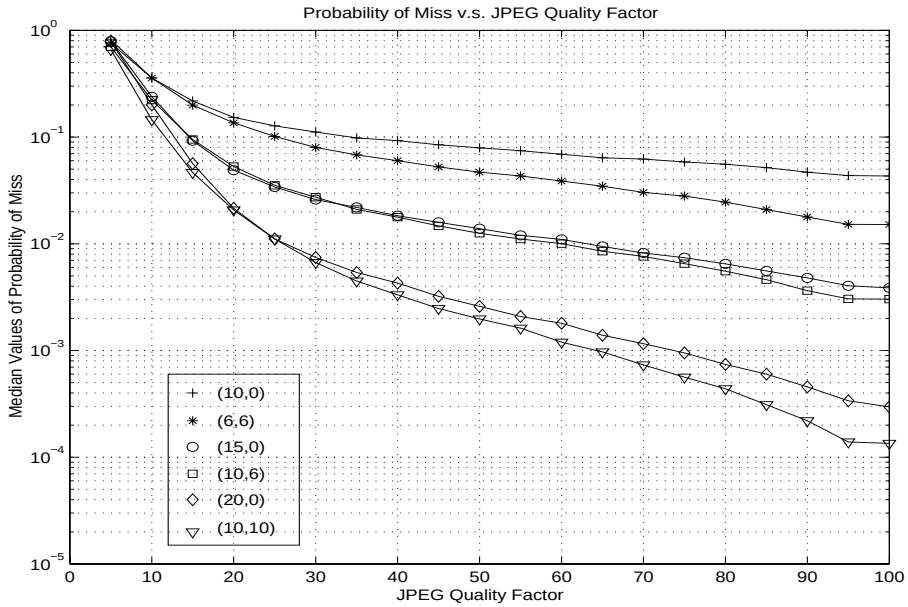



Figure 2-7: The Probability of Miss of images with different JPEG quality factors

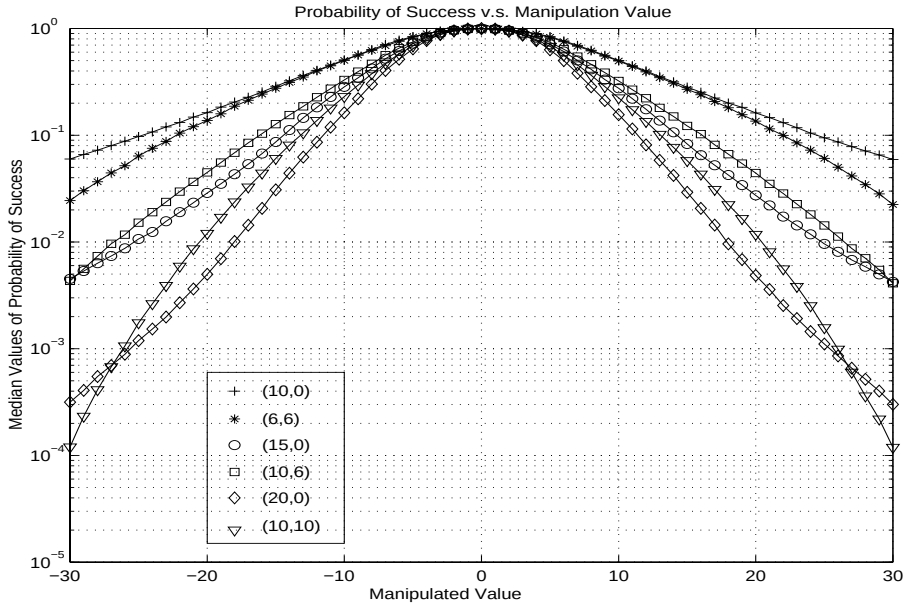



Figure 2-8: The Probability of Success with different manipulation values

| | Situation 1 | Situation 2 | Situation 3 | Situation 4 | Situation 5 |
|---|---|---|---|---|---|
| DCT (residual) coefficients | X (drop some coefficients) | X (requantization) | | X | |
| Motion Vectors | X | X | X | X | |
| Picture Type (I,P,B) | X | X | X | X (inconsistent in boundary) | |

Table 2.5: Consistent Properties of Transcoding and Editing Processing Situations

a block by 20, then the probability of sucess is about 0.03. In other words, assuming the attacker knows some authentication parameters ($\mathbf{Q}(\nu)$, $\tau$) but does not know which blocks are formed as pairs, his manipulation attack has a 0.03 probability of success.

Observing these figures, we know that the more bits used, the less the Probability of Miss will be. Also, we know that if the same number of bits was used, the performance of authentication signatures with two threshold sets will be better than those with only one set.

## 2.6 MPEG Video Authentication

In the following sections, we will focus on the issues and solutions of authenticating MPEG video. To extend our previous image authentication techniques, two important issues have to be noted: (1) transcoding and editing processes and (2) size of the digital signature. Since digital videos are seldom recorded in their raw format, we will consider the sources for authentication as being all in either the MPEG-1 or MPEG-2 format.

To design a system for the content authentication of compressed video, we have to know the types of possible acceptable manipulations that may be applied to the

video. In general, five acceptable transcoding or editing processing situations may be applied to the compressed video:

1. Dynamic Rate Shaping[38, 55]: A real-time rate-control scheme in the compressed domain. This technique sets dynamic control points to drop the high-frequency DCT coefficients on each $8 \times 8$ block in a macroblock. Motion vectors are not changed.

2. Rate Control without Drift Error Correction[128, 37]: This technique is also applied in the compressed domain. DCT coefficients are re-quantized to satisfy different bit-rate constraints. Motion vectors are not changed.

3. Rate Control with Drift Error Correction[125]: This technique improves the video quality after the requantization of DCT coefficients, but it needs more computation. DCT coefficients of the residue of intercoded blocks are modified to satisfy the change of the re-quantized intracoded blocks. Motion vectors are not changed in this case.

4. Editing with Mostly Consistent Picture Types[125, 138, 96]: The picture types (I, P and B) are kept unchanged in each editing generation. It may be used in creating a new sequence by cutting and pasting several video segments. The GOP (Group of Pictures) boundaries in each segment are not changed except those near the cut positions. Pixel values may be changed for video quality improvement such as intensity change, filtering, *etc.*.

5. Editing or Transcoding with Inconsistent Picture Types[125]: In some processes, the compressed videos are transformed to the uncompressed bitstreams which are then edited and re-encoded. The GOP structures and the motion vectors may change in this case. This kind of process includes format transmission between different compression standards and picture type conversion.

The first three processes are used for bitrate changes. They are all operated in the compressed domain. In other words, the structure of the MPEG *Program Streams* do not change. From Table 2.5, we can know that after these transcoding processes, the motion vectors and the picture types are preserved. The only change is on either the DCT coefficients of the *intra macroblocks* or the DCT residual coefficients of the *non-intra macroblocks*.

In studios, cutting and pasting several MPEG video segments to create a new video sequence is very common. It can be done with two different methods, Processing Situation 4 and Processing Situation 5. Their difference is basically whether the GOP structure is preserved through the editing process. In Situation 4, there are two kinds of GOP in the generated video sequence: original GOPs and created GOPs. An original GOP comes from an original video sequence with its structure intact. The created GOPs are generated from the boundary pictures of the original video sequence(s). There may be no created GOPs if the video sequence is not allowed to be cut inside a GOP. In practice, the number of created GOPs is much smaller than that of original GOPs (a typical GOP is about 0.5 second). For this situation, we focus on authentication of the original GOPs.

Video authentication signatures can be generated for different situations. We can find that for Situation 1-4, the GOP structure is not modified after transcoding or editing processes. Therefore, we can generate a robust digital signature which can survive these acceptable manipulations. We called this a Type I robust digital signature, which will be discussed in Section 2.7.1.

For Situation 5, because the GOP structure has been destroyed, only the pixel values of pictures will be preserved. Therefore, the video sequence is like a set of image frames, which can be authenticated by the image authentication that we proposed in Section 2.3. We call this a Type II robust digital signature. The

generation method is shown in Section 2.7.2.

### 2.6.1 Syntax of a MEPG Video Sequence

In the MPEG standard, each Video Sequence is composed of several sequential Group of Pictures (GOP). A GOP is an independent unit which includes several Pictures. In MPEG-1, each frame is a Picture. In MPEG-2, a Picture can be either a Field-Picture or a Frame-Picture. There are several Slices in a Picture. A Slice is a string of consecutive MacroBlocks (MBs) of arbitrary length running from left to right across the picture. The MB is the $16 \times 16$ motion compensation unit which includes several $8 \times 8$ Blocks. (An MB includes 6 blocks with the 4:2:0 chroma format, 8 blocks with the 4:2:2 chroma format, or 12 blocks with the 4:4:4 chroma format.) Each block is either Intra-coded or Non-Intra-coded. In MPEG, as with JPEG, Intra-coded blocks have their DC coefficients coded differently with respect to the previous block of the same YCbCr type, unless the previous block is Non-Intra, belongs to a skipped macroblock (MB), or belongs to another Slice[48]. The AC coefficients of each block in a macroblock is quantized by the *quantization_step_size* which is given by

$$(\kappa \cdot Q[m][n])/(8 \cdot v), \quad m, n = 0, 1, ..7, \quad m + n \neq 0, \tag{2.35}$$

where $\kappa$ is the *quantizer_scale* and $Q$ is the quantization matrix which is either the *Intra Qmatrix* for Intra blocks or the *NonIntra Qmatrix* for Non-Intra blocks. Both blocks may be defined in the *VideoSequence* Header if they are different from the default values. (In the 4:2:0 format, the luminance and chrominance Q-matrices are always the same. But they can be different in the 4:2:2 or 4:4:4 formats.) The parameter $v$ is equal to 1 for MPEG-1 video sequences, or 2 for MPEG-2 video sequences. The *quantizer_scale*, $\kappa$, is set for a Slice or a MB.

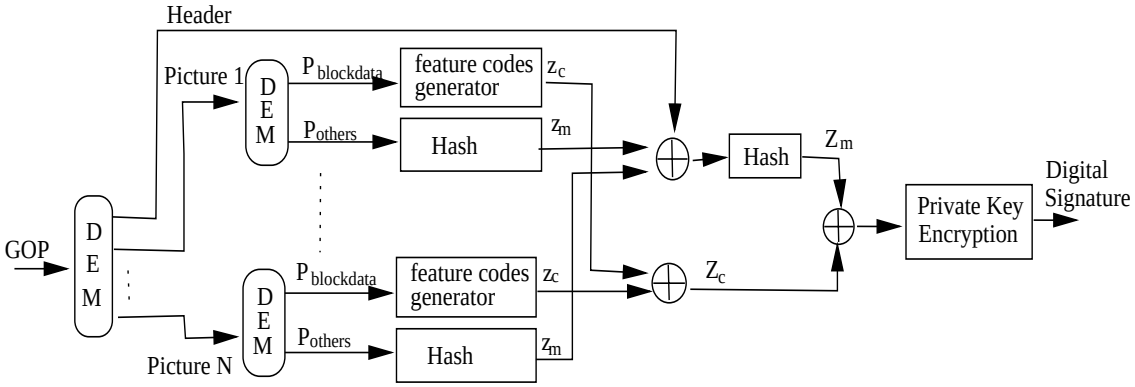



Figure 2-9: Robust Digital Signature : Type I

## 2.7 Robust Digital Signature

### 2.7.1 Robust Digital Signature: Type I

In Section 2.3, we have shown that the relationship between a coefficient pair, *i.e.*, two DCT coefficients of the same coordinate position, in any two $8 \times 8$ blocks of an image should remain the same or become equal after the re-quantization processes, if the same $quantization\_step\_sizes$ are applied on the blocks. We have also shown that the change of the difference value of a coefficient pair after re-quantization should be bounded in a range specified by the $quantization\_step\_sizes$, which can be different, of the blocks. Therefore, we can arrange all the blocks in an image to form block pairs, and generate some codes to represent the relationship status of coefficients in selected coordinate positions. The generated codes are then encrypted by public key method to form a digital signature.

To generate a robust digital signature for Processing Situations 1-4, we can use the quantized (*intra* or *non-intra*) DCT coefficients of the luminance and chrominance matrices in each macroblock to form comparison pairs. Since the $\kappa$ value as well as the $quantization\_step\_size$ is always the same in all blocks of a macroblock, the relative relationships of the coefficients at the corresponding positions of blocks

are invariant during transcoding. Therefore, similar to the signature generation process of images, we can use them to generate feature codes. First, the feature codes $Z_c$ of a macroblock can be written as,

$$z_c = VLC( \bigcup_{\mathbf{p}} \bigcup_{\mathbf{b}} sgn[\ \mathbf{f_p(b)} - \mathbf{f}_{W(\mathbf{p})}(\mathbf{b})\ ]\ ) \tag{2.36}$$

where

- **f** represents the quantized DCT coefficients in the compressed video sequence. They should be extracted from the bitstream and decoded with Variable Length Decoding (VLD).

- **p** is the set of the selected blocks in the macroblock, and W is the mapping function which maps each block at **p** to its corresponding block for forming a block pair. For instance, in a 4:2:0 format, if we label the 4 luminance blocks and the two chrominance as Block 1-6, then we can choose **p** as $\{1, 3, 5\}$ and a set $\mathbf{q} = W(\mathbf{p}) = \{2, 4, 6\}$ which forms three block pairs of Block $\{1, 2\}$, $\{3, 4\}$ and $\{5, 6\}$. For a macroblock of $\wp$ blocks, there will be $\wp!$ combinations.

- **b** is the set of the selected DCT coefficient positions. They are represents by the zig-zag order or alternative scan order whichever is used in the Video Sequence. For instance, if we choose to compare the DC values and the 1 - 5 AC coefficients in a block pair, then the **b** will be $\{1, 2, 3, 4, 5, 6\}$. The selection of **b** can vary for different block pairs.

- the sign function is defined as (1) $sgn(f) = 1,\ if\ f > 0$, (2) $sgn(f) = 0,\ if\ f = 0$, and (3) $sgn(f) = -1,\ if\ f < 0$.

It should be noted that here we use the sign function to represent the difference values because there are lots of zeros in the DCT coefficients of the compressed

video sequence. From the viewpoints of information, we should distinguish it from the other two situations, *i.e.*, positive and negative. This is different from what we have done for the images[73, 72]. Because there are lots of zeros in the coefficient comparison results, the VLC method can be applied to reduce the length of the feature codes.

In addition to the protection of DCT coefficients, we need to protect other information including the motion vectors and control codes as well. This can be done by adding the hash values of the remnant bitstreams in the video sequence to the feature codes. At the first step, assume a Picture P, $P$ includes $P_{block\_data}$ and $P_{others}$, where $P_{block\_data}$ includes the codes of DCT coefficients, the $quantizer\_scale$ in the Slice or MB header, and their control codes. $P_{others}$ includes all other codes in $P$. Then, we can get the hash values as,

$$z_m = Hash(P_{others}) \tag{2.37}$$

where $z_m$ is used for protecting other information of a Picture.

Because the GOP is the basic independent unit of a Video Sequence in the MPEG bitstream, we can encrypt the feature codes and the hash values of each pictures in a GOP to form a digital signature, *i.e.*,

$$DS = Private\ Key\ Encrypted(Z_c, Z_m) \tag{2.38}$$

where $Z_c = \bigcup_{Pictures} VLC(\bigcup_{MBs} z_c)$ is a combination of the feature codes $z_c$ of all the macroblocks in the GOP, and

$$Z_m = Hash(GOP\_Header, z_{m,1}, z_{m,2}, ..., z_{m,N}) \tag{2.39}$$

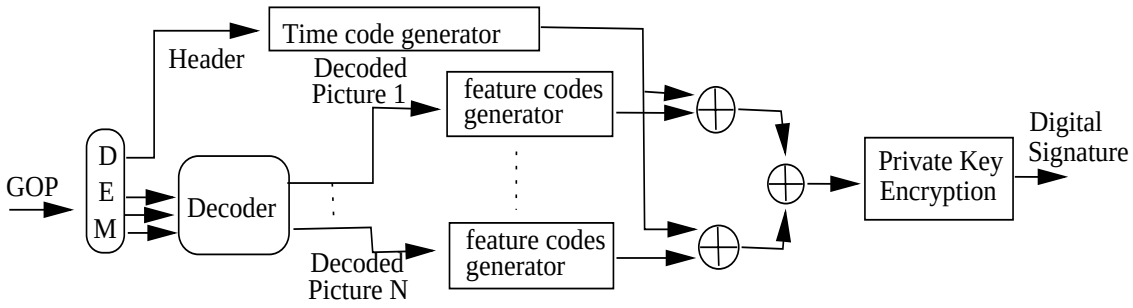



Figure 2-10: Robust Digital Signature : Type 2

where $N$ represents the total number of Pictures in a GOP. Eq.(2.39) indicates that, instead of using the combination of the hash values of each picture, the length of $Z_m$ can be further shortened by hashing the combination values, because all these information are fixed during the transcoding processes. Since $GOP\_Header$ includes the $time\_code$ which refers to the first picture to be displayed after the GOP header that has a $temporal\_reference$ of zero, it is important to include it to the digital signature for preventing temporal perturbation of GOPs. This digital signature, DS, can be placed in the $user\_data$ area of the GOP header. (In MPEG standards, $user\_data$ can be embedded in the Sequence Header, the GOP Header, or the Picture Header.)

### 2.7.2 Robust Digital Signature: Type II

The second type of robust digital signature is designed for surviving processes in Situation 5. Since the GOP structure, motion vectors, or DCT residual coefficients may change in this situation, the only consistent property is the pixel values of pictures. Therefore, we have to generate digital signature based on the pixel values of each picture. By using a similar authentication method for images, we can generate the digital signature picture by picture. The generation method is as follows:

1. Reconstruct the pixel values of the picture of any kind of picture type (I, P, B).

2. Generate feature codes by using exactly the same procedure as we proposed in Section 2.3, *i.e.*, dividing the image into $8 \times 8$ blocks, forming block pairs, comparing the DCT coefficients at the block pair, using one bit to represent each comparison.

3. Add time codes of each picture to the feature codes.

4. Using the Private Key Encryption to form the digital signature.

A diagram of the generating this type of robust digital signature is shown on Figure 2-10.

## 2.8 Authenticator

### 2.8.1 Authenticating Video Sequence after Transcoding (Situations 1-3)

The authenticator can be implemented as an augmentation of the general decoder. In the authenticator, the digital signature is extracted from the GOP header and decrypted to get the feature codes and the hash values. For examining the authenticity of a GOP in the video sequence, similar to the processes of signature generation, each picture in the GOP is divided into two parts: $P_{block\_data}$ and $P_{others}$. We then authenticate these two parts separately. To authenticate the hash values, we can get the $\hat{Z}_m$ of the GOP by using the same hash function(s) in the Eq. (2.37) and Eq. (2.39). Since this part of information is intact during the transcoding processes, $\hat{Z}_m$ is expected to be equal to $Z_m$. Otherwise, this GOP must have been modified by some other processes.

To authenticate the feature codes of GOP, the authenticator must first apply the VLC decoding to the feature codes to obtain the signs of the relationship of selected coefficients in each block pair. By applying a similar procedure of the authenticator we proposed on [73, 72], we can authenticate whether the DCT coefficients have been maliciously modified because:

- in Situation 1, some DCT high frequency coefficients in a block may be dropped and set to zero. Referring to the Theorem 1 in Section 2.3, if two DCT coefficients are both equal to zero after transcoding, the authenticator considers them as authentic. Because the lower frequency coefficients are preserved during transcoding, their relationships will be exactly the same as the original.

- in Situation 2, the DCT coefficients may be requantized to satisfy some bitrate constraints. Since all the DCT coefficients at the same position of the blocks in a MB are always quantized by the same $quantization\_step\_size$, according to the same theorem in Section 2.3, the possible changes of the sign values of the difference of a coefficient pair are: "positive to positive," "positive to zero," "zero to zero," "negative to negative," and "negative to zero." If we find the relationships of the coefficients do not satisfy this rule, we can claim that the video sequence has been modified by other manipulations.

- in Situation 3, the DCT coefficients of the intra blocks may be requantized. Also, the DCT residue coefficients of the non-intra blocks may be changed to compensate the quantization error introduced by the requantization of their reference blocks, and then be requantized again. To authenticate these blocks, we can introduce some tolerance bound to the authenticator. If we define $\Delta\mathbf{f}_{p,q}(b) = \mathbf{f}_p(b) - \mathbf{f}_{W(p)}(b)$, which is the difference of the coefficients at the

position $b$ in the block pair $(p, W(p))$ of the original video, and $\Delta\hat{\mathbf{f}}_{p,q}(b) = \hat{\mathbf{f}}_p(b) - \hat{\mathbf{f}}_{W(p)}(b)$, which is from the examined video. Then, the following property has to be satisfied,

$$if\ \Delta\mathbf{f}_{p,q}(b) > 0, \quad then \quad \Delta\hat{\mathbf{f}}_{p,q}(b) \geq -\tau, \tag{2.40}$$

$$else\ if\ \Delta\mathbf{f}_{p,q}(b) = 0, \quad then \quad \tau \geq \Delta\hat{\mathbf{f}}_{p,q}(b) \geq -\tau, \tag{2.41}$$

$$else\ if\ \Delta\mathbf{f}_{p,q}(b) < 0, \quad then \quad \Delta\hat{\mathbf{f}}_{p,q}(b) \leq \tau. \tag{2.42}$$

where

$$\tau = \begin{cases} 0, & intrablock, \\ 1 + \sum_{\mathbf{i}} \frac{\hat{\kappa}_{ref_i} \cdot Q_{ref_i}(b)}{\hat{\kappa} \cdot Q_{nonintra}(b)}, & nonintrablock \end{cases} \tag{2.43}$$

In Eq. (2.43), $\hat{\kappa}$ is the *quantizer_scale* of the nonintra blocks $p$ and $q$ in the examined video sequence. The set $\mathbf{i}$ represents the number of reference blocks, *e.g.*, $\mathbf{i} = \{1\}$ for a non-intra block in the first P picture of GOP, or $\mathbf{i} = \{1, 2\}$ for a non-intra block in the second P picture of GOP. The parameters $\hat{\kappa}_{ref_i}$ and $Q_{ref_i}$ are the *quantizer_scale* and the quantization matrix of the $i$-th reference block, respectively. (For a bi-directional predicted non-intra block, we have to use the average of the $\hat{\kappa}_{ref} \cdot Q_{ref}$ from its two reference blocks.) The proof of Eq.(2.43) is shown in [78]. Similar to the previous situations, the authenticator can examine the coefficients by Eq.(2.40)-(2.42). If they are not satisfied, we know that the video sequence must have been modified by unacceptable manipulations.

In addition to the manipulations within the GOPs, an attacker may perturb the temporal order of GOPs to change the meaning of video sequence. This manipulation can be detected by examining the time codes on the GOP Header that are protected in the digital signature. Changes of temporal order of pictures in a GOP

can be detected because both feature codes and hash values of the digital signature are generated in the order of pictures.

### 2.8.2 Authenticating Video Sequence after Editing (Situations 4 and 5)

The Type I robust digital signature is used in the Situation 4. In this situation, there are two kinds of GOP in the generated video sequence: original GOPs and created GOPs. For an original GOP that comes from an original video sequence with its structure intact, it has its independent digital signature which can be examined by the same authentication method described earlier. The created GOPs are generated from the boundary pictures of the segments of the original video sequence(s). There may be no created GOPs if we restrict the video sequence cannot be cut inside a GOP. This means splicing can only be performed to a resolution of about half a second[14]. If this restriction can not be satisfied, in a created GOP, type conversions may be applied on some pictures [96]. In an compressed video editor, if the digital signature of the corresponding source GOP is copied to the header of the created GOP, then those pictures without type conversions as well as all the intracoded macroblocks can be examined. The authenticator cannot examine those pictures with type conversions. Otherwise, if the digital signature is not copied to the created GOP, there is no clue for examining the authenticity. In general, we can neglect these boundary pictures and show that they are not examined. It addition to authenticating video sequences after cutting and pasting in the temporal segments, some other editing processes such as intensity enhancement, cropping, scaling, filtering, *etc.* may be applied in the video sequences. The robustness of our proposed digital signature towards these manipulations has been shown in [75].

For Situation 5 (video cut & paste or transcoding), all pixel values in each picture may change. However, the changes are like noises and are usually small

such that they do not change the meaning of video content. As we have discussed in [75], small noise-like changes in the spatial domain result in small changes in the DCT domain, too. Therefore, large changes in the DCT domain can be assumed to be from malicious manipulations. We can authenticate each picture by some predetermined tolerance values, $\tau$, in the authenticator. Applying Eq.(2.40)-(2.42), if all the coefficient pairs satisfy the equations, we can claim that authenticity of the examined video sequences.

Because there is no exact tolerance bound for changes caused by transcoding or editing of Situation 5, the authenticator can only indicate some areas of a picture "may have been" maliciously manipulated. This is done by observing the authentication result of the picture with different tolerance values. For instance, if $\tau = 0$, we may find the authenticator considers a large percentage of blocks in the picture as being manipulated. However, as $\tau$ increases, we can observe that most false-alarms will disappear and only areas that are actually maliciously manipulated are detected by the authenticator.

The time codes that are included in the digital signature can be used to detect changes in the temporal order and indicate the pixel values in the picture of the specific time. Since the video sequence is authenticated picture by picture, its authenticity can still be examined even if it was re-encoded with different temporal resolution.

## 2.9 Experimental Results and Discussion

Several video sequences have been tested with our proposed algorithms by using two different digital signatures. Our purpose is to evaluate the probability of missing a malicious manipulation by minimizing the probability of falsely reporting a manipulation. Through 20 more practical experiments on a video sequence "train"

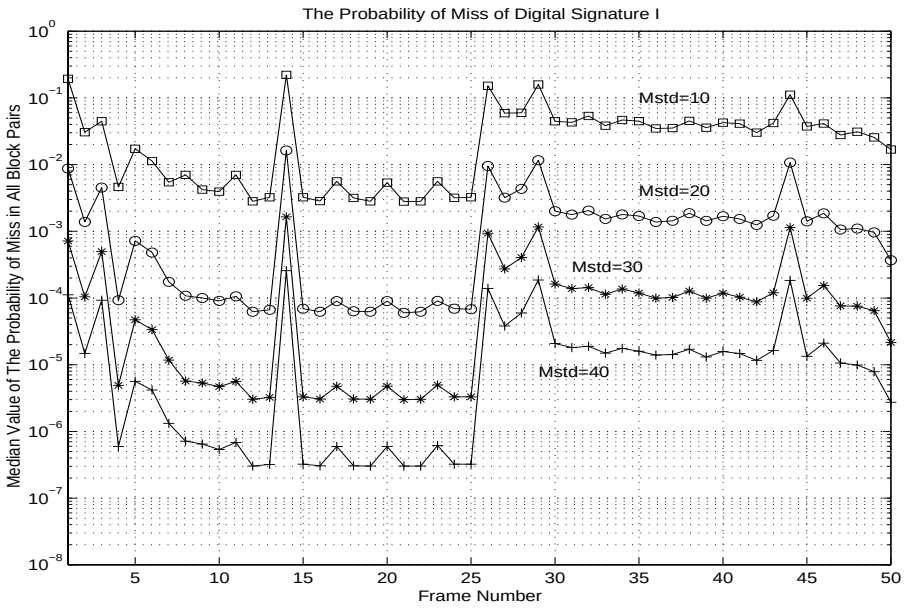



Figure 2-11: The Probability of Miss of Digital Signature I

(*e.g.*, transcoding in different rates, editing with cut and paste, object substitutions, *etc.*), we found that in Situations 1,2, and 4, there was no false alarm with tolerance values, $\tau = 0$, and in Situation 3 and 5, there was no false alarm with $\tau = 2$. With those settings, the authenticator can detect all object substitution manipulations.

Further system performance analysis can be done by estimating the probability of miss. The details of the statistical analysis are shown in [78]. In Figure 2-11, we show an example of the probability of miss of the video sequence "train", which includes 50 frames. From Figure 2-11, we can observe them with four different manipulation levels in terms of the standanrd deviation of Gaussian distributed manipulation changes in the blocks. In this example, we use six coefficients compared in a block pair. For the typical level of manipulation in the range of 30 - 40 (shown in Section 2.4), we can see the probabilities of miss of frames are within the range of $10^{-7}$ to $10^{-3}$. Those are all quite small. By examining the original video sequence, I frames or P frames with more intra blocks have larger probability of miss. That comes from the fact that intra blocks have more nonzero coefficients

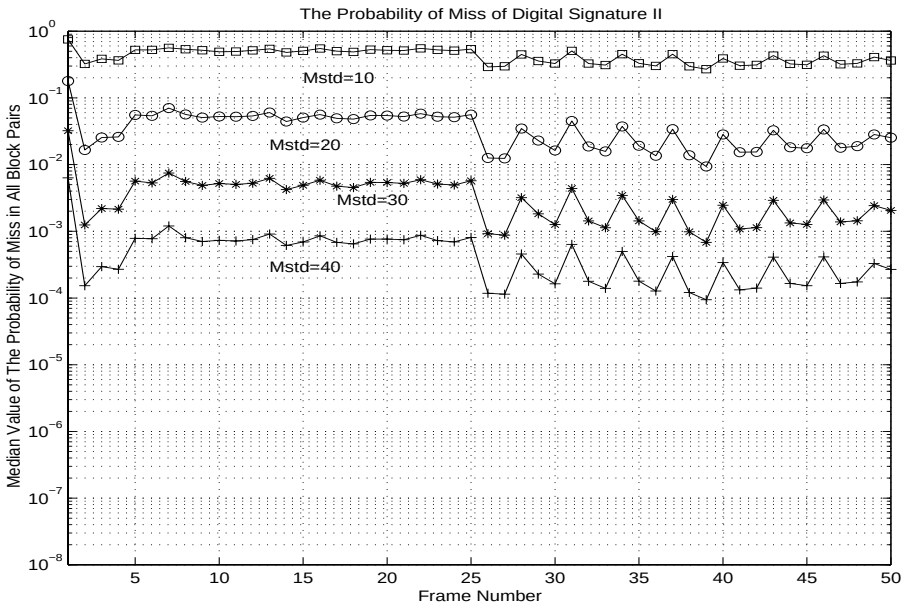



Figure 2-12: The Probability of Miss of Digital Signature II

that are more insensitivie to manipulations. From Figure 2-12, we can find that the probabilities of miss of Digital Signature Type II are larger than those in Figure 2-11. The probability of miss is in the range of $10^{-4}$ to $10^{-2}$. There are two reasons for this phenomeon. The first one is because all the blocks are decompressed and all of them are considered as the intra blocks. The second reason is the use of a larger tolerance value ($\tau = 2$), which reduces the probability of false alarm but also increases the probability of miss.

## 2.10 Conclusion

In this chapter, we have proposed an image authentication technique that distinguishes the JPEG lossy baseline compression from other malicious manipulations. In practical applications, images may be compressed and decompressed several times and still considered as authentic. Some manipulations, *e.g.*, integral value rounding, color space transformation and cropping, are also considered acceptable in some ap-

plications. We propose a technique that allows JPEG lossy compression but prevents malicious manipulations. Our proposed technique can be customized to accommodate different requirements and accept "desirable" manipulations. Our extensive analytic and empirical performance analysis has shown the effectiveness of this system.

Using practical simulations and mathematical analyses, we have examined the effectiveness of the proposed digital signature algorithms for MPEG video authentication. Our technique can distinguish compression from malicious manipulations. It solves the blind trustworthy problem of interim entities and makes video content authentication feasible. In the future, we will investigate issues in the MPEG audio content authentication for a complete multimedia content authentication system.

## 2.11 Proof of Theorems in Chapter 2

### 2.11.1 Proof of Theorem 1 and Theorem 2

**Proof 1** $\forall a, b, c \in \Re$, *assume* $a = A + r(a)$, $b = B + r(b)$, *and* $c = C + r(c)$, *where* $A, B, C \in Z$ *are the rounding integers of a,b,c, respectively, and* $-0.5 \leq r(a), r(b), r(c) < 0.5$.

*Assume* $a - b > c$, *then*

$$A + r(a) - B - r(b) > C + r(c). \tag{2.44}$$

*Therefore,*

$$A - B - C > r(c) + r(b) - r(a). \tag{2.45}$$

*If c is an integer,* i.e., $r(c) = 0$, *then*

$$A - B - C > -1.0, \tag{2.46}$$

*Since $A, B, C$ are integers,*

$$A - B \geq C. \tag{2.47}$$

*If $r(c) \neq 0$, then $-1.5 < r(c) + r(b) - r(a) < 1.5$. Since $A, B, C \in Z$,*

$$A - B \geq C - 1. \tag{2.48}$$

*Theorem 1 can be proved by substituting $a$ by $\frac{\mathbf{F_p}(u,v)}{\mathbf{Q}(u,v)}$, $A$ by $\frac{\tilde{\mathbf{F}}_\mathbf{p}(u,v)}{\mathbf{Q}(u,v)}$, $b$ by $\frac{\mathbf{F_q}(u,v)}{\mathbf{Q}(u,v)}$, $B$ by $\frac{\tilde{\mathbf{F}}_\mathbf{q}(u,v)}{\mathbf{Q}(u,v)}$, $c$ by 0, and with every parameter multiplied by $\mathbf{Q}(u,v)$. In Theorem 2, Eq.(3.3) can be proved by the same parameter substitutions except $c$ is replaced by $\frac{k}{\mathbf{Q}(u,v)}$ and $C$ is replaced by $\tilde{k}_{u,v}$. Eq.(3.4) and eq.(3.5) can be proved by using similar methods.*

□

In some software implementations, the integer rounding process is replaced by the truncation process. In this case, Theorem 1 and 2 are still valid. They can be proved by **Proof 2** with the same parameter substitutions as in **Proof 1**.

**Proof 2** $\forall a, b, c \in \Re$, *assume $a = A + r(a)$, $b = B + r(b)$, and $c = C + r(c)$, where $A, B, C \in Z$ are the truncated integers of a,b,c, respectively, and $0 \leq r(a), r(b), r(c) < 1$. Similarly, in the case that $a - b > c$, i.e., $A - B - C > r(c) + r(b) - r(a)$,*
*if $c$ is an integer, then $-1.0 < r(c) + r(b) - r(a) < 1.0$. Therefore,*

$$A - B \geq C. \tag{2.49}$$

*If $r(c) \neq 0$, then $-1 < r(c) + r(b) - r(a) < 2$. Since $A, B, C \in Z$,*

$$A - B - C \geq 0 > -1, \tag{2.50}$$

*therefore,*

$$A - B \geq C, \tag{2.51}$$

*which satisfies* $A - B \geq C - 1$.

□

### 2.11.2 Variable Quantization Tables

In some image/video compression techniques, different quantization tables are used in different image blocks for adaptive compression rate control, such as in MPEG or later JPEG standards. In these cases, the proposed image authentication techniques can be extended by the following theorem.

**Theorem 3** *Use the parameters defined in Theorem 1, except* $\tilde{\mathbf{F}}_{\mathbf{p}}$ *is defined as* $\tilde{\mathbf{F}}_{\mathbf{p}}(\nu) \equiv$ *Integer Round*$(\frac{\mathbf{F_p}(\nu)}{\mathbf{Q_p}(\nu)}) \cdot \mathbf{Q_p}(\nu)$ *and* $\tilde{\mathbf{F}}_{\mathbf{p}}(\nu) \equiv$ *Integer Round*$(\frac{\mathbf{F_q}(\nu)}{\mathbf{Q_q}(\nu)}) \cdot \mathbf{Q_q}(\nu)$, *where* $\mathbf{Q_p}$ *and* $\mathbf{Q_q}$ *are quantization tables for blocks* $\mathbf{F_p}$ *and* $\mathbf{F_q}$ *respectively. Assume a fixed threshold* $k \in \Re$. *The following properties hold:*

- *if* $\mathbf{\Delta F}_{\mathbf{p,q}}(\nu) \geq k$, *then* $\mathbf{\Delta}\tilde{\mathbf{F}}_{\mathbf{p,q}}(\nu) \geq k - \frac{1}{2}(\mathbf{Q_p}(\nu) + \mathbf{Q_q}(\nu))$,
- *else if* $\mathbf{\Delta F}_{\mathbf{p,q}}(\nu) < k$, *then* $\mathbf{\Delta}\tilde{\mathbf{F}}_{\mathbf{p,q}}(\nu) \leq k + \frac{1}{2}(\mathbf{Q_p}(\nu) + \mathbf{Q_q}(\nu))$.

□

We redefine Eq. (2.11) as

$$\hat{k} = \begin{cases} k + \frac{1}{2}(\mathbf{Q_p}(\nu) + \mathbf{Q_q}(\nu)), & if\ Z_n(\nu) = 0,\ i.e.,\ \mathbf{\Delta F}_{\mathbf{p,q}}(\nu) < k, \\ k - \frac{1}{2}(\mathbf{Q_p}(\nu) + \mathbf{Q_q}(\nu)), & if\ Z_n(\nu) = 1,\ i.e.,\ \mathbf{\Delta F}_{\mathbf{p,q}}(\nu) \geq k. \end{cases}$$

In other words, if $\mathbf{\Delta F}_{\mathbf{p,q}}(\nu) < k$, then $\mathbf{\Delta}\hat{\mathbf{F}}_{\mathbf{p,q}}(\nu) - \hat{k} \leq 0$ must be satisfied.

Except the above modifications, the authentication system designed for the variable quantization table cases would be the same as the proposed system for the case with equal quantization tables. A detailed discussion of this case is in [71].

# Chapter 3

# Using Semi-Fragile Watermarks to Generate Self-Authentication-and-Recovery Images

## 3.1 Introduction

In this chapter, 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 approximations 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 Self-Authentication-and-Recovery Images (SARI) authenticator is based on two invariant properties of quantization-based lossy compression. 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 relationship between two coefficients in a block pair before and after JPEG compression. We can use the second property to generate the authentication signature, and use the first property to embed it as a watermark. These properties provide solutions to two major challenges in developing authentication watermarks (*a.k.a.*, integrity watermarks): how to extract short, invariant and robust information to substitute fragile hash functions, and how to embed information that is guaranteed to survive quantization-based lossy compression to an acceptable extent. Because the first property almost reaches maximum zero-error embedding capacity, in addition 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. SARI has been extensively tested through real application software, *e.g.*, Photoshop, XV, Paint Shop Pro, *etc.*, in various testing environments. Experimental results showed the effectiveness of this system.

**• 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 [79]. Using digital signature, Schneider and Chang first proposed the concept of salient feature extraction and similarity measure for image content authentication [112]. They also discussed issues of embedding such signatures into the image. However, their work lacked a 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 [11]. They gen-

erated digital signature based on the location of these feature points. The advantage of this technique was the efficiency in its signature length. But it lacked a rigorous mechanism to select visually interesting points. This technique's ability to detect crop-and-replace manipulations and its robustness through lossy compressions was questionable.

Queluz proposed techniques to generate digital signature based on moments and edges of an image[108]. Using moments as features ignores 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 in Chapter 2. Our authentication signature is an encrypted feature vector generated from the invariant relationship between DCT coefficients in separate blocks of an image. We proved that 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 which measurement of the error between the watermarked image and the manipulated image determined

authenticity [149]. They estimated a masking function from the image, and used it to measure distortion. Their method added imperceptible changes to the image. It is not clear that whether this estimated masking function would 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 would change if the image is manipulated by pixel replacement.

Wolfgang and Delp developed an authentication method that embeds bipolar m-sequence into blocks[140]. 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 [42][43]. He divided images to 64 pixel $\times$ 64 pixel blocks. For each block, quasi-VQ codes were embedded using the spread spectrum method. This technique was robust to manipulations. But, comparing his experiments in [42] and in [43], 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 chapter, 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-watermarked image. To achieve this, some iterations may be needed in implementation. (2) The signature and the watermark consist of 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 processes. 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 approximations of the originals. 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 property 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 applies integral DCT and Inverse DCT until the designated DCT values can be directly obtained from integer pixel

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 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 3.1: The quantization tables, $\mathbf{Q_{50}}$, of JPEG compression with *Quality Factor(QF) = 50* : (a) luminance,(b) chromnance. 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.

values. These methods help avoid computation errors and false alarm in practical implementations.

This chapter is organized as follows. In section 3.2, we show two important properties mentioned above. In Section 3.3, we describe details of our authentication system. The performance of this authentication system is analyzed in Section 3.4. In Section 3.5, we show some testing results. Conclusion and discussion of future directions are shown in Section 3.6.

## 3.2 Two Invariant Properties in JPEG compression

In this section, we describe and demonstrate two invariant properties during JPEG compression. The first one is used for embedding watermark, and the second one is proposed in Chapter 2 and is used for generating authentication signature.

**Theorem 4** *Assume* $\mathbf{F_p}$ *is a DCT coefficient vector of an arbitrary* $8 \times 8$ *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. $\frac{\grave{\mathbf{F}}_{\mathbf{p}}(\nu)}{\mathbf{Q}'_{\mathbf{m}}(\nu)} \in Z$ *where* $\mathbf{Q}'_{\mathbf{m}}(\nu) \leq$

$\mathbf{Q_m}(\nu)$, *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:*

$$\textit{Integer Round}(\frac{\tilde{\mathbf{F}}_\mathbf{p}(\nu)}{\mathbf{Q'_m}(\nu)}) \cdot \mathbf{Q'_m}(\nu) = \grave{\mathbf{F}}_\mathbf{p}(\nu) \tag{3.1}$$

□

**Proof of Theorem 4:** See Section 3.7.

Theorem 4 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 3.1 shows that quantization tables of JPEG compression for all quality factors. From Table 3.1, we know that

$$\mathbf{Q_{QF}}(\nu) \leq \mathbf{Q_m}(\nu), \quad \forall \nu \in \{1, ..64\} \textit{ and } QF \geq m. \tag{3.2}$$

In other words, the higher QF (quality factor) is, the smaller the quantization step is. In Eq. (3.2), 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[60]. 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 or equal to $\mathbf{Q_m}(\nu)$ (note the greater than but not equal sign in Theorem 4). 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 4. 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 4 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 in Chapter 2** *Assume* $\mathbf{F_p}$ *and* $\mathbf{F_q}$ *are DCT coefficient vectors of two arbitrary* $8 \times 8$ *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* $\mathbf{\Delta F_{p,q}}(\nu) > k$,

$$\mathbf{\Delta \tilde{F}_{p,q}}(\nu) \geq \begin{cases} \tilde{k}_\nu \cdot \mathbf{Q}(\nu), & \frac{k}{\mathbf{Q}(\nu)} \in Z, \\ (\tilde{k}_\nu - 1) \cdot \mathbf{Q}(\nu), & elsewhere, \end{cases} \tag{3.3}$$

[1]This value was $\mathbf{Q_m}(\nu) + 1$ in [84]. We later found that Theorem 4 was also valid if $\mathbf{Q'_m}(\nu) = \mathbf{Q_m}(\nu)$. This is proved in Section 3.7.

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

$$\mathbf{\Delta \tilde{F}_{p,q}}(\nu) \leq \begin{cases} \tilde{k}_\nu \cdot \mathbf{Q}(\nu), & \frac{k}{\mathbf{Q}(\nu)} \in Z, \\ (\tilde{k}_\nu + 1) \cdot \mathbf{Q}(\nu), & elsewhere, \end{cases} \tag{3.4}$$

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

$$\mathbf{\Delta \tilde{F}_{p,q}}(\nu) = \begin{cases} \tilde{k}_\nu \cdot \mathbf{Q}(\nu), & \frac{k}{\mathbf{Q}(\nu)} \in Z, \\ (\tilde{k}_\nu \ or \ \tilde{k}_\nu \pm 1) \cdot \mathbf{Q}(\nu), & elsewhere. \end{cases} \tag{3.5}$$

□

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 $8 \times 8$ 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.

## 3.3 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.

### 3.3.1 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.

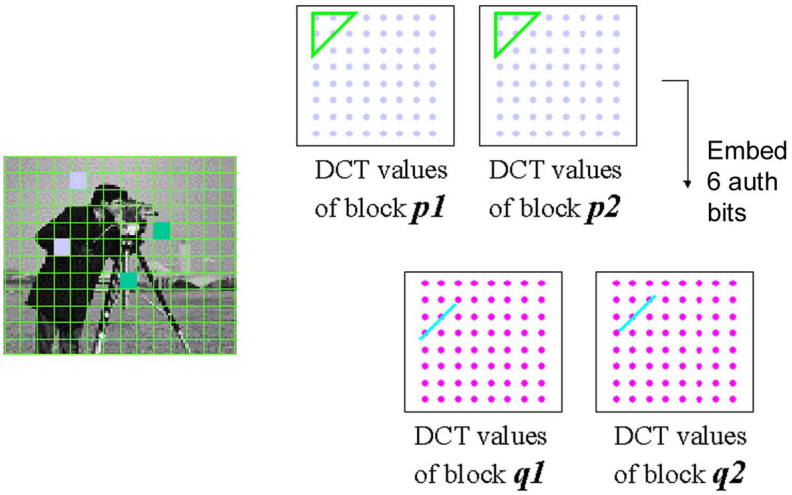



Figure 3-1: Embedding authentication bits in the image based on the mapping functions

We use a signature generation method we proposed in Chapter 2. Similar to the JPEG process, images are first divided to $8 \times 8$ 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 $\beta_a$ 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

$$\mathbf{\Phi}_p(\nu) = \begin{cases} 1, & \mathbf{\Delta F}_{p,q}(\nu) \geq 0 \\ 0, & \mathbf{\Delta F}_{p,q}(\nu) < 0, \end{cases} \tag{3.6}$$

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

Figure 3-1 shows the process for embedding authentication bits. 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 $\mathbf{f'}_p(\nu) = Integer\ Round(\frac{\mathbf{F}_p(\nu)}{\mathbf{Q'_m}(\nu)})$, where $\mathbf{Q'_m}(\nu) = \mathbf{Q_m}(\nu)$. Then we embed the authentication bits by modifying $\mathbf{F}_p(\nu)$ to $\grave{\mathbf{F}}_p(\nu)$ as follows

$$\grave{\mathbf{F}}_{\mathrm{p}}(\nu) = \begin{cases} \mathbf{f'}_p(\nu) \cdot \mathbf{Q'_m}(\nu), & LSB(\mathbf{f'}_p(\nu)) = \mathbf{\Phi}_{T_a(p)}(\nu) \\ (\mathbf{f'}_p(\nu) + sgn(\frac{\mathbf{F}_p(\nu)}{\mathbf{Q'_m}(\nu)} - \mathbf{f'}_p(\nu)) \cdot \mathbf{Q'_m}(\nu), & LSB(\mathbf{f'}_p(\nu)) \neq \mathbf{\Phi}_{T_a(p)}(\nu), \end{cases} \tag{3.7}$$

where $sgn(x) = 1$, if $x \geq 0$, and $sgn(x) = -1$, 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. 3.7 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 may carefully select the embedding method in the system. For instance, we could use even number

of $\mathbf{f'_P}(\nu)$ to represent the authentication bit "1," because the "flat" area images usually have a lot of AC coefficients equal to zero. Therefore, a large percentage of the authentication bits, "1", will be generated and embedded without much modifications. We have tested that this strategy can significantly reduce visual distortion in the synthetic or document images.

Another alternative is as follow. If we want to embed $\beta_a$ 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 $x_\nu$, of $LSB(\mathbf{f'}_p(\nu))$ and $LSB(\mathbf{f'}_q(\nu))$ to represent a bit, $b_\nu$. 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

$$(\grave{\mathbf{F}}_{\mathrm{p}}(\nu), \grave{\mathbf{F}}_{\mathrm{p}}(\nu)) = \begin{cases} (\mathbf{f'}_p(\nu),\ \mathbf{f'}_q(\nu) + sgn(\frac{\mathbf{F}_q(\nu)}{\mathbf{Q'_m}^{(\nu)}} - \mathbf{f'}_q(\nu)) \cdot \mathbf{Q'_m}(\nu), & if\ \mathbf{f'}_p(\nu) = 0,\ \mathbf{f'}_p(\nu) \neq 0, \\ (\mathbf{f'}_p(\nu) + sgn(\frac{\mathbf{F}_p(\nu)}{\mathbf{Q'_m}^{(\nu)}} - \mathbf{f'}_p(\nu)),\ \mathbf{f'}_q(\nu)) \cdot \mathbf{Q'_m}(\nu), & if\ \mathbf{f'}_p(\nu) \neq 0,\ \mathbf{f'}_p(\nu) = 0. \end{cases} \tag{3.8}$$

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

$$(\grave{\mathbf{F}}_{\mathrm{p}}(\nu), \grave{\mathbf{F}}_{\mathrm{p}}(\nu)) = \begin{cases} (\mathbf{f'}_p(\nu),\ \mathbf{f'}_q(\nu) + sgn(\frac{\mathbf{F}_q(\nu)}{\mathbf{Q'_m}^{(\nu)}} - \mathbf{f'}_q(\nu)) \cdot \mathbf{Q'_m}(\nu), \\ \qquad\qquad if\ |\frac{\mathbf{F}_p(\nu)}{\mathbf{Q'_m}^{(\nu)}} - \mathbf{f'}_p(\nu)| \geq |\frac{\mathbf{F}_q(\nu)}{\mathbf{Q'_m}^{(\nu)}} - \mathbf{f'}_q(\nu)|, \\ (\mathbf{f'}_p(\nu) + sgn(\frac{\mathbf{F}_p(\nu)}{\mathbf{Q'_m}^{(\nu)}} - \mathbf{f'}_p(\nu)),\ \mathbf{f'}_q(\nu)) \cdot \mathbf{Q'_m}(\nu), \\ \qquad\qquad if\ |\frac{\mathbf{F}_p(\nu)}{\mathbf{Q'_m}^{(\nu)}} - \mathbf{f'}_p(\nu)| < |\frac{\mathbf{F}_q(\nu)}{\mathbf{Q'_m}^{(\nu)}} - \mathbf{f'}_q(\nu)|. \end{cases} \tag{3.9}$$

Eq. (3.8) is used to avoid large modifications on the AC coefficients in the "flat" blocks. Eq. 3.9 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.

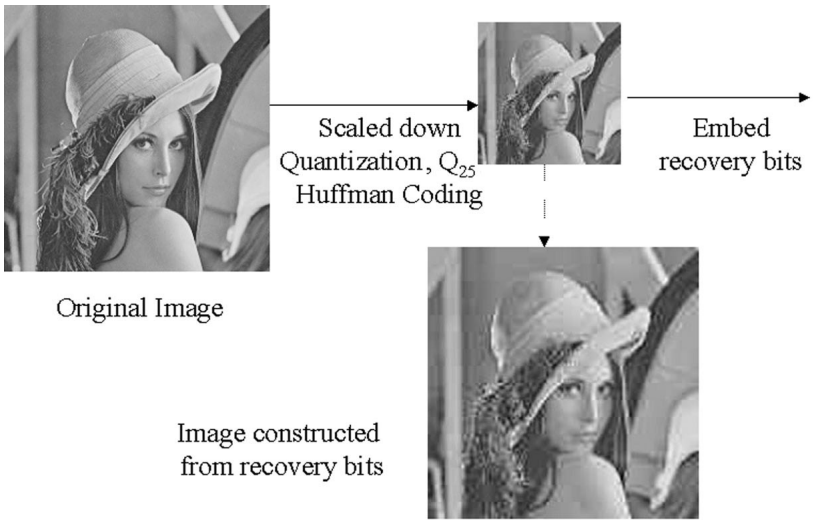



Figure 3-2: An example of embedding recovery bits

### 3.3.2 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. Figure 3-2 shows an example of the process for embedding recovery bits. We can see the quality of an image that is constructed from all recovery bits.

To generate the recovery bits, $\mathbf{\Psi}$, we first scale down the image by 2 along each axis, and divide the image into $8 \times 8$ 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 $\mathbf{Q'_{m_r}}(\nu) = \mathbf{Q_{m_r}}(\nu)$ 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 $\beta_r$ positions in a block. These parameters are image independent. Then, recovery bits are embedded in a similar way as in Eq. (3.7) (or Eq. (3.8) and Eq. (3.9)). 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.

### 3.3.3 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 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.

## 3.4 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*, $P_S$, as discussed in Chapter 2. The first two are from the authentication system developer. The last one is from the viewpoints of attacker. The last two are distinguished based on different information known to the developer 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

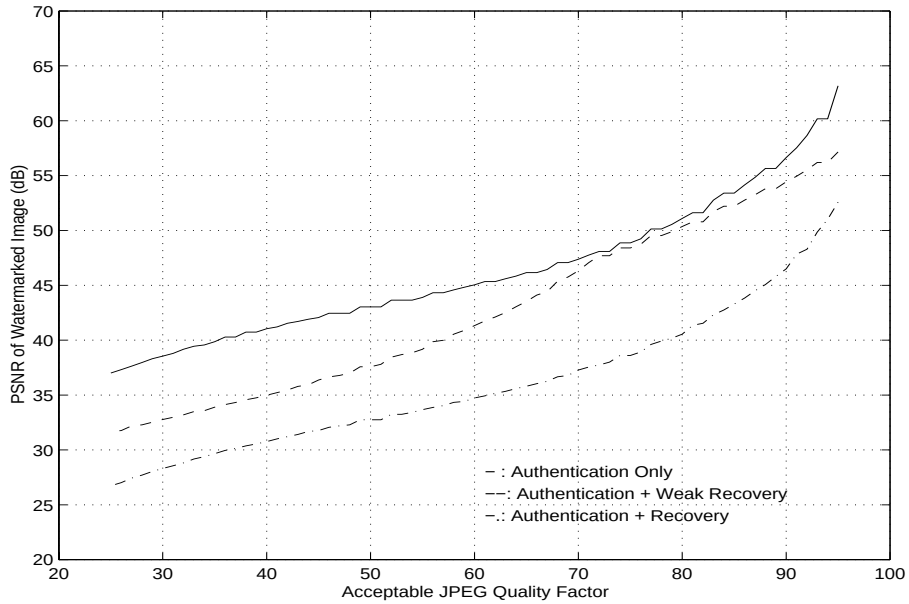



Figure 3-3: Expected value of PSNR of the watermarked image v.s. Acceptable JPEG Quality Factor. The embedded bits are: (1) Authentication Only: 3 bits/block, (2) Authentication + Weak Recovery: 9 bits/block , and (3) Authentication + Recovery: 9 bits/block.

implementation described in Eq. (3.6) and Eq. (3.7).

### 3.4.1 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,

$$E[{\sigma_{\mathbf{w}}}^2(\nu)] = \frac{1}{2} \cdot \int_0^{\mathbf{Q'_m}(\nu)} x^2 f(x) dx + \frac{1}{2} \cdot \int_0^{\mathbf{Q'_m}(\nu)} (\mathbf{Q'_m}(\nu) - x)^2 f(x) dx = \frac{1}{3} {\mathbf{Q'_m}}^2(\nu), \tag{3.10}$$

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

Then the expectation value of PSNR of a watermarked image is,

$$E[PSNR] = 10log_{10}\frac{64 \cdot 255^2}{\sum_{\nu_i \in \mathbf{E}} E[{\sigma_\mathbf{w}}^2(\nu_i)]}. \tag{3.11}$$

Applying Table 3.1, 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 3-3. In Figure 3-3, 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 3-3, we set the quality factor for weak recovery to be 25 larger than that for authentication.

### 3.4.2 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 4 and Theorem 2, the probability of false alarm,

$$P_{FA} = 0, \tag{3.12}$$

if the image goes through by the JPEG lossy compression. In practical systems, Eq. (3.12) 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 $\sigma_N^2$ instead of JPEG compression, then the probability of false alarm in a block pair,

$$P_{FA} = 1 - (1 - P_{FA,\mathbf{E}})(1 - P_{FA,\mathbf{B}}) \approx P_{FA,\mathbf{E}} + P_{FA,\mathbf{B}} \tag{3.13}$$

where

$$P_{FA,\mathbf{E}} = 1 - \prod_{\nu \in \mathbf{E}} [1 - \sum_{i=0}^{\infty} [erfc(\frac{(\frac{1}{2} + 2i)\mathbf{Q}'\mathbf{m}(\nu)}{\sqrt{2}\sigma_N}) - erfc(\frac{(\frac{3}{2} + 2i)\mathbf{Q}'\mathbf{m}(\nu)}{\sqrt{2}\sigma_N})]]. \tag{3.14}$$

where $erfc()$ is the complementary error function[58]. And

$$P_{FA,\mathbf{B}} = 1 - \prod_{\nu \in \mathbf{E}} [1 - \frac{1}{2} erfc(\frac{|\mathbf{\Delta F}_{p,q}(\nu)|}{2\sigma_N})]. \tag{3.15}$$

We can see that $P_{FA,\mathbf{E}}$ is image independent, but $P_{FA,\mathbf{B}}$ is image dependent. For instance, if we set $\mathbf{Q}'\mathbf{m} = \mathbf{Q_{50}}$ 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 $256 \times 256$ "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 (as in Chapter 2) and set the bound equal to $\mathbf{Q}'\mathbf{m}$, then $P_{FA,\mathbf{B}} = 9 \times 10^{-9}$ which decreases significantly.

### 3.4.3 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 point of view. In other words, that is a probability that the content distributor knows how each specific watermarked image may miss a manipulation. $P_{S,p}$ is the probability that, from the attacker's point of view, his estimation of successful attack without the information of $\mathbf{F}_p$, $\mathbf{F}_q$, and $\mathbf{B}_p$ but with a given attack scenario. These scenarios may 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.* Detailed discussion and derivation of these two probabilities are in Chapter 2.[2]

In this chapter, 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) without given any attack scenario. Here,

$$P_S \approx 2^{-\frac{3}{2}\cdot\beta_a\cdot N} \tag{3.16}$$

where N is the number of $8 \times 8$ blocks that are affected by the attack. If each block

[2]However, a complete derivation of $P_M$ and $P_S$, that includes the probability modeling of embedding bits, is not yet addressed in this thesis.

is protected by $\beta_a$ authentication bits, and $\frac{1}{2}\beta_a$ authentication bits from other coefficient pairs are embedded in this block, then changing the pixel values in the block will influence $\frac{3}{2}\beta_a$ bits totally. Assuming random pixel changes and the coefficient pairs are unknown, each manipulation may cause change of each authentication bit with a probability of $\frac{1}{2}$. Then, with $\frac{3}{2}\beta_a$ bits involved, the probability of successful attack ($P_S$) is approximately $2^{-\frac{3}{2}\cdot\beta_a}$. If $\beta_a = 6$, then $P_S$ is $\approx 2^{-9}$. In practical, because manipulation may cross the block boundary, if an attacker replace an area of $20 \times 20$, which may affect 16 blocks, then $P_S \approx 2^{-9\times16} \approx 10^{-43}$. Practically, Eq. (3.16) is an conservative estimation of the probability of successful attack (from attacker's point of view), because the probability that a manipulated coefficient pair pass the authenticator may be larger than $\frac{1}{2}$, and is image dependent (as discussed in Chapter 2).

### 3.4.4 Security

Security of the SARI system is achieved by using secret mapping functions, $T_a$ and $T_b$, as described in Section 3.3. Using a fixed and not publicized algorithm to generate $T_a$ and $T_b$ in both watermark embedder and authenticator, early versions of SARI system provide preliminary level of security against attacks. However, because the mapping functions may be found and maliciously published, this strategy is not secure enough. To meet the generic security requirements, we enhanced our system based on a secret mapping function generator and the common Public Key Infrastructure (PKI).

A secret function, $\Upsilon$, which is a hardware or software component in both watermark embedder and authenticator, is used to generate the mapping functions based on the seeds which includes user identification, time stamp, and a random

seed number generated by the embedder. That is,

$$\{T_a, T_b\} = \Upsilon\{EmbedderID + TimeStamp + RandomSeed\} \tag{3.17}$$

where $\Upsilon$ is designed by the system provider. It is similar to a pseudo number generator plus additional considerations of properties of mapping functions, *e.g.*, the minimum distance between locations where the information bits are generated from and embedded to. In real applications, the three input parameters of $\Upsilon$ have to be transmitted to the authenticator through the watermark. This is done by generating the third type of information bits, $\lambda$, based on,

$$\begin{aligned} \lambda_{\mathbf{i}} = \quad & PublicKey_{Auth}\{EmbedderID \\ & + PrivateKey_{Embedder}\{TimeStamp + RandomSeed + Hash[\hat{\mathbf{\Phi}}_{\mathbf{i}}]\}\} \end{aligned} \tag{3.18}$$

where $\lambda_{\mathbf{i}}$, called the security bits, are embedded into the $i-th$ non-overlapping section of the image. Here, we divide the image into several sections, and embed 1 bit of security bits into each block of the section. Each $\lambda_{\mathbf{i}}$ is generated based on the user ID, a time stamp, a random seed, and a hash of the embedded authentication bits, $\hat{\mathbf{\Phi}}_{\mathbf{i}} = \cup_{T_a(p) \in section\ i} \Phi_p$, in the section $i$. A section may include one or several rows, whose number depends on the image size and the length of security bits. Here, the purpose of generating multiple sets of security bits is for error resilience.

The design philosophy of the security system is based on three static security entities, the private key of the authenticator, the private key of the watermark embedder, and the mapping function generator $\Upsilon$. A secret dynamic information, the random seed, is generated by the embedder and is transmitted from the embedder to the authenticator. This secret random seed is used along with other transparent information of the embedder ID and the time stamp to generate the mapping functions. In Eq. (3.17) and (3.18), the time stamp does not play a security role, which

is used here simply to indicate the fact that other information can also be added here. If system users need not know the embedding time, it can be deleted in both equations. In SARI system, the authenticator would retrieve the embedder ID and the time stamp to the system user as part of authentication services. In other words, these information would be transparent to users. In Eq. (3.17), we include the hash values of the embedded authentication bits to make $\lambda_i$ content-dependent, *i.e.*, it cannot be copied from authentic images in faking the authenticity of other images. Note that, in Chapter 2, we cannot use the hash values of authentication bits in a robust digital signature, because the DCT relationships may introduce ambiguity in some "equal" cases. However, we can use the hash values of authentication bits here, because they are not used for content authentication and the embedded authentication bits can be exactly reconstructed from the watermarked SARI image.

The authentication process begins as follows. First, the authenticator decrypts the security bits of each section and extract the embedded ID, the time stamp, the random seed, and the embedded hash values. Only the intended authenticator devices or software systems can decrypt $\lambda_\mathbf{i}$ since it requires the matched private key. Once the embedder ID is known, the public key of the embedder can be used to extract the information inside the inner most bracket in Eq. (3.18). Without malicious manipulation, these parameters should be the same in all sections. If there is manipulation, they are determined based on the majority of the decrypted $\lambda_\mathbf{i}'s$. This is why we divide image into sections for reduplicate embedding. We should notice that, most cryptographic methods do not allow any single bit change in the cipher text. Therefore, only those sections that keep integrity of $\lambda_\mathbf{i}$ can provide correct parameters. After the parameters is retrieved, the authenticator can apply Eq. (3.17) to generate mapping functions and authenticate images as described in Section 3.3.3.

Adding the private key of the embedder in Eq. (3.18) is to avoid attacker's forgery of false input of $\Upsilon$. Without this digital signature-like functionality, once an attacker has found a relation between mapping functions and its corresponding inputs, he will be able to generate mapping functions as in Eq. (3.17) and embed authentication watermarks to forged images. Adding the hash of the authentication bits is to generate diverse and content-based security bits in the sections. It can also help to prove the integrity of the section, that sometimes is useful in helping allocating the manipulated positions.

Adding the public key of the authenticator in Eq. (3.18) is to protect the parameters from being known by the attacker. In the real cases, we assume that $\Upsilon$ is not publicized. Without the public key in Eq. (3.18), this system can still be secure. However, since $\Upsilon$ is a component in the hardware or software of embedder, it may be reverse engineered. In such a scenario, we need to protect the input parameters of $\Upsilon$ from being known, otherwise the attacker can use some hacked parameters to forge watermarks. In Eq. (3.18), only the intended authenticator can retrieve the parameters. Specifying which authenticator would authenticate the image a priori is a feasible solution in practice. For example, authentication services can be deployed on the World Wide Web (WWW) by the same watermark embedder provider or trusted agents. This additional public key enhances the security of the authentication system with the trade-off that the authenticator could not be locally operated by the recipient.

We trust the unknown private keys to provide the ultimate security protection. If they are known by the attacker, but the mapping function generator, $\Upsilon$ is still secret, the authentication system may be secure, unless the parameters and corresponding mapping functions have been found. At the end, if both the private keys and the mapping function generator, $\Upsilon$, are all known by the attacker, then the

(a) (b) (c)

Figure 3-4: (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).

authentication system is no longer secure.

## 3.5 Experimental Results

### 3.5.1 Example

We first use the $256 \times 256$ gray-level "Lenna" image to test our system. The original and watermarked images are shown in Figure 3-4. 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. (3.8) and Eq. (3.9). In Figure 3-4(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 3-3, because this enhanced method modifies more bits in each block. In Figure 3-4(c), we show the watermarked image with both authentication bits and weak recovery bits. For each block, in addition to the authentication bits, several recovery bits are embedded. These recovery bits survives JPEG compression to QF =75. There is visible distortion after embedding, but overall, the degradation is

(a) (b) (c)

Figure 3-5: (a) Manipulation on the watermarked image in Figure 3-2(b), (b) the authentication result of (a), (c) the authentication and recovery result from the manipulated image of Figure 3-2(c).

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 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 ma-

Figure 3-6: Image test set for SARI benchmarking

nipulated in a similar way by deleting the pin of Lenna's hat. The image manipulated from Figure 3-4(b) is shown in Figure 3-5(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 3-5(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 3-4(c)) if recovery bits are used.

### 3.5.2 Benchmarking

Benchmarking[3] of SARI utilizes common software and is performed from a consumer's perspective. We tested 9 images that belong to 5 different categories: Human (Lenna and Miss Tokiyo), Natural Scene (cafe and library), Still Objects (fruit and clock), Synthetic (reading and strike), and Document (insurance). These images are shown in Figure 3-6.

Our interests include image quality after watermark embedding, robustness of authentication bits to JPEG compression, authentication sensitivity to malicious

[3]This benchmarking represents joint work with L. Xie, K. Maeno and Q. Sun [144].

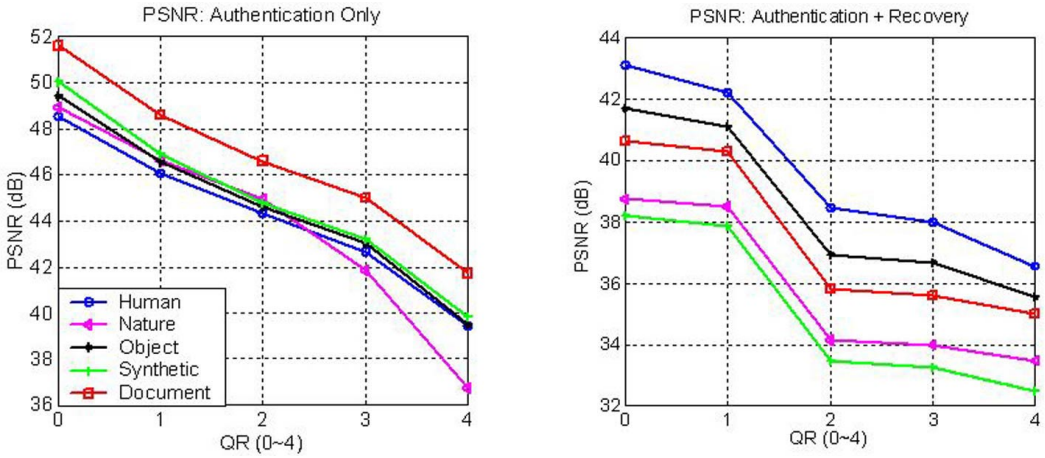



Figure 3-7: Average PSNR for different image types after watermarking

| | | |
|---|---|---|
| Viewer No.1 | image-processing expert | Trinitron 17' monitor |
| Viewer No.2 | image-processing expert | Sony Laptop LCD monitor |
| Viewer No.3 | no image-processing background | Trinitron 17' |
| Viewer No.4 | image-processing expert | Trinitron 17' monitor |

Table 3.2: Viewers' in the SARI subjective visual quality test

manipulation such as crop-and-replace. We show the benchmarking result in the following subsections. A complete benchmarking report of SARI system can be found in [144].

#### 3.5.2.1 Image Quality Test

There are five QR (Quality and Recovery) modes operated in a SARI system. They are fixed quality parameters predetermined by the system designer. Usually, $QR = 0$ is means the embedded watermark introduces minimum quality degradation, but has the lowest robustness toward compression. An objective test on the watermarked images is shown in Figure 3-7.

In addition to objective PSNR tests, we also conducted subjective tests to examine the quality of watermarked image toward human observers. Four viewers are used for this test. Their background and monitor types are listed in Table 3.2.

| Image Name | Lena | Tokiyo | Cafe | Library | Fruit |
|---|---|---|---|---|---|
| Image Type | Color | Color | Color | Color | Color |
| Image Size | 512*512 | 768*960 | 480*592 | 560*384 | 400*320 |
| Embedded Bits, Auth | 12,288 | 34,560 | 13,320 | 10,080 | 6,000 |
| Embedded Bits, A+R | 47,240 | 109,514 | 88,751 | 52,868 | 24,616 |
| Max Invisible QR, Auth | 3 | 3 | 4 | 2 | 4 |
| Max Invisible PSNR, Auth | 43.0 | 42.3 | 40.2 | 45.0 | 39.8 |
| Max Invisible QR, A+R | 1 | 1 | 3 | 1 | 3 |
| Max Invisible PSNR, A+R | 41.9 | 42.5 | 33.2 | 39.3 | 36.9 |

| Clock | Reading | Strike | Insurance |
|---|---|---|---|
| Gray | Color | Color | Color |
| 256*256 | 336*352 | 256*192 | 792*576 |
| 3,072 | 5,544 | 2,304 | 21,384 |
| 11,686 | 34,033 | 10,474 | 90,968 |
| 3 | 2 | 3 | 3 |
| 44.7 | 42.5 | 43.8 | 45.0 |
| 0 | 0 | 1 | 1 |
| 36.2 | 34.2 | 39.6 | 41.3 |

Table 3.3: SARI embedded bits and max invisible (MI) embedding strength referring to Subjective test. (A+R: embedding authentication and recovery bits, Auth: embedding authentication bits)

We use the average of subjective tests to show the maximum embedding strength for each image. This is shown in Table 3.3. From this table, we can see the number of bits embedded in each image. The number of authentication bits per $8 \times 8$ block is 3 bits, and the average number of recovery bits are 13.1 bits/block. We can also see that the maximum acceptable QR or PSNR vary according different image type. Through the objective and subjective tests, we observed that:

1. The changes are almost imperceptible for modest watermark strength $QR = 0 - 2$.

2. The embedding capacity of a natural image is generally larger than that of a synthetic image. This is because the former has more textural areas, thus the slight modification caused by authentication bits is less visible. The image quality of human, nature, and still object is generally better than that of

| Image Name | Lena | Tokiyo | Cafe | Library | Fruit | Clock | Reading | Strike | Insur. |
|---|---|---|---|---|---|---|---|---|---|
| Survive QF, MED | 3 | 3 | 3 | 4 | 1 | 4 | 3 | 3 | 4 |
| Survive QF, QR=4 | 1 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 |
| Detect M., 1-pix | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| Detect M., C&R | Y | Y | Y | Y | Y | Y | Y | Y | Y |

Table 3.4: SARI performance test under JPEG compression Quality Factor (in Photoshop 5.0) and Crop-Replacement (C&R) Manipulations (MED: watermarks are embedded under maximum invisible embedding strength)

synthetic and document image, and both the objective and subjective tests agree at this point.

3. The quality judgments vary among different viewers. This is because users pay attention to different features of an image and their tolerance bounds can be quite different. Moreover, different types of monitors have different display effects, e.g. the images that appear not acceptable on a Dell PC look just fine on a Sun Workstation.

#### 3.5.2.2 Sensitivity Test

Table 3.4 shows the benchmarking result of performance test. We tested the robustness against JPEG lossy compression by embedding the watermarks in two different QR modes. For JPEG Compression, we found that all the information bits embedded in the image can be exactly reconstructed without any false alarm after JPEG compression. We observed similar results from other JPEG testing using XV, PhotoShop 3.0, PaintShop Pro, MS Paint, ACD See32, Kodak Imaging, *etc.* Statistics here conform with the designed robustness chart (QR $0-4$). For instance, for image Lena, watermark with strength $QR = 4$ survives Photoshop 5.0 Quality Factor $1-10$. Watermarks embedded by using maximum invisible subjective embedding strength (MED) can survive JPEG compression Quality Factor $3-10$. This result is even better than predicted.

We embedded the watermarks in the $QR = 4$ mode to test its sensitivity to malicious manipulations. $QR = 4$ is the most robust mode to compression and is the least sensitive mode in detecting manipulations[4]. We found that even in this case, SARI authenticator is quite sensitive to this kind of manipulation. It can properly detect up to 1-pixel value changes, and it is very effective in detecting Crop-and-Replacement manipulations. In our tests, it properly detects all manipulations.

For recovery tests, we found that in all malicious manipulation cases, an approximation of the original pixels in the corrupted area can be properly reconstructed.

In addition to these two manipulations, we had also tested other image processing manipulations for reference. We found that the authenticator can detect the change resulted by blurring, median filtering. For Gaussian noises, the authenticator detects changes. But, if further compressed by JPEG, usually no change were detected because compression cancelled out the slight difference introduced by it. We also found that the robustness of noises or filtering can be increased through setting larger tolerance bound in the authentication process (the defaulted tolerance bound, $\tau$, is equal to 1). However, as discussed early in Chapter 2, the authenticator will be less sensitive to malicious manipulations.

## 3.6 Conclusion and Future Direction

In this chapter, 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 re-

[4]We use $QR = 2$ for the Insurance image because the visual degradation of $QR = 4$ is clearly visible.

cover 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. Our future research in this area includes: (1) considering 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.

## 3.7 Proof of Theorem 4 in Chapter 3

**Proof 4** *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,*

$$\mathbf{F}_p(\nu) - \frac{1}{2}\mathbf{Q}(\nu) \leq \tilde{\mathbf{F}}_{\mathbf{p}}(\nu) \leq \mathbf{F}_p(\nu) + \frac{1}{2}\mathbf{Q}(\nu). \tag{3.19}$$

*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. (3.19), the quantization result,* $\tilde{\mathbf{F}}'_p(\nu)$*, will be,*

$$\grave{\mathbf{F}}_p(\nu) - \frac{1}{2}\mathbf{Q}(\nu) \leq \tilde{\mathbf{F}}'_p(\nu) \leq \grave{\mathbf{F}}_p(\nu) + \frac{1}{2}\mathbf{Q}(\nu). \tag{3.20}$$

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

$$c \cdot \mathbf{Q}'_{\mathbf{m}}(\nu) - \frac{1}{2}\mathbf{Q}'_{\mathbf{m}}(\nu) < \tilde{\mathbf{F}}'_{\mathbf{p}}(\nu) < c \cdot \mathbf{Q}'_{\mathbf{m}}(\nu) + \frac{1}{2}\mathbf{Q}'_{\mathbf{m}}(\nu). \tag{3.21}$$

*If we quantize* $\tilde{\mathbf{F}}'_{\mathbf{p}}(\nu)$ *again using* $\mathbf{Q}'_{\mathbf{m}}(\nu)$, (i.e., *dividing all coefficients in Eq. (3.21) 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*

$$Integer\ Round(\frac{\tilde{\mathbf{F}}'_p(\nu)}{\mathbf{Q}'_{\mathbf{m}}(\nu)}) \cdot \mathbf{Q}'_{\mathbf{m}}(\nu) = c \cdot \mathbf{Q}'_{\mathbf{m}}(\nu) = \grave{\mathbf{F}}_p(\nu), \tag{3.22}$$

*which proves part of Theorem 4 in the case that* $\mathbf{Q}(\nu) < \mathbf{Q}'_{\mathbf{m}}(\nu)$.

*Assume another case that* $\mathbf{Q}(\nu) = \mathbf{Q}'_{\mathbf{m}}(\nu)$, i.e., $\grave{\mathbf{F}}_p(\nu)$ *is pre-quantized to the integer times of* $\mathbf{Q}'_{\mathbf{m}}(\nu)$. *In this case, the DCT coefficient is pre-quantized and quantized using the same quantization step. Therefore, the value will be exactly the same after quantization. This proves another part of Theorem 4 in the case of* $\mathbf{Q}(\nu) = \mathbf{Q}'_{\mathbf{m}}(\nu)$.

□

# Chapter 4

# Geometric Distortion Resilient Public Watermarking and Its Applications in Image Print-and-Scan Processes

## 4.1 Introduction

Today the print-and-scan (PS) process is commonly used for image reproduction and distribution. It is popular to transform images between the electronic digital format and the printed format. The rescanned image may look similar to the original, but may have been distorted during the process. For some image security applications, such as watermarking for copyright protection, users should be able to detect the embedded watermark even if it is printed-and-scanned. In image authentication cases, the rescanned image may be considered as authentic, because it is a reproduction of the original.

After the print-and-scan process, distortion occurs in both the pixel values and the geometric boundary of the rescanned image. The distortion of pixel values is caused by (1) the luminance, contrast, gamma correction and chromnance variations, and (2) the blurring of adjacent pixels. These are typical effects of the printer and scanner, and while they are perceptible to the human eye, they affect the visual

quality of a rescanned image.

This chapter includes two parts. In the first part, we propose a public watermark technqiue that is invariant to geometric distortions. In the second part, we propose a model of the print-and-scan (PS) process and present how we can modify the proposed watermarking technique to be applied in the PS process.

In Section 4.2, we make a brief introduction on the properties of the proposed watermark technique and an overview on the watermarking techniques that survive geometric distortions. In Section 4.3, we describe our watermarking algorithm. This is described in more detail in Section 4.4, including the iterative procedure for dealing with the one-to-many mapping from watermark to image space. Our solutions to a number of implementation issues are also discussed in Section 4.4. Section 4.5 describes the results of experiments on a large database.

In the second part of this chapter, we begin with the discussion of the characteristics of the PS process in Section 4.6. Then, in Section 4.7, we propose a model that can be used to analyze the distortion of a discretized digital image after the PS process in the spatial and frequency domain. Then, we will analyze the variations of DFT coefficients, leading to important properties for extracting invariants. In Section 4.8, we discuss several methods that can be used to extract invariants of the PS process. Some experimental results, including an analysis of the feature vector proposed in Section 4.3, are shown in Section 4.9. In Section 4.10, we make a conclusion and discuss some future work.

## 4.2 Properties of the Proposed Watermarking Technique

In this chapter, we propose a public watermark technique that is invariant to geometric distortions. Our method does not embed an additional registration pattern [103, 30] or embed watermark in a recognizable structure [68], so there is no need

to identify and invert them. In particular, we are concerned with distortions due to rotation, scale and/or translation (RST). While these geometric distortions have recently become of interest to the watermarking community, they have long been of interest to the pattern recognition community. A comprehensive discussion of the pattern recognition literature is outside the scope of this chapter. Hu [49] described the use of moment invariants for visual pattern recognition of planar geometric figures. It has been shown [120] that these classic moment invariants are equivalent to the radial moments of circular-harmonic functions (CHF's) that arise from a Mellin transform of the log-polar representation of an image when the complex Mellin radial frequency $s$, is a real integer $s \geq 1$.

The Fourier-Mellin transform is closely related to the algorithm described in this chapter. There are a variety of related ideas from pattern recognition. First, Casasent and Psaltis [16, 17] note that the signal-to-noise ratio of the correlation peak between two images decreases from 30db to 3dB with either a 2% scale change or a 3.5° rotation. Their proposal is essentially a hybrid opto-electronic implementation of the Fourier-Mellin transform. Altmann and Reitbock [4] and Altmann [5] discuss implementation issues related to the discrete Fourier-Mellin transform. These include interpolation, aliasing, and spectral border effects, which are discussed in detail in Section 4.4 of this chapter. Wechsler and Zimmerman [137] describe a conformal-log mapping that is very closely related to the Fourier-Mellin transform. Also, Lin and Brandt [86] discuss the use of the Fourier-Mellin and other transforms for pattern recognition. Lin and Brandt [86] describe a number of *absolute* or *strong* invariants based on the phase of the Fourier or Fourier-Mellin spectrums. The terms "absolute" and "strong" refer to the fact that all information about an image except that of position, orientation or scale is preserved. This may be important for recognition tasks, especially if the library of objects is large. Ferraro and Caelli [41]

discuss this issue in more detail. However, we do not believe that strong invariants are necessary for watermarking applications.

It is important to realize that watermark detection is different from the general problem of recognizing an object. First, an $N$-bit watermark only requires recognition of $N$ independent patterns. Since $N$ is typically between 32 and 64, this is considerably smaller than a practical object recognition database. Second, the watermark is not a naturally occurring object but is artificially inserted into an image. As such, the watermark can be designed to be easily represented. In particular, it is often advantageous to represent the watermark as a one-dimensional projection of the image space. If properly designed, this has the benefit of reducing a two-dimensional search to one dimension, thereby significantly reducing the computational cost. Finally, since the set of watermarks is small (compared with the number of naturally occurring objects in a scene) and artificially created, it is not necessary that the image transform be strongly invariant as it is not as important to be able to reconstruct the image modulo rotation, scale and/or translation from the parameterization.

O'Ruanaidh and Pun [100] first suggested a watermarking method based on the Fourier-Mellin transform. However, they note very severe implementation difficulties which we suspect have hampered further work in this area. They choose to use a transformation that is strongly invariant claiming that "it is more convenient to use strong invariants because the last stage of embedding a mark involves inverting the invariant representation to obtain the (marked) watermarked image". We believe that invertibility is not essential. Following the formulation in [26], suppose we have a non-invertible extraction function, $X(C)$, that maps a piece of media, $C$, into an extracted signal. Such a function would be used as part of a detection strategy. An

example extraction function found in the literature[62] is

$$X_i(C) = \sum_{j \in R_i} C(j) \qquad 1 \leq i \leq N \tag{4.1}$$

where $R_i$ are disjoint subsets of elements of the media, $C$. We can often define an embedding function, $Y(w, C)$, which finds a new piece of media, $C_w = Y(w, C_o)$, such that

$$X(C_w) \equiv X\left(Y(w, C_o)\right) \approx w \tag{4.2}$$

and $C_w$ is perceptually similar to $C_o$. In other words, the watermarked image looks like the original and the vector extracted during detection *looks like* the watermark vector. This function is sufficient for use in a watermark embedder. Our public watermarking algorithm differs from that of [100] in two primary ways. First, we choose to watermark a projection of the transform space. Second, the watermark embedding is based on the principle of communication with side information [26].

There have been a number of other recent watermarking algorithms designed to deal with geometric distortions. Of particular note is the recent work of Bas *et al* [9]. They describe an algorithm based on the detection of salient features in an image and the insertion of signals relative to these salient features. Experimental results indicate that the method is robust to mirror reflection and rotation. However, surprisingly, the system fails to survive other geometric distortions. A somewhat related set of methods is described by Maes and van Overveld [90] and Rongen *et al* [110]. These methods are based on geometrically warping local regions of an image onto a set of random lines. However, currently, these methods are not robust to geometric distortions, but rather, allow for a rapid, but exhaustive search through the possible set of geometric distortions.

## 4.3 Algorithm

Consider an image $i(x,y)$ and a rotated, scaled, and translated (RST) version of this image, $i'(x,y)$. Then we can write

$$i'(x,y) = i\left(\sigma\left(xcos\alpha + ysin\alpha\right) - x_0, \sigma\left(-xsin\alpha + ycos\alpha\right) - y_0\right) \tag{4.3}$$

where the RST parameters are $\alpha$, $\sigma$, and $(x_0, y_0)$ respectively.

The Fourier transform of $i'(x,y)$ is $I'(f_x, f_y)$, the magnitude of which is given by:

$$|I'(f_x,f_y)| = |\sigma|^{-2}\left|I\left(\sigma^{-1}(f_xcos\alpha + f_ysin\alpha), \sigma^{-1}(-f_xsin\alpha + f_ycos\alpha)\right)\right|. \tag{4.4}$$

Equation 4.4 is independent of the translational parameters, $(x_0, y_0)$. This is the well known translation property of the Fourier transform [12].

If we now rewrite Equation 4.4 using log-polar coordinates, i.e.

$$f_x = e^{\rho}cos\theta \tag{4.5}$$

$$f_y = e^{\rho}sin\theta \tag{4.6}$$

then the magnitude of the Fourier spectrum can be written as

$$\begin{aligned} |I'(f_x,f_y)| &= |\sigma|^{-2}\left|I\left(\sigma^{-1}e^{\rho}cos(\theta-\alpha), \sigma^{-1}e^{\rho}sin(\theta-\alpha)\right)\right| && (4.7)\\ &= |\sigma|^{-2}\left|I\left(e^{(\rho-\log\sigma)}cos(\theta-\alpha), e^{(\rho-\log\sigma)}sin(\theta-\alpha)\right)\right| && (4.8) \end{aligned}$$

or

$$|I'(\rho,\theta)| = |\sigma|^{-2}\left|I\left(\rho - \log\sigma, \theta - \alpha\right)\right|. \tag{4.9}$$

Equation 4.9 demonstrates that the amplitude of the log-polar spectrum is scaled by $|\sigma|^{-2}$, that image scaling results in a translational shift of $\log \sigma$ along the $\rho$ axis, and that image rotation results in a cyclical shift of $\alpha$ along the $\theta$ axis.

We need not be concerned with the amplitude scaling of the spectrum, since we intend to perform watermark detection using the correlation coefficient, which is invariant to this scaling. See Section 4.3.1 for more details.

Next, we define $g(\theta)$ to be a one-dimensional projection of $|I(\rho, \theta)|$ such that

$$g(\theta) = \sum_j \log\left(|I(\rho_j, \theta)|\right). \tag{4.10}$$

The reason for summation of the log values rather than the magnitudes themselves is discussed in Section 4.4.4. Due to the symmetry of the spectra of real images,

$$|F(x, y)| = |F(-x, -y)|, \tag{4.11}$$

we only compute $g(\theta)$ for $\theta \in [0°...180°)$.

We find it convenient to add the two halves of $g(\theta)$ together, obtaining

$$g_1(\theta') = g(\theta') + g(\theta' + 90°) \tag{4.12}$$

with $\theta' \in [0°...90°)$. The reasons for this are discussed in section 4.4.6.

Clearly, $g_1(\theta)$, is invariant to both translation and scaling. However, rotations result in a (circular) shift of the values of $g_1(\theta)$. If $\theta$ is quantized to the nearest degree, then there are only 90 discrete shifts, and we handle this by an exhaustive search.

### 4.3.1 Watermark detection process

In principle, detectors may be built that can handle watermarks encoding several bits. However, the present detector determines only whether or not a given watermark has been embedded in a given image. It takes as input, an image and a watermark and the output is a single bit indicating whether the image contains the watermark.

The watermark is expressed as a vector of length $N$. To determine whether the watermark is present, an "extracted signal" $v = g_1(\theta)$ is computed from the image, for $N$ values of $\theta$ evenly spaced between 0° and 90°. The extracted signal is then compared to the watermark using the correlation coefficient. If the correlation coefficient is above a detection threshold $T$, then the image is judged to contain the watermark.[1]

Thus, the basic algorithm for watermark detection proceeds as follows:

1. Compute a discrete log-polar Fourier transform of the input image. This can be thought of as an array of $M$ rows by $2N$ columns, in which each row corresponds to a value of $\rho$, and each column corresponds to a value of $\theta$.

2. Sum the logs of all the values in each column, and add the result of summing column $j$ to the result of summing column $j + N$ $(j = 0...N - 1)$ to obtain an invariant descriptor $v$, in which

$$v_j = g_1(\theta_j) \tag{4.13}$$

   where $\theta_j$ is the angle that corresponds to column $j$ in the discrete log-polar Fourier transform matrix.

[1]The use of correlation coefficient as a detection measure is recommended in [26]. One benefit of this metric is its independence to scaling of the signal amplitudes.

3. Compute the correlation coefficient $D$, between $v$ and the input watermark vector $w$, as

$$D = \frac{w \cdot v}{\sqrt{(w \cdot w)(v \cdot v)}} \tag{4.14}$$

4. If $D$ is greater than a threshold $T$, then indicate that the watermark is present. Otherwise, indicate that it is absent.

### 4.3.2 Watermark embedding process

Once a method for detecting watermarks has been defined, we can construct a watermark embedding algorithm which is similar to the methodology described in [26]. In that paper, watermarking is cast as a case of communications with side information at the transmitter, which is a configuration studied by Shannon [118]. The difference between this view of watermarking, and a more common view, is as follows.

In most public watermarking methods found in the literature, the original image is considered to be noise. The embedder adds a small-amplitude signal to this noise, and the detector must be sensitive enough to work with the small signal-to-noise ratio that results.

However, this common approach ignores the fact that the embedder has complete knowledge of the "noise" caused by the original image. If we view the embedder as a transmitter and the cover image as a communications channel, then this knowledge amounts to side-information about the behavior of that channel. When the transmitter knows ahead of time what noise will be added to the signal, its optimal strategy is to subtract that noise from the signal before transmission. The noise then gets added back by the communications channel, and the receiver receives a perfect reconstruction of the intended signal.

In the case of watermarking, it is unacceptable for the embedder to subtract

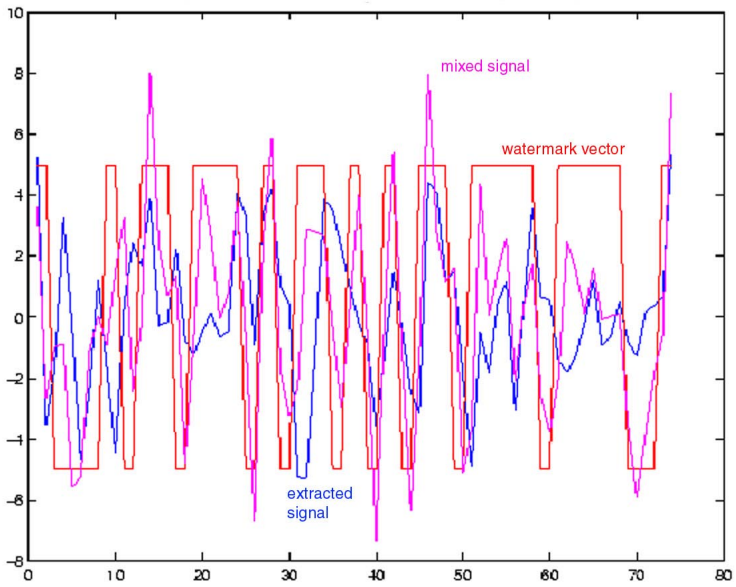



Figure 4-1: An example of feature vector shaping: the extracted signal is iteratively shaped to the mixed signal, according to the watermark signal

the original image from the watermark before embedding the watermark, because it would result in unacceptable fidelity loss. In fact, if the watermark is expressed as a pattern that is the same size as the image, then this strategy simply replaces the image with the watermark pattern, which is clearly too drastic. However, when the watermark is expressed as a signal in a lower-dimensional space, as is the case with the present system, the results need not be so drastic, since a wide variety of full-resolution images project into the same extracted signal and the embedder may choose the one that most resembles the original. But even in the case of lower-dimensional watermarks, it is not always possible to completely replace the extracted signal with the watermark signal while maintaining acceptable fidelity.

To make maximal use of the side-information at the embedder, while maintaining acceptable fidelity, [26] introduces the idea of a "mixing function", $f(v, w)$. This takes an extracted signal $v$, and a watermark vector $w$, as input, and the output

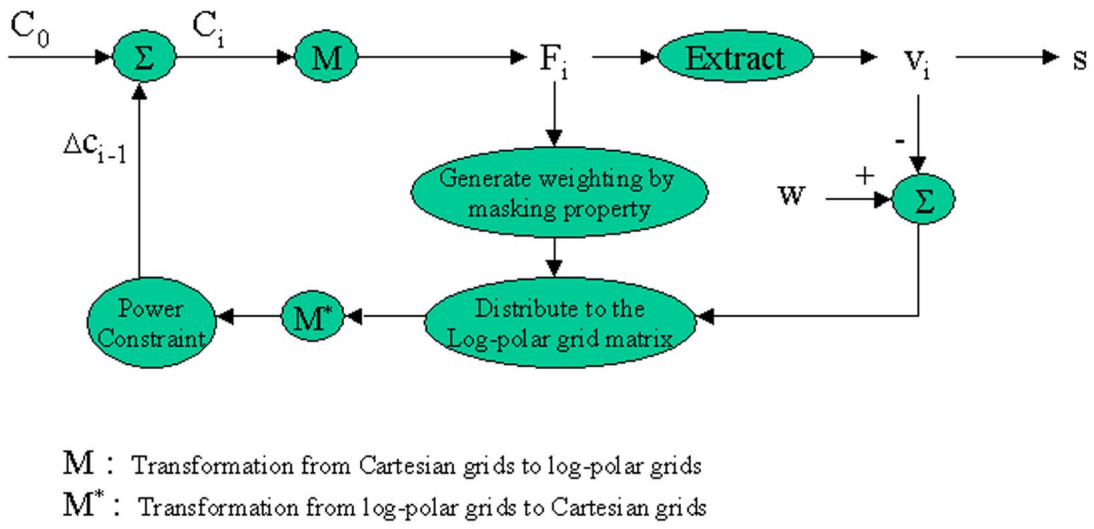



Figure 4-2: Proposed watermark embedding process

is a signal $s$, which is perceptually similar to $v$, and has a high correlation with $w$. Since $s$ is something between $v$ and $w$, it is referred to as the "mixed signal". It is this mixed signal that the embedder transmits, by modifying the image so that the extraction process in the detector will produce $s$. We may called this method as "feature vector shaping." An example of feature vector shaping is shown in Figure 4-1.

The watermark embedding process is shown in Figure 4-2. We should note there is a major difference between the proposed watermarking embedding process in this chapter and [26]. In [26], both mixing function and mixed signal are all determined in *a priori*. However, in this chapter, both of them are determined by the iteration result of the embedding process shown in Figure 4-2. They are conceptual descriptions of the iteration process.

Our watermark embedding process consists of four steps:

1. Apply the same signal-extraction process to the unwatermarked image as will be applied by the detector, thus obtaining an extracted vector, $v$. In our case,

this means computing $g_1(\theta)$. In Figure 4-2, $C_0$ is the DFT magnitudes of the unwatermarked image, $C_i$ is the DFT magnitudes of the watermarked image after $i$ times of iteration, and $F_i$ is the log-polar map magnitudes after $i-th$ iteration.

2. Compare the extracted vector, $v$, to the watermark vector, $w$. Calculate the differences between this two vectors, and then estimate the required change in the individual log-polar grids. The required changes are based on a weighting set generated according to the contrast masking property of the original log-polar coefficients. In this chapter, we use the weighting set which is proportional to $\log F_i$.

3. Transform the estimated changes from log-polar grids to Cartesian grids. They are performed using linear interpolation. We add an energy constraint on the estimated changes in the Cartesian DFT domain. Modify the original image so that, when the signal-extraction process is applied to it, the result will be $v_i$ instead of $v$.

4. Repeat the first three steps until the estimated changes are all smaller than a threshold, or until a defined maximum iteration number (which is 5 in our process).

Step 2 and 3 are the most difficult. A natural approach would be to modify all the values in column $j$ of the log-polar Fourier transform so that their logs sum to $s_j$ instead of $v_j$. This can be done, for example, by adding $(s_j - v_j)/K$ to each of the $K$ values in column $j$. Next, we would invert the log-polar resampling of the Fourier magnitudes, thus obtaining a modified, Cartesian Fourier magnitude. Finally, the complex terms of the original Fourier transform would be scaled to have the new magnitudes found in the modified Fourier transform, and the inverse

Fourier transform would be applied to obtain the watermarked image.

The main implementation issue in such an approach is the inherent instability in inverting the log-polar resampling. We therefore approximate this step with an iterative method in which a local inversion of the interpolation function is used for the resampling. The method is shown in Figure 4-2 and further discussed in Section 4.4.2.

## 4.4 Implementation problems and solutions

There are a number of problems that arise when implementing the algorithm of Section 4.3. Several of these are addressed below.

### 4.4.1 Rectilinear tiling implied by DFT

The log-polar Fourier transform of an image can be computed by resampling the image DFT with a log-polar grid. Some interpolation method must be used during the resampling, since the log-polar sample points don't generally coincide with the Cartesian sample points in the DFT.

The DFT is conventionally assumed to represent a tiled version of an image, as illustrated in Figure 4-3(a). Stone *et al* [123] have noted that this tiling pattern represents an inherent problem for any algorithm that relies on the rotational properties of Fourier transforms, since, when the content of an image is rotated, the rectilinear tiling grid is not rotated along with it. Thus, the DFT of a rotated image is not the rotated DFT of that image. The problem is illustrated in Figure 4-3(b) and (c).

One possible solution is to compute the log-polar Fourier transform directly, without using the Cartesian DFT as an intermediate step. In the continuous Fourier domain, each point has a value determined by correlating the image with a complex,

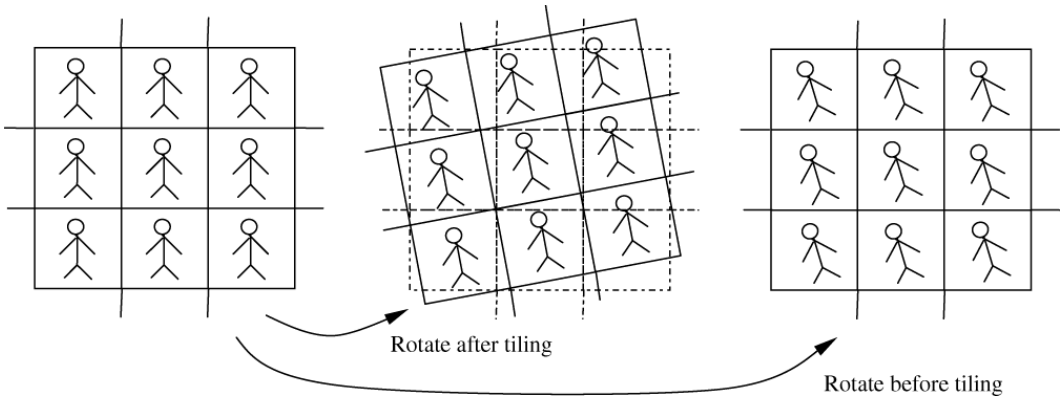



Figure 4-3: Rectilinear tiling and image rotation.

planar sinusoid. If we wish to obtain a value for a point between those that would be sampled in a DFT, we can find the corresponding sinusoid and directly compute its correlation with the image. This amounts to assuming that all the pixel values outside the bounds of the image are black, rather than assuming they are tiled copies of the image.

Of course, the direct approach described above doesn't take advantage of the efficient methods available for computing DFT's, and is thus likely to be prohibitively expensive[2]. Instead, we approximate the log-polar Fourier transform with the following steps:

1. Pad the image with black to obtain a larger image.

2. Take the DFT of the padded image. This yields a more finely sampled version of the continuous Fourier transform.

3. Resample in a log-polar grid, using an inexpensive interpolation technique. The technique we use is linear interpolation of the magnitudes of the coefficients.

[2]Alliney [3] presents a technique for the efficient direct computation of the polar Fourier transform of an image.

By padding with black to obtain a denser sampling of the Fourier transform, we reduce the distances between the DFT's sample points and the log-polar sample points, thus reducing the error introduced by inexpensive interpolation.

### 4.4.2 Difficulty of inverting log-polar mapping

Each element of the log-polar Fourier magnitude array is a weighted average of up to four elements of the Cartesian Fourier magnitude array. Thus, we can write

$$F = MC, \tag{4.15}$$

where $F$ is a column vector containing all the elements of the log-polar array, $C$ is a column vector containing the elements of the Cartesian array, and $M$ contains the weights used to perform interpolation. If we wish to modify the log-polar array so that it contains the watermark, and then find the corresponding Cartesian array, we have to find the inverse of $M$. Unfortunately, $M$ is ill-conditioned and it is not practical to perform this inversion precisely.

Instead, we use an iterative process to perform an approximate inversion. Let $F'$ be the modified version of $F$. We begin by observing that the four non-zero values in each row of $M$ sum to 1. Thus, if we add $F'_i - F_i$ to each of the elements $C_{j_1} ... C_{j_4}$, where $M_{i,j_1} ... M_{i,j_4}$ are non-zero, then the resulting Cartesian array will yield $F'_i$ in its log-polar mapping.

Unfortunately, if we try to apply this method to change all the elements of $F$, we'll have conflicting changes in the various elements of $C$. For example, both $M_{i,j}$ and $M_{k,j}$ might be non-zero, so that we'd need to change $C_j$ both when changing $F_i$ to $F'_i$ and when changing $F_k$ to $F'_k$. The desired changes are unlikely to be the same. We resolve this problem by using a weighted average of all the desired changes to

each element of $C$. So, in the above example, we would change the value of $C_j$ by

$$\frac{M_{i,j}(F'_i - F_i) + M_{k,j}(F'_k - F_k)}{M_{i,j} + M_{k,j}} \tag{4.16}$$

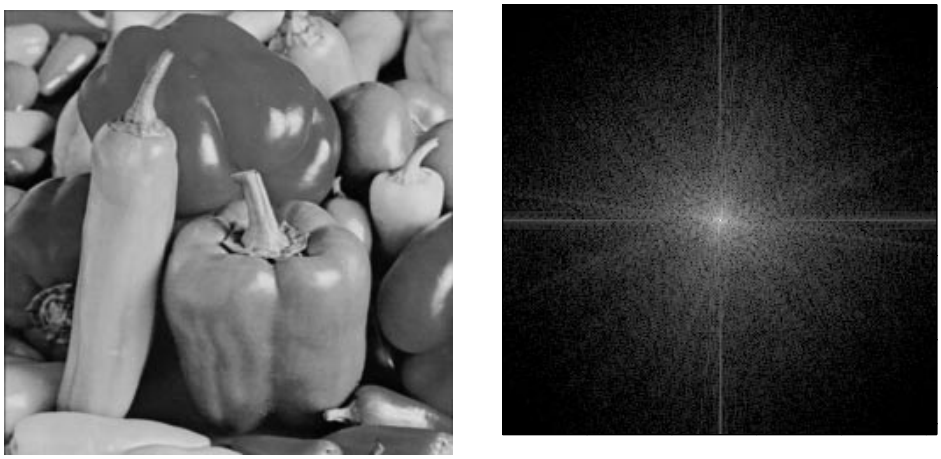
(assuming that $M_{i,j}$ and $M_{k,j}$ are the only nonzero elements of column $j$).

The above method results in a rough approximation to the desired inversion. We can obtain successively better approximations by applying the operation iteratively. In practice, we find it most effective to iterate the entire watermark embedding process described in section 4.3.2, using the above approximate log-polar inversion in each iteration. We have found that three or four iterations usually suffice to produce an approximation that can be detected.

### 4.4.3 Orientation of image boundaries

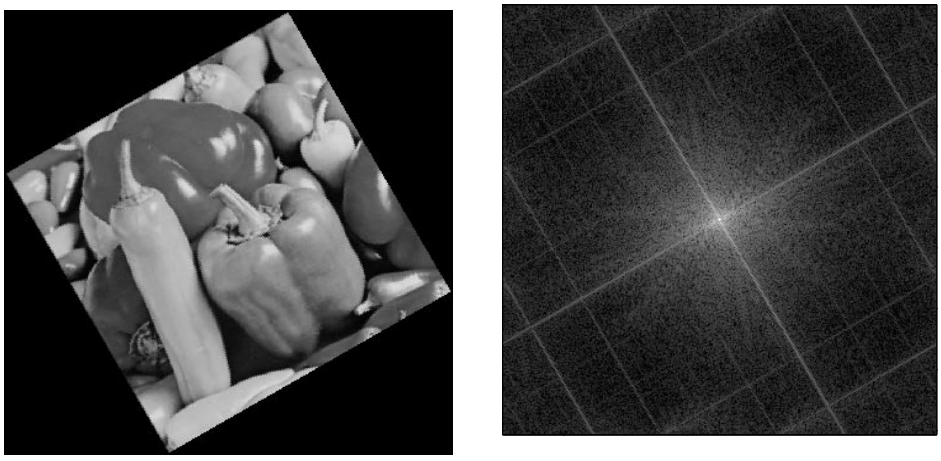
It is well known that the rectangular boundary of an image usually causes a "cross" artifact in the image's energy spectrum (see Figure 4-4). This happens because there is usually a large discontinuity at each edge of the image due to the implicit tiling. The DFT magnitude of such vertical and horizontal discontinuities has large energy in all the vertically and horizontally oriented frequencies, which results in the cross artifact.

If the image is rotated, but padded with black so that no image content is cropped, then the cross in the DFT magnitude will also rotate (Figure 4-5). If, on the other hand, the rotated image is cropped, so that no black is added, then the new image boundaries cause a horizontal and vertical cross similar to that found in the original image, even though the rest of the DFT magnitude is rotated (Figure 4-6). Since the cross has so much energy, it tends to cause two large bumps in the extracted watermark vector, which substantially reduce the correlation coefficient with the embedded watermark.



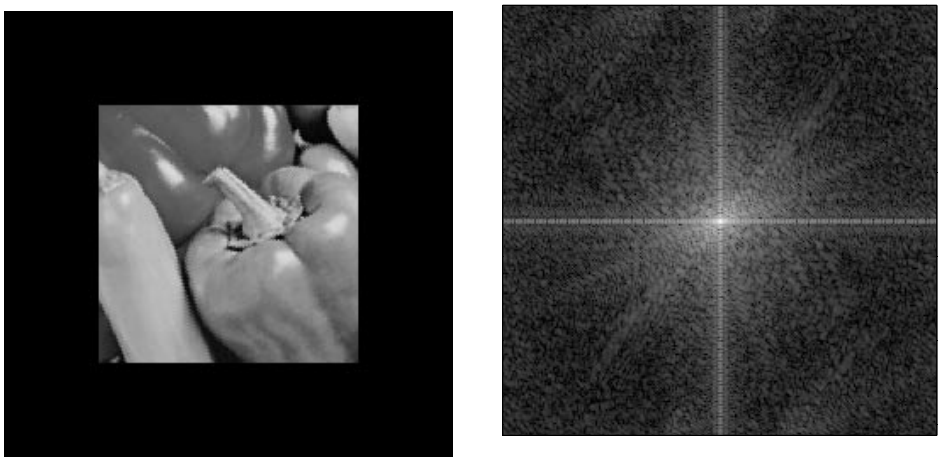
(a) (b)

Figure 4-4: An image and its Discrete Fourier Transform.



(a) (b)

Figure 4-5: DFT effects of rotation



(a) (b)

Figure 4-6: DFT effects of rotation and cropping

Our present solution to this problem is to simply ignore the bumps in the extracted signal by ignoring a neighborhood around each of the two highest-valued elements. Alternative solutions that appear in the literature include multiplication of the image by a circularly-symmetric window [32] and blurring of the image edges [92]. These solutions are probably more general than the one employed here, but would require modification to the watermark embedder, and has been left for future work.

### 4.4.4 Dynamic range of frequency magnitudes

The magnitude of low frequencies can be very much larger than the magnitude of mid and high frequencies. In these circumstances, the low frequencies can become overwhelming. To reduce this problem, we sum the logs of the magnitudes of the frequencies along the columns of the log-polar Fourier transform, rather than summing the magnitudes themselves.

A beneficial side-effect of this is that a desired change in a given frequency is expressed as a fraction of the frequency's current magnitude rather than as an

absolute value. This is better from a fidelity perspective.

### 4.4.5 Unreliability of extreme frequencies

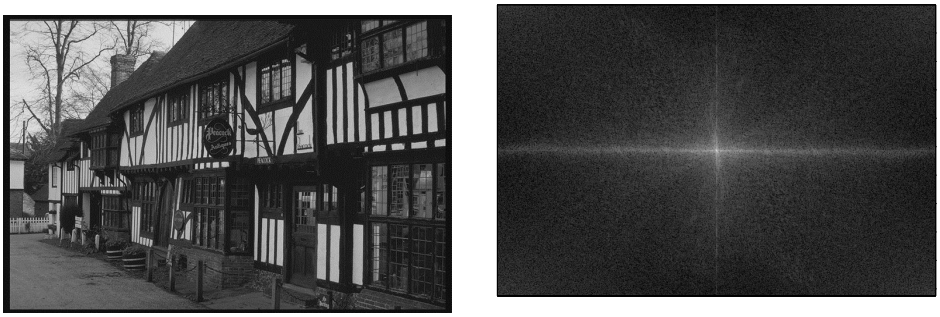
It is well known that the lowest and highest frequencies in an image are usually unreliable for watermarking. The low frequencies are unreliable because they are difficult to modify without making visible changes in the image. The high frequencies are unreliable because they can be easily modified by common processes such as compression, printing, and analog transmission. Our solution is to neglect these unreliable frequencies when extracting the watermark.

A better solution would be to use a perceptual model to estimate the maximum amount of change that can be applied to each frequency and a model of specific attacks to estimate the degree of robustness. The amount of watermark energy embedded into each frequency would then be proportional to this perceptual significance and robustness. Such an approach is discussed in [24, 25, 107, 141]. Application of this idea to the present watermarking method is a topic for future research.

### 4.4.6 Images are rotationally asymmetric

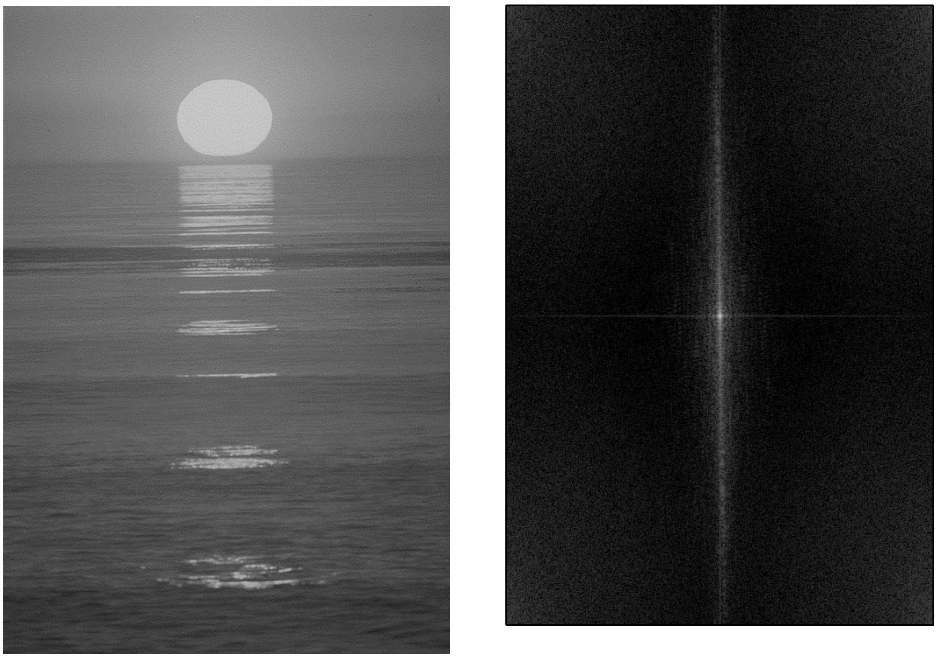
The energy in an image is seldom evenly distributed in angular frequency. Images frequently have a large amount of energy in one group of directions, while having much lower energy in an orthogonal group of directions. For example, images containing buildings and trees have significant vertical structure yielding more energy in the horizontal frequencies than in the vertical (Figure 4-7), while seascapes or sunsets are strongly oriented in the horizontal direction yielding higher vertical frequencies (Figure 4-8).

Spectra such as those of Figures 4-7 and 4-8 suggest an uneven masking ability



(a) (b)

Figure 4-7: Image with dominant vertical structure and its DFT.



(a) (b)

Figure 4-8: Image with dominant horizontal structure and its DFT.

in orthogonal directions. As a consequence, it may be much easier, from a fidelity perspective, to embed some portions of the watermark than others. For example, when watermarking the image of tall buildings, we can more easily hide noise with a strong vertical component than noise with a strong horizontal component. This can be a problem if the difficult-to-modify portions of the watermark are critical in differentiating it from other watermarks.

To reduce this problem, we divide the extracted signal into two halves, and add the two halves together. Thus, rather than using $g(\theta)$ of Equation 4.10, we use $g_1(\theta)$, of Equation 4.12.

If we want to modify an element of $g_1(\theta)$, we can do so by hiding noise that's oriented along either angle $\theta$ or angle $\theta + 90°$. This increases the likelihood that each element of the watermark can be embedded within the fidelity constraints.

### 4.4.7 High correlation between elements of extracted watermark

For natural images, $g_1(\theta)$ is likely to vary smoothly as a function of $\theta$. In other words, the extracted signal will have more low-frequency content than high-frequency content. This reduces the effectiveness of the correlation coefficient as a detection measure.

We improve the detection measure by applying a whitening filter to both the extracted signal and the watermark being tested for before computing the correlation coefficient. Note that the whitening filter is employed only in the watermark detector; the embedder is unchanged. The whitening filter was designed to decorrelate the elements of signals extracted from natural images, and was derived from signals extracted from 10,000 images from [21]. These images were not used in any of the subsequent experiments reported in Section 4.5.

The idea of using a whitening filter to improve watermark detection in this way

has been discussed in [33].

### 4.4.8 Interrelation between changes made in watermark elements

During watermark embedding, it is difficult to change the value of one element of the extracted watermark, without changing the values of its neighbors. This results primarily from the fact that any one frequency in the DFT can effect several values of $g_1(\theta)$, so changing that frequency can effect several elements of the watermark. Because of this, it is difficult to embed a watermark that varies wildly from one element to the next.

We reduce this problem by replicating elements of the desired watermark to obtain a lower-frequency watermark. For example, if the watermarks are extracted by computing 74 samples of $g_1(\theta)$ (after removing the samples that contain the "bumps" discussed in 4.4.3), then we would define our desired watermark as a vector of 37 values, and duplicate each of its 37 values to obtain a length 74 vector.

## 4.5 Experimental Results

The following results were obtained by extracting a length 90 vector from the image and neglecting the 16 samples surrounding the peak (assumed to correspond to the DFT cross artifact). This leaves a descriptor that is 74 samples in length. The detection process involves a comparison of the watermark with all 90 cyclic rotations of the extracted descriptor. In this section we examine the false positive behavior, effectiveness, and robustness of the proposed scheme. False positive measurements were collected on 10,000 unwatermarked images[3], and effectiveness and robustness measurements were collected on 2,000 watermarked images except that scale up

[3]The images used in this test were all different from, but from the same database as the 10,000 images that were used to generate the whitening filter.

with cropping used only 947 images and JPEG compression used 1909.

### 4.5.1 Probability of False Positive

We begin our evaluation of the new watermarking method by finding the relationship between the threshold and the probability of false positive. A false positive or false detection occurs when the detector incorrectly concludes that an unwatermarked image contains a given watermark. Thus, the probability of false positive is defined as

$$P_{fp} = P\{D_{max} > T\} \tag{4.17}$$

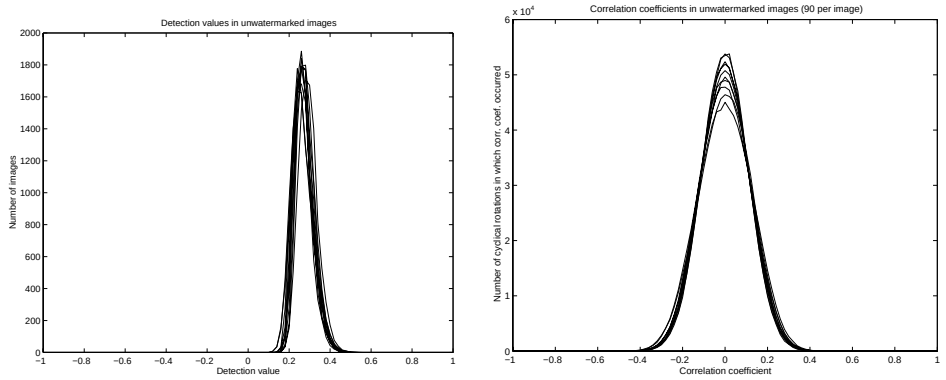
where $D_{max}$ is a detection value obtained by running the detector on a randomly selected, unwatermarked image and $T$ is the detection threshold. The subscript $max$ specifies the maximum detection value from all of the cyclical shifts examined.

This probability is estimated empirically by applying the detector to 10,000 unwatermarked images from [21], testing for 10 different binary watermarks in each. The 10 resulting histograms are shown in Figure 4-9(a) superimposed on one another. The probability of false positive is then plotted in Figure 4-10(b) as a function of threshold. Again, each trace corresponds to one of the 10 watermarks.

Figure 4-9(a) indicates that most detection values from unwatermarked images fall between 0.2 and 0.4. This might seem surprising, since we might expect unwatermarked images to yield detection values closer to zero. The reason the values are so high is that each one is the maximum of 90 different correlation coefficients, computed during the cyclical search (see section 4.3.1, step 3). This means that

$$P_{fp} = P\{D_{max} > T\} = P\{(D_0 > T) \ \ or \ \ (D_1 > T) \ \ or \ldots (D_{89} > T)\} \tag{4.18}$$

where $D_0 ... D_{89}$ are the 90 correlation coefficients computed during the search. Each




(a) (b)

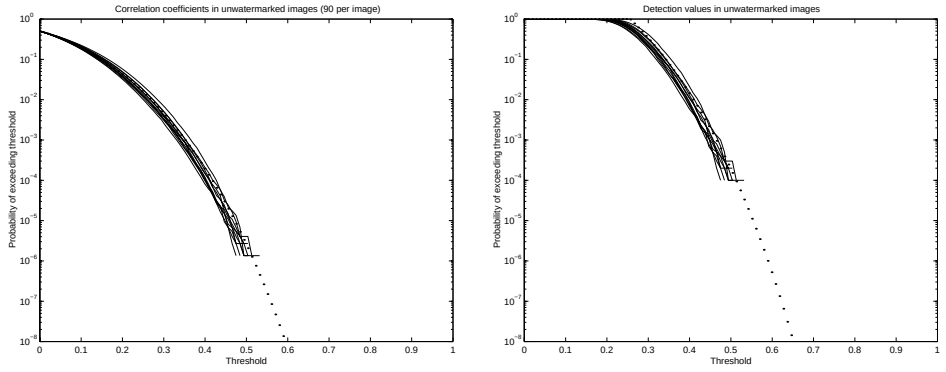
Figure 4-9: Detection value distributions for 10 watermarks in 10,000 unwatermarked images: (a) maximum detection value for each watermark/image pair and (b) all 90 detection values for each watermark/image pair.

of $D_0 ... D_{89}$ is drawn from a distribution that is centered around zero, as shown in Figure 4-9(b), which shows 10 superimposed histograms of the 90,000 correlation coefficients computed for each of the 10 watermarks during the experiment. The maximum of 90 values drawn from a distribution like that in Figure 4-9(b) is likely to be higher than zero.

During the experiment with unwatermarked images, the highest detection value obtained was 0.55. Thus, we have no data to estimate $P_{fp}$ for $T > 0.55$. To estimate this, we must employ a theoretical model, such as the one described in [97]. This model says that, if $D$ is the correlation coefficient between a preselected $d$-dimensional watermark vector and a random vector drawn from a radially-symmetric distribution, then

$$P\{D > T\} = R(T, d) = \frac{\int_0^{cos^{-1}(T)} \sin^{d-2}(u) du}{2 \int_0^{\pi/2} \sin^{d-2}(u) du}. \tag{4.19}$$

The whitening filter employed in our detector makes the distribution roughly spherical, so this model is expected to apply to the present system, with $d = 74$. The resulting false positive prediction is shown as a dotted line in Figure 4-10(a).




(a) (b)

Figure 4-10: False positive rates measured with 10,000 unwatermarked images, (a) individual correlation coefficients and (b) final detection value. Each solid trace corresponds to one of 10 different watermark vectors. Dashed line represents theoretical estimates.

The model predicts the probability that one correlation coefficient is greater than the threshold, not the probability that the maximum of several coefficients is greater. Thus, it predicts $P\{D_i > T\}, i \in [0...89]$, rather than $P\{D_{max} > T\}$. Figure 4-10(a) indicates how well the model predicted $P\{D_i > T\}$ in our experiment.

We obtain an estimated upper bound on $P\{D_{max} > T\}$ by observing that

$$P\{Q_0 \ or \ Q_1 \ or \ldots Q_{n-1}\} \leq \min\left(1, \sum_i P\{Q_i\}\right) \tag{4.20}$$

When $Q_i$ corresponds to the event $(D_i > T)$, and $n = 90$, we obtain

$$P\{D_{max} > T\} \leq \min\left(1, 90 \times R(T, 74)\right). \tag{4.21}$$

This prediction is shown in Figure 4-10(b) as a dotted line.

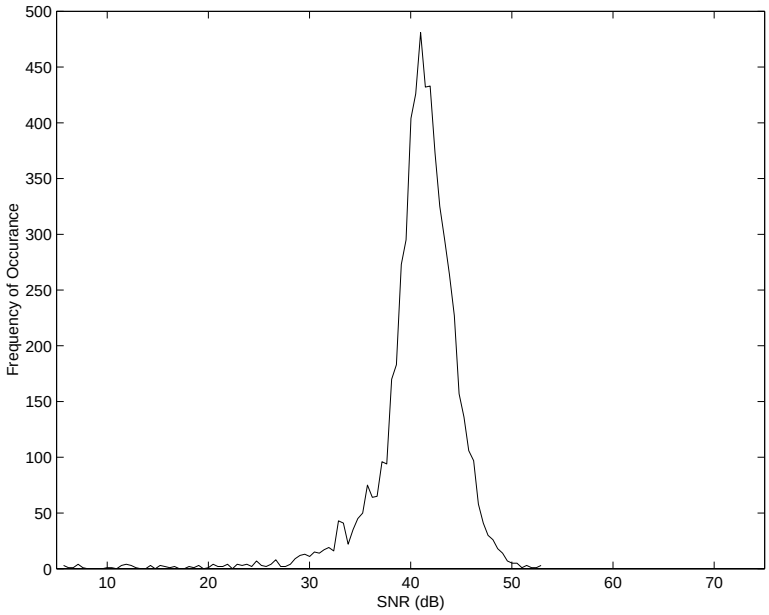



Figure 4-11: Signal-to-noise ratio

### 4.5.2 Fidelity

The tradeoff between fidelity and robustness is controlled by adjusting the relative weighting used in the mixing of the watermark signal and the extracted signal (see Section 4.3.2). As the relative weight assigned to the watermark signal is increased, the strength of the embedded watermark is increased at the expense of lower fidelity. Once chosen, the mixing weights were held constant over all experiments described in this section. These weights were empirically selected to yield an average signal-to-noise ratio of about 40dB.[4] Figure 4-11 shows a histogram of the ratios obtained. Figure 4-12 shows an example of a watermarked image with little impact on fidelity.

It must be noted, however, that signal-to-noise ratio is not a very effective predictor of perceptual quality. The fidelity of the image depends to a large degree on the perceptual relationship between the image and the noise. In general, noise that matches the underlying textures in an image is less perceptible than noise that is

[4]Here the "signal" is the image, and the "noise" is the watermark pattern.


Figure 4-12: Watermarking with little impact on fidelity

very different from the image, even at the same signal-to-noise ratios.

The present system generates watermark patterns by making small percentage adjustments to the powers of frequencies in the image's spectrum, so the resulting noise pattern is usually similar to the textures in the image. Thus, when we watermark an image that contains a homogeneous texture, the watermark is well-hidden. But when we mark an image that contains widely varying textures, the mark can become visible. Figure 4-13 illustrates the problem. The watermark strength in this figure was increased so that the problem should be visible after printing in a journal.

Solving the fidelity problem in non-homogeneous images would require a modification to the algorithm that attenuates or shapes the watermark according to local texture characteristics. This has been left for future work.

### 4.5.3 Effectiveness

The effectiveness of a watermarking scheme is measured as the probability that the output of the watermark embedder will contain the watermark, subject to constraints on the fidelity of the marked image and the detection threshold or probability of false positive. The effectiveness of the current scheme is measured and

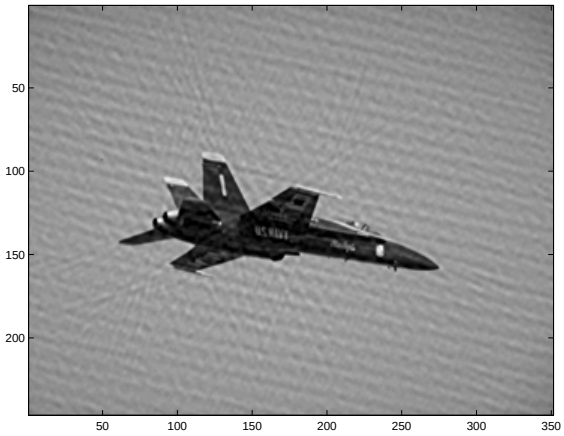



Figure 4-13: Character of the watermark noise when the strength is too high.The watermark strength in this figure was increased so that the problem should be visible after printing in a journal.

plotted as the dashed ROC curves in each of Figures 4-15 – 4-22.

### 4.5.4 Robustness

In a practical setting, RST distortions are usually accompanied by cropping. Figure 4-14(f), (g), and (i) show respectively rotation, scaling, and translation with the associated cropping. With the current algorithm, cropping can be viewed as distortion of the extracted signal by additive noise. As such, we expect cropping to degrade the detection value.

In this section seven geometric distortion attacks are examined; rotation with and without cropping, scaling up with and without cropping, translation with and without cropping, and scaling down. Note that scaling down does not imply cropping. In order to isolate the effects of rotation, scaling up, and translation from cropping, the images have been padded with gray as shown in Figure 4-14(a). The embedder has been applied to these expanded images and then the gray padding replaced with unwatermarked gray padding prior to detection or attack. The amount of padding is such that none of the rotation, scaling up, and translation experiments

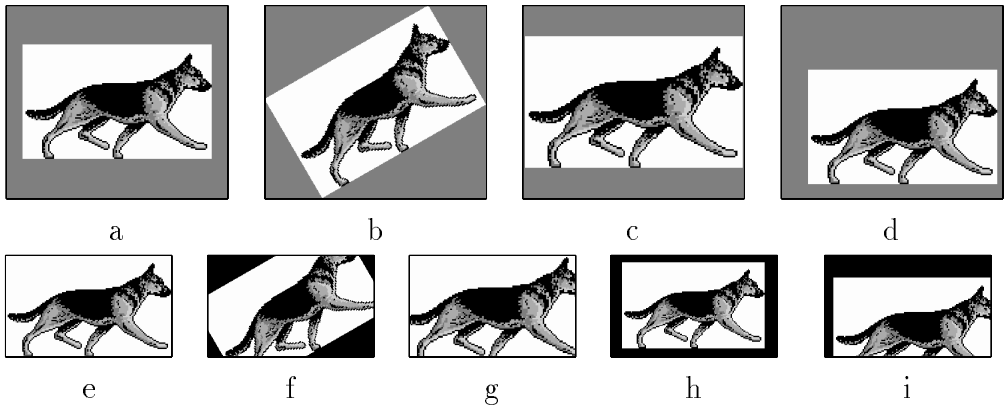



Figure 4-14: Examples of geometric attacks: (e) and (a) are the original and padded original respectively, (b)-(d) attacks without cropping, and (f)-(i) attacks with cropping

cause image data to be cropped. The only data that is cropped is unwatermarked padding. Thus, the differences between the detection values prior to rotation and those after rotation can be attributed solely to the rotation as the associated cropping of unwatermarked padding does not effect the detection value.

The detection value prior to attack is used to measure the effectiveness of the watermarking scheme. This effectiveness is likely to be reduced in the padded examples since a portion of the watermarked image (the watermarked gray padding) has been replaced with non-watermarked padding. However, the purpose of the experiments based on these padded geometric attacks, shown in Figure 4-14(b)-(d), is to isolate the effects due to geometric distortions from those due to cropping.

#### 4.5.4.1 Rotation

Two experiments were performed to test the watermark's robustness against rotation. The first experiment was designed to isolate the effects of rotation from all other types of attack. The second was a more realistic test of the effects of rotation with cropping.

Each trial of the first test comprised the following steps:

1. Pad an image with neutral gray, increasing its size. The amount of padding was chosen to allow rotation without any part of the original image going outside of the image boundaries (Figure 4-14(a)).

2. Embed a randomly-selected watermark in the padded image.

3. Replace the padding with neutral gray again. This removes any watermark information from the neutral gray area.

4. Run the watermark detector on the image to obtain a detection value before rotation.

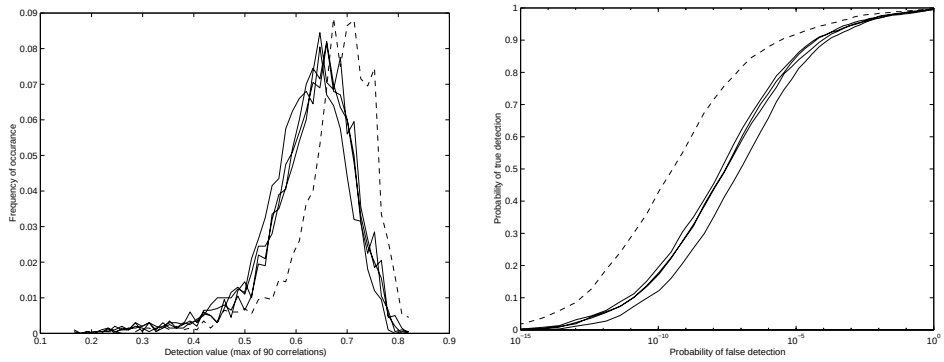
5. Rotate the image by a predetermined angle, and crop to the original size. Figure 4-14(b) shows what an image looks like after this step. Note that only the padding is cropped, so we do not crop off any of the watermark pattern.

6. Run the watermark detector on the image to obtain a detection value after rotation.

Since the padding that's cropped off during rotation contains no watermark pattern, any difference between the "before" value obtained in step 4 and the "after" value obtained in step 6 can only result from the effects of rotation.

This experiment was performed on 2,000 images with rotations of 4°, 8°, 30°, and 45°. We limited this test to a maximum rotation of 45° because rotations beyond 45° are equivalent to smaller rotations after a rotation of 90°. An image that has been rotated 90° yields exactly the same extracted vector as an unrotated image, so a rotation of greater than 45° should behave the same as a smaller rotation.

As indicated in Figure 4-15(a), the different rotations yielded essentially the same results. Figure 4-15(b) shows receiver-operating-characteristic (ROC) curves before



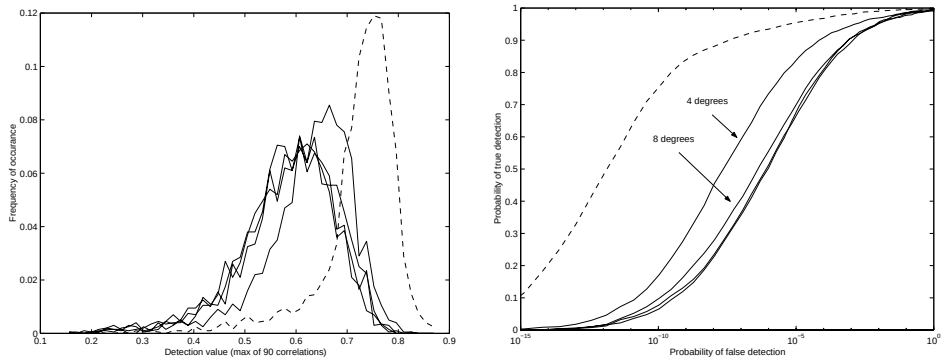

(a) (b)

Figure 4-15: Rotation without cropping, 4°, 8°, 30°, and 45°, (a) histogram and (b) ROC

and after rotation. For each of the ROC curves, the false-positive probabilities were estimated using the method described in Section 4.5.1. In the two plots of Figure 4-15, the dashed lines represent the detection values prior to attack, i.e. the effectiveness of the embedding. The deviations of the solid traces from the dashed represent the effects of the attack.

In the second experiment, we watermarked the original image without padding, and allowed part of the watermark pattern to be cropped off after rotation. Figure 4-14(f) shows an example of what an image looked like after the rotation. This experiment was performed on 2,000 images with rotations of 4°, 8°, 30°, and 45°. Figure 4-16 shows the results.

Three immediate observations based on the ROC curve of Figure 4-15(b) are that the effects of these four rotations are all similar, for a fixed fale positive probability, $P_{fp}$, (independent axis) rotation decreases the likelihood of detection (difference between the dashed and solid lines), and the effect of rotation on the probability of detection is dependent on the $P_{fp}$ or equivalently the threshold. For relatively high $P_{fp}$, for example $10^{-3}$ or one in a thousand, the current method is extremely robust to rotation. At higher values of $P_{fp}$, for example $10^{-8}$, rotation degrades the



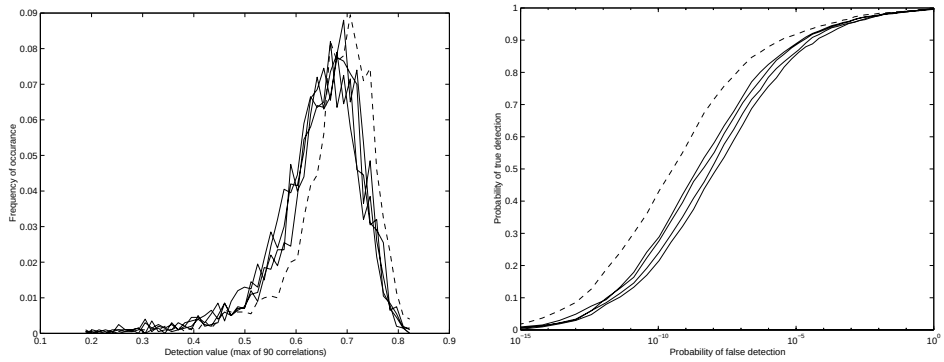

(a) (b)

Figure 4-16: Rotation with cropping, 4°, 8°, 30°, and 45°, (a) histogram and (b) ROC

detection value more significantly. Figure 4-16(b) further shows that the cropping that accompanies rotation has a significant, negative impact on detection (downward shift of the solid lines in Figure 4-16(b) from those in Figure 4-15(b)), and the deterioration of the detection value is more dependent on rotation angle (different rotations result in different amounts of cropping).

These ROC curves emphasize the importance of the baseline measurement (dashed lines), which serves as an upper bound on robustness. They also show that each of the two experiments begin from a different baseline. In the second experiment, the rotation attack is applied to images that have been much more effectively watermark. The lower effectiveness of the first experiment represents the cropping of watermarked data that occurs when the watermarked gray padding is replaced with unwatermarked gray padding. Recall that these somewhat artificial embedding conditions are in place to isolate the effects of rotation from any further degradation that may occur due to the cropping that normally accompanies rotation.

These results demonstrate that the current watermark, designed to be invariant to rotations, does exhibit a resilience to rotation. This watermark has not been explicitly designed to withstand cropping and the results highlight this fact.




(a) (b)

Figure 4-17: Scaling up without cropping, 5%, 10%, 15%, and 20%, (a) histogram and (b) ROC

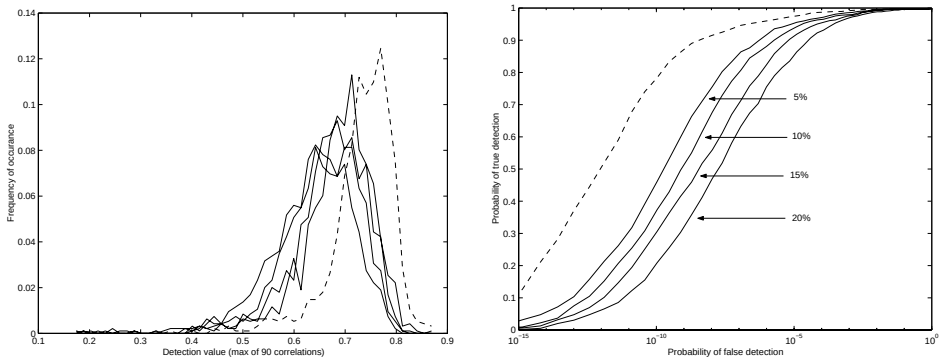
#### 4.5.4.2 Scale

To test robustness to scaling, we performed three experiments. The first and second test the effect of scaling up, with and without cropping. The third tests the effect of scaling down, with padding.

In the first scaling test, the steps performed for each trial were the same as those for the first rotation step, with the exception that instead of rotating the image we scaled the image up. Figure 4-14(c) shows an example of an image that has been scaled up after padding and watermarking. The test was performed on 2,000 images at scales 5%, 10%, 15%, and 20% larger than the original. The results are shown in Figure 4-17.

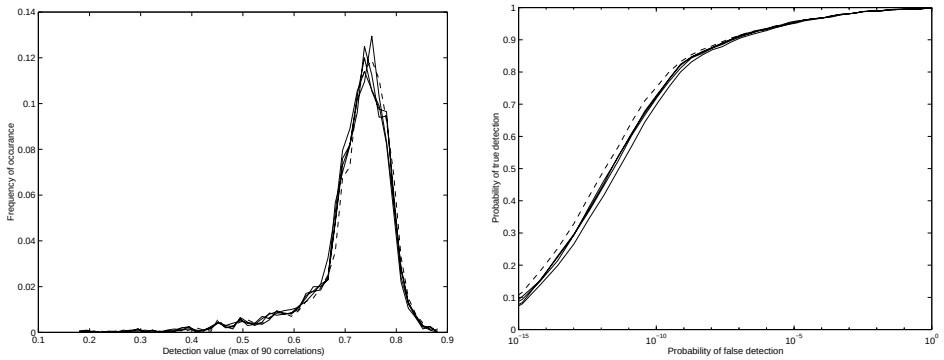
The second test was the same as the first except without padding the images before scaling, so part of the image was cropped off after scaling. Figure 4-14(g) illustrates the attack. The test was performed on 947 images at scales of 5%, 10%, 15%, and 20% larger than the original. The results are shown in Figure 4-18.

For the test of reduced scaling, we do not have to be concerned with cropping. Rather, after watermarking and scaling, the image is padded back to its original size. Since cropping is not an issue here, we only performed one version of this




(a) (b)

Figure 4-18: Scaling up with cropping, 5%, 10%, 15%, and 20%, (a) histogram and (b) ROC




(a) (b)

Figure 4-19: Scaling down, 5%, 10%, 15%, and 20%, (a) histogram and (b) ROC

experiment, in which the image was not padded before watermarking as shown in figure 4-14(h). The test was performed on 2,000 images at scales 5%, 10%, 15%, and 20% smaller than the original. The results are shown in figure 4-19.

As with rotation, the results show that scaling up, in general, degrades the probability of detection as a function of $P_{fp}$. For the relatively high $P_{fp} = 10^{-3}$, scaling has very little effect on the likelihood of detection while at $P_{fp} = 10^{-8}$ the effect is more significant. We also observe that the results differ slightly for different scale factors at these lower false positive rates.

The differences between the ROC curves in Figures 4-17 and 4-18 clearly show

the severe degradation due to the cropping that normally accompanies scaling. As expected, the effect of this cropping increases with the scale factor because higher scale factors imply more cropping.

Figure 4-19 shows that a decrease in scale has virtually no effect for $P_{fp} > 10^{-7}$ or so and for lower $P_{fp}$ the degradation is only slight.

The current watermark was designed to be invariant to changes in scale and these results demonstrate an excellent resilience to a decrease in scale and good resilience to an increase in scale. Again, these results highlight the negative impact of cropping.

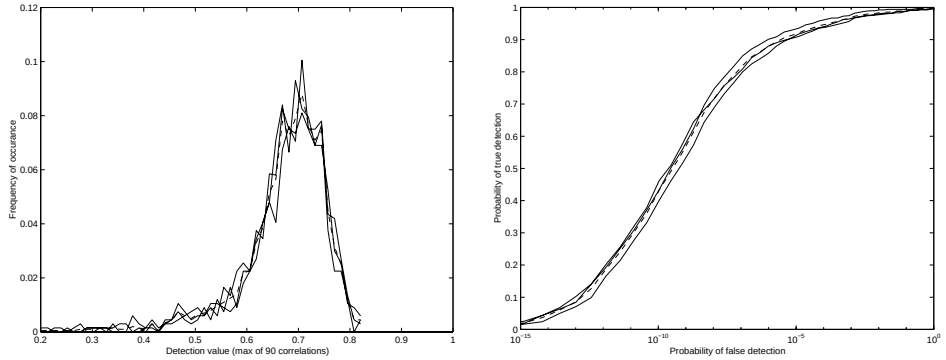
#### 4.5.4.3 Translation

We expect translation alone to have no effect on the watermark, since the watermark is computed from the magnitudes of the Fourier coefficients. To test this, we performed two experiments.

The first experiment was similar to the first rotation and scaling experiments, in that the image was padded before watermarking and the padding was replaced after watermarking. We then translated the image by cropping gray off the top and right, and padding gray onto the bottom and left. Figure 4-14(d) shows an example of such a translated image. The experiment was performed on 2,000 images at translations of 5%, 10%, and 15% of the image size. The results are shown in Figure 4-20.

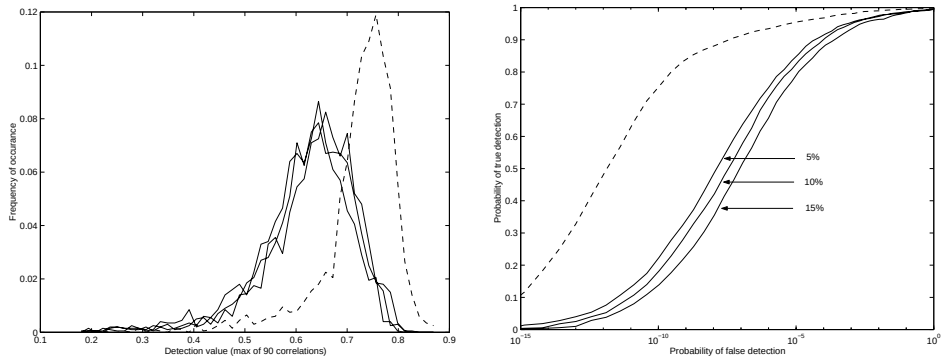
The second translation test was performed without padding the image before translation, so that part of the watermark pattern is cropped during translation. Figure 4-14(i) shows an example of this attack. Again, the experiment was performed on 2,000 images at translations of 5%, 10%, and 15% of the image size. The results are shown in Figure 4-21.

The results of the first experiment show that translation has negligible effect



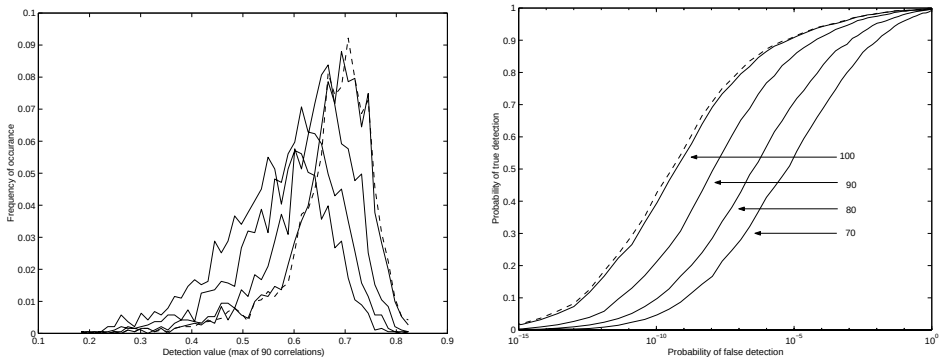

(a) (b)

Figure 4-20: Translation without cropping, 5%, 10%, and 15%, (a) histogram and (b) ROC




(a) (b)

Figure 4-21: Translation with cropping, 5%, 10%, and 15%, (a) histogram and (b) ROC




(a) (b)

Figure 4-22: JPEG compression, QF = 100, 90, 80, and 70, (a) histogram and (b) ROC

on probability of detection. This means that the second test is more a test of robustness to cropping than to translation, and we see the same sort of pattern that was observed in the second rotation and scaling experiments.

### 4.5.5 JPEG compression

While the purpose of the present watermark design is to survive RST transformations, it is, of course, important that the watermarks also survive other common types of image processing. We therefore conducted a test of robustness to JPEG compression.

After watermarking, images were JPEG compressed at quality factors of 100, 90, 80, and 70. The test was performed on 1,909 images. Figure 4-22 shows the results.

The results show that the likelihood of detection decreases with the amount of compression noise introduced and that this decrease is dependent on the $P_{fp}$. For relatively high $P_{fp} = 10^{-3}$ JPEG at QF = 70 yields a robustness of about 75%. At lower $P_{fp}$ the results degrade significantly.

### 4.5.6 Summary of the Experimental Results

Experimental results on a database of over 2,000 images demonstrate that the method is resilient to either rotations, scale changes or translations. The degree of resilience changes as a function of the probability of false positive. The results also demonstrate the weakness of this method to cropping and JPEG compression, attacks against which no steps have been taken in the design.

## 4.6 Properties of the Print-and-Scan Process

After the print-and-scan process, distortion occurs in both the pixel values and the geometric boundary of the rescanned image. The distortion of pixel values is caused by (1) the luminance, contrast, gamma correction and chromnance variations, and (2) the blurring of adjacent pixels. These are typical effects of the printer and scanner, and while they are perceptible to the human eye, they affect the visual quality of a rescanned image.

Distortion of the geometric boundary in the PS process is caused by rotation, scaling, and cropping (RSC). Although it does not introduce significant effects on the visual quality, it may introduce considerable changes at the signal level, especially on the DFT coefficients of the rescanned image.

It should be noted that, in general image editing processes, geometric distortion cannot be adequately modeled by the well-known rotation, scaling, and translation (RST) effects, because of the design of today's Graphic User Interface (GUI) for the scanning process. From Figure 4-23, we can see that users can arbitrarily select a range for the scanned image. We use "cropping" to describe this operation, because the rescanned images are cropped from an area in the preview window, including the printed image and background. The RST model, which has been widely used in pattern recognition, is usually used to model the geometric distortion on the image

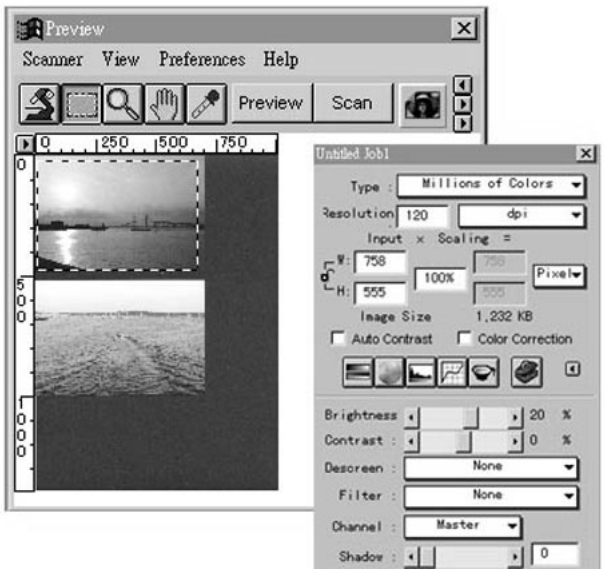



Figure 4-23: Typical control windows of scanning processes. Users have the freedom to control scanning parameters, as well as can arbitrarily crop the scanned image. [source: Microtek ScanWizard]

of an observed object. In those cases, the meaning of RST is based on a fixed window size, which is usually pre-determined by the system. However, in the PS process, the scanned image may cover part of the original picture and/or part of the background, and may have an arbitrarily cropped size. These changes, especially that of image size, will introduce significant changes of the DFT coefficients. Therefore, instead of RST, a RSC model is more appropriate to represent the PS process. We will discuss this in more detail in Section 4.7.

## 4.7 Modeling of the Print-and-Scan Process

In this section, we first propose a hypothetical model of the pixel value distortions. To our knowledge, there is no existing appropriate model in the literature to describe the pixel value distortions in PS process. Therefore, we propose the following hypothetical model based on our experiments and [40][142]. Although more

experiments are needed to verify its validity, we have found this model is appropriate in our experiments using different printers and scanners, as it shows several characteristics of rescanned images. In Section 4.7.2, we analyze the geometric distortion in the PS process, and then focus on the changes of DFT coefficients for invariants extraction. These models can be applied to general geometric distortions, although a special case (the PS process) is considered here.

### 4.7.1 Pixel Value Distortion

We are interested in modeling the variation of luminance values of color pixels before and after the PS process, because we only use luminance as the main place for embedding information (*e.g.*, watermarking) or extracting features in our system. Readers who are interested in color variation can find extensive references in [119]. Our focus is on the popular consumer PS devices such as color inkjet printers and flatbed scanners.

Consumer printers are based on halftoning, which exploits the spatial lowpass characteristics of the human visual system. Color halftone images utilize a large number of small colored dots. Varying the relative positions and areas of the dots produces different colors and luminance values. The lowpass property is usually shown in the spread function of the scanner.

Discrete images are converted to continuous images after printing. In the continuous physical domain, assume we have a virtual finite support image, $\mathbf{x}$, which is reconstructed from the original discrete image, $\mathbf{x_0}$,

$$\mathbf{x}(t_1,t_2) = \begin{cases} \sum\sum \mathbf{x_0}[n_1,n_2]\delta(t_1 - n_1T_{01}, t_2 - n_2T_{p2}), & t_1 \in [-\frac{T_1}{2},\frac{T_1}{2}], t_2 \in [-\frac{T_2}{2},\frac{T_2}{2}] \\ 0, & elsewhere, \end{cases} \tag{4.22}$$

where $T_{o1}$ and $T_{o2}$ are the inverse of DPI (dots per inch) values in the $t_1$ and $t_2$

directions, and $T_1$ and $T_2$ are the range of support of the image. Then, the printed image will be a dithered version of $\mathbf{x}$ with additional noises. Combining with scanning process, we assume the pixel value distortion in the PS process can be modeled as

$$\mathbf{x}(t_1, t_2) = K[\mathbf{x}(t_1, t_2) * \tau_1(t_1, t_2) + (\mathbf{x}(t_1, t_2) * \tau_2(t_1, t_2)) \cdot N_1] \cdot s(t_1, t_2), \qquad (4.23)$$

where $\mathbf{x}(t_1, t_2)$ is the output discrete image, $K$ is the responsivity of the detector, and $s(t_1, t_2)$ is the sampling function. There are two components inside the bracket. The first term models the system point spread function,

$$\tau_1(t_1, t_2) = \tau_p(t_1, t_2) * \tau_s(t_1, t_2), \qquad (4.24)$$

where $\tau_p(t_1, t_2)$ is the point spread function of printer, $\tau_s(t_1, t_2)$ is the detector and optical point spread function of scanner, and $*$ represents convolution. In the second term, $\tau_2$ is a high-pass filter, which is used to represent the higher noise variance near the edges, and $N_1$ is a white Gaussian random noise. The noise power is stronger in the moving direction of the carriage in scanner, because the stepped motion jitter introduces random sub-pixel drift. This indicates that $\tau_2$ is not symmetric in both directions.

In Eq. (4.23), the responsivity function, $K$, satisfies this equation,

$$K(x) = \alpha \cdot (x - \beta_x)^\gamma + \beta_K + N_2(x), \qquad (4.25)$$

which includes the combined AC, DC and gamma adjustments in the printer and scanner. $N_2$ represents that power of noises is a function of pixel value. It includes thermal noises and dark current noises. The variance of $N_2$ is usually larger on dark

pixels, because sensors are less sensitive to their low reflectivity.

From this model, we can analyze the low-pass filtering properties on the Fourier coefficients and describe the high noise variances in the high band coefficients. Some tests of its validity are shown in Section 4.9.

### 4.7.2 Geometric Distortion

In general, the scanning process follows a customary procedure. First, a user places a picture (or the printed original image) on the flatbed of the scanner. If the picture is not well placed, this step may introduce a small orientation, or a rotation of 90°, 180° or 270° on the scanned image with a small orientation1. Then, the scanner scans the whole flatbed to get a low-resolution preview of the image. After this process, the user selects a cropping window to decide an appropriate range of the picture. Usually, it includes only a part of the original image, or the whole picture with additional background (*a.k.a.* zero padding). The scanner then scans the picture again with a higher resolution to get a scanned image. The size of this image is usually different from the original, because the resolution in the scanner and the printer may be different. The final scanned discrete image is obtained by sampling the RSC version of the printing-distorted image with additional scanning noise.

Images are discretized at both ends of the PS process, while they are continuous in the intermediate stages of a printout. We should notice that images are first reconstructed to be continuous, then manipulated, and sampled again. Therefore, a continuous-domain definition of geometric distortions will be more appropriate. Examples of the images after general geometric distortions are shown in Figure 4-24.

In this section, we propose a general model, including multi-stage RSC in the continuous spatial domain, and discuss how to simplify it. We also show the change

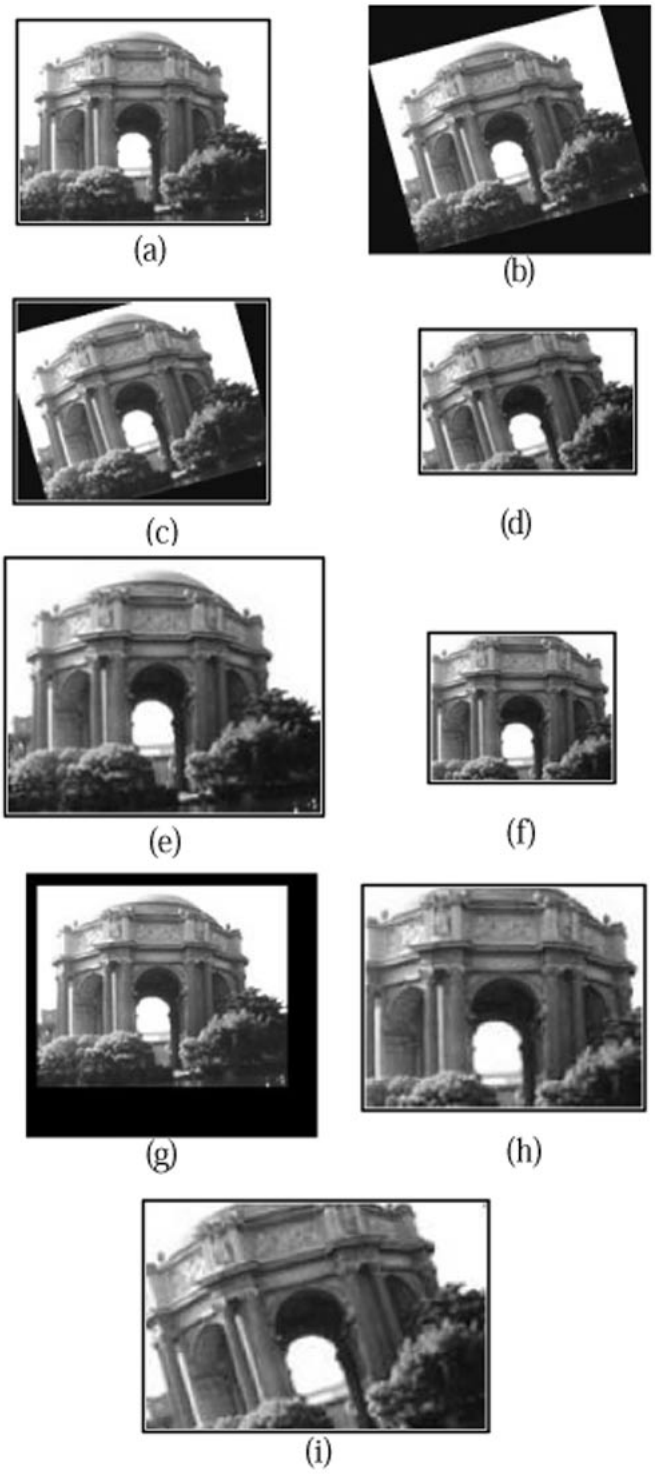



Figure 4-24: General geometric distortion of images: (a) original, (b) rotation and cropping with background and the whole image, (c) rotation and croppin with background and part of the image, (d) rotation and cropping with part of the image, (e) scaling, (f) croppin without background, (g) cropping with background, (h) scaling and cropping, and (i) rotation, scaling, and cropping

of Fourier coefficients after RSC. Since DFT is usually used for frequency-domain analysis of discrete images, we will discuss the impact of RSC in the DFT domain, and then show how to choose an appropriate method to calculate DFT coefficients for invariants extraction.

#### 4.7.2.1 Continuous-domain models for geometric distortion and the definition of RSC

Considering a general case of the geometric distortion introduced by multiple stages of rotation, scaling, and cropping, the distorted image can be represented as

$$\mathbf{x_G} = \mathbf{G} \cdot \mathbf{x}, \tag{4.26}$$

where $\mathbf{G}$ is the geometric distortion operator. For instance, $\mathbf{G}$ may equal to **RRSCSRCSRSSC**..., where $\mathbf{R}$, $\mathbf{S}$ and $\mathbf{C}$, are the operators of rotation, scaling and cropping, respectively.

We first show the individual effect of RSC. If the image is rotated by $\theta$ counterclockwisely, i.e., $\mathbf{x_R} = \mathbf{R} \cdot \mathbf{x}$, then

$$\begin{aligned} \mathbf{x_R}(t_1, t_2) &= \mathbf{x}(t_1 \cos\theta - t_2 \sin\theta, t_1 \sin\theta + t_2 \cos\theta) \\ &\longleftrightarrow \mathbf{X}(f_1 \cos\theta - f_2 \sin\theta, f_1 \sin\theta + f_2 \cos\theta) = \mathbf{X_R}(f_1, f_2) \end{aligned} \tag{4.27}$$

where $\mathbf{X}$ is the Fourier transform of $\mathbf{x}$. If the original image is scaled by $\lambda_1$ in the $t_1$-axis and $\lambda_2$ in the $t_2$-axis, i.e., $\mathbf{x_S} = \mathbf{S} \cdot \mathbf{x}$, then

$$\mathbf{x_S}(t_1, t_2) = \mathbf{x}(\frac{t_1}{\lambda_1}, \frac{t_2}{\lambda_2}) \longleftrightarrow \mathbf{X}(\lambda_1 f_1, \lambda_2 f_2) = \mathbf{X_S}(f_1, f_2). \tag{4.28}$$

We define cropping as the process that crops the image in a selected area (which may include part of background) at GUI window. Cropping introduces three effects

| | Operations in the continuous image domain | | |
|---|---|---|---|
| | **Scaling** | **Cropping** | **Rotation** |
| **Change of Fourier coefficients** | Scaling | Phase shift + (Information loss) | Rotation |

Table 4.1: Change of Fourier coefficients after operations in the continuous spatial domain.

on the image: (1) translation of the origin point of the image, (2) change of the support of image, and (3) information loss in the discarded area. They can be considered as a combination of translation and masking. It is well known that translation introduces only phase shift in the frequency domain. Masking includes the second and the third effects. In the continuous domain, the effect of changing support is not evident, because Fourier transform uses an infinite support, and ignores it. However, in the discrete domain, changing the support of image will change the image size. This results in significant effects on DFT coefficients. We will further discuss it in Section 4.7.2.2.

Changes of Fourier coefficients introduced by information loss can be considered in two ways. First, the cropped image could be a multiplication of the original image with a masking window, which introduces blurring (with the sinc function) in the Fourier domain. The other method is to consider the cropped image, $\mathbf{x_C}$, as a subtraction of the discarded area, $\mathbf{x_{\bar{C}}}$, from the original image, $\mathbf{x}$. Then, this equation,

$$|X_C(f_1, f_2)| = |X(f_1, f_2) - X_{\bar{C}}(f_1, f_2)| \tag{4.29}$$

represents the cropping effect in the continuous Fourier domain. We find that the second method is a better way to describe the cropping effect. From Eqs. (4.27), (4.28) and (4.29), we can see that rotation and/or scaling in the spatial domain re-

sults in rotation and/or scaling in the frequency domain, respectively, while cropping introduces phase shift and/or information loss. These are shown in Table 4.1.

Geometric distortion of RSC can also be represented by using coordinate mapping and masking. For instance, a geometric distortion of single rotation, scaling and cropping, sequentially, can be described by

$$\begin{bmatrix} t_1' \\ t_2' \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} + \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}, \tag{4.30}$$

and

$$\mathbf{x_G} = \begin{cases} \mathbf{x}', & (t_1, t_2) \in \mathbf{M} \\ 0, & elsewhere \end{cases} \tag{4.31}$$

$\mathbf{M}$ is a masking function and $\mathbf{x}$ is the image after coordinate mapping. Eqs. (4.30) and (4.31) show that RSC can be considered as RST + masking.

How to simplify Eq. (4.26)? One solution is to reduce multiple RSC operations to a combination of single rotation, scaling, and cropping. First, adjacent similar operations, e.g., **RRR**, can be represented by a single operation. Second, from Eq. (4.30), we can easily verify that **RC**, **SC** are all inter-changeable. In other words, a rotation operation after cropping can be substituted by a (different) cropping operation after rotation. We notice that **RS** is not inter-changeable unless the scaling factors in $t_1$ and $t_2$ dimensions are the same. Therefore, only in the case that images are scaled with the same aspect ratio can Eq. (4.26) be simplified. Or, Eq. (4.26) can also be simplified, if rotation is not allowed.

If we only focus on a simple print-and-scan process, then the geometric distortion of the image is a special case of Eq. (4.26). The manipulations are in the order of rotation, scaling, and cropping. We notice that, without deliberate adjustment, the scaling factor in this process is usually the same in both directions. Therefore, the

geometric distortion of PS process in the continuous domain can be described by Eq. (4.30) with the $\lambda_1 = \lambda_2$. In the continuous Fourier domain, the changes are a combination of Eqs. (4.27), (4.28), and (4.29). Unlike scaling, cropping usually results in a different image size that does not keep the aspect ratio of the original.

#### 4.7.2.2 Discrete-domain models for geometric distortion

We first define the geometric distortions in the discrete domain. The discretized image is sampled from distorted continuous image, $\mathbf{x_G}$. As we have mentioned, geometric distortion is better described in the continuous domain. Therefore, when we refer to a rotated discrete image, that means the image is converted to the continuous domain, then rotated, and sampled again using the original sampling rate. In practice, discrete images may not be really converted to the continuous domain, but it is possible to use interpolation to approximate this operation. The same definition applies to scaling and cropping. It should be noted that, because using a fixed sampling rate on the scaled continuous image is the same as using a different sampling rate on the original image, "change of sampling rate" and "scaling" indicate the same operation in the discrete-domain models.

It is well known that, in practical implementation, DFT coefficients can be obtained by using radix-2 FFT with zero padding. Some other fast methods of calculating DFT without using radix-2 FFT are also available. For example, Matlab calculates DFT coefficients by using the original size without zero padding. One of the two methods is usually used for calculating 2-D DFT of the sampled image. They are shown in Figures 4-25(a) and 4-25(c). Figures 4-25(b) and 4-25(d) show some alternatives mentioned in the literature. All of these methods can be used to obtain DFT coefficients. However, different calculation methods introduce different responses to the coefficients after geometric distortion. Unfortunately, this

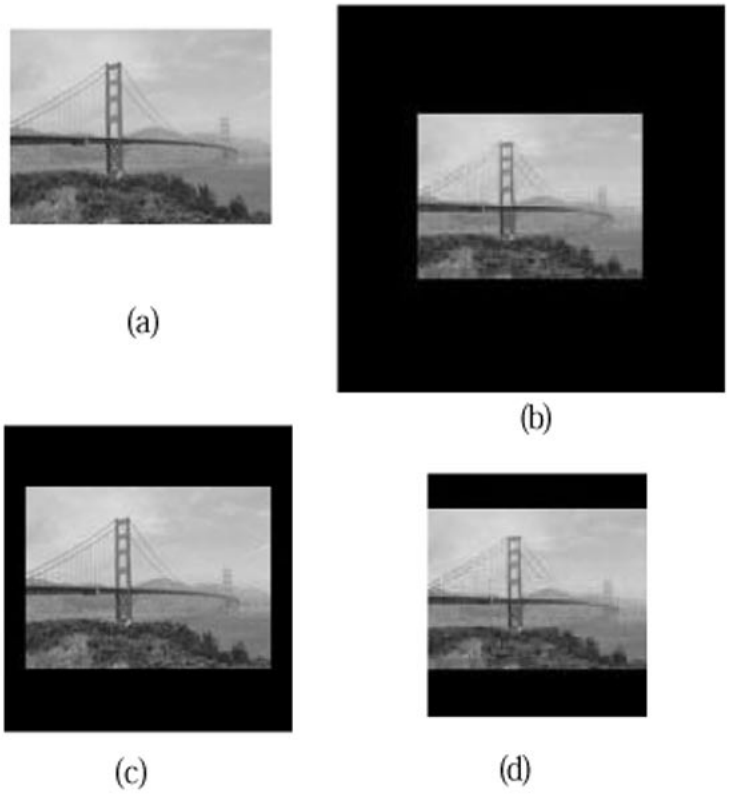



Figure 4-25: Four common methods to calculate DFT coefficients. The length and width of DFT window are: (a) the image size, (b) a fixed large rectangle, (c) the smallest rectangle with radix-2 width and height, or (d) the smallest square including the whole image.

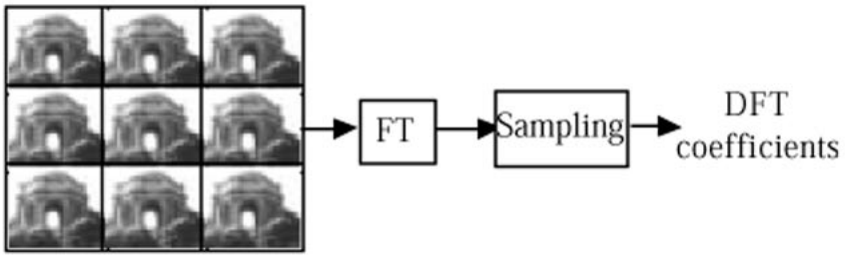



Figure 4-26: DFT coefficients are obtained from the repeated image.

phenomenon is usually overlooked. In the following paragraphs, we will show some general properties of DFT coefficients, and then analyze them.

- **General properties of DFT coefficients**

We first show the relationships between continuous Fourier coefficients and DFT. Once a continuous image is discretized, its Fourier coefficients become periodic (and are continuous). They are called the Discrete-Time Fourier Transform (DTFT) coefficients. For images, because their support is finite, we can periodically repeat it in the spatial domain. This will discretizes DTFT coefficients, and gets DFT coefficients. In other words, DFT coefficients are sampled from the Fourier spectrum of the repeated discrete image (see Figure 4-26). Alternatively, if we first consider the periodicity of an image and then consider its discrete property, DFT coefficients will be the same as Fourier Series (FS) coefficients, with additional noise introduced by aliasing effect.

Figure 4-27 shows how DFT coefficients change with different spatial sampling rate and different DFT size. Figure 4-27(a) is a continuous 1D signal and its corresponding Fourier coefficients. This signal is then discretized. The DFT coefficients (DFT window size $T_0$) of the discretized signal are the samples in the frequency domain. Figure 4-27(b) shows that the frequency sampling interval ($f_0 = \frac{1}{T_0}$) is determined by the repetition period ($T_0$), i.e., the size of DFT. It is obvious that

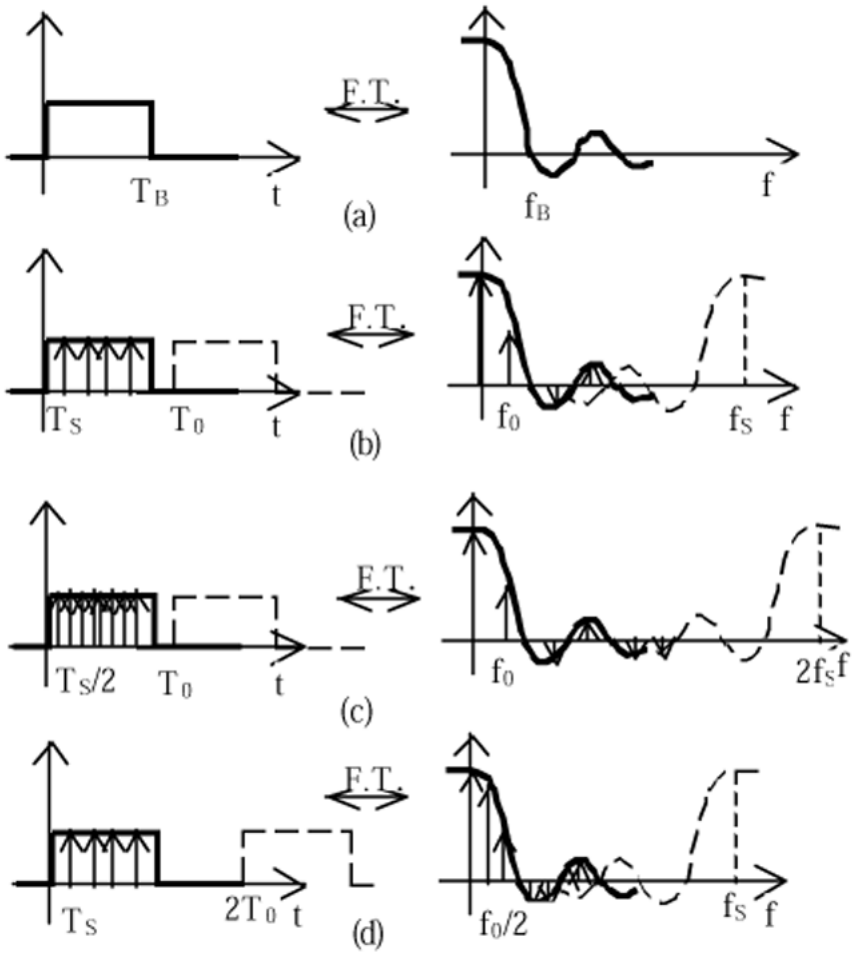



Figure 4-27: The relationship of DFT coefficients and Fourier coefficients: (a) the original continuous signal, (b) the discretized signal, (c) the up-sampled signal (or enlarged signal in a 2-D image), and (d) the zero-padded signal

DFT size plays an important role in the final coefficients. For example, consider the case when the DFT size keeps a fixed ratio to the signal/image size. Then, in Figure 4-27(c), if the signal is up sampled (or scaled) by 2, we can see that the sampling position of the DFT coefficients in Figure 4-27(b) and 4-27(c) are the same, with only difference in the aliasing effect. This is different from the continuous case, where scaling in the continuous domain results in scaling in the continuous Fourier domain. Figure 4-27(d) shows the effect of zero padding. The more we pad zeroes outside the image, the smaller the sampling interval in the frequency domain will be. Using these properties, we can model the change of DFT coefficients, which are calculated from the four cases in Figure 4-25, after geometric distortion.

**Case I: DFT size equals the image size**

In the first case, if the image is scaled, then the FS coefficients of the repeated original continuous image, $\tilde{\mathbf{X}}$, and the scaled image, $\tilde{\mathbf{X}}_\mathbf{S}$, should be the same at the same indices. That is,

$$\tilde{\mathbf{X}}_\mathbf{S}[n_1, n_2] = \mathbf{X}_\mathbf{S}(\frac{n_1}{T_{S1}}, \frac{n_2}{T_{S2}}) = \mathbf{X}(\frac{n_1\lambda_1}{T_{S1}}, \frac{n_2\lambda_2}{T_{S2}}) = \mathbf{X}(\frac{n_1}{T_1}, \frac{n_2}{T_2}) = \tilde{\mathbf{X}}[n_1, n_2], \quad (4.32)$$

where $T_{S1}$, $T_{S2}$ are the sizes of the scaled image, and $T_1$, $T_2$ are sizes of the original image. Adding the concern of discretization in the spatial domain, we can get the DFT coefficients in the scaled case, $\hat{\mathbf{X}}_\mathbf{S}$ as

$$\hat{\mathbf{X}}_\mathbf{S}[n_1, n_2] = \hat{\mathbf{X}}[n_1, n_2] + \mathbf{N}_{sampling}, \quad (4.33)$$

where $\hat{\mathbf{X}}$ is the DFT of original image. Eq. (4.33) indicates that, after scaling, the DFT coefficients at each indices are still the same as the original with only (sampling) aliasing noise. We can see this property from Figure 4-27(c). It should

be noted that $\hat{\mathbf{X}}_{\mathbf{S}} \subset \hat{\mathbf{X}}$ or $\hat{\mathbf{X}}_{\mathbf{S}} \supset \hat{\mathbf{X}}$, because the numbers of sampling points are difference in these two images. In Eq. (4.33), the power of sampling noise is larger, if the image is down-sampled.

In this case, the size of the cropped image will be the DFT size. If we assume this size to be $\alpha_1 T_1 \times \alpha_2 T_2$, then the DFT coefficients after scaling and cropping are,

$$|\hat{\mathbf{X}}_{\mathbf{SC}}[n_1, n_2]| = |\hat{\mathbf{X}}[\frac{n_1}{\alpha_1}, \frac{n_2}{\alpha_2}] + \hat{\mathbf{N}}_{\mathbf{SC}}[n_1, n_2]|, \tag{4.34}$$

where

$$\hat{\mathbf{N}}_{\mathbf{SC}}[n_1, n_2] = -\hat{\mathbf{X}}_{\bar{\mathbf{C}}}[\frac{n_1}{\alpha_1}, \frac{n_2}{\alpha_2}] + \mathbf{N}_{sampling} \tag{4.35}$$

In Eq. (4.34), if the cropped area include the entire original image, *i.e.*, $\alpha_1, \alpha_2 \geq 1$, then the effect of the discarded area can be ignored. If the cropping ratios are too small, then the power loss in the discarded area may not be just ignored as noises. The reliable minimum thresholds that can be considered as noises depend on the system design and specific images. In Eq. (4.35), strictly speaking, there is no definition in at the non-integer positions. But, since are samples of $\mathbf{X}$, we can set directly from the original Fourier coefficients. In practical applications, these values are generally obtained from interpolation.

In cases where DFT size equals image size, rotation in the spatial domain results in the same rotation in the frequency domain.

Several properties of the change of DFT coefficients after geometric distortions are listed in Table 4.2. In the other three cases, these properties can be readily verified by similar methods in the first case. Thus, we will only discuss them later.

**Case II: DFT size is a fixed large rectangle**

| | Operations in the discrete image domain | | |
|---|---|---|---|
| **DFT Size** | **Scaling** | **Cropping** | **Rotation** |
| **Case I** | Almost no effect* | Scaling + Phase shift + (Information loss) | Rotation |
| **Case II** | Scaling | Phase shift + (Information loss) | Rotation |
| **Case III** | Scaling | Phase shift + (Information loss) + (Scaling) | Rotation |
| **Case IV** | Scaling in one dimension and no effect* in the other dimension | Scaling + Phase shift + (Information loss) | Rotation |

*: No changes on sampling positions but may introduce different aliasing effect.

Table 4.2: Change of DFT coefficients after operations in the discrete spatial domain.

When calculating DFT, if the number of DCT coefficients is fixed, then the properties of RSC operations are the same in the DFT domain and the continuous Fourier domain. We can see it by comparing Table 4.1 and Table 4.2. In this case, previous discussions of the continuous cases are all valid in the DFT domain. However, this method is not practical because it requires a very large fixed-size DFT window for all images. In cases where DFT size is a fixed large rectangle, Eq. (4.34) and (4.35) are still applicable, but $\alpha_1$ and $\alpha_2$ should be replaced by $\lambda_1$ and $\lambda_2$.

**Case III: DFT size is the smallest rectangle with radix-2 width and height**

The third case in Figure 4-25(c) is widely used, but it introduces an unpredictable scaling effect, if image sizes change across the boundary of two radix-2 values, (*e.g.*, sizes changed from $127 \times 127$ to $129 \times 129$). This unpredictable property makes the invariant extraction process more difficult in practical applications. In this case, $\alpha_1$ and $\alpha_2$ in Eq. (4.34) and (4.35) should be replaced by other more complicated values that are functions of image sizes, scaling factors, and cropping factors.

**Case IV: DFT size is the smallest square including the whole image**

In this case, since cropping and scaling may also introduce unpredictable scaling effects in the DFT coefficients, similar problems occur as in Case III.

- **Rotation**

The DFT coefficients of the rotated image have two important properties: the 'cross' effect and the Cartesian sampling points. In Figure 4-28, we can see that the spectrum of the original image holds a strong cross, which is caused by the discontinuity of pixel values after the image is repeated as in Figure 4-26. After rotation, if the image includes the whole original and additional background, then this 'cross' will

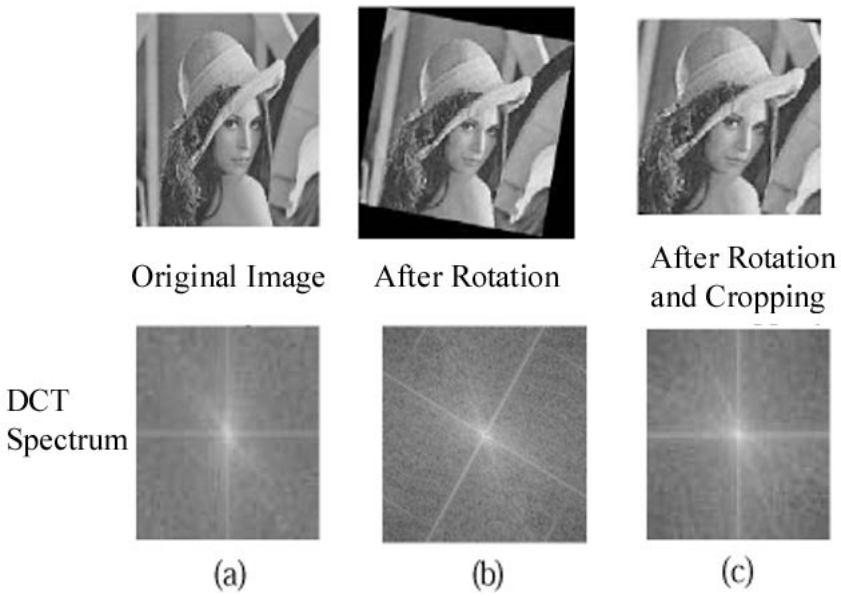



Figure 4-28: The spectrum of rotated-and-zero-padded image and rotated-cropped image

also rotate with the image. However, if the rotated image is cropped as in Figure 4-28, then this cross will not be rotated, while other coefficients are rotated. In other words, the shape of the support of image decides the angle of 'cross'. We found that this phenomenon becomes less noticeable, if images are subtracted by their mean values before calculating DFT coefficients. Observing from Figure 4-27(b) and 4-27(d), we can see the spectrum of a cropping mask The larger the distance of repetition period and the support of mask, the larger the magnitudes of the sidelobe pulses would be. Since these pulses convolve with all the DFT coefficients in the frequency domain, large DFT coefficients domain the values along the angle of the mask. In implementation, we have to notice this effect, and in acknowledge that some DFT values may be affected near the angle of mask.

DFT coefficients of a discrete rotated image are sampled from the Cartesian grid points of the rotated original continuous spectrum. Therefore, they are not the rotated original DFT coefficients. Two methods can be used to solve this problem in

practical cases. The first is to calculate DTFT coefficients at the rotated grid point positions. They are exactly the same sampling points as in the original. However, these calculations are time-consuming. The other method is to interpolate from the DFT coefficients. This method can take advantage of FFT, and can get reasonable results in experiments. To improve the accuracy, zero-padding may be applied to the image to conduct interpolation from denser frequency samples. In implementation, we chose to interpolate coefficients from the magnitudes of DFT coefficients, because phase shifting (introduced by translation) could have significantly changed the complex coefficients.

## 4.8 Extracting Invariants in the Print-and-Scan Process

**• Using scaled images for the DFT-domain analysis**

Using DFT as a frequency analysis tool, we can manipulate images a priori to make their DFT coefficients more predictable after geometric distortions. Here are two examples. We scale images uniformly or non-uniformly to a standard size (*e.g.*, $256 \times 256$), and then apply radix-2 FFT. (In the uniform scaling cases, we may need to pad zeros outside the scaled image.) From Eq. (4.33), we know that scaling introduces almost no effect on the DFT coefficients, if images are not extensively down-sampled.

As we discussed in Section 4.7.2.1, uniform scaling can be combined with the original single RSC in the PS process, and it still results in single RSC. Therefore, if both original and distorted images are uniformly scaled to a fixed size before calculating DFT, their DFT coefficients should demonstrate the same properties shown in the continuous Fourier domain. Therefore, we can conclude that the DFT coefficients obtained by this method only suffer single rotation, scaling, phase shift, and noises in the PS process. (Here, scaling and phase shifting in the DFT domain

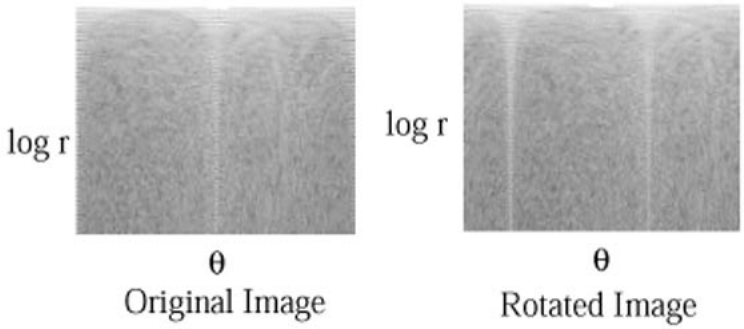



Figure 4-29: Log-polar map of DFT coefficients. RSC introduces simple shift on this map.

are introduced by cropping in the spatial domain, and noises are introduced by scaling and cropping in the spatial domain.)

An alternative method is to non-uniformly scale images to a standard size before calculating DFT. In some applications other than the PS process, such as operations in general image editing software, images may be cropped and scaled with an arbitrary aspect ratio but may not be rotated. This method can be applied to these applications. Examples can be found in [80].

**• Using log-polar or log-log map of DFT coefficients to extract invariants**

It is well known that the log-polar map of Fourier magnitudes possesses simple shifting properties, if images are rotated, translated and uniformly scaled. That is

$$|\mathbf{X_{RST}}(\log r, \theta)| = |\mathbf{X}(\log r + \log \lambda, \theta + \theta_R)| \tag{4.36}$$

where every coordinate point $(f_1, f_2)$ is represented by $(r\cos\theta, r\sin\theta)$. Eq. (4.36) can be easily verified from Eqs. (4.27) and (4.28). As we know, the DFT coefficients of a uniformly scaled image have similar properties as in the continuous Fourier coefficients. Therefore, Eq. (4.36) will be satisfied in the discrete DFT domain. We can use interpolation to obtain the coefficients at log-polar coordinate points.

Examples of the log-polar maps of Figures 4-28(a) and 4-28(b) are shown in Figure 4-29.

Since the log-polar map of the rescanned image is a translated version of the original (with noises), it is natural to expect that the 2D DFT magnitudes of this map should be invariant. Therefore, any function of them is expected to be invariant, and served as a feature vector. However, in practical cases, the noises introduced by scanning and cropping are too large to ignore. Also, in the discrete image cases, the invariant-magnitude property of Eq. (4.36) is valid only if images are cyclically shifted (because DFT is calculated from the repeated image, see Figure 4-26). This cyclic shift only happens at the axis of $\theta$, but not at the axis of $\log r$. Therefore, DFT magnitudes of this map usually do not possess a sufficient invariance.

An alternative method for generating feature vector has been developed in Section 4.3, and is summarized as follows. The basic idea is to project all log-polar coefficients along each angle, so that we can obtain a 1D signal that is invariant in the PS process except the cyclic shift introduced by rotation. The feature extraction process is shown in Figure 4-30. Images are first scaled to a standard size (*e.g.*, $256 \times 256$), then zero-padded to double the size (*e.g.*, $512 \times 512$). We can get the magnitudes of log-polar coefficients ($\mathbf{Fm}$) from DFT coefficients. The purpose of these steps is to get more accurate $|\mathbf{Fm}|$. The next step is to sum up the log values of $|\mathbf{Fm}|$ along each angle from $r_l$ to $r_u$, which includes mid-band coefficients. Log values of $|\mathbf{Fm}|$ are taken so that the summation will not be dominated by principal values, and the utilization of mid-band coefficients is for watermarking. This signal is then divided to two parts, and each value is summed up with the value at its orthogonal direction. There are two reasons. First, the resulted signal will be invariant if the rescanned image is rotated by 90°, 180°, or 270°. Second, its distribution will be more like white Gaussian, which is important to embedding watermark. The

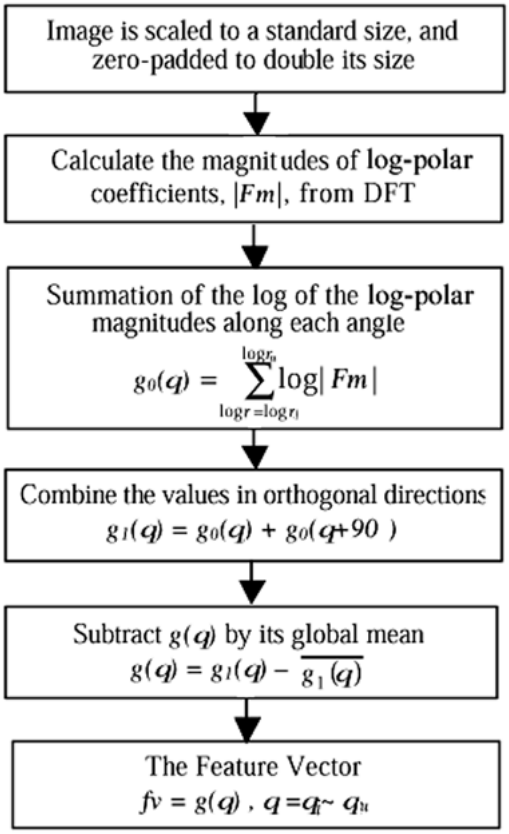



Figure 4-30: Extract invariants from log-polar map of DCT coefficients.

final feature vector is the AC component of this signal, which excludes the coefficients near the angle of the axes. This feature vector is very robust. We show some experimental results in Section 5. As mentioned above, rotation introduces cyclic shift to this feature vector. Therefore, in practical PS process, tests should base on shifting the original feature vector within a range, (*e.g.*, $\pm 5°$).

In addition to the method in Section 4.3, some other methods may be applied to extract invariants. The 1D DFT magnitude of the previous feature vector is an example, which is rotation invariant but less robust. Another example is to use a similar step, but sum up values along each $r$ or $\log r$. The resulted feature vector will be invariant to non-uniform scaling, rotation, and cropping to the scaled size. As we mentioned, in some cases, non-uniformly scaling and cropping is a more demanding process. We can use the log-log map instead of the log-polar map, because it only suffers simple shifting properties after general scaling and cropping [80].

## 4.9 Experiments

**• Pixel value Distortion**

We tested our models using the EPSON Stylus EX Inkjet printer and the HP Scanjet 4C scanner, both common commercial products. Five different images are tested, and they showed similar results. Here is an example. A color image of $384 \times 256$ was printed on the inkjet paper, with the physical size of $5.32" \times 3.54"$. Then it was scanned with 75 dpi [size: $402 \times 266$]. To isolate the inference of pixel value distortion, we crop, scale, and estimate its sub-pixel translation to register this image to the original. Experimental results are shown in Figures 4-31(a)- 4-31(e). We can see that the noises in the rescanned image are not like additive Gaussian noises. Instead, they depend on pixel values and the spatial distribution of pixels. In Figure 4-31(c), we show the mapping of the pixel values from the original image and

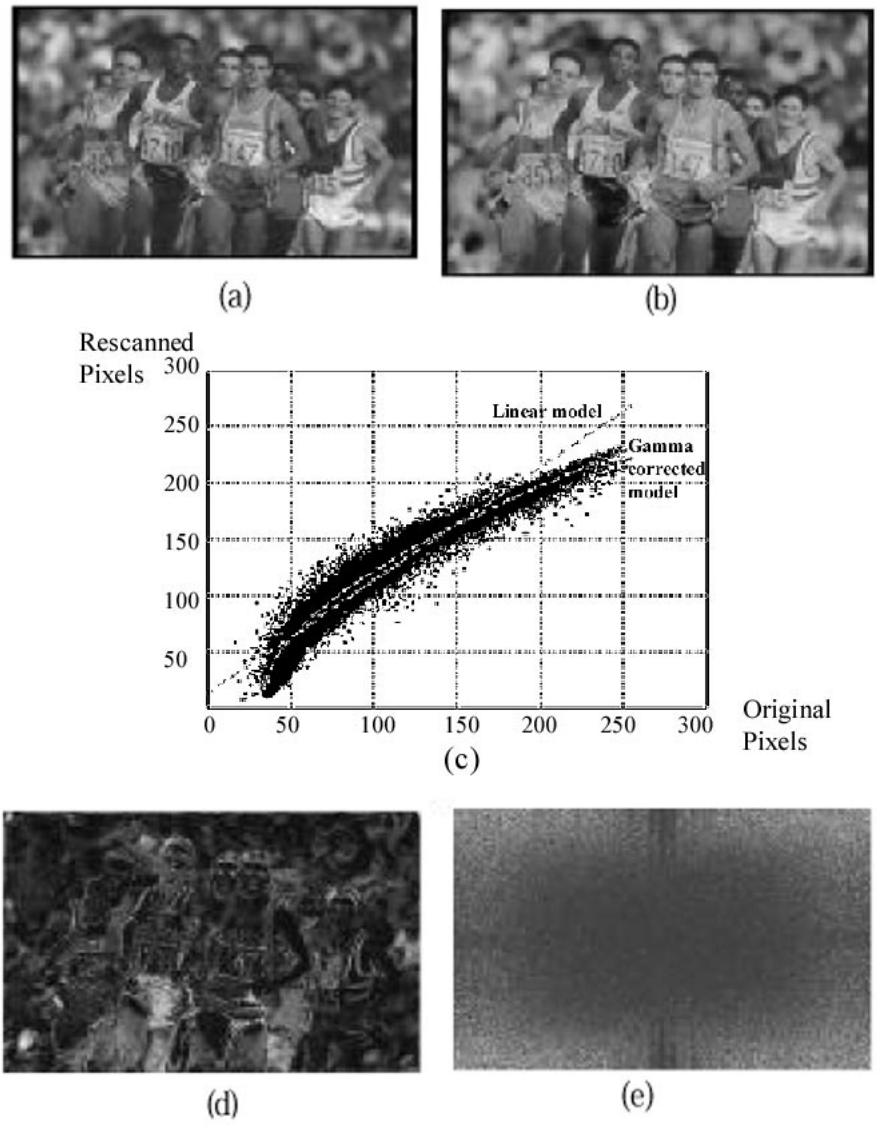



Figure 4-31: Pixel value distortion of rescanned image. (a) original image [384x256], (b) rescanned image [402x266], (c) corresponding pixel mapping and modeling, (d) noise in the spatial domain after gamma correction, (e) noise in the frequency domain after gamma correction

the registered rescanned image. We can see that Eq. (4.25) can suitably model the distribution of mapping function. We use optimum estimation methods to estimate $(\alpha, \gamma, \beta_x, \beta_K)$, which are (8.3, 0.6, 35, 20). The MSE of estimation noise is 73.58. At Figure 4-31(d), we show the difference between the original pixels and the gamma-corrected pixel values of the rescanned image. We can see that noises are larger in the edges and the dark areas. The former satisfies $N_1$ in Eq. (4.23), and the latter shows $N_2$ in Eq. (4.25). In Figure 4-31(e), we show the difference of the frequency spectrum of original image and gamma-corrected image. We can clearly see the lowpass filtering and high frequency noises in the spectrum.

The above experiment shows the effectiveness of our model of Eq. (4.23), (4.24) and (4.25). In the practical applications, however, if the original image is not available for registration, then we can not estimate the gamma corrected model. In that case, we can use a linear model, *i.e.*, $\gamma = 1$, in Eq. (4.25). In Figure 4-31(c), we can see the result of a linear model, which uses linear regression. The MSE of this model is 124.08. We can see that noises are larger when pixels are very bright or very dark. Noise distribution in the spatial domain is similar to Figure 4-31(d), but with larger variances in the bright areas Distribution of noises in the frequency domain is similar to Figure 4-31(e).

We also tested our models by scanning a photo 10 times, and comparing differences. The noise distribution satisfied $N_1$ in Eq. (4.23) of our model. Furthermore, we tested some of the cheapest consumer inkjet printers and scanners (which cannot print or scan images higher than 300 dpi), and found their quality is so bad that individual color halftone dots look very distinct in the rescanned image. In these cases, the rescanned images have to be further blurred by users to obtain merely acceptable images. Our hypothetical models are found sustainable with a lowpass filter of very low cutoff frequency.

- **Geometric Distortion**

We use the famous Lenna image [$512 \times 512$] as an example to show the properties of the described feature vector. The experimental results are shown in Figure 4-32. Correlation coefficients, $\rho$, of the feature vector extracted from the original image and the distorted image are used to measure the invariance. In our experience, if $\rho > 0.6$, then the extracted feature vector will be invariant enough to be applied to public watermarking. It should be noted that no information from the original image is needed for the feature vectors of rescanned image. In these experiments, $(r_l, r_u, \theta_l, \theta_u) = (34, 100, 8°, 81°)$.

In Figure 4-32(a), we show that the extracted feature vector is very robust to scaling. Testing is based on different scaling factors, $\lambda$ from 0.1 to 2.0. We found that $\rho > 0.98$ for all $\lambda > 0.25$. In other words, the feature vector extracted from a scaled image which is larger than 128x128 is almost the same as the original. Only if $\lambda < 0.12$, *i.e.*, 0.014 of the original area, then the correlation coefficient, $\rho$, becomes smaller than 0.6.

In Figure 4-32(b), we test its robustness against JPEG compression. The testing results are so good that $\rho > 0.988$ for all quality factors $> 30$, and $\rho > 0.947$ for all quality factors $> 10$.

In Figure 4-32(c), the cropped area only includes part of the original and no background (*a.k.a.*, strict cropping). We tested three ways: uniform, non-uniform, and one-side cropping. Uniform cropping means that the cropping ratios at both axes are the same. We choose cropping factors $0.6 \leq \alpha_1 = \alpha_2 \leq 1$, and show the result by the ratio of cropped area, *i.e.*, $\alpha_1 \times \alpha_2$. Non-uniform cropping uses different cropping ratios at axes. Their cropping factors are randomly chosen between 0.6 and 1. The one-side cropping method sets $\alpha_2 = 1$, and $\alpha_1$ from 0.6 to 1. These methods result in different image shapes, which affect the feature extraction process. For

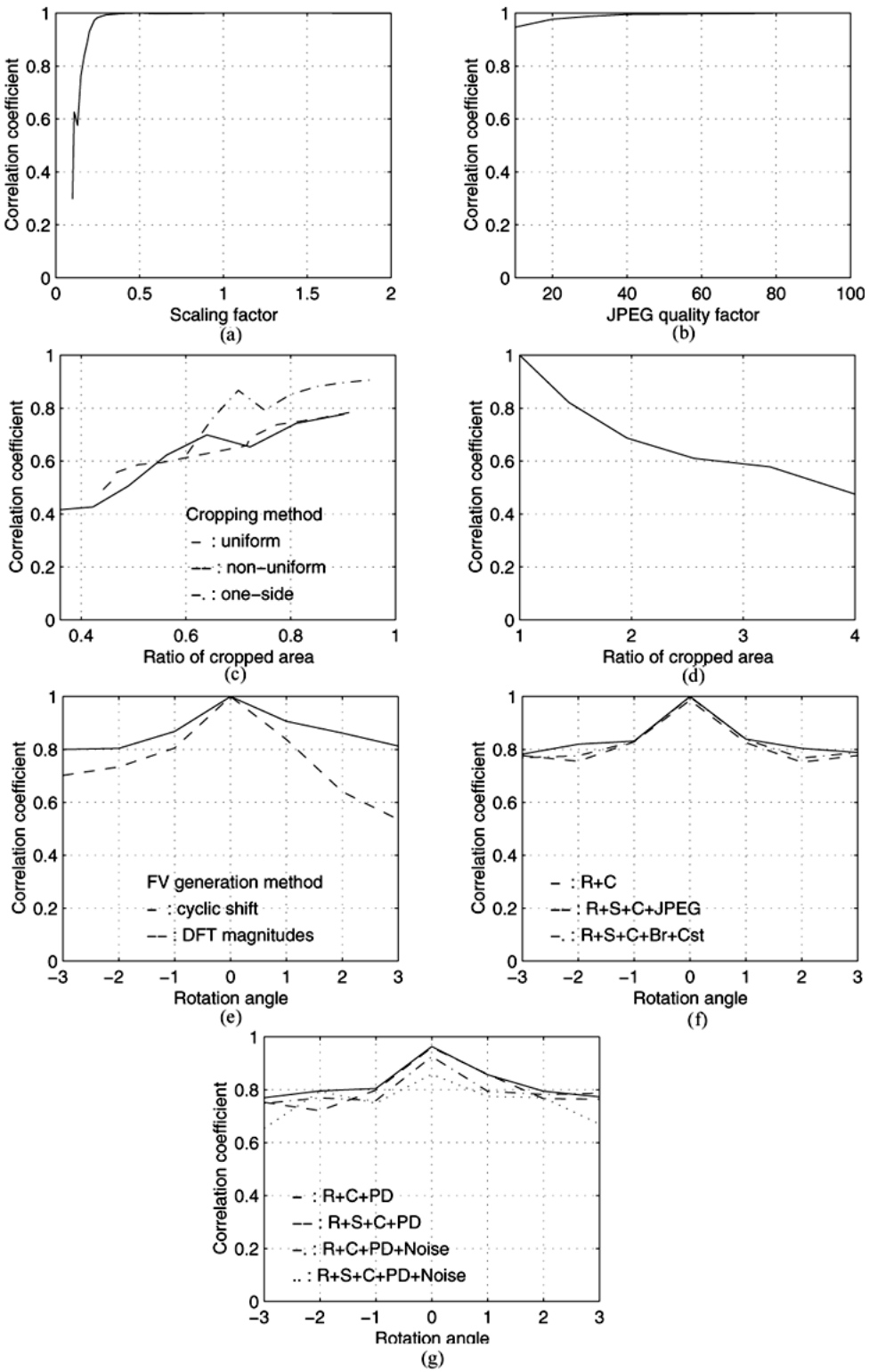



Figure 4-32: Robustness test of the extracted feature vector: (a) scaling, (b) JPEG, (c) strict cropping, (d) general cropping, (e) rotation with general cropping, and (f) rotation, strict cropping, scaling, and JPEG or brigtness/contrast adjustment, and (g) RSC, pixel distortion and noise.

instance, the largest size, *i.e.*, $max(width, height)$, of the distorted image after one-side cropping remains the same as the original. Because images are uniformly scaled to a standard size ($256 \times 256$) in the feature extraction process, the distorted image will be a subset of the original image. Therefore, only information loss results in the change of DFT coefficients. We can see that information loss is still acceptable if the ratio of cropped area $> 0.6$. The other two cases introduce scaling in the DFT coefficients, in addition to the information loss. We see that their correlation coefficients are smaller. But, no matter which method is used, a ratio of cropping area $> 0.6$ is usually acceptable.

In Figure 4-32(d), we show the test results of general cropping including background. We can see that $\rho < 0.6$ if the ratio of cropped area $> 2.5$. Distortions come from the scaling of DFT coefficients in the extraction process. Although it only introduces shifting in the log-polar map of the DFT coefficients, the loss of coefficients shifted outside the calculated range is too large to ignore. Experimental results are not good in this case. However, this kind of cropping is not common, and we can always crop the image again to obtain a better feature vector.

Figure 4-32(e) shows the experimental results of rotation. Images are rotated within $\pm 3°$, and then cropped to the original size. Because the extracted feature vector suffers cyclic shift, tests are based on the largest $\rho$ calculated by shifting the original feature vector in a range of $\pm 5°$. All $\rho$ values are all acceptable in these cases. To compare the extracted feature vector, we show the results of another method that uses the DFT magnitudes of the feature vector. This method does not require any cyclic test of the feature vector, but it is not as robust as the previous method.

- **Geometric distortion + pixel value distortion**

Figure 4-32(f) shows that the proposed feature vector is robust to a combined attack of rotation, scaling, cropping, JPEG, brightness and contrast adjustments. In this test, the image is rotated within $\pm 3°$, strictly cropped with the largest area that does not cover background, scaled with $\lambda_1 = \lambda_2 = 0.4$, and then either JPEG compressed ($qf = 75$) or brightness/contrast adjusted ($\alpha = 1.2$, $\gamma = 1$, $\beta_x = 0$, $\beta_K = 10$). Compared to Figure 4-32(e), we can see that distortion of feature vectors are mostly introduced by rotation and cropping, while the effects of scaling, JPEG compression, brightness/contrast adjustments are negligible. In Figure 4-32(g), we show the result of a combination of RSC and our pixel value distortion model. The parameters estimated in Figure 4-31 are used in these experiments: $(\alpha, \gamma, \beta_x, \beta_K) = (8.3, 0.6, 35, 20)$. We use an additive Gaussian noise ($\sigma = 8.5$) in these tests. Because it is distributed in all bands, it will be worse than the real situations in which noises only affect uncalculated high-band. We observed that noises have larger effect in downsized images. Comparing to Figure 4-32(f), we can see that their results are similar.

We tested the practical rescanned images in Figures 4-31(a) and 4-31(b), and obtained their correlation coefficient, $\rho = 0.915$. Applying the proposed feature vectors for watermarking, we have tested a large database of 20,000 color images from the Corel image library. Their results have proved the invariant properties of the feature vector (shown in Section 4.5). Also, a very low false positive rate in those experiments helped prove that the feature vectors from different images are mutually uncorrelated.

## 4.10 Conclusion

Geometric distortions continue to be a major weakness for many watermarking methods. We described a solution to the common problems of rotation, scale, and translation. This solution is related to earlier proposals in the pattern recognition literature regarding invariants of the Fourier-Mellin transform. However, unlike those proposals, we do not explicitly derive an invariance relationship.

Instead of creating a truly RST invariant signal, we create a signal that changes in a trivial manner as a result of rotation, scale, or translation. The calculation of this projection is performed by taking the Fourier transform of the image, performing a log-polar resampling and then integrating along the radial dimension. We note that an alternative implementation can be performed using the Radon transform [12]. We have investigated this implementation but do not report it here.

The one-dimensional watermark has a many-to-one mapping to the two-dimensional image space. This is advantageous, especially when the embedder is based on the principle of communications with side information. Our implementation is a very simple example of this principle and we believe that future work can lead to significant improvements.

Experimental results on a database of over 2,000 images demonstrate that the method is resilient to either rotations, scale changes or translations. The degree of resilience changes as a function of the probability of false positive. The results also demonstrate the weakness of this method to cropping and JPEG compression, attacks against which no steps have been taken in the design.

Future work will focus on more effective embedding and RST resilient watermarking designed to survive cropping and compression. Improvements in effectiveness are possible in the approximate inversion of the log-polar resampling and in the distribution of the difference signal to the log-polar coefficients. Methods based

on gradient descent will be investigated. Also, the current technique of uniform distribution does not fully exploit the visual properties of the host image.

We will examine techniques for building crop resistant watermarks that rely on first subdividing the image into a number of possibly overlapping tiles. The RST resilient watermark is then embedded in each of these tiles. The detection algorithm is applied to each tile and the results averaged together. With appropriate constraints on the tiling and the symmetry of the watermark this technique may provide the desired resilience to cropping.

Our contribution in this chapter also includes the new, extensive work on modeling the changes that digital images undergo in the print-and-scan process. We propose a model for the pixel value distortion, define the RSC-based geometric distortions, analyze the change of DFT coefficients after geometric distortion, and describe methods to extract invariant feature vector. Preliminary experimental testing of the pixel value distortion, as well as experimental analyses of the feature vector in Section 4.5, have indicated the effectiveness of the proposed models. In addition to the image watermarking applications, the proposed model may be used for other applications, such as image registration or authentication.

# Chapter 5

# Theoretical Watermarking Capacity of Images

## 5.1 Introduction

In watermarking schemes, multimedia data is considered as a communication channel to transmit messages. Users decode messages from the received data, which may have been distorted. In this chapter, we address the following important question: how much information can be reliably transmitted as watermarks without causing noticeable quality losses, under some distortions in the watermarked images? Our objective is to find theoretical bounds of image watermarking capacity based on the information theory and the characteristics of human vision systems.

A general watermarking model has been shown in Figure 1-3. If we ignore security issues, watermarking capacity is affected by invisibility and robustness requirements, as shown in Figure 1-5. There are three dimensions involved in this figure – visual quality, robustness, and amount of embedded information. Fixing any dimension, there exist tradeoff relationships between the other two dimensions. We say a watermark scheme is robust if we can extract embedded bits with an error probability deterministically equal to or statistically approaching zero. Visual quality represents the quality of watermarked image. In general, if we want to make the message bits more robust against attacks, a longer codeword or larger codeword

amplitudes will be necessary. However, visual quality degradation will become more significant. Similarly, given a fixed visual quality, there exists a trade-off between the information quantity of the embedded message and robustness. For instance, the fewer the message bits are embedded, the more redundant the codeword can be. Therefore, the codeword has better error correction capability against noises. It is our objective in this chapter to find the theoretic bounds of these trade-off curves.

### 5.1.1 Analysis of Watermarking Capacity Issues

In this subsection, we first analyze the characteristics of three watermarking capacity parameters: amount of embedded information, visual quality, and robustness. Then, in subsequent sections, we will present several analytic methods for analyzing the watermarking capacity.

- **How many message bits are transmitted in the watermark?**

As shown in Figure 1-5, messages are the information that the source want to transmit or preserve via watermarking. Messages are first encoded as codewords, which are embedded as a watermark. For example, a single bit message may be encoded as a pseudo-random sequence. The amount of message bits transmitted through watermarking is different from the length of codeword. Different applications may require different message lengths.

For copyright protection, two kinds of approaches have been proposed. The first approach considers copyright protection as a detection problem, in which the decision is made based on the correlation value between the extracted watermark and the original watermark bases. The second types of methods decode the transmitted message based on estimation, where the decoder estimates the message bits based on projection values of the received signal on several pre-agreed bases. In this method,

transmitting $n$ message bits requires minimally $n$ orthogonal bases. Although some authors called the first case 1-bit watermarking and the latter n-bit watermarking [44], the actual amount of information transmitted in the first case is not 1 bit. Instead, it should be considered $log_2(M)$ where $M$ is the number of bases used in the watermarking process. It is neither 1 bit nor the length of basis. On the other hand, the information transmitted in the second case is $n$ bits.

For authentication, the embedded watermark is not used for transmitting messages but for detecting forgery. If the authentication output is only an indication of whether this image is altered, then the transmitted information is 1 bit. If the authenticator can localize the altered area as small as $m_x \times m_y$ pixels, then we can consider that, at least, $\frac{I_x}{m_x} \times \frac{I_y}{m_y}$ bits of information are transmitted, where $I_x \times I_y$ is the image size.

- **What kind of changes are invisible?**

There has been much work in analyzing the "visibility" or "noticeability" of changes made to a digital image. We can roughly categorize them into three types. They vary in the extent of utilization of the Human Vision System (HVS) model.

Works of Type I consider that the just-noticeable changes are uniform in all coefficients in a specific domain, such as the spatial domain, frequency domain, or some transform domain. PSNR is a typical measure used in these works for assessing image quality. Works of Type II apply the human vision model to some extent. Works of Type III attempts to fully apply the HVS model to predict the visibility of changes.

The fact that some coefficient changes are not noticeable is due to the masking effect. The maximum un-noticeable changes (or equivalently the minimal noticeable changes) are sometimes called masks or the Just-Noticeable Distortion (JND). We

Figure 5-1: Binary noise pattern with strength equal to Chou's JND bounds

Figure 5-2: Sinusoidal pattern with strength smaller than or equal to Chou's JND bounds

should note that the meaning of "JND" is not consistent throughout the early literatures and some recent papers (especially later than 1997). In early literatures, the word "JND" is used as a measurement unit to indicate the visibility of the changes to a pixel or the whole image in two images[88, 133]. The measurement JND is a posterior measurement (*i.e.*, after changes are made). In some recent papers, JND is used to predict the maximum amount of invisible changes in a specific pixel or a transform coefficient of an image [18]. This is a measure made prior to the change. We found that several watermarking papers use the latter concept. However, no rigorous physical and psychological experiments have ever shown the existence of such a priori distortion estimation. In reality, whether a distortion is visible depends on both the image content and the distortion pattern. Therefore, a distortion model depending on the image content only may not be capable of predicting the visibility of distortion. An example is shown in Figure 5-1 and 5-2. We can see that different change patterns introduce different degrees of visibility (result in Figure 5-2 is more noticeable) although the additive changes are all within Chou's JND bounds [18].

Human Vision System models have been studied for over 30 years. These models were explored to describe human vision mechanisms such as spatial frequency-orientation channels, dependence of sensitivity on local contrast, adaptation, masking, spatial summation, and channel summation. In the literature, the most complete result involving the fields of image processing, image science, and vision science are two HVS models proposed by Lubin [88] and Daly [31]. Their models are built by matching many empirical experiments designed by vision scientists. Usually, these experiments are made by using grating (unidirectional sinusoidal signal) with a fixed contrast, instead of complex, natural image data. Although whether the results of these experiments can be directly applied to complex image data is still an

open issue [102], Lubin's and Daly's models could match these experimental results in a very good extent.

HVS models indicate that masking effects have different influences in different positions either in the spatial pixel domain, frequency domain, or frequency-orientation domain. Also, general distortions such as lossy compression, blurring, ringing, *etc.* do not generate uniform noises in these domains. Therefore, there are obvious limitations in using Type I analysis models to predict the visibility of changes. However, in practical applications, Type I analysis may be more computable and can be used to provide a generic lower bound for watermarking capacity. It is lower bound because Type I analysis usually utilizes the minimum of all the invisible change values of pixels or transform coefficients.

In image coding literatures, some research has applied human vision mechanism to some extent. We categorize them under the Type II approach. Works in [133, 135] include a series approaches in designing the quantization steps for the block-based DCT coefficients or wavelet coefficients . In [133], Watson *et. al.* proposed a content adaptive quantization method which applies some human vision mechanisms, such as local contrast and masking. These models are used to adaptively adjust quantization steps in each $8 \times 8$ block. In [135], they designed a quantization matrix for the wavelet coefficients that was conceptually similar to the role of the Quality Factor 50 matrix in JPEG. That matrix was not content dependent, and was determined by their initial limited experiments (with 3-5 people). They did try to estimate a content-adaptive quantization matrix, but no experiments were shown. These works may be useful for optimizing image coding parameters. However, much of characteristics in Human Vision Models derived from rigorous vision science experiments was not utilized. JPEG 2000 has considered this as an optional functionality [61]. An advantage of using wavelet coefficients or block-based DCT coefficients is

that decomposition is complete and orthogonal. Therefore, capacity estimations in individual coefficients can be summed up. We will discuss them in more details in Section 5.3.

- **How robust is the watermark?**

The robustness of a watermark is hard to define. It depends on what kind of applications the watermark is designed for. For instance, if a watermark is used for copyright protection, it has to survive different types of attacks, including filtering, noise, geometric distortion, non-linear distortion, digital-to-analog and analog-to-digital conversion, transcoding, *etc.* How to design a watermark to satisfy these requirements is still an open issue. A more feasible approach is to design watermarking methods that are robust in specific environments. One example is our work described in Chapter 4 that focuses on distortions caused by the print-and-scan processes. If a watermark is used for authentication, then acceptable manipulations may be explicitly constrained, *e.g.*, JPEG compression, as discussed in Chapter 3.

Depending on the availability of the source image in watermark decoding, watermarks can be private or public. With a private watermark, because the source image is also available in the decoder, the distorted watermarked image can usually be resynchronized and compared to the original image. Therefore, most distortions can be modeled as additive noises. In this case, watermarking capacity is much easier to find. In the case of public watermarking, there are many different designs, as we have discussed in Chapter 4. It is hard to find a universal model for public watermarking. In this case, we may ignore the geometric distortion that cannot be resolved due to the lack of source image and consider pixel value distortion only.

Another issue is that in an environment with finite states (*e.g.*, discrete value coefficients) and bounded-magnitude noises, transmission error can be actually zero,

instead of stochastically approaching zero. For example, if quantization (with a maximum quantization step) is the only source of distortion, since the added noise is finite, we can actually find the zero-error capacity of a digital image. We will prove this in Section 5.2.

### 5.1.2 Prior Works Based on the Information Theory

Previous works on watermarking capacity directly apply the work of Shannon [116] and Costa [20]. If the whole image has uniform watermark (or said codeword) power constraint and noise power constraint in all pixel locations, then the capacity problem of private watermarking is the same as the one solved by Shannon in his original information theory paper in 1948 [116]. With the same constraints, the capacity problem of public watermarking is the same as the problem that was described by Costa in 1983 [20]. In both cases, the capacity is the same. That is,

$$C = \frac{1}{2} log_2(1 + \frac{P}{N}) \ \ (bits/sample) \tag{5.1}$$

where P and N are the uniform power constraints of watermark and noise, respectively. With these uniform constraints, the image can be considered as a communication channel. Shannon also showed that the channel capacity is the same if the signal values are discrete [116].

In earlier watermarking works, public watermarking is sometimes considered as a special case of private watermarking while the power of source image is included as part of noise [23]. Therefore, intuitively, we can treat the source image as a noise in calculating the capacity. Costa's paper in 1983 [20] seemed to show that the capacity of public watermarking was the same as the private watermarking cases where the power of the source image can be excluded. Some recent watermarking papers have developed works based on this finding [19]. However, it is still an open

issue whether the above claim is theoretically correct. The reason is that, in Costa's work [20], the design of codewords has to depend on the source signal if we want to achieve the same capacity regardless of the existence of the source signal. In other words, codewords have to be specially designed for a specific source signal, and the codewords need to be transmitted to the decoder as well. In addition, in Shannon's theory, the channel capacity can only be achieved when the codeword length is infinite. Therefore, we think it is still too early to claim that, with uniform power constraint, public watermarking has the same capacity as private watermarking [19, 20].

Furthermore, we have shown, the Human Vision System (HVS) model tells us that different pixel locations have different sensitivity of noise. In other words, the power constraints of watermarks are actually not uniform. Also, general distortions such as lossy compression, blurring, ringing, *etc.* do not generate uniform noises in all pixel positions. Therefore, Shannon's and Costa's theories cannot be directly applied to estimate the watermark capacity.

Several works including Servetto [114] and Akansu [109], did consider the variant properties in different locations. They considered each pixel as an independent channel and then calculated the capacity based on the theory of Parallel Gaussian Channels (PGC) [22]. However, there are controversial issues with this treatment. Compared to the PGC model, images do not have the temporal dimension in each channel. In Shannon's channel capacity theory, the maximum information transmission rate of a noisy channel can be achieved only if the codeword length is large enough (*i.e.*, the number of samples in each channel is long enough). But given a single image, if we consider each pixel as a channel, then there is only one sample available in each channel and therefore it is impossible to achieve the estimated capacity.

The theoretical capacity analysis of an Arbitrarily Varying Channel (AVC) [28] looks more promising for this variant state problem. However, this theory dealt with cases in which the source power constraint in time (or in space) was described statistically, which is not the case for watermarking. An image may be a varying channel, but its power constraints on watermark are determined by the HVS model, and thus are not varying stochastically. Therefore, we have to develop a new information-theoretical framework for analyzing the watermarking capacity based on discrete-value and variant-state cases. We will present such a framework in Section 5.3.

### 5.1.3 Focus of this Chapter

In this chapter, we will investigate watermarking capacity in three directions:

1. *Watermarking capacity based on content-independent constraints on the magnitudes of watermarks and noises.* In Section 5.2, we will show that, in the case that the noise magnitudes are constrained, a capacity bound with "deterministic" zero error can be actually achieved. Specifically, we will find the zero-error capacity for private and public watermarking in a magnitude-bounded noisy environment. An example case is that, assuming the added noise is due to quantization (as in JPEG), we can calculate the zero-error capacity based on the setting of the magnitude constraints on watermark and noise. Note that we consider all pixels, watermarks and noises are discrete values, which occur in realistic cases.

2. *Watermarking capacity based on domain-specific masking effects.* In Section 5.3, we first show the capacity of private watermarking in which the power constraints are not uniform. Then, we apply several domain-specific HVS approximation models to estimate the power constraints and then show the

theoretical watermarking capacity of an image in a general noisy environment.

3. *Watermarking capacity issues based on actual Human Vision System models.* In Section 5.4, we first describe in details the most sophisticated Human Vision Systems developed by Daly and Lubin. Then, we discuss issues and possible directions in applying these models to estimation of the watermarking capacity.

## 5.2 Zero-Error Watermarking Capacity of Digital Image

In this section, we will show that, in an environment with finite states and bounded noises, transmission error can be actually zero, instead of approaching zero as contemplated in Shannon's channel capacity theory. We can find the zero-error capacity of a digital image if quantization is the only source of distortion such as in JPEG and the largest applicable quantization steps are determined in advance.

Shannon defined the zero-error capacity of a noisy channel as the least upper bound of rates at which it is possible to transmit information with zero probability of error [117]. In contrast, here we will show that rather than a probability of error approaching zero with increasing code length, the probability of error can be actually zero under the conditions described above. This property is especially needed in applications that no errors can be tolerated. For instance, in multimedia authentication, it is required that no false alarm occur under manipulations such as JPEG compression.

In this section, we will show that the semi-fragile watermarking method that we proposed in Chapter 3 is, in fact, one way of achieving the zero-error capacity. We will also show two sets of curves that represent the zero-error capacity. Although our discussion in this section will focus on image watermarking subject to JPEG manipulation, the zero-error capacity we showed here can be extended to other domains, such as the wavelet domain, as long as the noise magnitude is constrained.

### 5.2.1 Number of channels in an image

Here we consider the case that the maximal acceptable level of lossy compression is pre-determined. In such case, the maximal magnitude of added noise is bounded. As we mentioned in Section 5.1, JPEG is a bounded distortion process. Maximum distortion of each DCT coefficient is determined by the quantization step size. Since JPEG uses the same quantization table in all blocks, maximum distortion just depends on the position in the block and is the same for all coefficients from different blocks but at the same position.

Assume a digital image $\mathbf{X}$ has $M \times N$ pixels that are divided into $B$ blocks. Here, in the blocked-based DCT domain, $\mathbf{X}$ may be considered as

- Case 1: a variant-state discrete memoryless channel (DMC). Transmission utilizes this channel for $M \times N$ times.

- Case 2: a product of 64 static-state DMCs, in which all coefficients in the same position of blocks form a DMC. Each channel can be at most transmitted $B$ times. In other words, the maximum codeword length is $B$ for each channel.

- Case 3: a product of $M \times N$ static-state DMCs, in which each coefficient forms a DMC. Each channel can be at most transmitted once.

In most information theory research works, channel is usually considered as invariant in time and has uniform power and noise constraint. This is usually valid in communication. To the best our knowledge, time variant cases have been addressed in [1, 28, 29], called Arbitrarily Varying Channel (AVC). However, such a work on AVC may not be adequate to the watermarking problem because the channel does not vary in a statistically arbitrary way. We think that Case 2 is the best candidate for the capacity analysis problem if the image is only manipulated by JPEG. Assuming no error correction codes are used in this zero-error environment,

the codes in Case 2 will be sensitive to local changes. Any local changes may cause loss of the whole transmitted information in each channel. In applications that information bits have to be extracted separately from each block, Case 3 may be the best candidate. For instance, in the authentication case, some blocks of the image may be manipulated. By treating each coefficient as a separate channel (as in Case 3), we can detect such manipulations in a local range.

### 5.2.2 Zero-Error Capacity of a Discrete Memoryless Channel and a Digital Image

The zero-error capacity of discrete memoryless channel can be determined by applying adjacency-reducing mapping on the adjacency graph of the DMC (Theorem 3 in [117]). For a discrete-value channel, Shannon defined that two input letters are adjacent if there is a common output letter which can be caused by either of these two [117]. Here, in the JPEG cases, a letter means an integer value within the range of the DCT coefficient. An adjacency-reducing mapping means a mapping of letters to other letters, $i \rightarrow \alpha(i)$, with the property that if i and j are not adjacent in the channel (or graph) then $\alpha(i)$ and $\alpha(j)$ are not adjacent. In other words, it tries to reduce the number of adjacent states in the input based on the adjacency of their outputs. Adjacency means that $i$ and $j$ can be mapped to the same state after transmission. We should note that the problem of determining such a mapping function for an arbitrary graph is still wide open. Also, it is sometimes difficult to determine the zero-error capacity of even some simple channels [65]. For instance, it took more than 20 years and a brilliant idea of Lovasz [87] to show the Shannon's lower bound on the zero-error capacity for the pentagon graph was tight.

Fortunately, we can find an adjacency-reducing mapping and the zero-error capacity in the JPEG case. Assume the just-noticeable-change on a DCT coefficient is

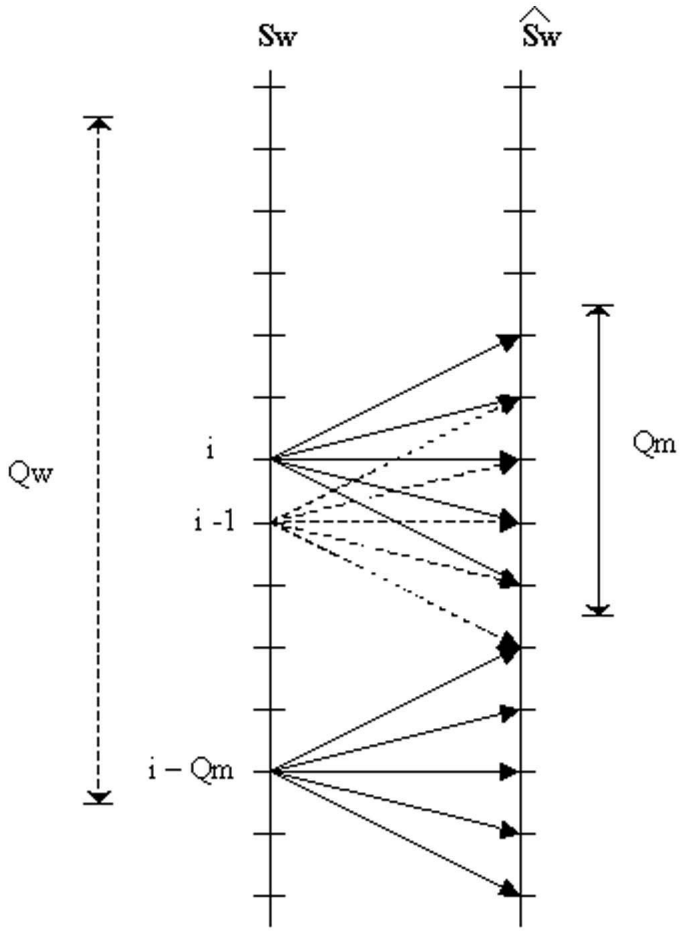



Figure 5-3: Adjacency-reducing mapping of discrete values in the appearance of quantization noise

$\frac{1}{2}Q_w$ [1] and the largest applicable JPEG quantization step to this coefficient is $Q_m$, then the zero-capacity of this channel will be

$$C(Q_w, Q_m) = \log_2(\lfloor \frac{Q_w}{Q_m} \rfloor + 1) \tag{5.2}$$

Eq. (5.2) can be proved by using the adjacency-reducing mapping as in [117]. Figure 5-3 shows an example to reduce adjacency points. Given a $Q_m$, which is the maximum quantization step that may be applied to the coefficient $S_w$, then the possible value $\hat{S}_w$ at the receiver end will be constrained in a range of $Q_m$ possible states. According to Shannon's adjacency-reducing mapping, we can find that the non-adjacent states have to separate from each other for a minimum of $Q_m$. For instance, assume the value of $S$ is $i$, then its closest non-adjacency states of $i$ are $i + Q_m$ and $i - Q_m$. To find out the private watermarking capacity, we assume that all the states within the range of $[i - \frac{1}{2}Q_w, i + \frac{1}{2}Q_w)$ are invisible. Therefore, there are $Q_w$ candidate watermarking states in this range. Since we have shown that the non-adjacent states have to separate from each other by $Q_m$, then there will be $\lfloor \frac{Q_w}{Q_m} \rfloor + 1$ applicable states in the $Q_w$ ranges that can be used to represent information without noticeable change. Therefore, from the information theory, we can get the capacity of this channel in Eq. (5.2). For instance, in Figure 5-3, $Q_w = 11$ and $Q_m = 5$. Using Eq. (5.2), we can obtain the capacity rate to be 1.59 $bits/sample$.

Eq. (5.2) is a bound for private watermarking with known source values $S$ in the receiver. However, in the public watermarking cases, $S$ is not known at the receiver end, *i.e.* $i$ is unknown. In this case, the number of applicable states in the just-noticeable range, $[i - \frac{1}{2}Q_w, i + \frac{1}{2}Q_w)$, will be larger than or equal to $\lfloor \frac{max(Q_w - Q_m, 0)}{Q_m} \rfloor + 1$.

[1] Note that $Q_w$ can be uniform in all coefficients in the same DCT frequency position, or they can be non-uniform if some human perceptual properties are applied. This is an open issue as discussed in Section 5.1. For Case 2, we assume the uniform property, while whether $Q_w$ is uniform or non-uniform does not affect our discussion in Case 3.

In other words, if $Q_w \geq Q_m$, then there is either $\lfloor \frac{Q_w}{Q_m} \rfloor + 1$ or $\lfloor \frac{Q_w}{Q_m} \rfloor$ applicable states. If $Q_w < Q_m$, then there may be no applicable state or only 1 state. Therefore, we can get the minimum capacity of public watermarking in this case,

$$\tilde{C}(Q_w, Q_m) = \log_2(\lfloor \frac{max(Q_w - Q_m, 0)}{Q_m} \rfloor + 1). \tag{5.3}$$

In Case 2, information is transmitted through $B$ parallel channels, whose capacity can be summed up [22]. The total zero-error capacity of an image surviving JPEG compression is, therefore,

$$C = \lfloor B \times \sum_{\nu \in V} \tilde{C}_\nu(\mathbf{Q_w}, \mathbf{Q_m}) \rfloor \tag{5.4}$$

where $V$ is a subset of $\{1..64\}$. Intuitively, $V$ is equals to the set of $\{1..64\}$. However, in practical situation, even though the changes are all within the JND of each coefficient, the more coefficients changed the more possible the changes are visible. Also, not always all the 64 coefficients can be used. We found that $V = \{1..28\}$ is a empirical reliable range that all coefficients are quantized as recommended in the JPEG standard by using some commercial software such as Photoshop and xv[2] Therefore, we suggest to estimate the capacity based on this subset. An empirical solution of $Q_w$ is $Q_{50}$, as recommended as invisible distortion bound in the JPEG standard. Although practical invisible distortion bounds may vary depending on viewing conditions and image content, this bound is considered valid in most cases [129]. Figure 5-4 shows the zero-error capacity of a gray-level $256 \times 256$ image.

In Case 3, we want to extract information through each transmission channel. Because the transmission can only be used once in this case, the information each

[2]Some application software may discard all the {29..64}th DCT coefficients regardless of their magnitudes.

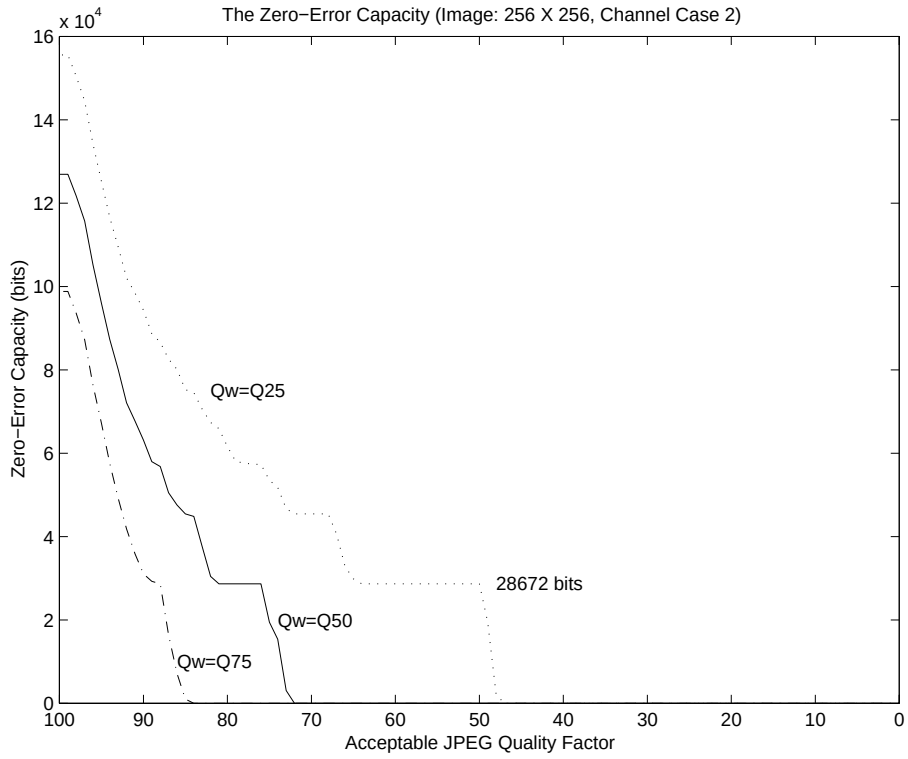



Figure 5-4: The Zero-Error Capacity of a $256 \times 256$ gray-level image for channel case 2

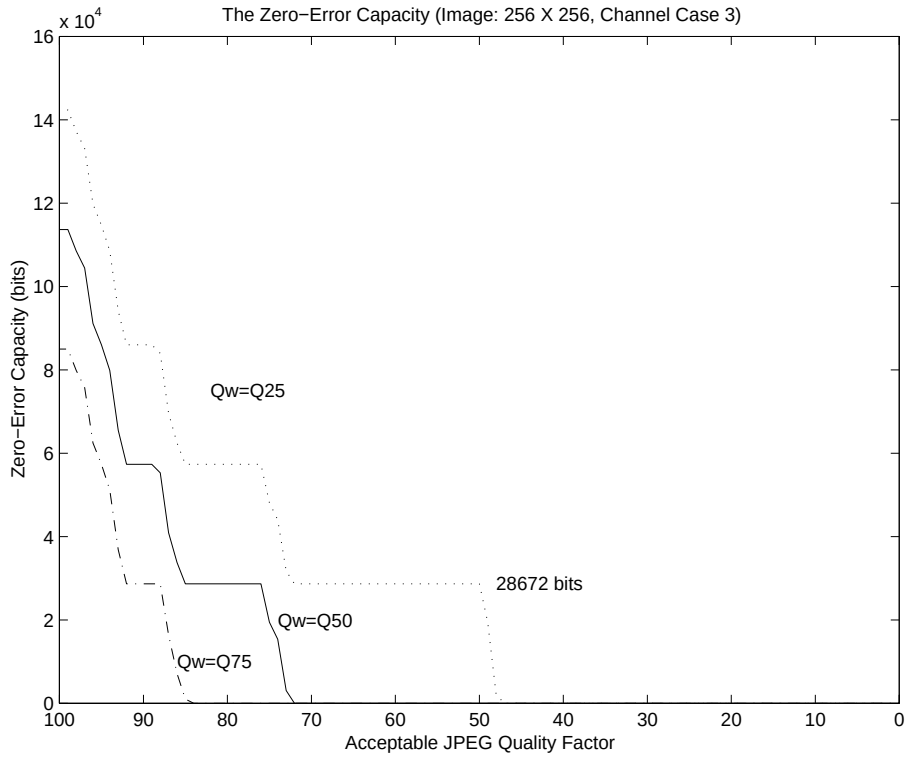



Figure 5-5: The Zero-Error Capacity of a $256 \times 256$ gray-level image for channel case 3

channel can transmit is therefore $\lfloor \tilde{C} \rfloor$. Similar to the previous case, summing up the parallel channels, then we can get the zero-error capacity of public watermarking in Case 3 to be

$$C = B \times \sum_{\nu \in V} \lfloor \tilde{C}_\nu(\mathbf{Q_w}, \mathbf{Q_m}) \rfloor \tag{5.5}$$

A figure of Eq. (5.5) is shown in Figure 5-5. These bits can be restored independently at each utilized coefficient. In other words, changes in a specific block would only affect its hidden information in that block.

### 5.2.3 Figures of Zero-Error Capacity Curve of Digital Images

In Figure 5-4 and Figure 5-5, we show the zero-error capacity of any $256 \times 256$ gray level image. Three different just-noticeable changes in the DCT coefficients are used. The curve $Q_w = Q_{50}$ is the just-noticeable distortion suggested by JPEG. In Figure 5-4, we can see that if the image is quantized by a JPEG quality factor larger or equal to 75, (*i.e.*, $Q_m \geq Q_{75} = \frac{1}{2}Q_{50}$) then the zero-error capacity of this image is at least 28672 *bits*, which is equal to 28 *bit/block*. This is a bound due to an empirical constraint that only the first 28 DCT coefficients are used in our calculation. It means that we can embed 28 bits of information in 28 coefficients of a block, and the message can be reconstructed without any error if the image is distorted by JPEG with quality factor larger or equal to 75. We can notice that when $75 < m \leq 72$, the capacity is not zero because some of their quantization steps in the quantization table are still the same as $Q_{75}$.

Comparing Eq. (5.5) with Theorem 4 in Chapter 3, we can see that Theorem 4 has utilized the zero-error capacity by embedding information similar to Case 3. The only difference is that, in SARI, because we fix the ratio of $Q_w = 2Q_m$, we embed one bit in each used channel. Therefore, the embedding rate is smaller or equal to

28 bits for each block[3]. Our experiments have shown that the previous estimated capacity bound can be achieved in realistic applications. More experimental results of this zero-error capacity analysis will be shown in [85].

## 5.3 Watermarking Capacity based on Domain-Specified Masking Effect

In this section, we investigate the watermarking capacity based domain-specified masking effects. We first derive the capacity of private watermarking when that power and noise constraints are not uniform across samples, *i.e.*, the capacity issue in a variant state channel. Then, we apply domain-specific HVS models to estimate the power constraints of watermarks. We will apply four models: Watson's DCT perceptual adaptive quantization method, Watson's wavelet quantization table, JPEG default quantization table and Chou's JND profile. We will show the theoretical private watermarking capacity based on these four methods.

### 5.3.1 Capacity of a Variant State Channel

We consider an image as a channel with spatial-variant states, in which the power constraint of each state is determined by HVS model or masking effect in some special domains. In this way, each coefficient is considered as an independent random variable with its own noise distribution. We will not consider a coefficient as a communication channel [109, 114] or sub-channel [67] because a channel usually indicates its reuse temporally, spatially, or in other domains.

Let $X_1, X_2, .., X_n$ be the changes of the coefficients in a discrete image due to watermarking. We first assume these values are continuous, and later we will show

[3]Because the length of recover bits are variant depending on the image content, some coefficients are not used for embedding

that the capacity is the same if they are quantized to discrete values. The power constraint of these values are the masking bounds determined by the source coefficient values $S_1, S_2, ..., S_n$. We define a masking function $f$ s.t. $E(\mathbf{X}\mathbf{X}^T) \leq f(\mathbf{S})$ where $\mathbf{X} = [X_1, X_2, .., X_n]^T$ and $\mathbf{S} = [S_1, S_2, .., S_n]^T$. Refer to Figure 1-3, $\mathbf{S_W} = \mathbf{S} + \mathbf{X}$. In the receiver end, consider $\mathbf{Y} = \hat{\mathbf{S}}_\mathbf{W} - \mathbf{S} = \mathbf{X} + \mathbf{Z}$ where $\mathbf{Z}$ are the noises added to the coefficients during transmission. Then, the maximum capacity of these multivariant symbols is

$$C = \max_{p(\mathbf{X}):E(\mathbf{X}\mathbf{X}^T)\leq f(\mathbf{S})} I(\mathbf{X};\mathbf{Y}) \quad given \;\; p(\mathbf{Z}) \tag{5.6}$$

where $p(.)$ represents any probability distribution and $I(.;.)$ represents mutual information.

From Eq. (5.6), because we can assume $\mathbf{X}$ and $\mathbf{Z}$ are independent, then

$$I(\mathbf{X};\mathbf{Y}) = h(\mathbf{Y}) - h(\mathbf{Y}|\mathbf{X}) = h(\mathbf{Y}) - h(\mathbf{Z}), \tag{5.7}$$

where $h(.)$ represents the differential entropy. According to Theorem 9.6.5 in [22], $\forall \mathbf{Y} \in \mathbf{R}^n$ with zero mean and covariance $\mathbf{K} = E(\mathbf{Y}\mathbf{Y}^T)$, the differential entropy of $\mathbf{Y}$, *i.e.*, $h(\mathbf{Y})$ satisfies the following

$$h(\mathbf{Y}) \leq \frac{1}{2} log(2\pi e)^n |\mathbf{K}|, \tag{5.8}$$

with equality iff $\mathbf{Y} \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$ and $|.|$ is the absolute value of the determinant. Here, this theorem is valid no matter what the range of $\mathbf{K}$ is.

Therefore, from Eq. (5.7), (5.8) and $|\mathbf{K}| = |E(\mathbf{Y}\mathbf{Y}^T)| = |E(\mathbf{X}\mathbf{X}^T) + E(\mathbf{Z}\mathbf{Z}^T)|$, we can see that

$$C = \frac{1}{2} log(2\pi e)^n |f(\mathbf{S}) + E(\mathbf{Z}\mathbf{Z}^T)| - h(\mathbf{Z}). \tag{5.9}$$

where we assume $f(\mathbf{S})$ is diagonal and nonnegative *s.t.* $|E(\mathbf{X}\mathbf{X}^T) + E(\mathbf{Z}\mathbf{Z}^T)| \leq$

$|f(\mathbf{S}) + E(\mathbf{Z}\mathbf{Z}^T)|$. This assumption means that embedded watermark values are mutually independent.

Eq. (5.9) is the watermarking capacity in a variant-state channel without specifying any type of noise. It is the capacity given a noise distribution. If we look at Eq. (5.9) and Theorem 9.6.5 in [22] again, for all types of noises, we can find that $C$ will be at least

$$\begin{aligned} C_{min} &= \tfrac{1}{2} log(2\pi e)^n |f(\mathbf{S}) + E(\mathbf{Z}\mathbf{Z}^T)| - \tfrac{1}{2} log(2\pi e)^n |E(\mathbf{Z}\mathbf{Z}^T)| \\ &= \tfrac{1}{2} |f(\mathbf{S}) + E(\mathbf{Z}\mathbf{Z}^T)^{-1} + \mathbf{I}|. \end{aligned} \tag{5.10}$$

when the noise is Gaussian distributed. If we further assume that noises are also independent in samples, then the watermarking capacity will be

$$C_{min} = \sum_{i=1}^{n} \frac{1}{2} log(1 + \frac{P_i}{N_i}) \tag{5.11}$$

where $P_i$ and $N_i$ are the power constraints in the $i-th$ coefficient, respectively. It is interesting that even though we use the multivariants to derive Eq. (5.11) instead of using Parallel Gaussian Channels, their results are the same in this special case.

For discrete values, we can apply Theorem 9.3.1 in [22], which shows the entropy of an n-bit quantization of a continuous random variable $X$ is approximately $h(X)+n$. Because, in general, only one kind of quantization would be used, in Eq. (5.6), we can see that the mutual information $I$ will be the same because $n$ will be deleted. Therefore, the capacity shown in Eq. (5.11) is still valid in the discrete value case.

### 5.3.2 Masking Effect in Specific Domains

General human vision mechanisms show that masking effects are decided by luminance, contrast, and orientation. Luminance masking, with its basic form of Weber's effect, describes that the brighter the background is, the higher the luminance

masking threshold will be. Detection threshold for a luminance pattern typically depends upon the mean luminance of the local image region. It is also known as light adaptation of human cortex. Contrast masking refers to the reduction in the visibility of one image component by the presence of another. This masking is strongest when both components are of the same spatial frequency, orientation, and location. Human vision mechanisms are sharply tuned to orientations. Orientation-selective channels in the human visual system were revealed through both human psychophysical means [15] and physical experiments on the visual cortex of cats [50, 51] and monkeys [106, 34].

Watson *el. al.* applied the two properties to coefficients in several different domains [104, 133, 134, 135]. Watson's model for DCT thresholds can be summarized as follows. First, an original just-noticeable-change, called mask, is assumed to be the same in all blocks. Then, these values are all adjusted by the DC values of the blocks, called luminance masking, and by the coefficients themselves, called contrast masking. Assume the original mask values are $t_{ij}, i, j = 0..7$ in all blocks, then, for the block $k$, set

$$t_{ijk} = t_{ij}(\frac{c_{00k}}{\bar{c}_{00}})^{a_T} \tag{5.12}$$

where $c_{00k}$ is the DC value of the block $k$ and $\bar{c}_{00}$ is the DC value corresponding to the mean luminance of the display ($\bar{c}_{00} = 128 \times 8 = 1024$ for an 8-bit gray level representation). The parameter $a_T$ is suggested by Ahumada and Peterson as 0.649 [2]. After luminance masking, we can then perform contrast masking to get the just-noticeable-change mask values, $m_{ijk}$, in the block $k$ as

$$m_{ijk} = max(t_{ijk}, |c_{ijk}|^{w_{ij}} t_{ijk}^{1-w_{ij}}) \tag{5.13}$$

where $w_{ij}$ is an exponent between 0 and 1. A typical empirical value of $w_{ij} = 0.7$

| | Level 1 | Level 2 | Level 3 | Level 4 |
|---|---|---|---|---|
| LL band | 14.05 | 11.11 | 11.36 | 14.5 |
| LH band | 23.03 | 14.68 | 12.71 | 14.16 |
| HH band | 58.76 | 28.41 | 19.54 | 17.86 |
| HL band | 23.03 | 14.69 | 12.71 | 14.16 |

Table 5.1: The quantization factors for four-level biorthogonal 9/7 DWT coefficients suggested by Watson et. al.

for $(i,j) \neq (0,0)$ and $w_{00} = 0$. Eq. (5.13) was derived based on Legge and Foley in [69]. In [133], Watson also defined a variable, just-noticeable differences (JND), as a measurement of coefficient distortion based on the mask and the distortion value of the coefficient. If necessary, the JNDs of coefficients in the image can be combined as a single perceptual distortion metric using pooling method based on Minkowski metric [133].

In [135], Watson *et. al.* proposed a method to estimate the mask values of wavelet coefficients. These mask values depend on the viewing environment, but are independent of content. Table 5.1 shows a list of recommended mask values in [135].

Chou and Li proposed a JND profile estimation method based on the luminance masking effect and the contrast masking effect [18]. This model is as follows:

$$JND(x,y) = max\{f_1(bg(x,y), mg(x,y)), f_2(bg(x,y))\} \tag{5.14}$$

where

$$f_1(bg(x,y), mg(x,y)) = mg(x,y) \cdot \alpha(bg(x,y)) + \beta(bg(x,y)) \tag{5.15}$$

$$f_2(bg(x,y)) = \begin{cases} T_0 \cdot (1 - (bg(x,y)/127)^{0.5}) + 3 & for\ bg(x,y) \leq 127 \\ \gamma \cdot (bg(x,y) - 127) + 3 & for\ bg(x,y) > 127 \end{cases} \tag{5.16}$$

$$\alpha(bg(x,y)) = bg(x,y) \cdot 0.0001 + 0.115 \tag{5.17}$$

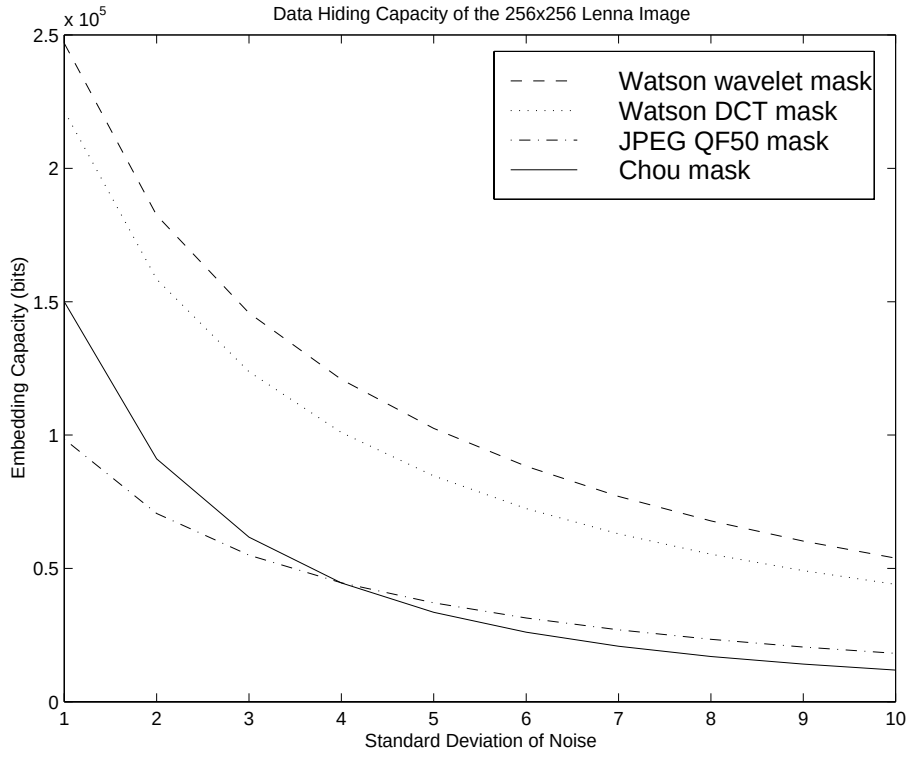



Figure 5-6: The estimated watermarking capacity based on four domain-specified masks

$$\beta(bg(x,y)) = \lambda - bg(x,y) \cdot 0.01 \tag{5.18}$$

The experimental result of the parameters are, $T_0 = 17$, $\gamma = \frac{3}{128}$, and $\lambda = \frac{1}{2}$. In this model, $bg(x,y)$ is the average background luminance, and $mg(x,y)$ is the contrast value calculated from the output of high-pass filtering at four directions. $f_1$ and $f_2$ model the contrast and luminance masking effects, respectively.

### 5.3.3 Experiments of Watermarking Capacity Based on Domain-Specified Masking Effects

In Figure 5-6, We show the estimated watermarking capacity based on Watson's DCT masking model, Watson's wavelet coefficient quantization table, JPEG recommended quantization table (*i.e.*, $\mathbf{Q}_{50}$) and Chou's JND profile. These four different

kinds of just-noticeable-change masks provide estimation of the watermarking power constraint $P_i$ in Eq. (5.11). Noises are assumed to be white Gaussian with its standard deviation from the range of $[1, .., 10]$. From Figure 5-6, we can see that the theoretical watermarking capacity indicates that this image can embed tens of thousand bits in the private watermarking cases. For instance, in the case where the standard deviation of noise is equal to 5 (PSNR = 34 dB), the theoretical estimated value are 84675, 102490, 37086, and 33542 bits, in the order of Watson's DCT mask, Watson's wavelet mask, JPEG quantization table, and Chou's JND profile, respectively.

## 5.4 Watermarking Capacity based on Human Vision System Model

In this section, we are interested in more complicated HVS, unlike the approximate ones used in the previous section. Specifically, we first introduce and compare two Human Vision System models that are developed by Lubin and Daly. Then, we will discuss how to calculate watermarking capacity based on these models. Because of the complexity of these models, there are some practical obstacles in directly applying them. Therefore, we do not be able to derive analytical results and show numerical results as prior sections. Instead, we point out interesting implications of these models and possible approaches for future works.

### 5.4.1 Human Vision Model

On interesting aspect of the HVS model is to predict whether an error in a picture is visible to an observer. The capability of allowing such an invisible error was later called "masking effect" [131]. This is primarily due to the optics of the eye, the sampling aperture of the cone photoreceptor, and both passive and active neural

| | Amplitude Nonlinearity | Intra-eye blurring | Re-sampling | CSF |
|---|---|---|---|---|
| Daly's VDP | Local Normalization | N/A | N/A | SQRI |
| Lubin's VDM | N/A | optics | 120 pxs/deg | SQRI |

| | Subband Decomposition | Masking Function | Pooling |
|---|---|---|---|
| Daly's VDP | Cortex Filters | coherence/learning effect | Probability Map |
| Lubin's VDM | Steerable Filters | dipper effect | JND Map |

Table 5.2: A comparison of two HVS models developed by Lubin and Daly

connections. From a historic review of HVS development, we see these models are derived by image processing scientists, first in the spatial pixel domain and later applying more human vision mechanisms such as the selective sensitivity of spatial-frequency and orientation. Later models incorporate more consideration of matching the experiments by vision scientists who used grating and contrast. In 1972, Stockham proposed a vision model for image processing, which is based on the nonlinear brightness adaptation mechanism of human vision [122]. Later additions on the HVS model that include transforming images into the frequency domain, color information, and orientation can be found in [47, 39, 88, 31]. In literature, the most complete results involving the fields of image processing, image science, and vision science are two HVS models proposed by Lubin [88] and Daly [31].

Both Lubin's and Daly's models include three steps. A calibration step, a masking measurement step in subbands, and a pooling step to generate a error-visibility profile in each spatial location of images. A comparison of the functionalities used in both models is shown in Table 5.2.

First, a calibration step is conducted. In Daly's model, this step includes a substep of pixel amplitude normalization using a nonlinear curve, which is based on the luminance adaption property of human retinal neurons, and a human contrast sensitivity function (CSF) calibration, which is a complex alternative to modulation transfer function (MTF). The amplitude nonlinearity is based on a shift-invariant,

simple point normalization curve derived by Daly *et. al.* [115]. The name "contrast" is basically used in the visual sensitivity experiments using gratings. In a complex image, the contrast is referred to as the value of spatial frequency coefficients. The CSF used in Daly's model is derived from the SQRI method of Barten [7, 8]. It was described in [8] as:

$$CSF(\nu) = a\nu \ exp(-b\nu)(1 + c \ exp(b\nu))^{\frac{1}{2}} \tag{5.19}$$

where $\nu$ is the angular spatial frequency in cycles/degree (cpd), and

$$a = \frac{540(1 + 0.7/L)^{-0.2}}{1 + 12/[w(1 + \nu/3)^2]}, \tag{5.20}$$

$$b = 0.3(1 + 100/L)^{0}.15, \tag{5.21}$$

$$c = 0.06. \tag{5.22}$$

where $L$ and $w$ are display luminance in $cd/m^2$ and display width in degrees. In Lubin's model, the calibration step includes a blurring function, which simulates the intra-eye optical point spread function (PSF) when the fixation distance differs from the image distance (*i.e.*, a calibration of myopia or hyperopia), and a sampling function to scale the image to a fixed square grid of 120 pixels/degree, which simulates the fixed density of cones in the fovea, based on the biological experiments on monkeys [146].

The second step of the models consider the masking functions. In both models, masking functions are applied to the intensity of spatial-frequency coefficients obtained by orientation-related filter banks. Daly uses Watson's cortex filters [132], which are performed in the DFT domain, as the filter bank. The basic idea of cortex filters is to divide the whole DFT spectrum into 5 circular subbands, and each subband is divided into 6 orientation bands. The boundary of subbands are step

functions convolved with Gaussian. These, with the orientation-independent base band, give a total of 31 subbands. Lubin uses the steerable myramid filters [63] as the filter bank. These filters are similar to an extended wavelet decomposition. They includes the 7 spatial-frequency decomposition and 4 orientation decomposition, with a total of 28 subbands. After subband decomposition, masking functions are applied to predict the tolerable changes in both models. Daly uses a masking function that is controlled by the type of image (noise-like or sine-waves) and the number of learning (*i.e.*, the visibility of a fixed change pattern would increase if the viewer observes it for multiple times). Lubin uses a masking function considering the dipper effect [99]. This masking function is,

$$T(e_i) = \frac{(k+2)|e_i|^n}{k|e_i|^{n-w} + |e_i|^m + 1}, \tag{5.23}$$

where $e_i$ is the coefficient value in subbands, $n = 1.5$, $m = 1.1$, $w = 0.068$, and $k = 0.1$. In Lubin's model, he applied the CSF (Eq. 5.19) in the subbands before this masking function, and called the two steps as a "transducer" stage. We should notice that this is an important difference between Lubin's model from Daly's model, in which CSF is applied in the DFT coefficients of the image. CSF and masking functions are the most important parameters in deciding the masking effect of images. CSF can be interpreted as a calibration function which is used to normalize the different perceptual importance in different spatial-frequency location. Masking functions determine how much change is allowed in each spatial-frequency location based on its value. We should note that masking function actives like a normalization function in predict the just-noticeable-change masks from the magnitude in a non-linear way.

The third step of the models is a mechanism to convert the measures of coefficient distortions into a number to indicate the human visual discretion capability toward

two images. The Daly's model shows a map of the probability of visibility. The Lubin's model shows a map of the JND unit value of each pixel, e.g., a JND value of 1 in a location means the change in this position is at about the visibility threshold. The distance measure is calculated based on the Minkowski metric of the output of masking function:

$$D_j = \{\sum_{k=1}^{m} |T_{j,k}(s_1) - T_{j,k}(s_2)|^Q\}^{\frac{1}{Q}} \tag{5.24}$$

where j indexes over spatial position, k indexes over the 28 frequencies and orientations, $s_1$ and $s_2$ are the two input images, T is the transducer output, and Q is a parameter set to 2.4. The Daly's model emphasizes threshold accuracy by duplicating psychophysical results concerning the visual system. The Lubin's model focuses on modeling the physiology of the visual pathway. An experimental comparison of these two HVS models can be found in [70]. In their tests, the Lubin's model was somewhat more robust giving better JND maps and requiring less re-calibration. The Lubin's model had better execution speed than the Daly's model but at the expense of using significantly more memory.

### 5.4.2 Capacity Measurements and Future Directions

We tried to measure the capacity based on Lubin's HVS model, because: 1) it has better experimental results in [70] and 2) there is no physical evidence showing that human eyes perform Fourier Transform [102] as in Daly's model. Lubin's wavelet-like steerable pyramid filters may be closer to the real physical function of human vision, in which high frequency components are affected by only adjacent areas instead the whole image as in DFT. To find out the capacity, we first need to rescale the image to $512 \times 512$. Then, we apply the 3rd-order directional derivatives of a circular-symmetric function as the steerable pyramid filter bank [63] to filter the rescaled image. We then get 28 subband images. To estimate the just-noticeable-

change from the masking output of image $s_1$, *i.e.*, $T_{j,k}(s_1)$, we first assume that all these 28 images introduce the same distortion in the masking function output. That means $T_{j,k}(s_2) = T_{j,k}(s_1) \pm (\frac{1}{28})^{\frac{1}{2.4}}$ if $D = 1$ in Eq. (5.24). After we get $T_{j,k}(s_2)$, we can estimate the just-noticeable magnitudes using some numerical method to get it from Eq. (5.23), and then normalize it based on the CSF (Eq. (5.19)).

Theoretically, we can use Eq. (5.10) to estimate the capacity, because Eq. (5.10) was valid in the non-orthogonal multivariant cases. We can then draw the capacity figures using similar methods as in Section 5.3. Some simple values have been calculated. However, to draw the whole figure, we found that this may not be realistic and is not computationally feasible because of its non-linearly in Eq. (5.23) and non-orthogonality in subband decomposition. This is a main reason why current HVS model may not be a good candidate in estimating the watermarking capacity.

From the discussion of the two most complete HVS models, we found that although they are constructed based on extensive physical and psychological experiments, we might be able to modify some model components so that they are more suitable for watermark capacity analysis. For instance, to satisfy the orientation-selective property of human vision, Lubin and Daly used specific filters, which were not justified with strong evidence. Because these orientation filters are main reason why the decomposed coefficients are not orthogonal, as an alternative, we may choose LH, HH, and HL bands of wavelet decomposition to substitute these orientation filters and derive its corresponding CSF and masking function based on the experiments in the vision field. We think this research will help us in designing a generic image quality assess method as well as estimating watermarking capacity.

## 5.5 Conclusion

In this chapter, we have presented several approaches to the challenging problem in estimating the theoretical bounds of image watermarking capacity. We analyze this problem with three considerations: the amount of embedded information, invisibility and robustness. First, we derived and demonstrated the zero-error capacity for public watermarking in environments with magnitude-bounded noise.[4] Because this capacity can be realized without using the infinite codeword length and can actually accomplish zero error, it is very useful in real applications. This work has also provided an theoretical framework for our semi-fragile watermarking work in Chapter 3, which has achieved the maximum amount of information that can be embedded if quantization is the only distortion noise to images.

In the second part of this chapter, we change the finite magnitude constraints to power constraints on watermark and noise. These constraints can be variant based on HVS. We have calculated the capacity of private watermarking with its power constraint estimated by domain-specific masking function. We analyzed the capacity by considering image pixels as multivariant, instead of parallel Gaussian channels. Using this model, the power constraint and noise constraint can be different in different positions, which is an important characteristic of human vision.

In the third part, we have introduced and compared the details of two complicated Human Vision Systems. We have pointed out the commonalities and major differences of these two models. We have also discussed and shown the difficulties in directly applying these HVS models to analyze watermarking capacity. Based on our analysis, we proposed potential future directions in which these models may be

[4]Note that the zero-error capacity exist because the magnitude of noises is bounded. Capacity is calculated by considering coefficients as multivariate. The power constraints of watermarking signal and noises need not be uniform.

modified to estimate watermark capacity.

There are still many issues about watermarking capacity not addressed in this chapter. For instance, in the second part, we still do not know how to estimate the capacity of public watermarking (with or without geometric distortion). As we mentioned in Section 5.1, it is still an open issue whether Costa's capacity is valid for public watermarking.

# Chapter 6

# Conclusions and Future Work

A new economy based on information technology has emerged. People create, sell, and interact with multimedia content. The Internet provides a ubiquitous infrastructure for e-commerce; however, it does not provide enough protection for its participants. Lacking adequate protection mechanisms, content providers are reluctant to distribute their digital content, because it can be easily re-distributed. Content receivers are skeptical about the source and integrity of content. Current technology in network security protects content during one stage of transmission. But, it cannot protect multimedia data through multiple stages of transmission, involving both people and machines. These concerns have hindered the universal acceptance of digital multimedia. At the same time, they also stimulate a new research field: multimedia security.

In this thesis, we have successfully developed a robust digital signature algorithm and a semi-fragile watermarking algorithm. These algorithms help design the Self-Authentication-and-Recovery Images (SARI) system, demonstrating unique authentication capacities missing in existing systems. SARI is a semi-fragile watermarking technique that gives "life" to digital images. Like a gecko can recover its cut tail, a watermarked SARI image can detect malicious crop-and-replacement manipulations and recover an approximated original image in the altered area. Another

important feature of SARI is its compatibility to JPEG lossy compression. SARI authenticator is the only system that can sensitively detect malicious changes while accepting alteration introduced by JPEG lossy compression. The lowest acceptable JPEG quality factor depends on an adjustable watermarking strength controlled in the embedder. SARI images are secure because the embedded watermarks are dependent on their own content (and on their owner).

Extending the robust digital signature technique to video, we have also developed the first video authentication system that can accept some video transcoding operations but is able to reject malicious attacks.

We have also successfully developed a public watermarking, surviving pixel value distortion as well as geometric distortion. The watermark is embedded by shaping a 1-dimensional signal obtained by first taking the Fourier transform of the image, resampling the Fourier magnitudes into log-polar coordinates, and then summing a function of those magnitudes along the log-radius axis. We can therefore compensate for rotation with a simple search, and compensate for scaling by using the correlation coefficient as the detection metric. Our tests of false positive and robustness on more than 2,000 images have shown that the watermark is robust to rotation, scale and translation. In addition, we have tested its robustness to cropping and JPEG compression. We have also discussed and presented its applications for the print-and-scan applications.

In addition, we have addressed the important question of how much information can be reliably transmitted as watermarks without causing noticeable quality losses, while being robust to some manipulations. We demonstrated the zero-error capacity for public watermarking in a magnitude-bounded noisy environment. We studied the capacity of private watermarks, based on predictable human vision properties in specific domains. We showed the capacity by deriving the capacity in a spatial-

variant discrete channel, and estimated the distortion bounds by several domain-specific models. We also included some preliminary studies with respect to the watermarking capacity issues based on Human Vision System models.

- **Future Related Research Issues**

There are many more topics waiting to be solved in the field of multimedia security. We briefly describe some of them in the following.

In the area of *public watermarking* schemes for multimedia copyright protection, we need to (1) model multimedia signal distortion after various D/A transcoding processes and compression schemes, and (2) design public watermarking schemes for various multimedia formats based on the above distortion models and feature vector shaping.

For *multimedia authentication*, open issues include:

- Content-based Multimedia Authentication: The objective here is to develop multimedia authentication methods based on content analysis techniques. The focus should be on the mechanism of authenticating multimedia data using different representation layers and the stability of automatic feature extraction.

- Document Authentication: Documents include combinations of text, pictures, and graphics. This task may include two directions: authentication of digital documents after they are printed-and-scanned, and authentication of paper documents after they are scanned-and-printed or photocopied. The first direction is to develop watermarking or digital signature techniques for the continuous- tone images, color graphs, and text. The second direction is to develop half-toning techniques that can hide information in the bi-level half-tone document representations.

- Audio Authentication: The idea here is to study the state-of-the-art speech and speaker recognition techniques, and to embed the speaker (or his/her vocal characteristics) and speech content in the audio signal. This research also includes the development of audio watermarking techniques surviving lossy compression.

- Image/Video/Graph Authentication: This project will focus on developing authentication techniques to accept new compression standards (such as JPEG-2000) and general image/video processing operations, and reject malicious manipulations.

Another interesting application is *fingerprinting*, which embeds multiple receiver identifications as tracing evidence in content distribution. The difficulties are to design co-existing watermarks and to enlist the cooperation of the web server providers. Development of such a system will involve two aspects. The first aspect is to design multiple watermarking techniques. It may be derived from the existing TDMA, FDMA, and CDMA methods. The second aspect is to develop software for the web server and to design protocols for tracing illegal distribution.

For *information hiding* applications, interesting issues exist in developing high capacity information hiding techniques in the digital domain as well as in the traditional domain such as printed documents or magnetic audio/video tapes. It may be used for any purpose, *i.e.*, hidden annotation in controlling the usage of multimedia content. Related issues exist in broadcast monitoring.

For *multimedia security infrastructure* issues, there have been extensive discussion in the MPEG-4 and MPEG-21 communities on developing universal infrastructures for multimedia data. In our view, the infrastructure development involves several parts. The first part is to develop authentication and copyright protection protocols in cooperation with the Public Key Infrastructure (PKI) used for net-

work security. The second part is to design protocols for specific applications such as MPEG-4 video or JPEG-2000 images/documents. The third part is to develop compatible watermarking methods for hidden annotations or copyright declaration.

In addition to multimedia security projects, ideas in several directions have been derived throughout our research and are not covered in this thesis due to space constraint. For instance, we have investigated the possibility in using the self-authentication-and-recovery images for error concealment in wireless and Internet transmission. We think that the SARI technique may be a promising solution in solving the complex session control problem in multimedia multicasting system or the high re-transimission delay in a network with high packet loss rate. Through the research of human vision models in Chapter 5, we have also found the important relationships of the models that may help in developing a simple and good image quality measurement model in the future.

Our works in developing watermarking and digital signature techniques for multimedia authentication and copyright protection have demonstrated that, although there are still a lot of open issues, trustworthy multimedia data is a realistic achievable goal.

# References

[1] R. Ahlswede, "Arbitrarily Varying Channels with States Sequence Known to the Sender," *IEEE Trans. on Information Theory*, Vol. 32, No. 5, pp. 621-629, September 1986.

[2] A. J. Ahumada Jr. and H. A. Peterson, "Luminance-Model-Based DCT Quantization for Color Image Compression," *SPIE Human, Vision, Visual Processing, and Digital Display III*, 1992.

[3] S. Alliney, "Digital Analysis of Rotated Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 15, pp. 499-504, 1993.

[4] J. Altmann and H. J. P. Reitbock, "A Fast Correlation Method for Scale- and Translation-Invariant Pattern Recognition," *IEEE Trans. on Pattern Analysis and Machine Intlligence*, Vol. 6, No. 1, pp. 46-57, 1984.

[5] J. Altmann, "On the Digital Implementation of the Rotation-Invariant Fourier-Mellin Transform," *Journal of Information Processing and Cybernetics*, pp. 13-36, 1987.

[6] M. Barni, F. Bartolini, A. De Rosa, and A. Piva, "Capacity of the Watermark-Channel: How Many Bits Can Be Hidden Within a Digital Image?," *SPIE Conf. on Security and Watermarking of Multimedia Contents*, Vol. 3657, pp. 437-448, San Jose, January 1999.

[7] P. Barten, "The SQRI Method: a New Method for the Evaluation of Visible Resolution on a Display," *Proceedings of the Society for Information Display*, Vol. 28/3, pp. 253-262, 1987.

[8] P. Barten, "Subjective Image Quality of High-Definition Televesion Pictures," *Proceedings of the Society for Information Display*, Vol. 31/3, pp. 239-243, 1990.

[9] P. Bas, J.-M. Chassery and F. Davoine, "A Geometrical and Frequential Watermarking Scheme using Similarities," *SPIE Conf. on Security and Watermarking of Multimedia Contents*, Vol. 3657, pp. 264-272, San Jose, January 1999.

[10] D. Bearman, and J. Trant, "Authenticity of Digital Resources: Towards a Statement of Requirements in the Research Process," *D-Lib Magazine*, June 1998.

[11] S. Bhattacharjee and M. Kutter, "Compression Tolerant Image Authentication," *IEEE International Conf. on Image Processing*, Chicago, October 1998.

[12] R. N. Bracewell, "The Fourier Transform and Its Applications", *McGraw-Hill*, 1986.

[13] G. W. Braudaway, K. A. Magerlein and F. Mintzer, "Protecting Publicly-Available Images with a Visible Image Watermark," *IBM T.J. Watson Research Center Research Report*, RC 20336, January 1996.

[14] P. J. Brightwell, S. J. Dancer and M. J Knee, "Flexible Switching and Editing of MPEG-2 Video Bitsreams," *International Broadcasting Convention*, Amsterdam, Netherlands, pp. 547-552, September 1997.

[15] F. W. Campbell and J. J. Kulikowski, "Orientational Selectivity of the Human Visual System," *J. Physiol. (London)*, No. 187, pp. 437-445, 1966.

[16] D. Casasent and D. Psaltis, "Position, Rotation, and Scale Invariant Optical Correlation," *Applied Optics*, Vol. 15, No. 7, pp. 1795-1799, 1976.

[17] D. Casasent and D. Psaltis, "New Optical Transforms for Pattern Recognition," *Proc. of the IEEE*, Vol. 65, No. 1, pp.77-84, 1977.

[18] C.-H. Chou and Y.-C. Li, "A Perceptually Tuned Subband Image Coder Based on the Measure of Just-Noticeable-Distortion Profile," *IEEE Trans. on Circuits and Systems on Video Technology,* VOl. 5, No. 6, pp. 467-476, December 1995.

[19] J. Chou, S. S. Pradhan and K. Ramchandran, "On the Duality between Data Hiding and Distributed Source Coding," *Proceedings of the 33rd Annuanl Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, November 1999.

[20] M. H. M. Costa, "Writing on Dirty Paper," *IEEE Trans. on Information Theory*, Vol. IT-29, No. 3, pp. 439-441, May 1983.

[21] Corel Stock Photo Library 3, Corel Corportation, Ontario, Canada.

[22] T. M. Cover and J. A. Thomas, "Elements of Information Theory," *John Wiley & Sones, Inc.*, 1991.

[23] I. J. Cox, J. Kilian, T. Leighton and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," *NEC Research Institute Technical Report*, 95-10, 1995.

[24] I. J. Cox, J. Kilian, T. Leighton and T. Shamoon, "Secure Spread Spectrum Watermarking for Images, Audio and Video," *IEEE Trans. on Image Processing*, Vol. 6, No. 12, pp. 1673-1687, December 1997.

[25] I. J. Cox and M. L. Miller, "A Review of Watermarking and the Importance of Perceptual Modeling," *Proceedings of SPIE, Human Vision & Electronic Imaging II*, Vol. 3016, pp. 92-99, 1997.

[26] I. J. Cox, M. L. Miller and A. McKellips, "Watermarking as Communications with Side Information," *Proceedings of the IEEE*, Vol. 87, No. 7, pp. 1127-1141, 1999.

[27] S. Craver, N. Memon, B. L. Yeo and M. Yeung, "Can Invisible Watermarks Resolve Rightful Ownerships?," *IBM T.J. Watson Research Center Research Report*, RC 20509, July 1996.

[28] I. Csiszar and P. Narayan, "Capacity and Decoding Rules for Classes of Arbitrarily Varying Channels," *IEEE Trans. on Information Theory*, Vol. 35, No. 4, pp. 752-769, July 1989.

[29] I. Csiszar and P. Narayan, "Capacity of the Gaussian Arbitrarily Varying Channel," *IEEE Trans. on Information Theory*, Vol. 37, No. 1, pp. 18-26, January 1991.

[30] G. Csurka, F. Deguillaume, J. J. K. O'Ruanaidh and T. Pun, "A Bayesian Approach to Affine Transformation Resistant Image and Video Watermarking," *Proc. of the 3rd Int. Information Hiding Workshop*, pp. 315-330, 1999.

[31] S. Daly, "The Visible Differences Predictor: An Algorithm for the Assessment of Image Fidelity," *Digital Images and Human Vision*, A. B. Watson, ed., pp. 179-206, MIT Press, 1993.

[32] E. De Castro and C. Morandi, "Registration of Translated and Rotated Images using Finite Fourier Transforms," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 9, pp. 700-703, 1987.

[33] G. Depovere, T. Kalker and J.-P. Linnartz, "Improved Watermark Detection using Filtering before Correlation," *IEEE Intl. Conf. on Image Processing*, Vol. 1, pp. 430-434, Chicago, October 1998.

[34] R. L. De Valois, E. W. Yund and N. Hepler, "The Orientation and Direction Selectivity of Cells in Macaque Visual Cortex," *Vision Research*, Vol. 22, pp. 531-544, 1982.

[35] W. Diffle and M. E. Hellman, "New Directions in Cryptography," *IEEE Trans. on Information Theory*, Vo. IT-22, No. 6, pp.644-654, November 1976.

[36] W. Diffie and M. E. Hellman, "New Directions in Cryptography," *IEEE Trans. on Information Theory*, Vol. 22, No. 6, pp.644-654, Nov 1976.

[37] W. Ding and B. Liu, "Rate Control of MPEG Video Coding and Recording by Rate-Quantization Modeling," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 6, No. 1, pp. 12-19, Feb 1996.

[38] A. Eleftheriadis and D. Anastassiou, "Constrained and General Dynamic Rate Shaping of Compressed Digital Video," *Proceedings of the 2nd IEEE International Conference on Image Processing (ICIP 95)*, Arlington, VA, USA, Oct 1995.

[39] O. D. Faugeras, "Digital Color Image Processing within the Framework of a Human Visual Model," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. 27, pp. 380-393, 1979.

[40] X. Feng, J. Newell and R. Triplett, "Noise Measurement Technique for Document Scanners," *SPIE vol. 2654 Solid State Sensor Arrays and CCD Cameras*, Jan. 1996.

[41] M. Ferraro and R. M. Caelli, "Lie Transform Groups, Integral Transforms, and Invariant Pattern Recognition," *Spatial Vision*, Vol. 8, No. 1, pp. 33-44, 1994.

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

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

[44] J. Fridrich and M. Goljan, "Comparing Robustness of Watermarking Techniques," *Proc. of SPIE Security and Watermarking of Multimedia Contents*, Vol. 3657, pp. 214-225, San Jose, CA, January 1999.

[45] G. L. Friedman, "The Trustworthy Digital Camera: Restoring Credibility to the Photographic image," *IEEE Trans. on Consumer Electronics*, Vol.39, No.4, pp.905-910, November 1993.

[46] R. Gennaro and P. Rohatgi, "How to Sign Digital Streams," *Proceeding of CRYPTO 97*, Santa Barbara, CA, USA, August 1997.

[47] C. F. Hall and E. L. Hall, "A Nonlinear Model for the Spatial Characteristics of the Human Visual System," *IEEE Trans. on Systems, Man. and Cyber.* Vol. 7, pp. 161-170, 1977.

[48] B. G. Haskell, A. Puri and A. N. Netravali, "Digital Video: An Introduction to MPEG-2," *Chapman & Hall*, 1997.

[49] M. K. Hu, "Visual Pattern Recognition by Moment Invariants," *IEEE Trans. on Information Theory*, Vol. IT-8, pp. 179-187, 1962.

[50] D. H. Hubel and T. N. Wiesel, "Receptive Fields of Single Neurons in the Cat's Striate Cortex," *J. Physiol. (London)*, No. 148, pp. 574-591, 1959.

[51] D. H. Hubel and T. N. Wiesel, "Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex," *J. Physiol. (London)*, No. 160, pp. 106-154, 1962.

[52] IETF Networking Group, "Internet X.509 Public Key Infrastructure Certificate Management Protocols," *Internet Engineering Task Force (IETF)*, RFC 2510, http://www.ietf.org /html.charters/pkix-charter.html, March 1999.

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

[54] International Business Machines, "Public Key Infrastructure: The PKIX Reference Implementation Project," *IBM SecureWay White Paper*, http://www.ibm.com/security/html/wp-pkix.pdf , August 1998.

[55] S. Jacobs and A. Eleftheriadis, "Streaming Video Using Dynamic Rate Shaping and TCP Flow Control," *Visual Communication and Image Representation Journal*, Jan 1998.

[56] A. Jaimes and S.-F. Chang, "A Conceptual Framework for Indexing Visual Information at Multiple Levels," *SPIE Internet Imageing 2000*, San Jose, CA, January 2000.

[57] A. K. Jain, "Fundamentals of Digital Image Processing," *Prentice-Hall Inc.*, pp. 80-99, 1989.

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

[59] N. F. Johnson and Z. Duric and S. Jajodia, "Recovery of Watermarks from Distorted Images," *Proc. of the 3rd Int. Information Hiding Workshop*, pp. 361-375, 1999.

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

[61] JPEG, "JPEG 2000 Part I Final Committee Draft Version 1.0," *ISO/IEC JTC1/SC 29 WH1*, April 2000.

[62] T. Kalker, G. Deprovere, J. Haitsma and M. J. Maes, "Video Watermarking System for Broadcast Monitoring," *Proceedings of the ICIP*, pp.103-112, October 1999.

[63] A. Karasaridis and E. Simoncelli, "A Filter Design Technique for Steerable Pyramid Image Transforms," *IEEE International Conf. on ASSP*, Atlanta, GA, May 1996.

[64] C. Kaufman, R. Perlman and M. Speciner, "Network Security: Private Communication in a Public World," *Prentice-Hall Inc.*, pp.129-162, 1995.

[65] J. Korner and A. Orlitsky, "Zero-Error Information Theory," *IEEE Trans. on Information Theory*, Vol. 44, No. 6, October 1998.

[66] D. Kundur and D. Hatzinakos, "Digital Watermarking for Telltale Tamper-Proofing and Authentication," *Proceedings of the IEEE – Special Issue on Identification and Protection of Multimedia Information*, Vol. 87, No. 7, pp. 1167-1180, July 1999.

[67] D. Kundur, "Water-Filling for Watermarking?," *IEEE Intl. Conf. on Multimedia & Expo*, New York, June 2000.

[68] M. Kutter, "Watermarking Resistance to Translation, Rotation, and Scaling," *SPIE Conf. on Multimedia Systems and Applications*, Vol. 3528, pp. 423-31, 1998.

[69] G. E. Legge and J. M. Foley, "Contrast Maskin in Human Vision," *Journal of the Optical Society of America*, Vol. 70, No. 12, pp. 1458-1471, 1980.

[70] B. Li, G. W. Meyer and R. V. Klassen, "A Comparison of Two Image Quality Models," *SPIE Conf. on Human Vision and Electronic Imaging III*, Vol. 3299, San Jose, January 1998.

[71] C.-Y. Lin and S.-F. Chang, "An Image Authenticator Surviving DCT-based Variable Quantization Table Compressions," *CU/CTR Technical Report 490-98-24*, November 1997.

[72] 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; also appear on *IEEE Trans. on Circuit and System for Video Technology*, 2001.

[73] C.-Y. Lin and S.-F. Chang, "A Robust Image Authentication Method Surviving JPEG Lossy Compression," *SPIE Storage and Retrieval of Image/Video Databases*, San Jose, January 1998.

[74] C.-Y. Lin and S.-F. Chang, "A Watermark-Based Robust Image Authentication Method Using Wavelets," ADVENT Project Report, Columbia University, April 1998.

[75] C.-Y. Lin and S.-F. Chang, "Generating Robust Digital Signature for Image/Video Authentication," *Multimedia and Security Workshop at ACM Multimedia 98*, Bristol, UK, Sep 1998.

[76] C.-Y. Lin and S.-F. Chang, "Issues and Solutions for Authenticating MPEG Video," *SPIE Conf. on Security and Watermarking of Multimedia Contents*, Vol. 3657, pp. 54-65, San Jose, January 1999.

[77] C.-Y. Lin and S.-F. Chang, "Issues and Solutions for Authenticating MPEG Video," *Proc. of SPIE Security and Watermarking of Multimedia Contents*, Vol. 3657, San Jose, CA, January 1999.

[78] C.-Y. Lin and S.-F. Chang, "Issues and Solutions for Video Content Authentication," *CU/ADVENT Technical Report*, Jan 1999.

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

[80] C.-Y. Lin, "Public Watermarking Surviving General Scaling and Cropping: An Application for Print-and-Scan Process," *Multimedia and Security Workshop at ACM Multimedia 99*, Orlando, FL, October 1999.

[81] C.-Y. Lin and S.-F. Chang, "Distortion Modeling and Invariant Extraction for Digital Image Print-and-Scan Process," *Intl. Symp. on Multimedia Information Processing*, Taipei, December 1999.

[82] C.-Y. Lin, M. Wu, M. L. Miller, I. J. Cox, J. Bloom and Y. M. Lui, "Geometric Distortion Resilient Public Watermarking for Images," *SPIE Security and Watermarking of Multimedia Content II*, San Jose, pp. 90-98, January 2000.

[83] C.-Y. Lin, M. Wu, Y. M. Lui, J. Bloom, M. L. Miller and I. J. Cox, "Rotation, Scale, and Translation Resilient Public Watermarking for Images," *IEEE Trans. on Image Processing*, 2001.

[84] C.-Y. Lin and S.-F. Chang, "Semi-Fragile Watermarking for Authenticating JPEG Visual Content," *SPIE Security and Watermarking of Multimedia Content II*, San Jose, pp. 140-151, January 2000.

[85] C.-Y. Lin and S.-F. Chang, "Zero-Error Data Hiding Capacity of Ditial Images," *IEEE Intl. Conf. on Information Technology: Coding and Computing*, Las Vegas, April 2001.

[86] F. Lin and R. D. Brandt, "Towards Absolute Invariants of Images under Translation, Rotation, and Dilation," *Pattern Recognition Letters*, Vol. 14, No. 5, pp. 369-379, 1993.

[87] L. Lovasz, "On the Shannon Capacity of a Graph," *IEEE Trans. on Information Theory*, Vol. IT-25, No. 1, pp. 1-7, January 1979.

[88] J. Lubin, "The Use of Psychophysical Data and Models in the Analysis of Display System Performance," *Digital Images and Human Vision*, A. B. Watson, ed., pp. 163-178, MIT Press, 1993.

[89] B. M. Macq and J. J. Quisquater, "Cryptology for Digital TV Broadcasting," *Proceedings of the IEEE*, vol. 83, No. 6, pp. 944-957, June 1995.

[90] M. J. Maes and C. W. van Overveld, "Digital Watermarking by Geometric Warping," *IEEE Intl. Conf. on Image Processing*, Vol. 2, pp. 424-426, 1998.

[91] K. Matsui and K. Tanaka, "Video-Steganography: How to Secretly Embed a Signature in a Picture," *IMA Intellectual Property Project Proceedings*, Vol. 1, pp. 187-206, 1994.

[92] M. McGuire, "An Image Registration Technique for Recovering Rotation, Scale and Translation Parameters," *Technical Report 98-018*, NEC Research Institute, 1998.

[93] N. D. Memon and P. W. Wong, "Protecting Digital Media Content," *Communications of the ACM*, Vol. 41, No. 7, July 1998.

[94] N. D. Memon and P. Wong, "Secret and Public Key Authentication Watermarking Schemes that Resist Vector Quantization Attack," *SPIE Intl. Conf. on Security and Watermarking of Multimedia Contents II*, Vol. 3971, San Jose, CA, January 2000.

[95] N. D. Memon and J. Fridrich, "Further Attacks on the Yeung-Mintzer Fragile Watermark," *SPIE Intl. Conf. on Security and Watermarking of Multimedia Contents II*, Vol. 3971, San Jose, CA, January 2000.

[96] J. Meng and S.-F. Chang, "CVEPS – A Compressed Video Editing and Parsing System," *Proceedings of ACM Multimedia 96*, Boston, MA, USA, Nov 1996.

[97] M. L. Miller and J. A. Bloom, "Computing the Probablity of False Watermark Detection," *Proceedings of the Third Intl. Workshop on Information Hiding*, 1999.

[98] M. L. Miller, I. J. Cox and J. A. Bloom, "Informed Embedding Exploiting Image and Detector Information during Watermark Insertion," *IEEE Intl. Conf. on Image Processing*, Vol. 3, pp.1-4, September 2000.

[99] J. Nachmias and R. V. Sansbury, "Grating Contrast: Discrimination May Be Better than Detection," *Vision Research*, Vol. 14, pp. 1039-1042, 1974.

[100] J. J. K. O'Ruanaidh and T. Pun,"Rotation, Scale and Translation Invariant Spread Spectrum Digital Image Watermarking," *Signal Processing*, Vol. 66, No. 3, pp. 303-317, 1998.

[101] The Oxford English Dictionary, 2nd Ed., *Oxford Univ.*, pp. 795-796, 1989.

[102] E. Peli, "Contrast in Comples Images," *Journal of Optical Society of America*, A/Vol. 7, No. 10, pp. 2032-2040, October 1990.

[103] S. Pereira and T. Pun, "Fast Robust Template Matching for Affine Resistant Image Watermarks," *Proc. of the 3rd Int. Information Hiding Workshop*, pp. 207-218, 1999.

[104] H. A. Peterson, A. J. Ahumada, and A. B. Watson, "An Improved Detection Model for DCT Coefficient Quantization," *Proceedings of SPIE*, Vol. 1913, pp. 191-201, 1993.

[105] F.A.P. Petitcolas and R.J. Anderson and M.G. Kuhn, "Attacks on Copyright Marking Systems," *Workshop on Information Hiding*, Portland, OR, 15-17 April 1998.

[106] G. C. Phillips and H. R. Wilson, "Orientation Bandwidths of Spatial Mechanisms Measured by Masking," *Journal of Optical Society of America*, A/Vol. 1, No. 2, February 1984.

[107] C. I. Podilchuk and W. Zeng, "Image-Adaptive Watermarking using Visual Models," *IEEE Trans. on Selected Areas of Communications*, Vol. 16, No. 4, pp. 525-539, 1998.

[108] 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.

[109] M. Ramkumar and A. N. Akansu, "A Capacity Estimate for Data Hiding in Internet Multimedia," *Symposium on Content Security and Data Hiding in Digital Media*, NJIT, Jersey City, May 1999.

[110] P. M. Rongen and M. J. Maes and C. W. van Overveld, "Digital Image Watermarking by Salient Point Modification Practical Results," *SPIE Conf. on Security and Watermarking of Multimedia Contents*, Vol. 3657, pp. 273-282, San Jose, January 1999.

[111] L. L. Scharf, "Statistical Signal Processing – Detection, Estimation, and Time Series Analysis," *Addison Wesley Inc.*, pp. 103-178, 1991.

[112] M. Schneider and S.-F. Chang, "A Robust Content Based Digital Signature for Image Authentication," *IEEE International Conf. on Image Processing*, Laussane, Switzerland, October 1996.

[113] B. Schneier, "Applied Cryptography," *John Willey & Sons. Inc.*, pp.461-502, 1996.

[114] S. D. Servetto, C. I. Podilchuk and K. Ramchandran, "Capacity Issues in Digital Image Watermarking," *IEEE Intl. Conf. on Image Processing*, Chicago, October 1998.

[115] M. I. Sezan, K. L. Yip, and S. Daly, "Uniform Perceptual Quantization: Applications to Digital Radiography," *IEEE Trans. on System, Man, and Cybernetics*, Vol. 17, No. 4, pp. 622-634, 1987.

[116] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, Vol. 27, pp. 373-423, 623-656, 1948.

[117] C. E. Shannon, "The Zero-Error Capacity of a Noisy Channel," *IRE Trans. on Information Theory*, IT-2: 8-19, 1956.

[118] C. E. Shannon, "Channels with Side Information at the Transmitter," *IBM Journal of Research and Development*, pp. 289-293, 1958.

[119] G. Sharma and H. Trussell, "Digital Color Imaging," *IEEE Trans. on Image Processing*, Vol. 6, No.7, July 1997.

[120] Y. Sheng and H. H. Arsenault, "Experiments on Pattern Recognition using Invariant Fourier-Mellin Descriptors," *J. Opt. Soc. Am. A*, pp. 771-776, 1986.

[121] S. D. Silvey, "Statistical Inference," *Chapman & Hall Inc.*, pp. 94-107, 1975.

[122] T. G. Stockham, Jr., "Image Processing in the Context of a Visual Model," *Proceedings of the IEEE*, Vol. 60, No. 7, July 1972.

[123] H. S. Stone and B. Tao and M. McGuire, "Analysis of Image Registration Noise Due to Rotationally Dependent Aliasing," *Technical Report 99-057R*, NEC Research Institute, 1999.

[124] Q. Sun, D. Zhong, S.-F. Chang and A. Narasimhalu, "VQ-based Digital Signature Scheme for Multimedia Content," *SPIE Intl. Conf. on Security and Watermarking of Multimedia Contents II*, Vol. 3971, San Jose, CA, January 2000.

[125] P. N. Tudor and O. H. Werner, "Real-Time Transcoding of MPEG-2 Video Bit Streams," *International Broadcasting Convention (IBC 97)*, Amsterdam, Netherlands, pp. 286-301, Sep 1997.

[126] Unzign, "http://altern.org/watermark."

[127] R. G. van Schyndel, A. Z. Trikel, and C. F. Osborne, "A Digital Watermark," *IEEE International Conf. on Image Processing*, Austin, Texas, Nov 1994.

[128] E. Viscito and C. Gonzales, "A Video Compression Algorithm with Adaptive Bit Allocation and Quantization," *SPIE Visual Communication and Image Processing 91*, Vol. 1605, 1991.

[129] G. K. Wallace, "The JPEG Still Picture Compression Standard," *Communications of the ACM*, Vol.34(4). pp. 30-44, April 1991.

[130] S. Walton, "Image Authentication for a Slippery New Age," *Dr. Dobb's Journal*, pp. 18-26, April 1995.

[131] A. B. Watson, "Estimation of Local Spatial Scale," *Journal of the Optical Society of America*, A 4, pp. 1579-1582, 1987.

[132] A. B. Watson, "The Cortex Transform: Rapid Computation of Simulated Neural Images," *Computer Vision, Graphics, and Image Processing*, Vol. 39, pp. 311-327, 1987.

[133] A. B. Watson, "DCT Quantization Matrices Visually Optimized for Individual Images," *Proceeding of SPIE*, Vol. 1913, pp. 202-216, 1993.

[134] A. B. Watson, "Perceptual optimization of DCT Color Quantization Matrices," *IEEE International Conf. on Image Processing*, Austix, TX, November 1994.

[135] A. B. Watson, G. Y. Yang, J. A. Solomon, and J. Villasenor, "Visibility of Wavelet Quantization Noise," *IEEE Trans. on Image Processing*, Vol. 6, No. 8, August 1997.

[136] The Webster's New 20th Century Dictionary.

[137] H. Wechsler and G. L. Zimmerman, "2-D Invariant Object Recognition Using Distributed Associative Memory," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 6, pp. 811-821, 1988.

[138] O. H. Werner, "Generic Quantiser for Transcoding of Hybrid Video," *Proceedings of the 1997 Picture Coding Symposium*, Berlin, Germany, Sep 1997.

[139] P. Wohlmacher, "Requirements and Mechanisms of IT-Security Including Aspects of Multimedia Security," *Multimedia and Security Workshop at ACM Multimedia 98*, Bristol, UK, Sep 1998.

[140] R. B. Wolfgang and E. J. Delp, "A Watermark for Digital Images", *IEEE International Conf. on Image Processing*, Laussane, Switzerland, Oct 1996.

[141] R. B. Wolfgang, C. I. Podilchuk and E. J. Delp, "Perceptual Watermarks for Digital Images and Video," *Proceedings of the IEEE*, Vol. 87, No. 7, pp. 1108-1126, 1999.

[142] H. Wong, W. Kang, F. Giordano and Y. Yao, "Performance Evaluation of A High-Quality TDI-CCD Color Scanner," *SPIE vol. 1656 High-Resolution Sensors and Hybrid Systems*, Feb 1992.

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

[144] L. Xie, K. Maeno, Q. Sun, C.-Y. Lin and S.-F. Chang, "Benchmarking for SARI Image Authentication System," *http://www.ctr.columbia.edu/sari/benchsari*, Oct. 2000.

[145] M. Yeung and F. Mintzer, "An Invisible Watermarking Technique for Image Verification," *IEEE International Conf. on Image Processing*, Santa Barbara, October 1997.

[146] R. W. Young, "The Renewal of Rod and Cone Outer Segments in the Rhesus Monkey," *Journal of Cell Biology*, Vol. 49, pp. 303-318, 1971.

[147] H. Yu, "Content-Based Graph Authentication," *Multimedia and Security Workshop at ACM Multimedia 99*, Orlando, FL, USA, October 1999.

[148] J. Zhao and E. Koch, "Embedding Robust Label into Images for Copyright Protection," *Proc. of the International Congress on Intellectual Property Rights for Specialized Information, Knowledge and New Technologies*, Vienna, Austria, August 1995.

[149] 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.