Abstract

Title of dissertation: MULTIMEDIA PROTECTION USING
CONTENT AND EMBEDDED FINGERPRINTS

Avinash Laxmisha Varna, Doctor of Philosophy, 2011

Dissertation directed by: Professor Min Wu
Department of Electrical and Computer Engineering

Improved digital connectivity has made the Internet an important medium for multimedia distribution and consumption in recent years. At the same time, this increased proliferation of multimedia has raised significant challenges in secure multimedia distribution and intellectual property protection. This dissertation examines two complementary aspects of the multimedia protection problem that utilize content fingerprints and embedded collusion-resistant fingerprints.

The first aspect considered is the automated identification of multimedia using content fingerprints, which is emerging as an important tool for detecting copyright violations on user generated content websites. A *content fingerprint* is a compact identifier that captures robust and distinctive properties of multimedia content, which can be used for uniquely identifying the multimedia object. In this dissertation, we describe a modular framework for theoretical modeling and analysis of content fingerprinting techniques. Based on this framework, we analyze the impact of distortions in the features on the corresponding fingerprints and also consider the problem of designing a suitable quantizer for encoding the features in order to improve the identification accuracy. The interaction between the fingerprint designer and a malicious adversary seeking to evade detection is studied under a game-theoretic framework and optimal strategies for both parties are derived. We then focus on analyzing and understanding the matching process at the fingerprint level. Models for fingerprints with different types of correlations are developed and the identification accuracy under each model is examined. Through this analysis we obtain useful guidelines for designing practical systems and also uncover connections to other areas of research.

A complementary problem considered in this dissertation concerns tracing the users responsible for unauthorized redistribution of multimedia. *Collusion-resistant fingerprints*, which are signals that uniquely identify the recipient, are proactively embedded in the multimedia before redistribution and can be used for identifying the malicious users. We study the problem of designing collusion resistant fingerprints for embedding in compressed multimedia. Our study indicates that directly adapting traditional fingerprinting techniques to this new setting of compressed multimedia results in low collusion resistance. To withstand attacks, we propose an anti-collusion dithering technique for embedding fingerprints that significantly improves the collusion resistance compared to traditional fingerprints.

MULTIMEDIA PROTECTION USING CONTENT AND EMBEDDED FINGERPRINTS

by

Avinash Laxmisha Varna

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2011

Advisory Committee:

Professor Min Wu, Chair/Advisor
Professor K. J. Ray Liu
Professor Rama Chellappa
Professor Adrian Papamarcou
Professor Lise Getoor

To my family

Acknowledgments

I am deeply indebted to my advisor Prof. Min Wu for her guidance during my PhD studies. She has always been supportive and has helped me navigate my way through graduate school from start to finish. She has lead me through a carefully crafted path, patiently, step by step, that has helped me learn the process of research. She has encouraged me to think creatively and maintain high standards in research. She has inculcated a questioning and exploring spirit in me, which is invaluable in research. From her, I have learnt that one should always be critical of one's own research and should not be afraid of facing the difficult questions. I have also learnt that through hard work and diligence, we can achieve even the most difficult tasks. She will always be an inspiration and role model for me.

I thank Prof. Rama Chellappa, Prof. Lise Getoor, Prof. Ray Liu, and Prof. Adrian Papamarcou for serving on my dissertation committee and their valuable comments on my thesis. I would also like to thank all the faculty at the University of Maryland, with whom I have taken courses and who have helped me at various stages of my studies. The courses provided me with a strong foundation which I could build upon in my research.

I also thank all my friends at the University of Maryland, who have made my stay at the university very pleasant and have been companions in this journey through graduate school. I have greatly benefited from their friendship, good will and support. Unfortunately, the limited space here does not allow me to exhaustively enumerate all of them, but thanks everyone! A special thanks to my colleagues in the MAST and SIG groups, with whom I have had many stimulating discussions on various aspects of academics and life. I also thank all the friends whom I came into contact with through Samskrita Bharati, who have helped me nurture and develop my hobby of learning Samskritam.

Finally, I would like to express my deepest gratitude to my parents and brother, who have always supported me in all my endeavors. From my childhood, my parents have taught me the value of education and instilled in me a sense of discipline and diligence. While giving me full freedom in choosing my path, they have inspired me to always aim higher, and have been a source of strength in achieving my goals. My brother has been a constant source of encouragement and has always kept me on my toes. My extended family, including my cousins and my relatives, have also supported me through this journey. Without all their support and encouragement, it would not have been easy for me to finish my graduate studies. I dedicate this thesis to my family.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Multimedia consumption via the Internet has increased radically over the last few years. The Internet has also emerged as an important medium for distribution of multimedia content such as video and audio. Video streaming services are available from such providers as Netflix, Blockbuster, Hulu, and Amazon. Services such as Google TV and Apple TV that are being planned will further strengthen this trend. Fueling this trend is the technological improvement in the bandwidth of network connections, and the growing popularity of user-generated content (UGC) websites, such as YouTube, which have changed the perspectives of both content providers and consumers with regards to the Internet.

At the same time, this ease of access has brought significant challenges to intellectual property protection, as the improved technology has made it easier to redistribute copyrighted multimedia content to a large number of users. The popularity of UGC websites has also raised concerns about the posting of copyrighted content by users. The movie industry estimated that piracy and illicit redistribution have caused over \$6 billion loss annually in terms of lost revenue [80, 82].

The problem of secure distribution and protection of multimedia in the Internet age raises significant technical challenges and various new technologies have been developed to address these issues. In this dissertation, we examine two important aspects of multimedia protection that rely on *digital fingerprints*.

1.1 Content Fingerprints for Multimedia Identification

One approach to protecting intellectual property rights is to reduce the number of avenues available for illicit multimedia redistribution. Most UGC websites have adopted rules and policies that discourage users from uploading copyrighted content and are seeking technological solutions for implementing these policies. A technology that automatically identifies videos can enable UGC websites to filter out copyrighted content and protect themselves from costly lawsuits and litigation. One promising solution to this problem relies on a technology called *content fingerprints*. A content fingerprint, in this context, is a compact signature that represents robust and unique characteristics of the multimedia and can be used to identify the document. For each video uploaded, the service provider can compute the fingerprint and compare it with a database of fingerprints of copyrighted content, to decide whether the video in question is copyrighted or not.

Content fingerprints are also used for identifying multimedia in a variety of other applications in multimedia management. Fingerprints are employed by such services as Shazam, Midomi, VCAST, etc. to perform automatic music identification. Given a noisy recording of an audio captured using a mobile device, these services identify the original audio track and provide metadata information, such as the album, options to buy the track, etc. Fingerprints have also been used to perform automatic tagging of audio collections and create automatic playlists based on user preferences [12].

The fingerprints utilized for identification should be robust, so that they are not altered by benign processing or minor distortion of the multimedia, but should be discriminative enough to distinguish between millions of different multimedia objects. In many practical applications, fingerprint extraction and matching are performed in real-time, which places constraints on the computation and memory requirements of the identification system. A significant amount of research effort has been focused on designing fingerprinting schemes with various robustness properties, that employ different kinds of features, and provide varying tradeoffs between robustness, discrimination, and computational requirements. These schemes are then evaluated using experiments on moderate sized databases.

While such experimental evaluations are important, they may not provide a complete picture of how the performance scales when the same algorithm is used to identify videos from a very large database with millions of hours of video. Also, the understanding obtained from evaluating a particular fingerprinting scheme may not help in predicting the performance of another scheme. To complement these experimental evaluations, there is a strong need for a systematic analysis of fingerprinting schemes that can help us obtain deeper insights and uncover connections to related areas. Analysis can help us answer such fundamental questions as - "What is the best possible identification accuracy achievable using *any* fingerprinting scheme?", "What properties should an optimal fingerprinting scheme have?", "How should the fingerprint be designed to resist attacks?", "What is the impact of using module X as opposed to module Y?", and others. Such an analysis can also guide the design of better fingerprinting schemes. A systematic study would also help identify weaknesses of fingerprinting systems that may be exploited by smart attackers to circumvent the system and allow suitable counter-attack strategies to be devised. In this dissertation, we describe a modular framework for analyzing fingerprinting systems that can help answer some of these questions.

1.2 Collusion-Resistant Fingerprints for Traitor Tracing

A complementary aspect of multimedia protection is to identify the user responsible for redistributing or pirating the content. This issue is not only important in the context of movie piracy, but becomes a critical necessity in many applications involving highly secure and classified documents. Consider, for example, a classified video that only a select group of users has access to. An untrustworthy user may leak this video to an outsider, or publish it via the internet. It then becomes necessary to identify the user responsible for this leak. The presence of such a tracing mechanism can also serve to deter and prevent users from redistributing classified information in the first place.

Embedded fingerprints have emerged as an important forensic tool to combat such illegal redistribution of protected content. The main idea behind fingerprinting is to embed a fingerprint signal in every legally distributed copy of the content that uniquely identifies the recipient. When an unauthorized copy is discovered, the embedded fingerprint can be extracted and used to identify the source of the leak. While such a system may be effective at identifying single adversaries, multiple malicious users may collaborate to launch powerful collusion attacks against the fingerprinting system [103]. By comparing their different versions, the colluders can attempt to identify the locations containing the fingerprint signal, remove the information from these locations and thereby create a copy that cannot be traced back to any of them.

Collusion-resistant fingerprints designed to withstand such attacks have been proposed in the literature [83, 98]. These designs provide good collusion resistance when embedded in uncompressed multimedia. In practical applications, multimedia is widely stored and transmitted in compressed format, and it is often necessary to embed fingerprints in compressed multimedia. Existing techniques, which are suitable for uncompressed multimedia, do not provide good resistance when adapted to compressed host signals, and novel designs that explicitly account for the compressed nature of the host are needed [86]. In this dissertation, we describe an *Anti-Collusion Dither* based technique that can significantly improve the collusion resistance of fingerprints embedded in compressed signals.

1.3 Organization of the Dissertation

As described in the previous sections, the first contribution of this dissertation is to develop models for and study the performance of content fingerprints. Chapter 2 describes the overall modeling framework for theoretical analysis of content fingerprints. Under this framework, modules that are typically utilized in fingerprinting algorithms are individually analyzed to understand their impact on the overall identification performance. This can also be thought of as a layered or hierarchical approach to modeling and designing content fingerprints, in terms of the underlying multimedia content, the features extracted from the multimedia, encoding the features to obtain compact fingerprints and the matching process. In the same chapter, some aspects of the relations between the features used in constructing the fingerprints and the final fingerprints obtained are examined. Specifically, the impact of distortion in the feature domain on the fingerprint bits is studied. An algorithm to optimize the design of the feature quantizer to improve the identification performance is described. Lastly, the interaction between the fingerprint designer and the adversary is studied under a game-theoretic framework and the optimal fingerprint bit distribution from the designer's perspective is determined.

Chapters 3-5 then focus on, and examine in detail, the fingerprint matching process, with progressively more complex models. In Chapter 3, a simple i.i.d. model is adopted for the fingerprint bits. Fingerprint matching is modeled as a hypothesis testing problem, and the best performance achievable using i.i.d. bits is determined. Bounds are also derived on the length of the fingerprint needed to achieve a desired performance. As practical fingerprints have correlated components, Chapter 4 describes a Markov Random Field based model that can capture these correlations. A statistical physics based approach is developed to estimate the identification accuracy under this model. Chapter 5 examines models to capture the temporal correlations among fingerprints that reflect the correlations of the underlying multimedia. An adaptive detector is then proposed that improves the matching accuracy.

In Chapters 6 and 7, the problem of designing collusion resistant fingerprints for compressed multimedia is studied. In Chapter 6, the performance of traditional fingerprints designed for uncompressed multimedia when applied to compressed multimedia is first examined. As the existing techniques result in low collusion resistance, an Anti-Collusion Dithering(ACD) technique is described to improve the overall collusion resistance, and the performance is studied via simulations and experiments. Chapter 7 then performs a theoretical analysis of the collusion-resistant fingerprints for compressed multimedia from different perspectives, and demonstrates that the proposed ACD technique is advantageous under each of these criteria.

The thesis concludes with Chapter 8, which summarizes the contributions of the dissertation and discusses future perspectives.

Chapter 2

Theoretical Modeling and Analysis of Content Fingerprinting

In recent years, user generated content (UGC) websites such as Youtube have grown in popularity and revolutionized multimedia consumption and distribution. Increasingly, the Internet is being seen as a medium for delivering multimedia content to consumers. These new distribution channels have made it easier to access multimedia in digital form and redistribute it via the internet. Concerns have been raised about the redistribution of copyrighted content, especially through UGC websites [81]. To identify and filter such copyrighted videos, several UGC websites are deploying content filtering schemes that rely on an emerging technology called content fingerprinting. A *content fingerprint*, is a compact identifier that represents robust and unique characteristics of the multimedia and can be used to identify the document. In this respects, it is similar in usage to a human fingerprint, and hence the name. The fingerprint of the uploaded video can be compared to a database of fingerprints of copyrighted content to identify whether it is copyrighted or not.

Watermarking, which is a proactive technique wherein a special watermark signal is embedded into the host at the time of content creation, can also be used for content identification. This embedded signal can later be extracted and used to identify the content and retrieve associated metadata [5]. Watermarking techniques are suitable if the embedder has control over the content creation stage. This requirement may be difficult to satisfy in many practical applications, including content filtering on UGC sites. In particular, a large volume of existing multimedia does not have embedded watermarks and cannot be identified using this approach. Content fingerprints, on the other hand, do not require access to the content at the time of creation and can be used to identify existing multimedia content that does not have embedded information.

Content fingerprints are designed to be robust to minor content preserving operations while being able to discriminate between different multimedia objects. At the same time, the fingerprints must be compact to allow for efficient matching. In this respect, content fingerprinting shares similarities with robust hashing [26, 78]. Traditionally, robust hashing was studied in the context of authentication, where the main objective was to prevent an adversary from forging a valid hash for a given image, and also prevent him/her from obtaining an image that has the same hash as the given image. In contrast, while collisions or false alarms are also a concern in content fingerprinting, the main threat model is an adversary making minor modifications to a given multimedia document that would result in a significantly different fingerprint and prevent identification. Another difference between fingerprinting and robust hashing is that fingerprinting applications typically involve large databases with millions of hours of video and audio, whereas traditional applications of image hashing typically focus on authenticating a smaller set of images. However, many hashing schemes with good robustness properties can be adapted for content identification purposes and hence the terms "content fingerprinting" and "robust hashing" are often used interchangeably in the literature.

Content fingerprinting has received a lot of interest from the research community and different approaches for fingerprinting have been proposed, some of which are reviewed in Section 2.1. Most of these works address the problem of designing fingerprinting schemes that are robust to different kinds of processing and achieve various tradeoffs between robustness, discrimination, and computational complexity. Typically, these algorithms are designed based on heuristics and are evaluated through experiments on moderately large databases. Some studies have also focused on the modeling and analysis of certain modules employed for designing fingerprints, but no overarching framework has been developed.

There is a strong need for theoretical modeling and analysis of the fingerprinting problem that can provide deeper understanding and insight. For example, in the field of data-hiding, the modeling of watermarking as communications with side information [18] led to the development of schemes inspired by communication approaches with provable optimality properties [11]. A similar framework for content fingerprints, that uncovers connections with other areas of research, can guide the design of better fingerprinting algorithms. In this dissertation, we develop a framework that can provide a basis for further study and analysis, and answer some of the fundamental questions regarding content fingerprints.

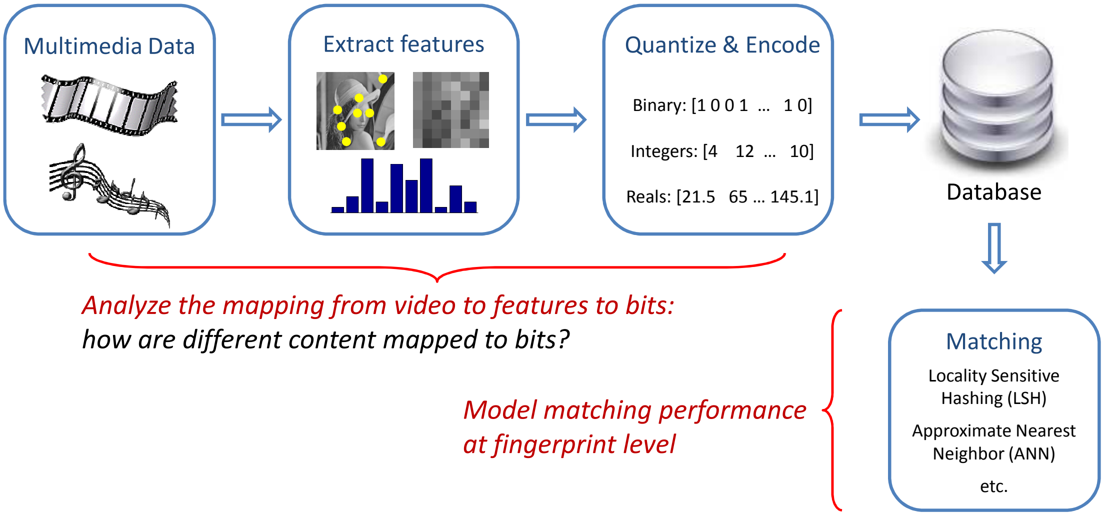
2.1 Prior Work on Content Fingerprinting

Content fingerprinting has attracted a lot of research and several audio and video fingerprinting techniques have been proposed in the literature. A robust fingerprinting technique for audio identification based on the signs of the differences between the energy in different frequency bands of overlapping frames was proposed in [30]. A similar approach for video, coupled with efficient indexing strategies was proposed in [64]. Ranks of the block average luminance of sub-sampled frames were used as fingerprints in [60], while signs of significant wavelet coefficients of spectrograms were used to construct fingerprints in [3]. Moment invariants that capture appearance and motion were proposed as features for fingerprints in [68].

In the robust hashing literature, hash generation by quantizing projections of images onto smooth random patterns was proposed in [26], which is used as a building block in many fingerprint constructions such as [68]. Hashes resilient to geometric transforms based on properties of Fourier transform coefficients were proposed in [78]. Spatiotemporal video hashes based on 3-D transforms were proposed in [15]. Several other hashing schemes with different robustness properties have been proposed in the literature. A comparative study of a few representative algorithms was performed in [46]. A survey of various fingerprinting and hashing algorithms proposed in the literature may be found in [53].

Regarding theoretical aspects of fingerprinting, qualitative guidelines for designing multimedia hash functions were provided in [56], with a focus on bit assignment and the use of suitable error-correcting codes to improve the robustness. Robust hashing was considered as a classification problem in [95]. As a null-hypothesis and false alarms were not explicitly considered in the formulation of [95], the analysis cannot be directly applied to the problem of content identification. In the related field of biometrics, the capacity of biometrics-based identification was studied in [101]. Capacity was defined as the maximum rate R such that 2^{LR} distinct biometrics could be identified with an asymptotic error probability of zero, as the length of the fingerprints $L \to \infty$. However, as noted in [101], while designing practical systems, we are more interested in determining the best performance obtainable using a given length of the fingerprint, which is one of the contributions of our study. Subsequent to the results described in Chapter 3, which were summarized in [88], a similar analysis applicable to fingerprints over general alphabets was described in [62].

Most of these prior works focused on the design and analysis of particular modules used in fingerprint and robust hash designs. In this dissertation, we describe a holistic framework for modeling and analysis of fingerprints, and study different aspects of the fingerprint problem.



Multimedia Data
Extract features
Quantize & Encode
Binary: [1 0 0 1 ... 1 0]
Integers: [4 12 ... 10]
Reals: [21.5 65 ... 145.1]
Database
Analyze the mapping from video to features to bits:
how are different content mapped to bits?
Model matching performance
at fingerprint level
Matching
Locality Sensitive Hashing (LSH)
Approximate Nearest Neighbor (ANN)
etc.

Figure 2.1: Framework for modeling content fingerprints.

2.2 Framework for Modeling Content Fingerprints

In this section, we describe the overall framework for analyzing various content fingerprinting schemes [84]. Practical algorithms for fingerprinting proposed in the literature may employ different building blocks, but they typically follow the general framework shown in Figure 2.1. Given the multimedia data, features that capture distinctive properties of the multimedia are extracted. These features should be robust to distortions of the underlying signal, so that they can be reliably extracted from a distorted version of the original content, while being distinctive enough to distinguish diverse multimedia objects. Robust properties of the signal in various domains, such as spatial, temporal or transform domain, may be used as features [53]. In many practical applications with stringent requirements on the computational complexity, simpler, easy-to-compute features may be preferred [53]. Some commonly used image/video features, such as interest point based features, block average luminance, and color histograms are illustrated in the second block of Figure 2.1.

The extracted features are typically quantized and compactly encoded as bits or integers to obtain the fingerprint, as illustrated in the third block of Figure 2.1. The computed fingerprints are then stored in the reference database for later use. Given a multimedia object that needs to be identified, features are extracted and encoded to form the query fingerprint, which is then matched with the fingerprints in the reference database to identify the multimedia. For the matching process, exhaustively comparing the query with every fingerprint in the database is optimal, but may incur high computational cost. Instead, efficient approximate matching schemes, such as Locality Sensitive Hashing (LSH) and k-D trees may be used in practice.

The accuracy of a given fingerprinting algorithm for different genres of multimedia is influenced by each module. To understand the overall matching accuracy, it is necessary to understand the contribution of each of these individual modules to the performance. In particular, through our analysis, we wish to understand how different multimedia are mapped into features, how distortion of the multimedia changes the feature values extracted, how these changes in the features affect the final fingerprint, and how these distortions of the fingerprint in turn impact the matching and the overall identification accuracy.

In the subsequent sections and chapters, we describe the analysis of some modules used in fingerprinting algorithms under this framework. We first examine different aspects of the mapping of the features to fingerprint bits in the remainder of this chapter. In Section 2.3 we examine the minimum amount of distortion that an attacker needs to introduce into the features to change a certain number of fingerprint bits, and in Section 2.4, we examine how the quantizer should be designed to improve the overall identification accuracy. Section 2.5 sheds light on the optimal choice of the fingerprint distribution to resist attacks. Chapters 3-5 are devoted to analyzing the matching performance at the binary fingerprint stage.

2.3 Distortion in Features Reflected in the Fingerprints

In this section, we study how distortion in the features translate into changes in the fingerprint bits. First, we study the probability of a fingerprint bit changing when the features are distorted, and examine the influence of the variance of the distortion and the correlation between the features and distortion on this probability. We then determine the minimum amount of distortion required to cause a unit change in the fingerprint, and the influence of the fingerprint distribution on this quantity.

2.3.1 Problem Setup

In many fingerprint constructions, the final binary fingerprint is obtained by comparing each feature component to a threshold. For example, in [30], the signs of differences between the average energy of adjacent frequency bands across frames are used as fingerprints. Similarly, in schemes employing random projection as the final step, such as [26, 68], the projection of the feature on a random vector is compared to a threshold. We adopt a similar model for our analysis of the fingerprints, where a set of features are extracted from a multimedia document such as image/video and are then quantized to 1-bit accuracy by comparing with a threshold. The feature used could be average block luminance, differences between average block luminance of adjacent blocks, or functions of transform coefficients.

Suppose that L features $X_1, X_2, \ldots, X_L$ are extracted from a given multimedia. For simplicity, we assume that the features can be modeled as i.i.d. random variables with a common p.d.f. f and corresponding c.d.f. F. Let the ith bit of the fingerprint $B_i \in \{0, 1\}$ be obtained by comparing the ith feature value to a threshold τ, $B_i = Q(X_i) = U(X_i - \tau)$, where $U(\cdot)$ is the unit step function. Typically, the threshold is set to be equal to the median value of the distribution f, so that the resulting bits are equally likely to take the values 0 and 1 [26]. Henceforth, without loss of generality, we assume that the features have been centered so that the median is equal to 0 and that the threshold is chosen to be equal to the median, so that $\tau = 0$.

2.3.2 Probability of Flipping a Fingerprint Bit

In this subsection, we examine how distortions of the multimedia are reflected as changes in the binary fingerprint in terms of the probability of a bit being flipped. Due to distortions of the underlying signal, the features extracted from a query may be different from the original features $\{X_i\}$. Denote the features obtained from the distorted content by $X_i + Z_i$, where $\{Z_i\}$ is the distortion in the features. In general, the distortion Z_i may be correlated with the original feature X_i. We assume that the noise in each feature Z_i is independent of the remaining $\{X_j\}_{j \neq i}$ and $\{Z_j\}_{j \neq i}$. The probability of a fingerprint bit flipping may be written as:

$$p = \Pr(B_i' \neq B_i) = \Pr(X_i > 0, X_i + Z_i < 0) + \Pr(X_i < 0, X_i + Z_i > 0), \qquad (2.1)$$

where $B_i' = Q(X_i + Z_i)$ is the ith fingerprint bit obtained from the distorted content. For many distributions of interest, the probability in Eqn. (2.1) can be evaluated numerically. As an example, we consider the case of Gaussian distributed features and distortion. The inferences obtained from this example would apply similarly to the case of other distributions.

Suppose that $[X_i \; Z_i]^T$ is jointly Gaussian with mean μ and covariance $\mathbf{\Sigma}$:

$$\mu = \begin{bmatrix} 0 \\ \mu_n \end{bmatrix}, \quad \mathbf{\Sigma} = \begin{bmatrix} 1 & \rho\sigma_z \\ \rho\sigma_z & \sigma_z^2 \end{bmatrix},$$

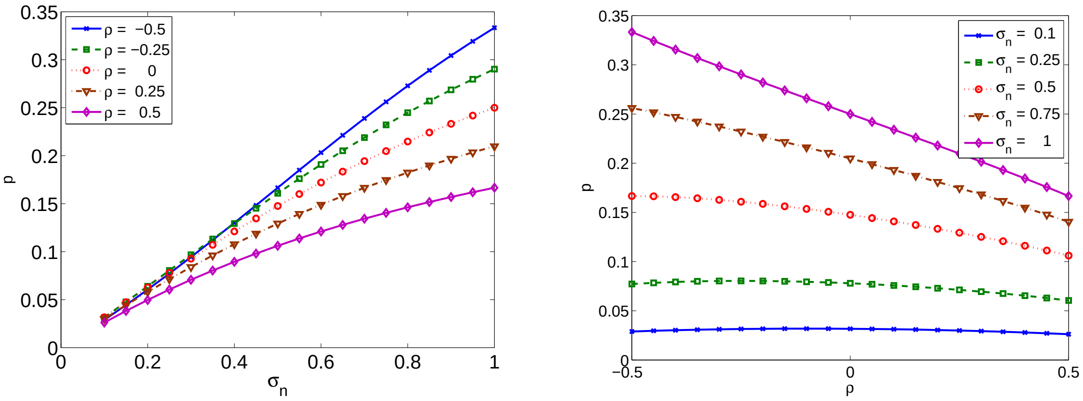
where for convenience we have normalized the mean and variance of the feature X_i to 0 and 1, respectively. The probability p can then be expressed as:

$$\begin{aligned} p &= \int_{x>0} f_X(x) \Pr(Z_i < -X_i | X = x) \, \mathrm{d}x + \int_{x<0} f_X(x) \Pr(Z_i > -X_i | X = x) \, \mathrm{d}x \\ &= \int_{x>0} f_X(x) \Phi(-x; \mu_{z|x}, \sigma_{z|x}^2) \, \mathrm{d}x + \int_{x<0} f_X(x)(1 - \Phi(-x; \mu_{z|x}, \sigma_{z|x}^2)) \, \mathrm{d}x, \end{aligned}$$

where $\Phi(x; \mu, \sigma^2)$ is the c.d.f. of a Gaussian distribution with mean μ and variance σ^2, and the mean and variance of the conditional distribution of $Z_i | X = x$ are given by:

$$\mu_{z|x} = \mu_z + \rho\sigma_z x, \quad \sigma_{z|x}^2 = \sigma_z^2(1 - \rho^2).$$

The probability p can then be evaluated numerically. Figure 2.2 examines the influence of the noise power σ_n^2 and the correlation ρ on the probability of a bit flipping,



(a)

(b)

Figure 2.2: Relation between the distortion in features and the probability of a fingerprint bit changing.

when $\mu_z = 0$. From Figure 2.2(a), we observe that for a fixed correlation, as the distortion power increases, the probability of a bit flipping increases, as expected. Figure 2.2(b) indicates that a negatively correlated distortion has a higher probability of altering a fingerprint bit.

2.3.3 Minimum Distortion Required to Alter Fingerprint Bits

The probability of a bit flipping for a given type of distortion as evaluated above is useful for modeling benign processing and gives a sense of the *average* distortion needed to change a certain fraction of the fingerprint bits. However, a malicious adversary seeking to evade detection could distort the features in a smart manner to cause large changes in the fingerprint while minimizing the amount of distortion introduced. The minimum amount of distortion that needs to be introduced to change fingerprint bits is thus an indicator of the robustness of the feature against malicious adversaries. We now derive an analytical expression for this metric.

The fingerprint bit obtained by quantizing a given feature X_i can be altered by adding a value $-X_i$ to it, so that the amount of squared distortion introduced is $|X_i|^2$. Given L features, m bits of the fingerprint can be changed with minimum distortion by altering the m features with the smallest absolute values. Let $Y_i = |X_i|$ and $Y_{(1)}, Y_{(2)}, \ldots, Y_{(L)}$ denote the order statistics of $\{Y_i\}$, where $Y_{(1)}$ corresponds to the minimum value and $Y_{(L)}$ corresponds to the maximum value. To change m fingerprint bits, it is sufficient to modify the features corresponding to $Y_{(i)}, i = 1, 2, \ldots, m$ and change their sign. The mean squared distortion $D'_L(m)$ thus introduced in the features is:

$$D'_L(m) = \frac{1}{L} \sum_{i=1}^{m} Y_{(i)}^2$$

Let $p = \frac{m}{L}$ denote the fraction of fingerprint bits changed, and $D_L(p)$ denote the minimum expected mean squared distortion needed to change a fraction p of the bits:

$$D_L(p) = \mathbb{E}[D'_L(\lceil Lp \rceil)] \tag{2.2}$$

$D_L(p)$ represents the best tradeoff, from an attacker's perspective, between the amount of distortion introduced and the changes in the fingerprint. In this sense, $D_L(p)$ is similar to the rate-distortion function in information theory [16].

2.3.4 Evaluation of the Distortion Function

The distortion function $D_L(p)$ defined in Eqn. (2.2) can be expressed as

$$\begin{aligned} D_L(p) &= \frac{1}{L}\mathbb{E}\left[\sum_{i=1}^{\lceil Lp \rceil} Y_{(i)}^2\right] \\ &= \frac{1}{L}\sum_{i=1}^{\lceil Lp \rceil} \mathbb{E}[Y_{(i)}^2] \\ &= \frac{1}{L}\sum_{i=1}^{\lceil Lp \rceil} \left(\mathbb{E}[Y_{(i)}]^2 + \mathrm{Var}(Y_{(i)})\right), \end{aligned}$$

where $\mathbb{E}[Y_{(i)}]$ and $\mathrm{Var}(Y_{(i)})$ represent the mean and variance of the order statistics. It can be shown that for large L, the order statistics converge in distribution to a Gaussian random variable:

$$X_{(\lceil Lp \rceil)} \xrightarrow{D} \mathcal{N}\left(F^{-1}(p), \frac{p(1-p)}{L[f(F^{-1}(p))]^2}\right),$$

where, $\xrightarrow{D}$ denotes convergence in distribution and $F^{-1}(\cdot)$ denotes the inverse c.d.f. [20]. The convergence to the Gaussian distribution, combined with the fact that the variance reduces to 0 as L increases, implies that the $\lceil Lp \rceil$th order statistic converges in probability to the mean, $X_{(\lceil Lp \rceil)} \xrightarrow{P} F^{-1}(p)$.

Using the above results for the mean and variance of order statistics, we have:

$$\begin{aligned} D_L(p) &\approx \frac{1}{L}\sum_{i=1}^{\lceil Lp \rceil}\left(\left[F_Y^{-1}\left(\frac{i}{L}\right)\right]^2 + \frac{\frac{i}{L}\left(1-\frac{i}{L}\right)}{L[f(F_Y^{-1}\left(\frac{i}{L}\right))]^2}\right) \\ &\longrightarrow \frac{1}{L}\sum_{i=1}^{\lceil Lp \rceil}\left(\left[F_Y^{-1}\left(\frac{i}{L}\right)\right]^2\right), \end{aligned}$$

as the variance term decays to zero for large L. Define $m_i = \frac{i}{L}$ and $\Delta m = m_i - m_{i-1} = \frac{1}{L}$. Using these variables, the expression for the distortion function can be written as

$$\begin{aligned} D_L(p) &= \sum_{m_i=\frac{1}{L}}^{p} \left[F_Y^{-1}(m_i)\right]^2 \Delta m \\ &\longrightarrow \int_0^p \left[F_Y^{-1}(m)\right]^2 \, \mathrm{d}m = D(p), \end{aligned} \tag{2.3}$$

for large L, where $F_Y^{-1}(\cdot)$ corresponds to the inverse c.d.f. of the absolute value of the features. Using the above expression, the distortion function can be evaluated for any distribution for which the inverse c.d.f. can be computed efficiently. Further, we observe that the the number of features L does not appear in the above expression. This implies that the minimum expected mean squared distortion needed to alter a fixed fraction of the fingerprint bits is independent of the number of bits.

2.3.5 Numerical Results

To verify our theoretical derivations, we perform experiments with synthetic data. We generate random variables with a specified distribution, alter the signs of Lp features with the minimum absolute values, and compute the average distortion introduced to estimate $D(p)$.

Figure 2.3 shows the distortion function for the case when the features $\{X_i\}$ extracted from the image/video are distributed as standard Gaussian random variables. From the figure, we observe that for $L \geq 64$, the distortion function is independent of the number of features. The difference between the distortion function for $L = 16$ and for larger L is small, especially in the region of interest, where $p < 0.5$. Further, to completely change the fingerprint, the mean squared distortion introduced is equal to the variance of the features. This is due to the fact

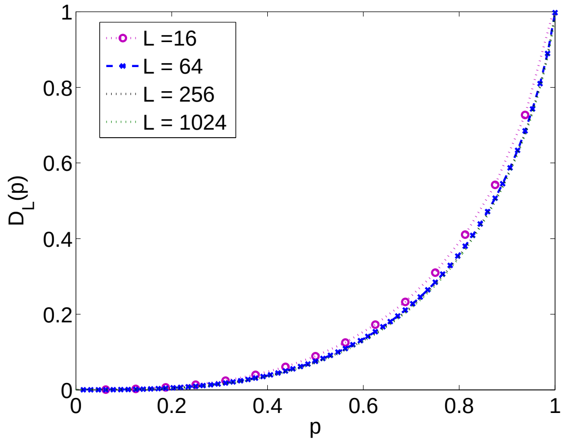


Figure 2.3: Influence of the number of features N on the distortion function.

that to change every single fingerprint bit, the distortion needed will be equal to the negative of the features. As a result, the average distortion introduced will be equal to the variance of the feature. We next compare the simulation results with the analytical expression in Eqn. (2.3), which is evaluated by numerical integration. Figure 2.4 shows that the theoretical expression and the simulation results agree very well.

2.3.6 Influence of Different Feature Distributions

We next consider the effect of the distribution of the features on the distortion function. As indicated by Eqn. (2.3), the inverse c.d.f. of the absolute value of the feature determines the distortion function. We consider three commonly encountered distributions for image features - uniform, Laplacian, and Gaussian. To enable a fair comparison, we normalize the variance of the distributions.

The uniform distribution on $[-\frac{\Delta}{2}, \frac{\Delta}{2}]$ with $\Delta = \sqrt{12}$ has unit variance. The

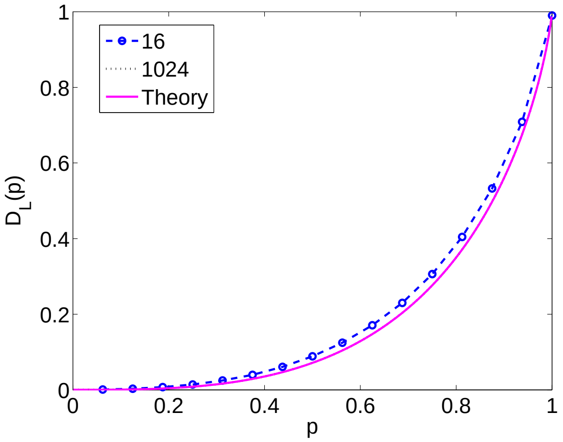


Figure 2.4: Comparison of the distortion function obtained via theory and simulation.

c.d.f. of $Y_i = |X_i|$ is then given by $F_Y^{(U)}(y) = \frac{2y}{\Delta}, 0 \leq y \leq \frac{\Delta}{2}$, and the inverse c.d.f. $F_Y^{-1\,(U)}(y) = \frac{\Delta y}{2}, 0 \leq y \leq 1$. The distortion function for the uniform distribution can then be derived as:

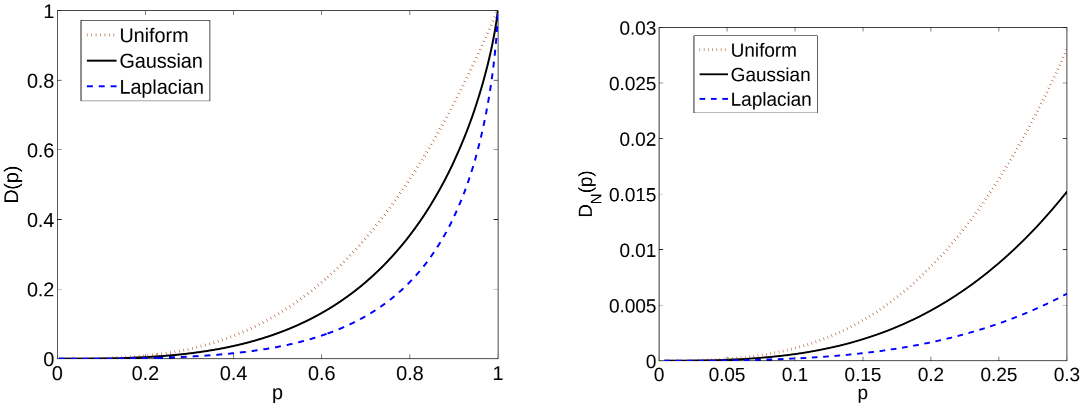
$$D^{(U)}(p) = \frac{\Delta^2 p^3}{12} = p^3$$

as the variance $\frac{\Delta^2}{12} = 1$.

For the Laplacian distribution defined as $f_L(x) = \frac{1}{2b}\exp(-\frac{|x|}{b})$, the variance is given by $\sigma_L^2 = 2b^2$, so that for unit variance, $b = \sqrt{\frac{1}{2}}$. The c.d.f. of the corresponding $Y_i = |X_i|$ is then given by $F_Y^{(L)}(y) = 1 - \exp(-\frac{y}{b}), y \geq 0$ and the corresponding inverse c.d.f. is $F_Y^{-1\,(L)}(y) = -b\ln(1-y), 0 \leq y \leq 1$. Integrating the square of the inverse c.d.f. to obtain the distortion function gives

$$D^{(L)}(p) = 0.5[2p - (1-p)((\ln(1-p))^2 - 2\ln(p))]$$

where we have used the fact that $2b^2 = 1$. For the Gaussian distribution, no closed form expression exists for the inverse c.d.f, and we use numerical techniques to



(a)

(b)

Figure 2.5: Influence of the feature distribution on the distortion function. (b) displays an enlarged portion of the distortion function to highlight the region $0 < p < 0.5$.

compute the distortion function.

Figure 2.5(a) shows the distortion function for random variables with the uniform, Gaussian, and Laplacian densities and Figure 2.5(b) shows an enlarged portion of the distortion function in the region of practical interest. From these figures, we observe that the uniform distribution requires the highest distortion to change the fingerprint by a fixed fraction, followed by the Gaussian distribution. The Laplacian distribution requires the least amount of distortion to be introduced. These results can be explained by examining the inverse c.d.f. of the absolute value $F_Y^{-1}(y)$, shown in Figure 2.6. We observe that for small y, the inverse c.d.f. corresponding to uniform distributed random variables has the highest value compared to the Gaussian and Laplacian distributions, which results in a corresponding highest value for the distortion function. Intuitively, since the Gaussian and Laplacian distributions are

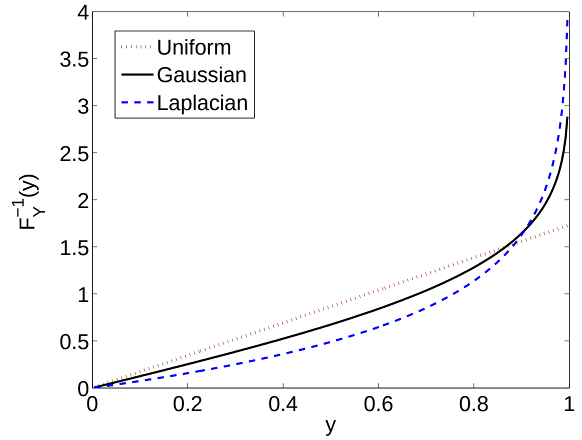


Figure 2.6: Comparison of the inverse c.d.f. of the absolute value for different distributions.

more likely to produce values closer to 0, the average amount of distortion needed to change these values is lower, compared to the uniform distribution. Thus, a distribution that is more spread out, or has lesser probability mass close to 0, would give a better performance from this viewpoint. In terms of the kurtosis, which is a measure of the "peakedness" of a distribution, having a lower value for the kurtosis implies that the distribution is more robust under this metric. If a designer has an option to choose between features with different distributions, then this metric could be used as a guideline for choosing a robust feature.

2.4 Optimal Quantizer Design for Content Identification

As described in Section 2.2, features extracted from multimedia are often quantized and encoded to obtain compact fingerprints, which require lesser storage. The quantization can also serve to improve the robustness against noise, but should be designed carefully to preserve the discriminative ability of the features. 1-bit quantizers are commonly used in fingerprinting applications [30]. Another common quantization and encoding technique is ordinal ranking [60], which has been studied and analyzed in [13, 14].

The problem of designing quantizers to optimize various criteria has been studied in the literature in different contexts. Quantizing and reconstructing a signal to achieve the minimum mean squared reconstruction error was studied in [51]. The optimal quantizer for this problem is the well known Lloyd-Max quantizer [36, 51]. Vector quantization techniques with applications in signal compression have also been studied in the literature [28]. Quantizer design for encoding a signal in noise to achieve minimum reconstruction error with respect to the noiseless source was studied in [2]. Quantizer design in a joint source channel coding framework was considered in [24]. In this setting, a source signal is quantized, and the binary codeword representing the quantized signal is transmitted over a binary symmetric channel. The decoder uses the channel output to reconstruct the source signal. The goal is to design a quantizer that minimizes the mean squared error between the source and the reconstructed signal.

As we will show below, quantizer design for content identification shares some similarities with the joint source channel coding problem considered in [24]. However, the main difference is that in [24] the binary encoded signal is transmitted over the noisy channel, whereas in the fingerprinting problem, the original signal is itself transmitted over a channel, which induces a noisy channel between the corresponding quantized versions or fingerprints.

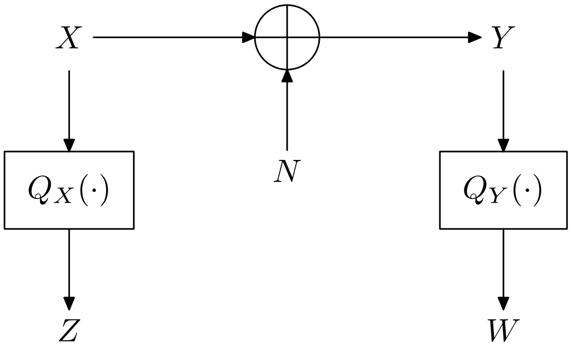


Figure 2.7: Quantizer design for content identification.

2.4.1 Problem Description

The problem setup for quantizer design in identification applications is shown in Figure 2.7. A source X is quantized using a scalar quantizer $Q_X(\cdot)$, to obtain $Z = Q_X(X)$. The random variable X may correspond to features derived from images or video in content identification applications or may correspond to features derived from biometric data in biometric identification applications. The quantized value Z which corresponds to the fingerprint is then stored in the database. In the identification stage, a noisy version of X denoted by Y is observed. To compute the fingerprint W, the feature Y is quantized using a quantizer $Q_Y(\cdot)$ so that $W = Q_Y(Y)$. The quantizers Q_X and Q_Y have L_X and L_Y reconstruction values respectively. The objective is to choose the quantization thresholds of Q_X and Q_Y so as to achieve the best identification performance.

The performance of an identification system is typically measured in terms of the probabilities of false alarm and correct identification. However, closed form expressions for these quantities in terms of the distributions of X and Y cannot be easily obtained. Instead, we use the identification capacity C_{id} [101] as a measure of the identification performance. The identification capacity for the system described in Figure 2.7 is given by $C_{id} = I(Z;W)$, where $I(Z;W)$ is the mutual information between the random variables Z and W [101]. The identification capacity is an important parameter, as the maximum number of distinct fingerprints that can be identified $N_{\max}$ is related to the capacity C_{id} as $N_{\max} \approx 2^{LC_{id}}$, where L is the length of the fingerprint.

Let $-\infty = t_0 < t_1 < \ldots < t_{L_X} = \infty$ be the quantization thresholds of Q_X and $-\infty = t'_0 < t'_1 < \ldots < t'_{L_Y} = \infty$ be the thresholds of Q_Y. The problem can now be stated as: Given the distribution of $X \sim p_X(x)$ and the conditional distribution of $Y|X \sim p_{Y|X}(y|x)$, choose the values $\{t_i\}_{i=1}^{L_X-1}$ and $\{t'_j\}_{j=1}^{L_Y-1}$ so as to maximize the mutual information $I(Z;W)$. Note that as the mutual information depends on the joint probability distribution of Z and W and not on the actual values of the random variables, the reconstruction points of the quantizer need not be considered in the optimization problem.

2.4.2 Necessary Condition for Optimality

In this section, we derive a necessary condition for optimality by setting the first derivative of the objective function to zero. For convenience, denote the reconstruction points of Q_X and Q_Y by $\{r_i\}_{i=1}^{L_X}$ and $\{r'_j\}_{j=1}^{L_Y}$, respectively, so that

$$Q_X(x) = r_i \quad \text{if } x \in (t_{i-1}, t_i].$$

Q_Y can be defined in a similar way. The probability mass function (p.m.f.) of Z is then given as

$$p_Z(r_i) = \int_{t_{i-1}}^{t_i} p_X(x)\ \mathrm{d}x$$

and the joint p.m.f. of Z and W is given by

$$p_{ZW}(r_i, r'_j) = \int_{t_{i-1}}^{t_i} p_X(x) \int_{t'_{j-1}}^{t'_j} p_{Y|X}(y|x)\ \mathrm{d}y\ \mathrm{d}x.$$

The mutual information $I(Z;W)$ can then be computed as

$$I(Z;W) = \sum_{r_i, r'_j} p_{ZW}(r_i, r'_j) \log \frac{p_{ZW}(r_i, r'_j)}{p_Z(r_i)\, p_W(r'_j)}.$$

At the optimal solution, the derivative of the objective function should be zero. Setting $\frac{\partial}{\partial t_i} I(Z;W) = 0, \forall i = 1, 2, \ldots, L_X - 1$ implies that:

$$\log \frac{p_Z(r_i)}{p_Z(r_{i-1})} - \sum_{j=1}^{M} p(W = r'_j | X = t_i) \log \frac{p_{ZW}(r_i, r'_j)}{p_{ZW}(r_{i-1}, r'_j)} = 0 \tag{2.4}$$

$\forall i = 1, 2, \ldots, L_X - 1$. Similarly, setting $\frac{\partial}{\partial t_i} I(Z;W) = 0, \forall j = 1, 2, \ldots, L_Y - 1$, we have

$$\log \frac{p_W(r'_j)}{p_W(r'_{j-1})} - \sum_{i=1}^{L} p(Z = r_i | Y = t'_j) \log \frac{p_{ZW}(r_i, r'_j)}{p_{ZW}(r_i, r'_{j-1})} = 0, \tag{2.5}$$

where we have assumed that $p_X(t_i) \neq 0$ and $p_Y(t'_j) \neq 0$, Eqns. (2.4) and (2.5) give $L_X + L_Y - 2$ simultaneous nonlinear equations in $L_X + L_Y - 2$ variables. Unfortunately, these equations cannot be solved analytically to obtain closed form expressions for the optimal solution. As an alternative, in the next section, we propose an iterative algorithm for determining the quantizer thresholds.

2.4.3 Iterative Algorithm

In many quantizer design problems, it is often not possible to analytically solve for the optimum thresholds and reconstruction values, and iterative algorithms are used instead. The Lloyd-Max algorithm for designing quantizers with minimum mean-square reconstruction error [51] is a classical example of such an iterative algorithm. This algorithm alternates between optimizing the quantizer thresholds and the reconstruction points at each iteration. We adopt a similar approach to iteratively optimize the quantizer thresholds.

The thresholds $\{t_i\}$ and $\{t'_j\}$ are randomly initialized. At each step of the iteration, the algorithm alternates between estimating the optimal values for $\{t_i\}$ and $\{t'_j\}$ assuming that the other set of values is fixed. For example, first, the thresholds $\{t_i\}$ for the Q_X quantizer are fixed and the values of $\{t'_j\}$ that maximize the objective function $I(Z;W)$ are found numerically. Subsequently, these $\{t'_j\}$ values are kept constant and the optimal values of $\{t_i\}$ are determined. This process is repeated until the value of the objective function does not change significantly with each iteration. As the value of the objective function is non-decreasing in each step of the iteration, the algorithm is guaranteed to converge to a local minimum. Convergence to the global minimum is typically not guaranteed in many quantizer design problems such as [51], due to the multi-dimensional optimization and the non-convex objective function which may have several local minima.

2.4.4 Results

In this subsection, we present experimental results using i.i.d. Gaussian signals as an example. The original feature components are assumed to be i.i.d. $\mathcal{N}(0,1)$, and the noise components are assumed to have components that are i.i.d. $\mathcal{N}(0,\sigma_n^2)$. The iterative algorithm described in the previous subsection is used to design the quantizers to maximize the mutual information between the original and noisy fingerprints. As a baseline for comparison, the Lloyd-Max quantizer which is the optimal MMSE quantizer is used to obtain the quantization thresholds for $Q_X(\cdot)$ and $Q_Y(\cdot)$.

The number of quantizer levels is set to 4 in both cases, $L_X = L_Y = 4$, so that three thresholds need to be determined for each of the quantizers. As the distribution is symmetric about zero, the second threshold obtained by both algorithms turns out to be zero, so that the quantizers are mid-rise. Further, the first and third thresholds are negatives of each other, implying that the quantizers can be characterized by one parameter, say, the value of the third threshold. Figure 2.8 compares the thresholds determined by the Lloyd-Max algorithm and the proposed method for Q_X and Q_Y. We observe that when the value for the noise is small, each algorithm chooses similar values for the threshold of Q_X and Q_Y. This is due to the fact that when the noise is small, the distribution of the noisy features Y is similar to the distribution of the original features X. We also observe that as the noise power increases, the thresholds chosen by both algorithms for Q_Y becomes similar, as the noise distribution tends to dominate. For intermediate values, we observe that the proposed algorithm chooses

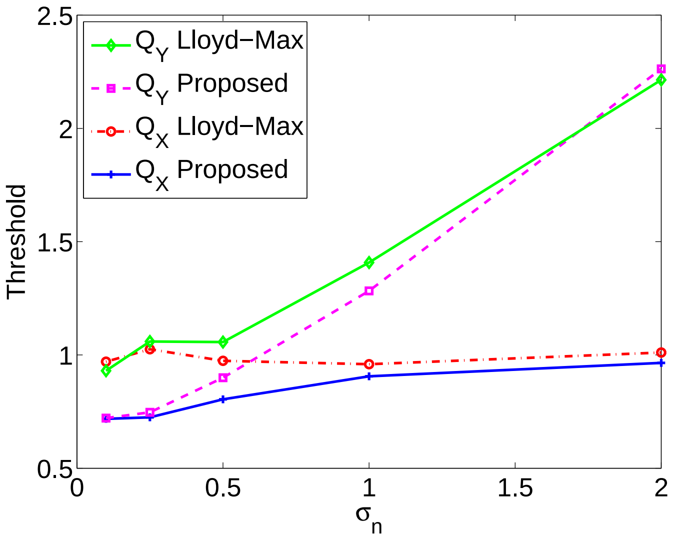


Figure 2.8: Comparison of the quantization thresholds obtained by the proposed method and the Lloyd Max algorithm.

thresholds that are somewhat smaller than the corresponding ones for the Lloyd-Max quantizer.

Figure 2.9 compares the identification accuracy in terms of the receiver operating characteristic (ROC) curves, where we plot the probability of correctly identifying a query P_c as a function of the probability of false alarm P_f on a semi-log scale, when the noise standard deviation $\sigma_n = 0.5$. We also compare the accuracy achievable using two different detectors. The optimal Maximum Likelihood (ML) detector utilizes the knowledge of the joint p.m.f. of the quantized original and noisy features to compute the likelihood and compares it to a threshold. In practical applications, this joint p.m.f. may not be available and it may be simpler to use the minimum distance detector instead. The minimum distance detector finds the fingerprint in the database that is closest to the query and compares this distance to

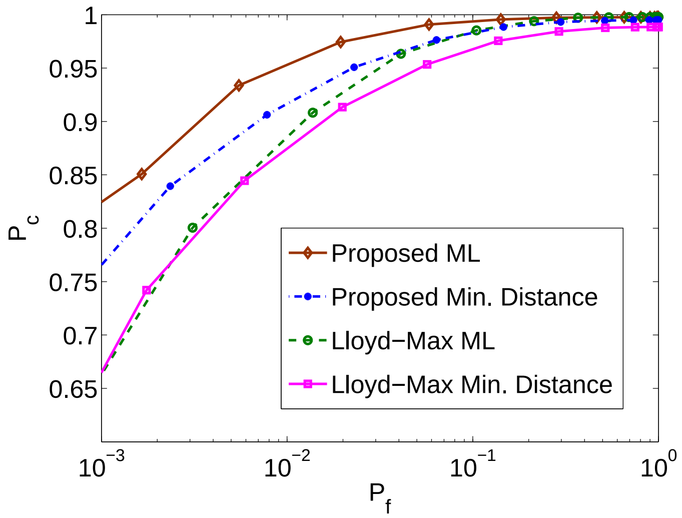


Figure 2.9: Comparison of the identification accuracy using the ML and minimum distance detector using features quantized by the proposed and Lloyd-Max algorithms.

a threshold. In our simulations, we use the L_1 distance as the distance metric. From the figure, we observe that using the ML detector in conjunction with the quantizer from the proposed algorithm gives the best performance. When the minimum distance detector is used instead, the P_c reduces, but is still better than the accuracy achievable using the Lloyd-Max quantizer. We observe that features quantized using the proposed algorithm yield consistently higher performance, especially at low P_f. For example, when $P_f = 10^{-3}$, fingerprints obtained using the proposed algorithm yield $10 - 15\%$ higher P_c compared to the Lloyd-Max quantizer.

As the proposed quantizer design algorithm utilizes the knowledge of the noise distribution, it is interesting to examine how changes in the noise parameters would impact the performance. To understand the generalization capability of the quan-

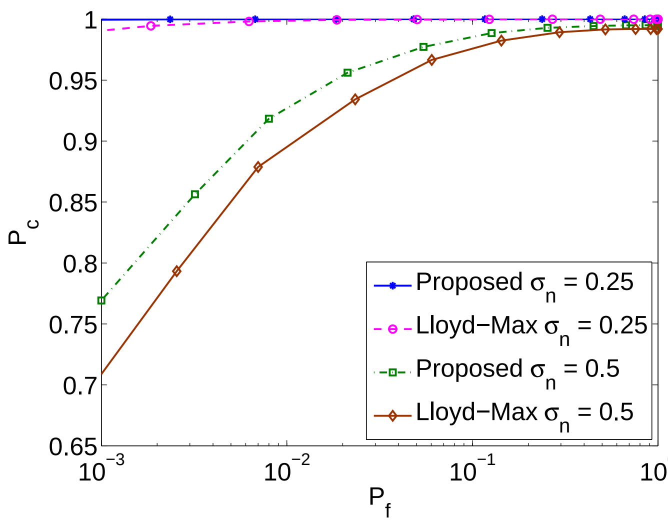


Figure 2.10: Generalization capabilities of the quantizers.

tizer, we perform the following experiment. The quantizers are designed assuming that the noise standard deviation is $\sigma_n = 0.25$ and the resulting quantizers are used to encode the features and obtain the fingerprints. Subsequently, noise with standard deviation $\sigma_n = 0.25$ and 0.5 is added to the features and these noisy features are quantized to obtain the distorted fingerprints. The identification accuracy when the distorted fingerprints are used as queries is shown in Figure 2.10. We observe that for features quantized using both the quantizers, the performance reduces when $\sigma_n = 0.5$ compared to when $\sigma_n = 0.25$ due to the stronger noise. However, the fingerprints obtained using the proposed method lead to a higher accuracy. For example, the proposed method results in a 5% higher P_c at $P_f = 10^{-3}$ compared to the Lloyd-Max quantizer.

2.5 Game-Theoretic Analysis of Content Fingerprinting

In Section 2.3, we examined the relation between the feature distribution and the amount of distortion needed to change a given fraction of the fingerprint bits. In this section, we study the optimal choice for the distribution of the fingerprint bits from the system designer's perspective, in the presence of such distortion that may be introduced by adversaries.

In the content identification problem, the system designer and the adversary modifying the content have conflicting objectives. The adversary's goal is to upload content and avoid detection, while the designer's goal is to identify modified content and minimize the probability of misclassification and false alarm. These conflicting objectives can be modeled under the framework of game theory [65]. In this section, we model the dynamics between the designer and the adversary by a two-player game between the adversary $\mathcal{A}$ and the system designer $\mathcal{D}$. In this game, the designer $\mathcal{D}$ designs the fingerprinting scheme and the adversary chooses the attack, to maximize their respective payoff functions. We illustrate this model using the example of binary fingerprints, which are commonly used for content identification [3, 15, 30]. We focus on the design and possible attacks on the fingerprints at the binary level. To specify the game, we describe below the strategy spaces and payoff functions for the two players.

2.5.1 Strategy Space

In the content identification game using binary fingerprinting, the strategy space of the designer consists of possible choices for the distribution of the fingerprint bits. For simplicity, here we consider fingerprint bits that are i.i.d. Under this setting, the designer chooses a value $0 \leq q_0 \leq 0.5$ as the probability that a fingerprint bit is 0 and $q_1 = 1 - q_0$ is the probability that a bit is 1. Thus, the strategy space for the designer S_D is the interval $[0, 0.5]$.

The strategy space for the adversary consists of possible modifications of the content that do not introduce excessive distortion and render the content unusable. Denote the probability of a fingerprint bit 0 being changed to a 1 after modification of the video by p_{01} and the probability that a bit 1 changes to 0 by p_{10}. As the adversary chooses these parameters, his strategy space is given by $S_A = 0 \leq p_{01}, p_{10} \leq 1$.

2.5.2 Payoff functions

Designer's Payoff Function

At the detection stage, for each content $\mathbf{V}^{(i)}$ in the database, the detector has to decide whether the query content denoted by $\mathbf{Z}$ is a distorted version of $\mathbf{V}^{(i)}$, by comparing their fingerprints. Let $\mathbf{X}^{(i)}$ and $\mathbf{Y}$ be the fingerprints of $\mathbf{V}^{(i)}$ and $\mathbf{Z}$, respectively. If $\mathbf{Z}$ is indeed a modified version of $\mathbf{V}^{(i)}$, then the fingerprints $\mathbf{X}^{(i)}$ and $\mathbf{Y}$ are dependent and their joint distribution is $p(\mathbf{Y}|\mathbf{X}^{(i)})q(\mathbf{X}^{(i)})$, where $q(\cdot)$ is the marginal distribution of the fingerprints and $p(\mathbf{Y}|\mathbf{X}^{(i)})$ is the conditional distribution representing the modification. If $\mathbf{Z}$ is *not* a modified version of $\mathbf{V}^{(i)}$, then the fingerprints $\mathbf{X}^{(i)}$ and $\mathbf{Y}$ are independent and their joint distribution is $q(\mathbf{Y})q(\mathbf{X}^{(i)})$.

The identification system's performance can be characterized by the probability P_f of incorrectly deciding that $\mathbf{X}^{(i)}$ and $\mathbf{Y}$ are dependent when they are actually independent, which corresponds to a false alarm, and the probability P_m of deciding that $\mathbf{X}^{(i)}$ and $\mathbf{Y}$ are independent when they are actually dependent which corresponds to a missed detection. As the designer's objective is to achieve low values for P_f and P_m, a suitable function of these quantities can be chosen as the payoff for the designer. However, as in any detection problem, these error probabilities are not independent of each other. In many practical applications, it is common to fix one of these error probabilities, say P_f, to be less than a threshold α and then minimize the other type of error. From the Chernoff-Stein Lemma [16], we know that the best asymptotic error exponent that can be achieved under this setting is given by the Kullback-Leibler (KL) distance between the distributions under the two hypotheses $D(p(\mathbf{Y}|\mathbf{X}^{(i)})q(\mathbf{X}^{(i)})||q(\mathbf{Y})q(\mathbf{X}^{(i)}))$. As the fingerprint bits are i.i.d., the KL distance between the distributions is LD_{KL}, where $D_{KL} = D(p(y|x)q(x)||q(y)q(x))$, $p(\cdot|\cdot)$ is the conditional distribution representing the modification of one bit and $q(\cdot)$ is the common distribution of the individual fingerprint bits. By choosing q_0 appropriately to maximize the KL distance, the designer can reduce the probability of making an error. Thus, we choose the KL distance between the two distributions as the payoff (utility) function for the designer $U_D(q_0, p) = D_{KL} = D(p(y|x)q(x)||q(y)q(x))$.

Adversary's Payoff Function

The adversary's main goal is to evade detection while minimizing the amount of distortion introduced into the content. By choosing the parameters p_{01} and p_{10} to minimize the KL distance D_{KL} between the two distributions, the adversary can reduce the probability of being detected. Hence, we choose $-D_{KL}$ as the adversary's payoff function. We also add a penalty term to the adversary's payoff based on the amount of distortion introduced into the video, to incorporate the adversary's goal of minimizing the perceptual distortion. We assume that the distortion of the original video can be equivalently represented in terms of the change in the fingerprint of the video, as analyzed in Section 2.3. For simplicity, we assume that the perceived commercial value of the distorted content reduces as a linear function of the Hamming distance between the fingerprints of the original and modified content. The analysis can be performed similarly for other models relating the distortion to the reduction in commercial value.

Under this setting, the expected utility for the adversary can be given as $U_A(q_0, p) = -D_{KL} - c_d \frac{1}{L}\mathbb{E}[d_H(\mathbf{X}^{(i)}, \mathbf{Y})]$, where $\mathbb{E}[d_H(\mathbf{X}^{(i)}, \mathbf{Y})]$ is the expected Hamming distance between the fingerprint $\mathbf{X}^{(i)}$ of the original content and the fingerprint $\mathbf{Y}$ of the distorted content, and c_d is the rate at which the perceived value of the content reduces as a function of the average Hamming distance. Since the fingerprint bits are i.i.d., the average Hamming distance can be written as $\frac{1}{L}\mathbb{E}[d_H(\mathbf{X}^{(i)}, \mathbf{Y})] = q_0 p_{01} + q_1 p_{10}$ and the expected payoff for the adversary is given by $U_A(q_0, p) = -D_{KL} - c_d(q_0 p_{01} + q_1 p_{10})$. We see that the adversary can reduce the probability of being detected by reducing D_{KL}, but this would increase the distortion and hence reduce the value of the content. The adversary has to find the optimal tradeoff between these conflicting objectives.

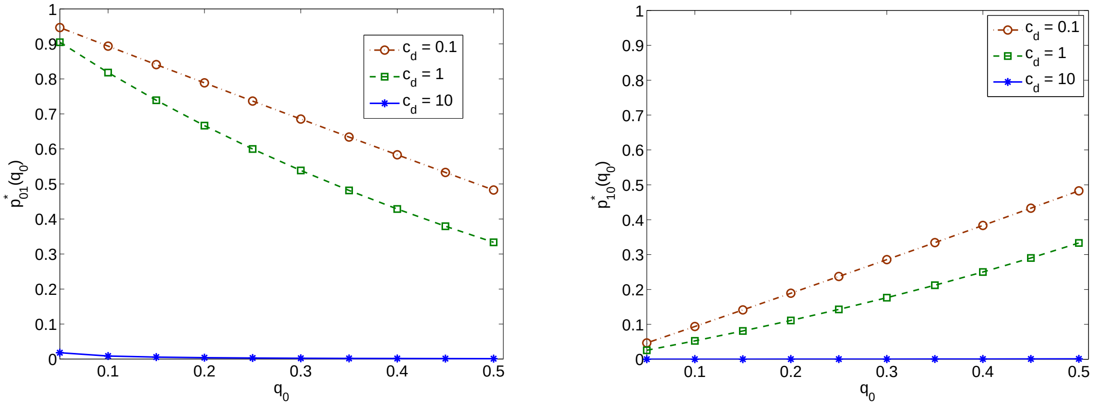
2.5.3 Subgame Perfect Nash Equilibrium

We recognize that under the above settings, the game corresponds to a two player sequential game with perfect recall [65]. In such sequential games, the optimal strategies for the players are given by Subgame Perfect Nash Equilibria (SPNE). The SPNE are similar to saddle-points and correspond to strategies from which neither player has incentive to deviate, given that the other player plays his equilibrium strategy. In other words, given that the designer plays his part of the equilibrium solution, the adversary cannot obtain a higher payoff by playing any strategy other than his equilibrium strategy, and vice versa. The SPNE of this game are given by points $(q_0^*, p^*(q_0^*))$, such that

$$\begin{aligned} p^*(q_0) &= \arg \max_{0 \leq (p_{01}, p_{10}) \leq 1} U_A(q_0, p) \\ q_0^* &= \arg \max_{0 \leq q_0 \leq 0.5} U_D(q_0, p^*(q_0)). \end{aligned} \tag{2.6}$$

These equations indicate that for each q_0 chosen by the designer, the attacker chooses the strategy that maximizes his/her payoff function, called the best response strategy. The designer chooses the strategy that maximizes his/her payoff given that the attacker chooses the best response strategy.

The maximum expected payoff that the adversary can achieve, given that the



(a) $p^*_{01}(q_0)$

(b) $p^*_{10}(q_0)$

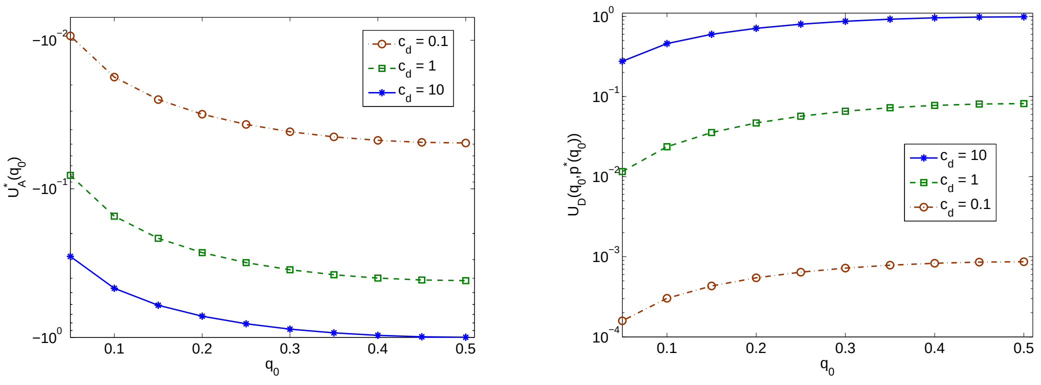
Figure 2.11: Adversary's best response strategy: Optimum choices of (a) p_{01} and (b) p_{10} as a function of the system designer's choice of q_0.

designer chooses q_0 is given by

$$U^*_A(q_0) = \max_{0 \le p_{01}, p_{10} \le 1} -D_{KL} - c_d(q_0 p_{01} + q_1 p_{10}).$$

As $-D_{KL}$ is concave in p [16], the utility function U_A is concave in p. As the constraints are also concave, there is a unique maximizer which is determined as $p^*_{01}(q_0) = \frac{q_1 2^{-c_d}}{q_0 + q_1 2^{-c_d}}$ and $p^*_{10}(q_0) = \frac{q_0 2^{-c_d}}{q_1 + q_0 2^{-c_d}}$. Using the above values for p^*_{01} and p^*_{10}, the maximum value of the expected utility for the adversary is found to be $U^*_A(q_0) = q_0 \log_2(q_0 + q_1 2^{-c_d}) + q_1 \log_2(q_1 + q_0 2^{-c_d})$.

Figure 2.11 shows the optimal values for p_{01} and p_{10} as a function of q_0 for various values of the degradation parameter c_d. We observe that when c_d is small, e.g. $c_d = 0.1$, which implies that the value of the distorted content reduces slowly as a function of the distortion introduced, the adversary can choose large values for p_{01} and p_{10}, corresponding to making large changes to the content so as to evade



(a) $U^*_A(q_0)$

(b) $U_D(q_0, p^*(q_0))$

Figure 2.12: Maximum payoffs for the players: (a) Maximum payoff for adversary as a function of q_0. (b) Designer's payoff when the adversary plays his best strategy.

detection, without incurring a significant reduction in the commercial value. If the parameter c_d is large, e.g. $c_d = 10$, the adversary cannot introduce much distortion into the content, as the value reduces rapidly and is restricted to modifications that result in a very small fraction of the fingerprints bits being altered. The maximum payoff that the adversary can obtain by playing his optimal strategy, in response to the designer's choice of q_0 is shown in Figure 2.12(a). For any fixed value of q_0, the adversary obtains a higher payoff when c_d is small, as he can introduce distortion without reducing the value of the content significantly.

When the adversary plays his best response strategy $p^*(q_0)$ shown in Figure 2.11, the payoff for the designer is found to be

$$\begin{aligned} U_D(q_0, p^*(q_0)) &= -q_0 \log_2(q_0 + q_1 2^{-c_d}) \\ &\quad -q_1 \log_2(q_1 + q_0 2^{-c_d}) \\ &\quad -c_d q_0 q_1 \frac{1 + 2^{-c_d}}{(q_0 + q_1 2^{-c_d})(q_1 + q_0 2^{-c_d})}, \end{aligned}$$

and is shown in Figure 2.12(b). We observe that when c_d increases, the designer can obtain a higher payoff, as the adversary can make limited changes to the content. This indicates that the fingerprint algorithm should be designed carefully, so that it is not easy to alter fingerprint bits without causing a lot of distortion. From the figure, we also see that for a fixed c_d, the payoff for the designer is an increasing function of q_0, attaining a maximum at $q_0 = 0.5$. Thus, the optimal strategy for the designer is to choose the fingerprint bits to be 0 or 1 with equal probability, while the corresponding best strategy for the adversary is $p_{01} = p_{10} = \frac{1}{1+2^{c_d}}$. If $2^{c_d} \gg 1$, $p_{01} = p_{10} \approx 2^{-c_d}$ indicating that the optimal choice for the adversary is to modify a very small fraction of the bits. If $2^{c_d} \ll 1$, then $p_{01} = p_{10} \approx 1$ signifying that the adversary can cause large changes to the hash and easily evade detection.

2.6 Chapter Summary

In this chapter, we described a framework for analyzing content fingerprinting by developing models for the various commonly used modules. By such an analysis we can gain understanding of how different types of multimedia processing affects the fingerprints and how such changes in turn affect the identification performance.

Under this framework, we studied how distortion of the features affects the fingerprint bits extracted from the features. We derived a closed form expression for the minimum amount of distortion needed to change a certain fraction of bits. Our analysis showed that to change a fixed fraction of bits, the minimum mean squared distortion needed is independent of the number of features or equivalently, the length of the fingerprint. We also studied the influence of the feature distribution on this metric and found that distributions that produce values close to zero with higher probability or have high kurtosis, are not favorable from the designer's perspective.

We next studied the design of a quantizer for use in fingerprint schemes to achieve the best identification performance. We derived sufficient conditions for the optimality of the quantizer, and proposed an iterative algorithm to determine the quantization thresholds. Through experiments, we showed that the proposed quantizer can improve the identification performance by around 10% at low P_f values.

We also modeled the dynamics of the interaction between the fingerprint system designer and an adversary seeking to evade detection under the framework of game theory. Using the example of binary fingerprint-based content identification, we illustrated the model and obtained strategies for designing the fingerprints to achieve the best possible performance. We showed that the optimal strategy for the system designer is to design the fingerprinting scheme such that the resultant bits are equally likely to take values 0 and 1 and also highlighted the benefit of designing robust schemes such that the content has to be distorted significantly in order to cause changes to the fingerprint.

Having gained some understanding of different modules involved in translating video features to fingerprint bits, in the next three chapters, we focus on the modeling and analysis of fingerprints in the bit domain. Chapter 3 examines the best possible identification accuracy achievable using binary i.i.d. fingerprints. In Chapter 4 we refine this model to allow correlations among the fingerprints and examine their performance. We examine the impact of correlations among fingerprints extracted from successive temporal frames in Chapter 5.

Chapter 3

Analysis of Binary Fingerprints with i.i.d. Components

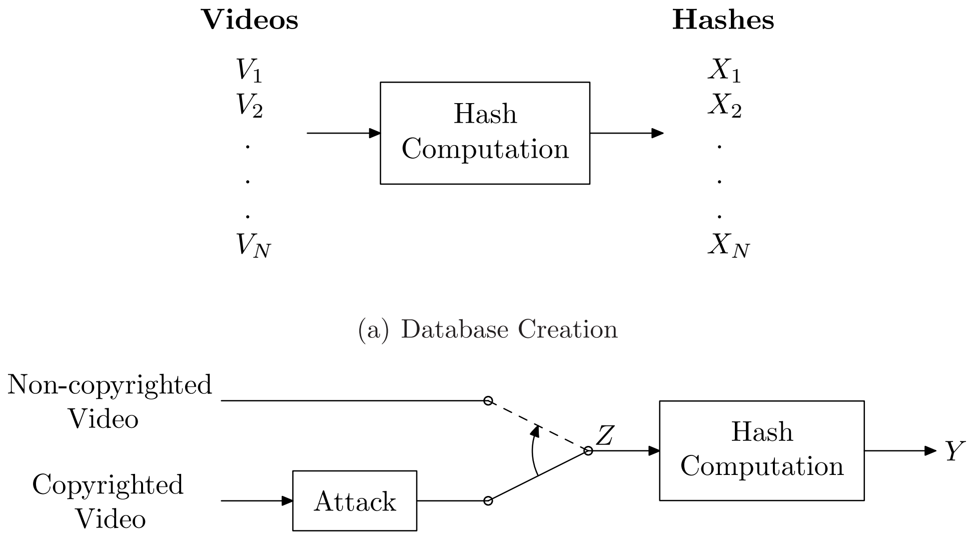
As discussed in the previous chapter, theoretical analysis of content fingerprinting approaches can provide insights into the performance of various algorithms, and enable us to predict how the performance would scale as system parameters, such as the size of the database, increase. In this chapter, we first describe a hypothesis testing framework for evaluating content fingerprinting schemes in Section 3.1. Subsequently, in Section 3.2 we focus on binary fingerprints with i.i.d. bits and derive expressions for the probability of correctly identifying the content. We then use these expressions to derive bounds on the error probabilities in Section 3.3 and examine how these probabilities depend on factors such the fingerprint length and the size of the database. A lower bound on the fingerprint length needed to achieve a desired identification performance is obtained. This analysis also uncovers relations between video fingerprinting and the problem of decoding with errors and erasures. In Section 3.4, we validate our theoretical predictions using an image database and a simple fingerprinting scheme.

3.1 Hypothesis Testing Framework

Hypothesis testing has been commonly used to model identification and classification problems [67] and a similar framework is adopted in this dissertation for analyzing content identification. To establish the notation that will be used in this and the subsequent chapters, we summarize the hypothesis testing framework in this subsection. For ease of presentation, we describe the framework using the example of a video identification application, but the analysis and results apply to other identification tasks as well.

The system model for a fingerprint-based video identification scheme is shown in Figure 3.1. Suppose that the detector has a collection of N videos $\mathbf{V}^{(1)}, \mathbf{V}^{(2)}, \ldots, \mathbf{V}^{(N)}$ which would serve as a reference database for identifying query videos. For example, in a UGC website application, the videos $\{\mathbf{V}^{(i)}\}$ may correspond to copyrighted videos that should be identified and filtered. In the initial creation stage, compact fingerprints $\{\mathbf{X}^{(i)}\}$ corresponding to the videos $\{\mathbf{V}^{(i)}\}$ are computed and stored in the database as shown in Figure 3.1(a). Note that the dimension of the fingerprint $\mathbf{X}^{(i)}$ is usually much smaller than the dimensionality of the video $\mathbf{V}^{(i)}$.

Given a query video $\mathbf{Z}$ that needs to be identified, the detector computes the fingerprint $\mathbf{Y}$ of the query and compares $\mathbf{Y}$ with the fingerprints $\{\mathbf{X}^{(i)}\}_{i=1}^{N}$ stored



(a) Database Creation

(b) Detection Stage

Figure 3.1: System Model

in its database, which are available at the detector. In general, the query $\mathbf{Z}$ may be some video $\mathbf{W}$ that does not correspond to any video in the database or a possibly distorted version of some video $\mathbf{V}^{(i)}$ in the database. These distortions may be caused by incidental changes that occur during transmission and storage, such as compression and transcoding, or they may be intentional distortions introduced by an attacker to prevent the identification of the content.

We consider two different detection objectives based on the requirements of different applications. In some applications, such as a video sharing website implementing content filtering, it may be sufficient to determine if the content is subject to copyright protection or not. In this case, the detector is only interested in determining whether a given video is present in a database of copyrighted material or not. We refer to this scenario as the *detection problem*, which can be formulated as a binary hypothesis test:

$$\begin{aligned} H_0 &: \mathbf{Z} \text{ does not correspond to any video in } \{\mathbf{V}^{(1)}, \mathbf{V}^{(2)}, \ldots, \mathbf{V}^{(N)}\}, \\ H_1 &: \mathbf{Z} \text{ corresponds to } \textit{some} \text{ video in } \{\mathbf{V}^{(1)}, \mathbf{V}^{(2)}, \ldots, \mathbf{V}^{(N)}\}. \end{aligned} \tag{3.1}$$

Under this setting, the performance of a particular fingerprinting scheme with the associated decision rule $\delta_D(\cdot)$ can be evaluated using the probability of false alarm $P_f = \Pr(\delta_D = 1|H_0)$ and the probability of correct detection $P_d = \Pr(\delta_D = 1|H_1)$ or equivalently, the probability of missed detection $P_m = 1 - P_d$.

In some applications, such as automatic tagging of content, the detector is further interested in identifying the original video corresponding to a query video. We refer to this scenario as the *identification problem*, which can be modeled as a multiple hypothesis test with each hypothesis corresponding to one original video and a null hypothesis corresponding to the case that the uploaded video is not present in the database:

$$\begin{aligned} H_0 &: \mathbf{Z} \text{ is not from the database } \{\mathbf{V}^{(1)}, \mathbf{V}^{(2)}, \ldots, \mathbf{V}^{(N)}\}, \\ H_i &: \mathbf{Z} \text{ is a (possibly distorted) version of } \mathbf{V}^{(i)}, i = 1, 2, \ldots, N. \end{aligned} \tag{3.2}$$

In this scenario, the probability of correctly identifying a query video P_c, the probability of misclassifying a video P_{mc}, and the probability of false alarm P_f can be used to quantify the performance of a given fingerprinting scheme and the corresponding detector $\delta_I(\cdot)$. In the remainder of this chapter, we examine the performance of i.i.d. binary fingerprinting schemes under this hypothesis testing framework.

3.2 Fingerprints with Independent Bits

Binary strings are commonly employed in fingerprinting schemes such as [30, 64] since comparison of binary strings can be performed efficiently. From the designer's point of view, it is desirable for the fingerprint bits to be independent of each other, so that an attacker cannot alter a significant number of fingerprint bits at once by making minor changes to the content. Further, if the bits are equally likely to be 0 or 1, the overall entropy is maximized and each bit conveys the maximum amount of information. If the bits are not equally likely to be 0 or 1, they can be compressed into a shorter vector with equiprobable bits, in order to meet the compactness requirement of the fingerprint. As shown in Section 2.5, from a game-theoretic perspective also, using equally likely bits is advantageous for the designer [90]. Binary strings with independent and identically distributed (i.i.d.) bits also arise in biometric identification [94]. Hence, in this chapter, we focus on the performance of fingerprinting schemes with i.i.d. equally likely bits and assume that each fingerprint $\mathbf{X}^{(i)}$ consists of L bits that are distributed i.i.d. according to a Bernoulli(0.5) distribution. Binary fingerprints with correlated bits will be examined in Chapter 4.

Distortions introduced into the content translate into changes in the fingerprint of the content. By a suitable choice of features used for constructing the fingerprint and appropriate preprocessing and synchronization, such attacks can be modeled as additive noise $\mathbf{n}$ in the hash space [56]. Since the hash bits considered in this section are designed to be i.i.d., we model the effect of attacks on the multimedia content as altering each bit of the hash independently with probability $p < 0.5$, i.e. the components of $\mathbf{n}$ are i.i.d. Bernoulli(p). The maximum possible value of p is proportional to the maximum amount of distortion that may be introduced into the multimedia content and will be referred to as the distortion parameter.

3.2.1 Detection Problem

Under the assumptions outlined above, the *detection problem*, where the detector is only interested in identifying whether a given content is present in a database or not, becomes:

$$\begin{aligned} H_0 &: \mathbf{Y} \neq \mathbf{X}^{(i)} + \mathbf{n} \text{ for } i = 1, 2, \ldots, N, \\ H_1 &: \mathbf{Y} = \mathbf{X}^{(i)} + \mathbf{n}, \text{ for some } i \in \{1, 2, \ldots, N\} \end{aligned} \tag{3.3}$$

where $\mathbf{Y}$, $\mathbf{X}^{(i)}, i = 1, 2, \ldots, N$ and the noise $\mathbf{n}$ are all binary vectors of length L. Under hypothesis H_0, $\mathbf{Y}$ can take any value with equal probability, since the fingerprint bits are i.i.d. with equal probability of being 0 or 1, so that $\Pr(\mathbf{Y} = \mathbf{y}|H_0) = \frac{1}{2^L}, \forall \mathbf{y} \in \{0, 1\}^L$. The distribution of the fingerprint $\mathbf{Y}$, given that it is a modified version of $\mathbf{X}^{(i)}$, $\Pr(\mathbf{Y}|\mathbf{X}^{(i)})$ can be specified by considering their Hamming distance. Let $d_i = d(\mathbf{Y}, \mathbf{X}^{(i)})$ be the Hamming distance between the fingerprint of the query video and a given fingerprint $\mathbf{X}^{(i)}$ in the database. Since the probability of a bit being altered due to the noise is p, the probability that exactly d_i bits are altered is $\Pr(\mathbf{Y}|\mathbf{X}^{(i)}) = p^{d_i}(1-p)^{L-d_i}$.

The alternative hypothesis H_1 is thus a composite hypothesis, as the computed fingerprint $\mathbf{Y}$ can have different distributions depending on which original fingerprint it corresponds to. The optimal decision rule for composite hypothesis testing is given as [67]:

$$\text{Decide } H_1 \text{ if } \quad \frac{p(\mathbf{Y}|H_1)}{p(\mathbf{Y}|H_0)} > \tau'' \tag{3.4}$$

where the threshold τ'' can be chosen to satisfy some optimality criterion. If the priors of the hypotheses and the associated costs are known, then τ'' can be computed so as to minimize the expected Bayes risk. If the costs are known, but the priors are unknown, the threshold τ'' can be chosen to minimize the maximum expected risk. We use a Neyman-Pearson approach [67] to maximize the probability of detection P_d subject to the constraint that the probability of false alarm $P_f \leq \alpha$.

To simplify the analysis, we assume that all videos in the database are equally likely to correspond to a query. In situations where some popular videos may be queried more often than others, the analysis can be applied by appropriately modifying the prior probabilities. With this assumption, the likelihood ratio test in Eqn. (3.4) becomes:

$$\frac{\sum_{i=1}^{N} p(\mathbf{Y}|\mathbf{X}^{(i)}) p(\mathbf{X}^{(i)}|H_1)}{p(\mathbf{Y}|H_0)} > \tau''.$$

Substituting $p(\mathbf{Y}|H_0) = \frac{1}{2^L}$, $p(\mathbf{Y}|\mathbf{X}^{(i)}) = p^{d_i}(1-p)^{L-d_i}$, and $p(\mathbf{X}^{(i)}|H_1) = \frac{1}{N}$, we get:

$$\sum_{i=1}^{N} \left(p^{\frac{d_i}{L}} (1-p)^{1-\frac{d_i}{L}} \right)^L > \tau' \tag{3.5}$$

where the constants have been absorbed into the threshold τ'. We note that the left hand side is a sum of exponentials, and for a reasonably large L, only the largest term would be relevant. Further, since $p^x(1-p)^{1-x}$ is a decreasing function of x for $p < 0.5$, the largest term in the left hand side of Eqn. (3.5) would be the one with the smallest value of d_i. Thus, we arrive at the decision rule:

$$\delta_D = \begin{cases} 1 & \text{if } d_{\min} < \tau, \\ 1 \text{ with probability } q & \text{if } d_{\min} = \tau, \\ 0 & \text{otherwise,} \end{cases} \tag{3.6}$$

where $d_{\min} = \min\limits_{i=1,2,\ldots,N} d_i$. Here τ is an integer threshold expressed in terms of the Hamming distance, and τ and q are chosen to achieve a desired probability of false alarm α. Based on this decision rule, the query is detected as being present in the database ($\delta_D = 1$), if the minimum Hamming distance between the fingerprint of the query and the fingerprints in the database is less than a specified threshold τ.

Computing P_d and P_f

The probability of false alarm P_f for a threshold τ is given by $P_f(\tau) = \Pr(d_{\min} < \tau | H_0) + q \Pr(d_{\min} = \tau | H_0)$. To compute the value of $P_f(\tau)$, consider the Hamming distance between $\mathbf{Y}$ and $\mathbf{X}^{(i)}$, which can be expressed as $d_i = d(\mathbf{Y}, \mathbf{X}^{(i)}) = \text{wt}(\mathbf{Y} \oplus \mathbf{X}^{(i)})$, where $\text{wt}(\cdot)$ denotes the Hamming weight of a binary vector and $\oplus$ denotes addition over the binary field (XOR). Under H_0, since each bit of $\mathbf{Y}$ and $\mathbf{X}^{(i)}$ are equally likely to be 0 or 1, each component of $\mathbf{Y} \oplus \mathbf{X}^{(i)}$ is also Bernoulli(0.5). The probability distribution of $d_i = \text{wt}(\mathbf{Y} \oplus \mathbf{X}^{(i)})$ thus corresponds to the weight of a random binary vector with i.i.d. uniform entries, which is a binomial distribution with parameters L and 0.5. Denote the probability mass function (p.m.f.) of a binomial random variable with parameters L and 0.5 by $f_0(k) \triangleq \frac{1}{2^L}\binom{L}{k}$ and the tail probability by $F_0(k) \triangleq \sum_{j=k}^{L} f_0(j)$. Then $\Pr(d_i = k | H_0) = f_0(k)$ and

$\Pr(d_i \geq k | H_0) = F_0(k)$.

As the fingerprints $\mathbf{X}^{(i)}, i = 1, 2, \ldots, N$ are independent, we have $\Pr(d_{\min} \geq \tau | H_0) = \prod_{i=1}^{N} \Pr(d_i \geq \tau | H_0) = [F_0(\tau)]^N$. The probability of false alarm can now be written as

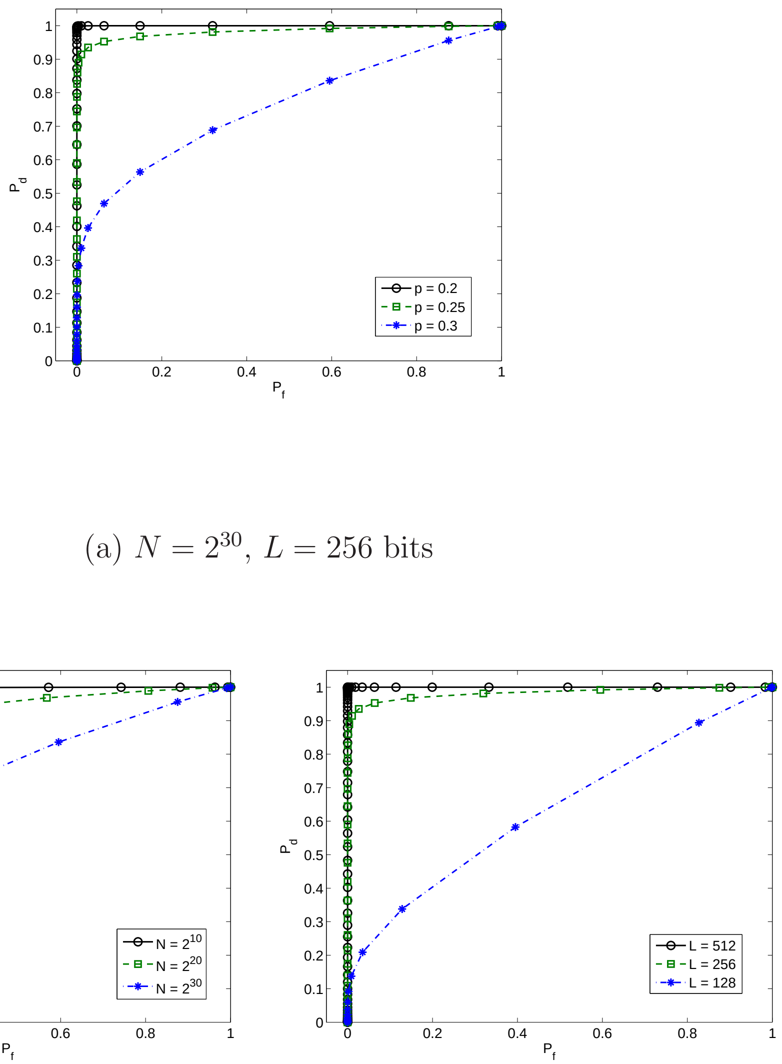
$$\begin{aligned} P_f(\tau) &= (1 - [F_0(\tau)]^N) + q([F_0(\tau)]^N - [F_0(\tau+1)]^N) \\ &= 1 - (1-q)[F_0(\tau)]^N - q[F_0(\tau+1)]^N \qquad (3.7) \end{aligned}$$

To compute the probability of detection, denote the p.m.f. of a binomial random variable with parameters L and p by $f_1(k) \triangleq \binom{L}{k} p^k (1-p)^{L-k}$ and the tail probability by $F_1(k) \triangleq \sum_{j=k}^{L} f_1(j)$. The probability of detection is given as $P_d(\tau) = \Pr(d_{\min} < \tau | H_1) + q \Pr(d_{\min} = \tau | H_1)$. Suppose that H_1 is true and that the query video is actually a distorted version of video V_s. As the noise is assumed to change each fingerprint bit independently with probability p, $\Pr(d_s = k | H_1, s) = f_1(k)$ and $\Pr(d_s \geq \tau | H_1, s) = F_1(k)$. For $i \neq s$, since $\mathbf{X}^{(i)}$ is independent of $\mathbf{Y}$ and has i.i.d. equally likely bits, $\mathbf{Y} \oplus \mathbf{X}^{(i)}$ has i.i.d. Bernoulli(0.5) components. Thus the distance $d_i = \mathrm{wt}(\mathbf{Y} \oplus \mathbf{X}^{(i)}), i \neq s$ follows a binomial distribution with parameters L and 0.5, which is the same as the distribution under H_0. Now consider

$$\begin{aligned} \Pr(d_{\min} \geq \tau | H_1, V_s) &= \Pr(d_s \geq \tau | H_1, V_s) \prod_{i \neq s} \Pr(d_i \geq \tau | H_1, V_s) \\ &= F_1(\tau)[F_0(\tau)]^{N-1}. \end{aligned}$$

The probability of detection can then be written as

$$\begin{aligned} P_d(\tau) &= 1 - [F_1(\tau)][F_0(\tau)]^{N-1} + q(\, [F_1(\tau)][F_0(\tau)]^{N-1} - [F_1(\tau+1)][F_0(\tau+1)]^{N-1}\,) \\ &= 1 - (1-q)[F_1(\tau)][F_0(\tau)]^{N-1} - q[F_1(\tau+1)][F_0(\tau+1)]^{N-1}. \qquad (3.8) \end{aligned}$$



(a) $N = 2^{30}$, $L = 256$ bits

(b) $p = 0.3$, $L = 256$ bits

(c) $N = 2^{30}$, $p = 0.3$

Figure 3.2: ROC for the binary hypothesis testing problem obtained from theoretical analysis.

Numerical Results

In Figure 3.2, we show the receiver operating characteristics (ROC) computed using Eqns. (3.7) and (3.8) for various values of the parameters L, N, and p. Figure 3.2(a) shows the ROC curves as the distortion parameter p is increased from 0.2 to 0.3 for $N = 2^{30}$ fingerprints in the database each of length 256 bits. We observe that as the distortion parameter p increases, the probability P_d of detecting a copyrighted video reduces for a given probability of false alarm P_f. As p approaches 0.5, the probability of detection approaches the lower bound $P_d = P_f$. Figure 3.2(b) examines the influence of the number of fingerprints in the database N on the detector performance for a fixed fingerprint length $L = 256$ bits and distortion parameter $p = 0.3$. As N increases, the probability of false alarm increases. As a result, for a given P_d, the P_f is higher, or equivalently, for a fixed P_f, the probability of detection is lower. Figure 3.2(c) shows that under a given distortion, the detector performance can be improved by using a longer fingerprint. As the fingerprint length is increased, P_d increases for a given P_f.

3.2.2 Identification Problem

We now consider the *identification problem* for binary fingerprinting schemes, where the detector is interested in identifying the specific video that the query corresponds to. As discussed in Section 3.1, this scenario can be modeled as a multiple hypothesis test:

$$\begin{aligned} H_0 &: \quad \mathbf{Y} \neq \mathbf{X}^{(i)} + \mathbf{n}, \text{ for } i = 1, 2, \ldots, N, \\ H_i &: \quad \mathbf{Y} = \mathbf{X}^{(i)} + \mathbf{n}, \quad i = 1, 2, \ldots, N. \end{aligned} \tag{3.9}$$

As before, we assume that the fingerprint bits are i.i.d. and equally likely to be 0 or 1, the noise independently changes each bit with probability p and that the prior probability of each hypothesis is the same. Under this model, the Maximum Likelihood (ML) decision rule can be derived as:

$$\delta_I = \begin{cases} i & \text{if } d_i \leq \tau \text{ and } i = \underset{j=1,2,\ldots,N}{\arg\min}\, d_j, \\ \\ 0 & \text{otherwise,} \end{cases} \tag{3.10}$$

where $d_i = d(\mathbf{Y}, \mathbf{X}^{(i)})$. If fingerprints of several copyrighted videos have the same distance to the fingerprint of the query video $\mathbf{Y}$, one of them is chosen randomly as the match.

We now compute the performance metrics for the ML detector δ_I. The probability of false alarm P_f is given by

$$\begin{aligned} P_f(\tau) &= \Pr(\text{at least one of } d_1, d_2, \ldots, d_N \leq \tau | H_0), \\ &= 1 - \Pr(\text{none of } d_1, d_2, \ldots, d_N \leq \tau | H_0), \\ &= 1 - [F_0(\tau + 1)]^N. \end{aligned}$$

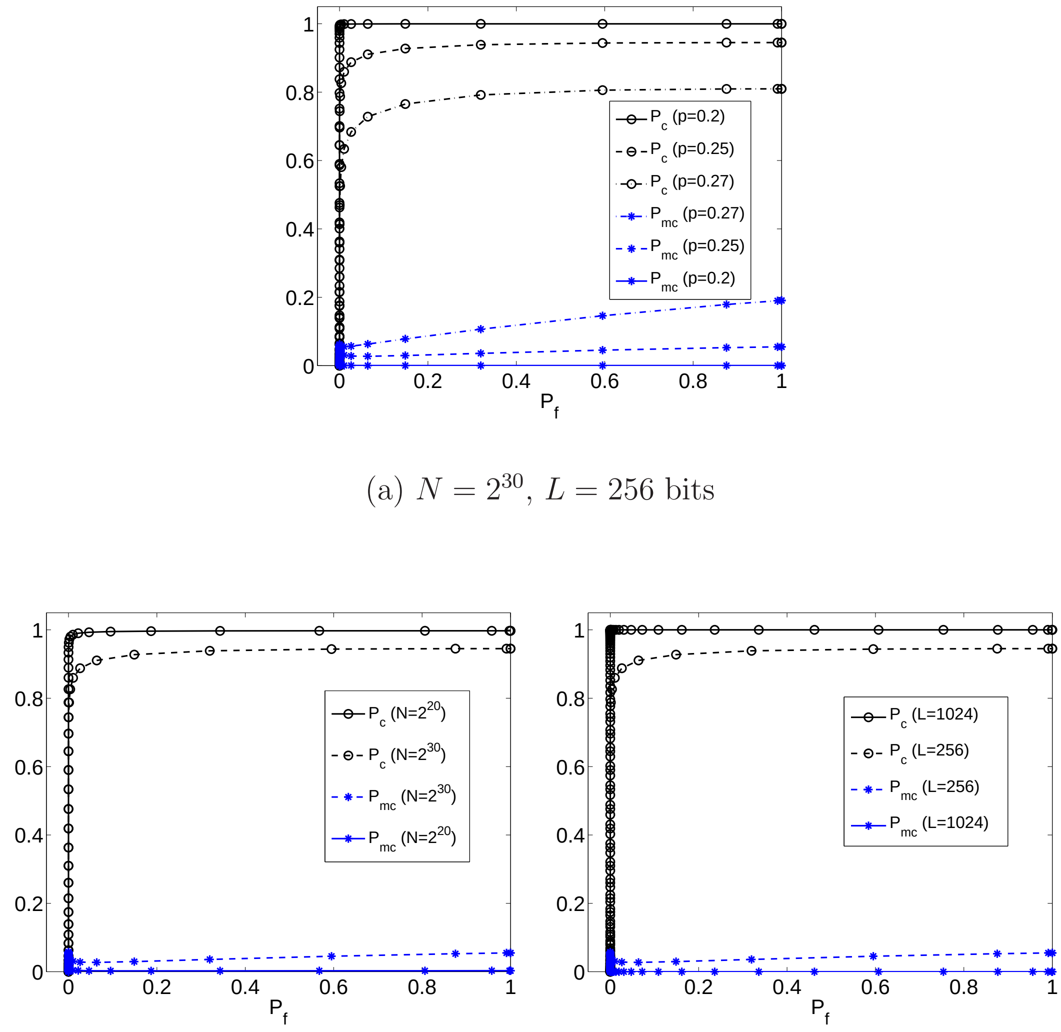
As the fingerprints $\{\mathbf{X}^{(i)}\}$ are identically distributed and equally likely to be queried, and the distribution of the noise $\mathbf{n}$ under each of the hypotheses is the same, the overall probability of correct identification P_c will be equal to the probability of correct identification under any given hypothesis, for example H_1. Under this hypothesis, d_1 has p.m.f. f_1 and $d_i, i \neq 1$ has p.m.f. f_0, so that:

$$\begin{aligned} P_c(\tau) &= \Pr(\delta_I = 1 | H_1) \\ &= \Pr(d_1 \leq \tau \bigwedge d_1 < \min_{i>1} d_i | H_1) + \Pr(\min_{i>1} d_i = d_1 \bigwedge d_1 \leq \tau \bigwedge \delta_I = 1 | H_1), \\ &= \sum_{j=0}^{\tau} f_1(j) \left[\{F_0(j+1)\}^{N-1} + \sum_{k=1}^{N-1} \frac{1}{k+1} \binom{N-1}{k} [f_0(j)]^k [F_0(j+1)]^{N-1-k} \right]. \end{aligned}$$

Similarly, the probability of misclassification can be computed as:

$$\begin{aligned} P_{mc}(\tau) &= \Pr(\delta_I \in \{2, 3, \ldots, N\} | H_1), \\ &= \Pr(\min_{i>1} d_i \leq \tau \bigwedge \min_{i>1} d_i < d_1 | H_1) + \Pr(\min_{i>1} d_i = d_1 \bigwedge d_1 \leq \tau \bigwedge \delta_I > 1 | H_1), \\ &= \sum_{j=0}^{\tau} \left[\sum_{k=1}^{N-1} \left\{ \binom{N-1}{k} f_0(j)^k [F_0(j+1)]^{N-1-k} \times \left(F_1(j+1) + \frac{k}{k+1} f_1(j) \right) \right\} \right]. \end{aligned}$$

Figure 3.3 shows the influence of the various parameters on the identification accuracy of the ML detector in Eqn. (3.10). Figure 3.3(a) shows the influence of the distortion parameter p. We observe that as p increases, the probability of correct identification P_c at a given false alarm probability P_f reduces, and the probability of misclassification P_{mc} increases. The influence of the number of videos N on the accuracy of identification is shown in Figure 3.3(b). As the number of videos in the database increases, the probability of false alarm increases, or equivalently, at a given P_f, the value of P_c is lower. Figure 3.3(b) shows that the probability of correct identification under a given distortion p and a given P_f can be increased by increasing the hash length. Thus, given the number of videos N and a desired probability of false alarm P_f, the content identification system can be made more robust by choosing a longer hash length L. These results are similar to that obtained



(a) $N = 2^{30}$, $L = 256$ bits

(b) $p = 0.25$, $L = 256$ bits

(c) $N = 2^{30}$, $p = 0.25$

Figure 3.3: ROC curves for the multiple hypothesis testing problem obtained from theoretical analysis.

for the detection problem in the previous section.

3.3 Error Exponents and Performance Bounds

In Section 3.2, we have derived expressions for the probability of correct identification and false alarm for a given set of parameters and examined the tradeoff between identification accuracy, robustness and the fingerprint length. In practice, we are often interested in choosing system parameters to ensure that the probability of error is below a certain threshold. While the expressions for P_d and P_f in Section 3.2 can be used to choose the parameters, the equations are non-linear and cannot be solved easily. Hence, in this section, we derive bounds on the achievable error probabilities using fingerprints of a given length and provide guidelines for choosing the fingerprint length required to achieve a desired detection accuracy. We provide an intuitive interpretation of these bounds and show that content identification with a false alarm requirement shares some similarities with the problem of joint source channel coding.

3.3.1 Error Exponents

Consider the detection problem where the detector is only interested in deciding whether a query video is a modified version of some video in the database or not. As before, we examine the case of i.i.d. binary fingerprints with the corresponding decision rule given by Eqn. (3.6). As we are interested in deriving bounds, we assume, for simplicity, that $q = 1$ in the decision rule. The probability of false alarm is given by

$$
\begin{aligned}
P_f(\tau) &= \Pr\left(\bigcup_{i=1}^{N}\{d(\mathbf{Y},\mathbf{X}^{(i)}) < L\lambda\}|H_0\right), \\
&\leq \sum_{i=1}^{N}\Pr(d(\mathbf{Y},\mathbf{X}^{(i)}) < \tau|H_0), \\
&= N\Pr(d(\mathbf{Y},\mathbf{X}^{(1)}) < \tau|H_0), \qquad (3.11)
\end{aligned}
$$

where we have used the union bound and the fact that the fingerprints $\mathbf{X}^{(i)}$ are i.i.d. As discussed in the previous section, under H_0, $\mathbf{Y}$ and $\mathbf{X}^{(1)}$ are independent with each component being equally likely to be 0 or 1. Thus, the XOR of $\mathbf{Y}$ and $\mathbf{X}^{(1)}$ is uniformly distributed over all binary strings of length L. The Hamming distance $d(\mathbf{Y},\mathbf{X}^{(1)}) = \mathrm{wt}(\mathbf{Y}\oplus\mathbf{X}^{(1)})$ and as a result, $\Pr(d(\mathbf{Y},\mathbf{X}^{(1)}) < \tau|H_0) = \frac{1}{2^L}\sum_{\mathbf{x}\in\{0,1\}^L}\mathbb{1}(\{\mathrm{wt}(\mathbf{x}) < \tau\}) = \frac{1}{2^L}S_{L,\tau}$, where $\mathbb{1}(\cdot)$ is an indicator function and $S_{L,\tau}$ is the number of binary vectors within a sphere of radius τ in $\{0,1\}^L$. Let $\lambda = \frac{\tau}{L}$ be the normalized radius. The volume of the sphere $S_{L,L\lambda}$, for $\lambda \leq \frac{1}{2}$ can be bounded as

$$
S_{L,L\lambda} \leq 2^{Lh(\lambda)},
$$

where $h(p) = -p\log_2 p - (1-p)\log_2(1-p)$ is the entropy function [69]. By combining this result with Eqn. (3.11), the probability of false alarm can be bounded from above as

$$
\begin{aligned}
P_f(L\lambda) &\leq N2^{-L}S_{L,\tau} \\
&\leq N2^{-L(1-h(\lambda))} \qquad (3.12)
\end{aligned}
$$

where $\tau = L\lambda$. The same result can been obtained by applying the Chernoff bound to upper bound $\Pr(d(\mathbf{Y},\mathbf{X}^{(1)}) < L\lambda)$ for $\lambda < \frac{1}{2}$, with $d(\mathbf{Y},\mathbf{X}^{(1)})$ being a binomial random variable with parameters L and $\frac{1}{2}$ [73]. However, we prefer this approach as it provides an intuitive explanation of the bounds, which is discussed in Section 3.3.3.

We next consider the probability of a missed detection $P_m = 1 - P_d$. Suppose that $\mathbf{X}^{(i)}$ is the fingerprint of a video $\mathbf{V}^{(i)}$ in the database and that $\mathbf{Y}$ is the fingerprint of a modified version of $\mathbf{V}^{(i)}$. A missed detection occurs if no fingerprint in the database is within a distance τ of the query fingerprint $\mathbf{Y}$. The probability of a missed detection can thus be bounded by the probability that the distance between $\mathbf{Y}$ and the original fingerprint $\mathbf{X}^{(i)}$ is larger than τ:

$$\begin{aligned} 1 - P_d(\tau) = P_m(\tau) &= \Pr\left(\bigcap_{j=1}^{N}\{d(\mathbf{Y}, \mathbf{X}^{(j)}) > L\lambda\}|H_i\right), \\ &\leq \Pr(d(\mathbf{Y}, \mathbf{X}^{(i)}) > L\lambda|H_i) \end{aligned}$$

Since $\mathbf{Y}$ is generated by flipping each bit of $\mathbf{X}^{(i)}$ with a probability p, $d(\mathbf{Y}, \mathbf{X}^{(i)})$ is distributed according to a binomial random variable with parameters L and p so that $P_m \leq \Pr(B > \tau)$, where B is a binomial random variable with distribution $\text{Binomial}(L, p)$. By the Chernoff bound [73], the tail probability of the binomial distribution can be bounded as

$$\Pr(B \geq L\lambda) \leq 2^{-LD(\lambda||p)}$$

where $D(\lambda||p)$ is the Kullback-Leibler distance between two Bernoulli distributions with parameters λ and p respectively. Thus, the probability of missed detection when $\tau = L\lambda$ can be bounded as

$$P_m(L\lambda) \leq 2^{-LD(\lambda||p)}. \tag{3.13}$$

These bounds on P_f and P_m may be interpreted as consequences of the large deviations principle [21]. As the Chernoff bound is asymptotically tight, the bounds presented above are also asymptotically tight on the exponential scale, for example, $\lim_{L\to\infty} \frac{1}{L} \log P_m(L\lambda) = D(\lambda||p)$.

Eqns. (3.12) and (3.13) show the tradeoff between the probability of false alarm P_f, the probability of missed detection P_m and the number of fingerprints N in the database. For example, given N videos, reducing the P_f would require $1 - h(\lambda)$ to be as large as possible, or equivalently, λ must be as small as possible. However, reducing λ leads to an increase in the P_m. To further examine this tradeoff, let us define the rate R as $N = 2^{LR}$, the false alarm error exponent as $E_f = 1 - h(\lambda) - R$, and the missed detection error exponent as $E_m = D(\lambda||p)$, so that $P_f \leq 2^{-LE_f}$ and $P_m \leq 2^{-LE_m}$. In the Neyman-Pearson setting, given a certain number of videos N and fingerprint length L, suppose we wish to ensure that $P_f \leq \epsilon = 2^{-L\Delta}$ and minimize P_m. This is equivalent to maximizing E_m for a fixed rate R while ensuring that $E_f \geq \Delta$:

$$\max_{\lambda} E_m = D(\lambda||p) \quad \text{subject to } 1 - h(\lambda) - R \geq \Delta. \tag{3.14}$$

As the objective function is increasing in λ, while the constraint is decreasing in λ, the maximum is achieved when $1 - h(\lambda) - R = \Delta$. Under this setting, Figure 3.4 shows the maximum achievable missed detection error exponent E_m as a function of the false alarm error exponent Δ, for a fixed rate R, when $p = 0.3$. From the figure, we observe that at a given rate R, E_m reduces as a function of Δ, which implies that for a fixed number of fingerprints in the database, reducing the false alarms leads to an increase in the number of missed detections, and vice versa. From the

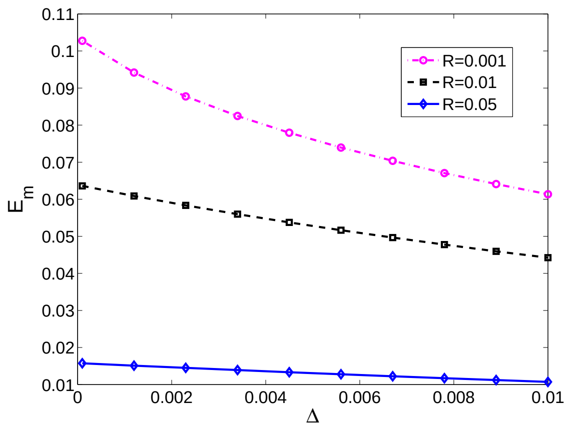


Figure 3.4: Error exponent for missed detection as a function of the rate for different values of the false positive error exponent.

figure, we also observe that for a fixed value of Δ, E_m reduces as N increases. This trend matches the results presented in Section 3.2.

To ensure that $P_m < 0.5$, the decision threshold $\tau = L\lambda$ should be greater than the mean of the binomial distribution Lp. As the entropy function $h(\lambda)$ is monotonically increasing for $\lambda < 0.5$, this would in turn imply that the false alarm exponent $\Delta = 1 - h(\lambda) - R \leq 1 - h(p) - R$. Hence, to ensure that $P_f \leq \epsilon = 2^{-L\Delta}$, we require that $R + \Delta \leq 1 - h(p)$, or equivalently,

$$\frac{1}{L} \log_2 \frac{N}{\epsilon} \leq 1 - h(p). \tag{3.15}$$

Thus, given a video database of size N, to ensure that the probability of false alarm $P_f \leq \epsilon$ when the attack alters on average a fraction p of the hash bits, the length of the fingerprints used for identification should be chosen large enough to satisfy Eqn. (3.15). The corresponding probability of missed detection is then less than 2^{-LE_m}, where E_m can be computed from Eqn. (3.14). It should be noted that at extremely small values of the probability of false alarm, the model mismatch between the i.i.d. model and the practical fingerprint distribution can cause discrepancies between the predicted and practical values of P_f. The required fingerprint length derived from this bound can serve as a guideline for choosing the fingerprint length in a practical system, with suitable compensations to allow for model mismatch.

3.3.2 Bounds on the Error Probabilities for the Identification Problem

Similar bounds on the various errors may be derived in the identification problem. As the expression for the false alarm probability in the identification problem is identical to that in the detection problem, it can be bounded by $P_f \leq 2^{-L(1-h(\lambda)-R)}$. Now, consider the probability of misclassification P_{mc}:

$$\begin{aligned}
P_{mc} &= \Pr(\delta_I \in \{2, 3, \ldots, N\} | H_1), \\
&\leq \Pr(\bigcup_{i=2}^{N} \{d(\mathbf{Y}, \mathbf{X}^{(i)}) < L\lambda\} | H_1) \\
&\leq \sum_{i=2}^{N} \Pr(d(\mathbf{Y}, \mathbf{X}^{(i)}) < L\lambda | H_1), \\
&= (N-1) \Pr(d(\mathbf{Y}, \mathbf{X}^{(1)}) < L\lambda | H_1), \\
&\leq 2^{-L(1-h(\lambda)-R)}
\end{aligned}$$

where we have used the Chernoff bound and replaced $N-1$ by N. Now let $P' = \Pr(d(\mathbf{Y}, \mathbf{X}^{(i)}) > L\lambda | H_i)$. As discussed in the previous subsection, this probability can be bounded using the Chernoff bound as $P' \leq 2^{-LD(\lambda || p)}$. The probability of not making a correct decision $1 - P_c$ is then bounded by:

$$\begin{aligned} 1 - P_c &\leq P_{mc} + P' \\ &\leq 2^{-L(1-h(\lambda)-R)} + 2^{-LD(\lambda \| p)}. \end{aligned}$$

These results are similar to the bounds derived in [25] and [27, Problems 5.14 and 5.15] for the problem of decoding error correcting codes with an erasure option.

3.3.3 A Sphere Packing Perspective

In the previous subsections, we have examined the relation between the rate R, the missed detection error exponent E_m, and the false alarm error exponent Δ. We now provide an intuitive explanation of the theoretical results obtained.

Consider the space of all binary strings of length L, represented by the dashed circle in Figure 3.5. Let the N binary fingerprints $\mathbf{X}^{(i)}, i = 1, 2, \ldots, N$ present in the database be represented by the solid dots in the figure and the circles around the dots represent the detection regions for the respective fingerprints. Any query fingerprint that falls within such a sphere is identified as the fingerprint represented by the center of the sphere. The number of such spheres controls the rate R, and the volume of the spheres determines the probability of false alarm and missed detections.

To ensure a low probability of missed detection when the probability of a bit flipping is p, the detection region around each fingerprint should include all binary strings that are within a Hamming distance Lp from the fingerprint. The volume of such a sphere of radius Lp is $S_{L,Lp}$, which for large L is approximately $S_{L,Lp} \approx 2^{Lh(p)}$.

As we have assumed that in the null hypothesis, the fingerprints of the videos absent from the database are uniformly distributed over the entire space of binary strings of length L, the probability of false alarm is approximately

$$P_f = \frac{N \times S_{L,Lp}}{2^L} \Rightarrow \epsilon \approx \frac{N 2^{Lh(p)}}{2^L} \tag{3.16}$$

which upon rearrangement gives Eqn. (3.15). To achieve a higher rate, we would like to pack more such spheres into the binary space, but this would increase the probability of false alarms. Similarly, to reduce the probability of missed detection, the volume of the decoding region around each fingerprint has to be increased, which would also increase P_f and reduce the number of spheres that can be packed into the binary space.

We see that the fingerprinting problem shares some analogies with source and channel coding. In channel coding, to achieve capacity, we are interested in packing as many spheres as possible into the binary space such that their overlap is minimum. In source coding with a fidelity criterion (rate-distortion theory), we are interested in covering the entire space with as few spheres of fixed size as possible. Here, to minimize the probability of false alarms, we would like to cover the space as sparsely as possible, but the conflicting objective of increasing the rate requires packing as many spheres as possible. Thus, fingerprinting can be thought of as being similar to joint source-channel coding.

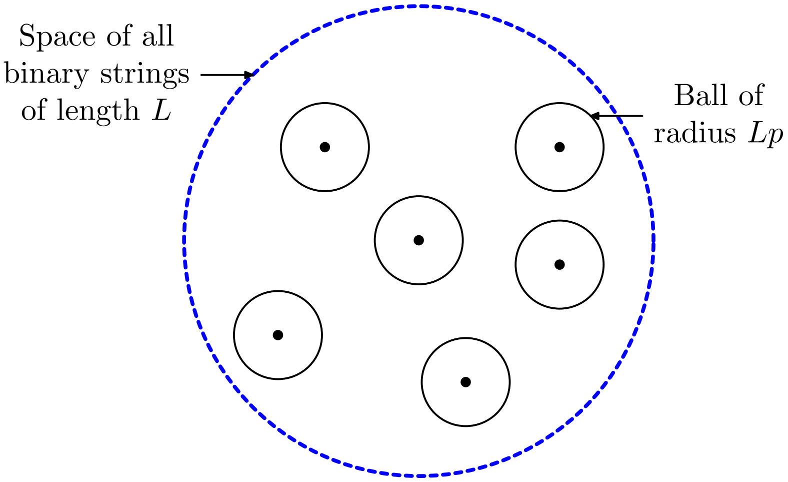


Figure 3.5: Sphere packing perspective of content fingerprinting.

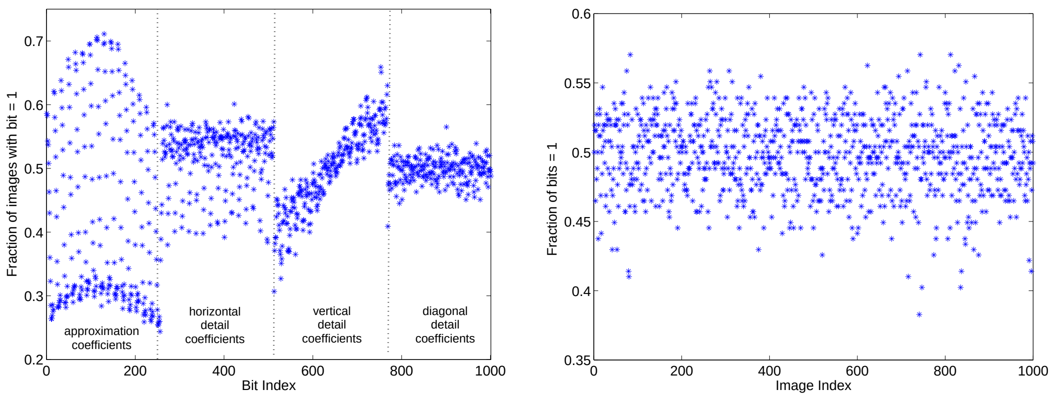
3.4 Evaluation of a Practical Fingerprinting Scheme

In this section, we examine the applicability of our theoretical results to a practical identification scheme. We use a simple image fingerprinting scheme based on the wavelet transform coefficients [55] as an example. A similar scheme for video fingerprinting based on DCT coefficients has been proposed in [15]. We present results for image identification, but the results can be easily extended to the case of video or audio identification using schemes such as [15].

3.4.1 Fingerprint Generation

Wavelet coefficients, and in particular, signs of wavelet coefficients have been used for content identification [3], retrieval of similar images [35], and to generate fingerprints for image authentication [59]. It has been shown that detail coefficients of the wavelet transform are symmetric around zero and can be modeled as i.i.d. generalized Gaussian random variables [54]. Thus, quantizing wavelet detail coefficients to 1 bit would yield i.i.d. equiprobable bits, which could be used as fingerprints to represent the image.

We decompose a 512×512 image using five levels of the wavelet transform using the Haar wavelet [55], which is chosen because of the low cost for computing the transform. Each of the four subbands at the coarsest level of decomposition thus has coefficients of size 16×16. We retain only the signs of the coefficients belonging to these subbands to obtain a 1024-bit sequence. A '1' at a particular location indicates a positive coefficient, whereas a '0' indicates a negative coefficient. Figure 3.6 shows the distribution of the bits comprising this bit sequence estimated from 1000 grayscale images of size 512×512. In Figure 3.6(a), we show the fraction of images (out of 1000) that have a '1' at a particular location. The first 256 bits correspond to the signs of the approximation coefficients, followed by 256 bits for each of the horizontal, vertical and diagonal detail coefficients. From this figure, we observe that the signs of the approximation coefficients are not independent and equally likely. This is due to the fact that the approximation coefficients for natural images are likely to be correlated with each other. The same holds true for the horizontal and vertical detail coefficients, since coefficients which correspond to strong horizontal or vertical edges would lie along the same row or column, respectively. The signs of the diagonal detail coefficients, however, appear to be less correlated and approximately equally likely to be '0' or '1'. Figure 3.6(b) shows the fraction of bits that are '1' for a given image. We observe that approximately half the bits are '1', indicating that these bits are approximately independent and



(a)

(b)

Figure 3.6: Distribution of fingerprint bits obtained by quantizing wavelet coefficients. (a) Fraction of images with a bit '1' at a given location and (b) Fraction of bits that are '1' for a given image.

equally likely. The coefficients at the lowest level of decomposition are also expected to be robust to common signal processing operations and can be used as fingerprints for image identification.

In summary, given an image, we resample it to size 512×512, perform wavelet decomposition up to 5 levels, and extract the diagonal detail coefficients. We then retain the signs of these coefficients to form a 256-bit fingerprint for the given image.

3.4.2 Attacks

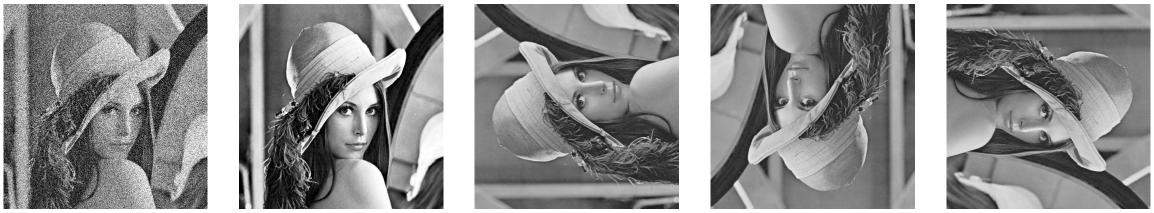
We evaluate the ability of these fingerprints to correctly identify an image after it has undergone the potential malicious attacks listed in Table 3.1. As the image pixel values are normalized to lie between 0 and 1, addition of zero mean Gaussian noise with standard deviation $\sigma = 0.2$ represents a strong attack and introduces

Table 3.1: List of attacks tested

Attack No.	Attack	Parameters
1-4	Zero-mean Gaussian Noise Addition	$\sigma = 0.05, 0.1, 0.15, 0.2$
5-8	Uniform Noise Addition $[-\frac{\Delta}{2}, \frac{\Delta}{2}]$	$\Delta = 0.05, 0.1, 0.15, 0.2$
9	Histogram Equalization	
10-19	Gamma Correction	$\gamma = 0.75 : 0.05 : 1.25 \backslash \{1\}$
20-28	Average, Median, and Gaussian Filtering	Filter Size $= 3, 5, 7$
29-31	JPEG Compression	Quality Factor $= 25, 50, 75$
32-34	Rotation by multiples of 90°	

a lot of distortion, as shown in Figure 3.7. Rotation by multiples of 90° (Attack No. 32-34) are very strong attacks that may be of concern if the image/video is being viewed on a portable device, which provides freedom in adjusting the orientation.

The strength of an attack can be measured in terms of the probability (p) of a fingerprint bit being altered after the attack. Figure 3.8 shows the probability of a fingerprint bit being changed as a result of each attack, averaged over 1000 images. We observe that the rotation attacks are devastating, and the probability of a fingerprint bit being altered is almost 0.5 for each of them. Our analysis of the probability of correct identification and false alarm in Section 3.2 suggests that



Attack 4 Attack 9 Attack 32 Attack 33 Attack 34

Figure 3.7: Some attacked versions of the Lena image. (The list of attacks is provided in Table 3.1).

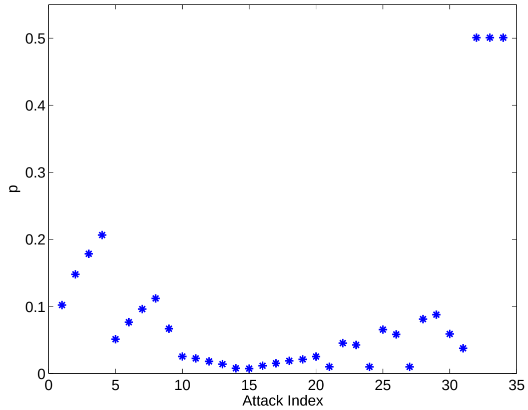


Figure 3.8: Probability of a fingerprint bit flipping averaged over 1000 images for each attack.

the fingerprinting scheme will not accurately identify the images after these attacks due to the high value of p. Among the other attacks, Gaussian noise addition with standard deviation of 0.2 (Attack No. 4) causes the highest number of changes to the fingerprint bits.

3.4.3 Performance Evaluation

We now evaluate the accuracy of the content identification system under these attacks. Our database consists of $N = 1000$ grayscale images of size 512×512. The attacks in Table 3.1 are applied to each of these images to obtain a set of $34,000$ attacked images. The length of the fingerprint used is $L = 256$ bits. The threshold for detection τ is chosen to achieve a probability of false alarm $\epsilon = 10^{-6}$. From Eqn. (3.15), the maximum attack strength that can be resisted under these settings is found to be $p = 0.3$. Thus, we expect that the rotated images (Attack No. 32-34) which have $p = 0.5$ will not be detected correctly. The other attacks No. $1-31$ have $p < 0.3$ and hence we expect the probability of detection P_d to be close to 1.

For the detection problem, we compute the fingerprint of an attacked image and compare it with each fingerprint in the database. We then use the decision rule described in Eqn. (3.6) to perform the classification. If the minimum distance $d_{\min} < \tau$, we declare the image to be present in the database. Figure 3.9 shows the probability of detection obtained using this decision rule under each of the attacks. As expected, the images which correspond to rotated versions of images in the database are almost never detected (Attacks No. $32 - 34$). This problem can be alleviated by suitably designing the fingerprints, as discussed in Section 3.4.5.

Under most of the other attacks, the probability of detection P_d is close to 1, except for addition of Gaussian noise with large variance (attacks no. 2-4). Under these attacks, the fraction of fingerprint bits altered for some images is larger than 0.3. Thus, according to our theoretical analysis, these images cannot be identified

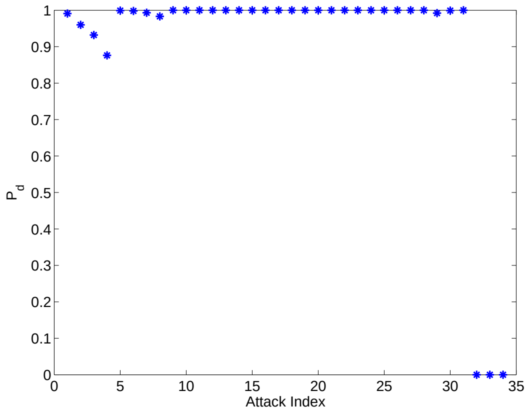


Figure 3.9: Probability that an attacked image is detected as a modified version of an image in the database P_d.

and the probability of detection, P_d, is less than 1 for these attacks. The overall probability of detection for attacks no. 1-31 was 0.991.

For the identification problem, we use the ML detector in Eqn. (3.10) to perform the classification. We found that *every* image that was detected as being present in the database in the detection problem was correctly identified, so that $P_c = 0.991$ and the probability of misclassification $P_{mc} = 0$.

The probability of false alarm P_f was estimated using the leave-one-out procedure in both the detection and identification problems. Every image in the database was treated as a probe image and compared with the remaining images. If the minimum distance of the fingerprint $d_{\min} < \tau$, the image constituted a false alarm. Using the fingerprint of length 256 bits, no false alarms were observed in our experiments.

3.4.4 Influence of the Fingerprint Length

Based on the analysis in Section 3.3, we know that longer fingerprints can resist stronger attacks. In this subsection, we perform simulations to determine the influence of the fingerprint length on the detection performance.

To generate fingerprints of different lengths, the number of levels of the wavelet decomposition is varied. For example, to generate fingerprints of length 1024 bits, we resample the image to size 512×512 and decompose it to four levels using the Haar wavelet. We then extract the signs of the diagonal detail coefficients at the coarsest level of decomposition. As the number of levels of decompositions becomes smaller, the diagonal detail coefficients correspond to higher frequencies and we expect these features to be less robust to modifications. Figure 3.10 shows the probability of a fingerprint bit flipping after attacks, for fingerprints of length 64, 256, and 1024 corresponding to 6, 5, and 4 levels of decomposition respectively. We observe that as the number of decomposition levels decreases (corresponding to longer fingerprints), the probability that a fingerprint bit changes increases, indicating that these coefficients are less robust to modifications.

From Eqn. (3.15), we find that for $N = 1000$ and $\epsilon = 10^{-6}$, the maximum probability of a bit flipping (p) that can be tolerated by fingerprints of length 64, 256, and 1024 bits is 0.1, 0.3, and 0.4, respectively. Thus, we expect the fingerprint with length 1024 bits to have a higher value of P_d, as it can resist stronger attacks.

In Figure 3.11, we examine the influence of the fingerprint length L on the probability of detection P_d under various attacks. In each case, the threshold for

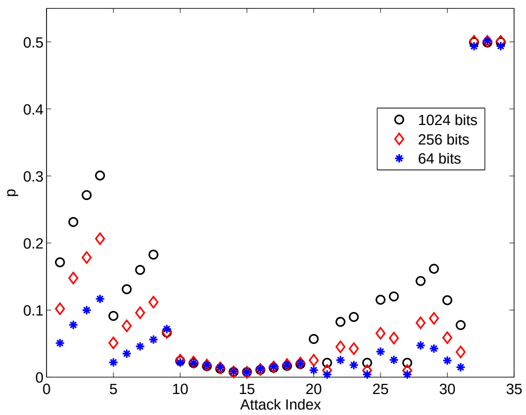


Figure 3.10: Probability of a fingerprint bit flipping under an attack on the image as a function of the fingerprint length.

detection τ was chosen to attain the desired value of $P_f = \epsilon = 10^{-6}$ as given by Eqn. (3.15). We observe that the fingerprint with length 1024 bits has the highest probability of detection. Even though the probability of a bit being altered after an attack p is higher for the 1024-bit fingerprint than the other fingerprints, the longer length of the fingerprint compensates for the reduced robustness of each individual bit, and leads to a higher overall probability of detection.

In Table 3.2, we compare the overall probability of detection under attacks no. $1-31$ as a function of the fingerprint length. We observe that as the fingerprint length increases, P_d also increases. There was only one case of false alarm when using fingerprints of length 1024 bits. Upon closer observation, it was found that these two images actually corresponded to the same scene, but the number of objects and illumination conditions in the picture were slightly different. These two images can be regarded as being obtained from each other after significant modification, such as insertion or deletion of objects, change in brightness, and modification of

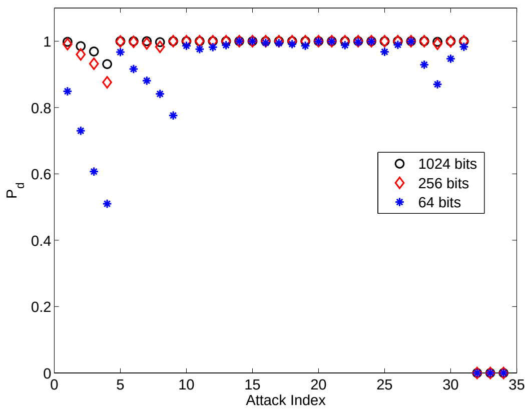


Figure 3.11: Probability of detection under various attacks as a function of the fingerprint length.

the details in the image. The overall attack would change a large fraction of the fingerprint bits, and is hence not identified using the shorter fingerprints. Since the 1024 bit fingerprint is more robust against changes in the fingerprint bits, it is able to determine that these two images are not independent of each other, and could have originated from the same source. Thus, the length of the fingerprint plays a crucial factor in determining the performance of the fingerprinting scheme, as predicted by our theoretical analysis in Section 3.3.

Under the identification problem, every image that is detected as having originated from an image present in the database is also correctly identified, so that $P_c = P_d$. Thus, the probability of misclassification as obtained from our experiments is $P_{mc} = 0$ and the probability of correct identification is the same as the second column in Table 3.2.

Table 3.2: Overall P_d and P_f obtained against Attacks No. 1-31 as a function of the fingerprint length.

Hash Length (bits)	P_d	P_f
64	0.924	0
256	0.991	0
1024	0.996	0.002

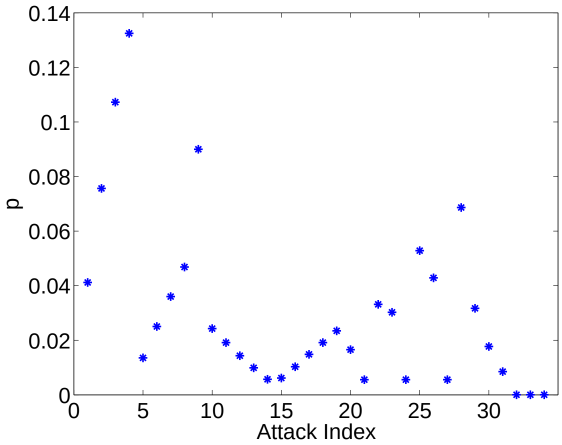


Figure 3.12: Probability of a bit flipping p for the rotationally invariant fingerprints under various attacks.

3.4.5 Proper Choice of Hash Features

Our attack model assumes that most attacks on multimedia can be modeled as additive noise in the fingerprint space. For some fingerprinting schemes, desynchronization attacks, including rotation, cropping, and geometric attacks, may not be directly modeled as additive noise fingerprint space. However, by suitably designing the features and applying appropriate preprocessing, it is possible to reduce these attacks to the additive noise model. We briefly illustrate the importance of appropriate choice of features using the example of the rotation attacks studied in Section 3.4.2. If robustness against rotations by multiples of 90° is desired, the following modification of the fingerprint scheme in Section 3.4.1 can improve the robustness against rotations.

Given a 512×512 image, we obtain four images corresponding to rotations by multiples of 90°, which are then summed pixel-wise. The resulting image is decomposed up to four levels using the Haar wavelet and the signs of the 1024 diagonal detail coefficients at the coarsest level of decomposition are extracted. As these bits are dependent, we retain only 25% of the bits that correspond to the coefficients in the upper left corner of the subband. The 256 bits thus obtained form the fingerprint for the image, which is invariant under rotations of the original image by multiples of 90°.

Figure 3.12 shows the probability of a bit flipping under the attacks listed in Table 3.1 for this modified scheme. We observe that none of the bits are altered under rotations by multiples of 90°. The fingerprint bits are also moderately robust under the other attacks no. 1-31. Under the detection problem we obtained $P_d = 1$ under the rotation attacks no. 32-34, while the overall P_d for attacks no. $1 - 31$ was 0.99. Thus, a suitable choice of the fingerprint features can enhance the robustness against attacks.

3.5 Chapter Summary

In this chapter, we presented a decision theoretic framework for analyzing binary fingerprint-based content identification schemes. We formulate the problem of detecting whether a given video or audio is present in a database of copyrighted material as a binary hypothesis test and the problem of correctly identifying the original content corresponding to a given query object as a multiple hypothesis test. Under this framework, we considered the case of fingerprinting schemes that generate i.i.d. equally likely bits and modeled distortions on the host content as altering each fingerprint bit independently with probability p. We derived expressions for the probability of correct identification under this model and studied the tradeoff between the number of fingerprints, the robustness, the identification performance and the length of the fingerprints. To understand the fundamental limits on the identification capability, we next derived bounds on the achievable error probabilities and characterized the tradeoff between the detection probability and the number of fingerprints in terms of the error exponents. We then derived guidelines for choosing the fingerprint length to attain a desired performance objective and provided an interpretation of our results from a joint source-channel coding perspective.

Under the proposed framework, we also examined a practical binary hash-based content identification scheme which utilizes the signs of the diagonal detail coefficients in the wavelet decomposition of the image. The simulation results confirm our theoretical predictions. We also briefly discussed the importance of choosing appropriate hash features to achieve robustness against attacks.

Our analysis provides a quantitative evaluation of how various system parameters influence the identification accuracy, and guidelines to choose these parameters to achieve a desired accuracy. It also reveals connections between the fingerprinting problem and other areas such as sphere-packing, joint source channel coding, and errors and erasures decoding.

The results in this chapter have been mainly presented in the context of content fingerprinting, but they are also applicable in many other applications, such as biometrics based identification. For example, Vetro et al. recently showed that by suitably transforming features extracted for human fingerprint matching using minutiae, the biometric data can be transformed into i.i.d. Bernoulli(0.5) bits and that the distortions caused while recapturing the fingerprint can be modeled by a binary symmetric channel [94]. The results and bounds derived would be applicable in this setting.

Chapter 4

Binary Fingerprints with Correlated Bits

The analysis and results described in the previous chapter have been focused on fingerprints with i.i.d. components and provides useful performance bounds and guidelines for designing fingerprinting schemes. Many practical fingerprinting schemes, however, generate fingerprints with correlated components. While it is possible to include an explicit decorrelation stage to remove such dependencies and obtain a shorter fingerprint with independent bits, to meet stringent computational requirements in large-scale practical deployments, it may be preferable to use the correlated fingerprint bits directly without incurring the additional computational cost for decorrelation. Another important reason that correlated fingerprints are used in practice is to cope with desynchronization. For example, fingerprints may be extracted from overlapping segments of multimedia to deal with cropping issues, which results in correlated components.

We use a Markov Random Field to model such correlations in the fingerprints and the noise introduced through distortion of the content. We then describe an approach inspired by statistical physics to compute the probability of errors and study the impact of the correlation on the identification performance of different detectors. Although we use a simple 2-D Ising model as a concrete example to illustrate our results, the analysis and the proposed method are quite general and can be applied to any Markov Random Field. We begin with a brief review of Markov Random Fields in the next subsection.

4.1 Markov Random Fields

Markov Random Fields (MRFs) are a generalization of Markov chains in which time indices are replaced by space indices [40]. MRFs are undirected graphical models and represent conditional independence relations among random variables. MRFs have been used in image processing [36, Chapter 6], image modeling [8, 39], and computer vision [9, Chapter 8] to model correlated random variables.

An MRF is represented by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a set of nodes $\mathcal{V} = \{1, 2, \ldots, N\}$ and a set $\mathcal{E}$ of edges between nodes, where an edge is represented by an unordered pair of nodes. Each node $i \in \mathcal{V}$ represents a random variable X_i and the vector $\mathbf{X}$ denotes all random variables represented by the MRF. Two nodes i and j are said to be neighbors if there is an edge between them, i.e. $(i, j) \in \mathcal{E}$. A set of nodes $\mathcal{C}$ is called a maximal clique if every pair of nodes in $\mathcal{C}$ are neighbors and no node in $\mathcal{V}\backslash\mathcal{C}$ is a neighbor of *every* node in $\mathcal{C}$. Denote the set of variables corresponding to the nodes in $\mathcal{C}$ by $\mathbf{X}_{\mathcal{C}}$ and their realizations by $\mathbf{x}_{\mathcal{C}}$. The set of all the maximal cliques in the graph $\mathcal{G}$ is denoted by $\mathbb{C}$.

An energy function $E_{\mathcal{C}}(\mathbf{x}_{\mathcal{C}})$ is associated with every maximal clique $\mathcal{C} \in \mathbb{C}$ that maps the values $\mathbf{x}_{\mathcal{C}}$ of the nodes in $\mathcal{C}$ to a real number. The energy of the entire configuration $\mathbf{x}$ is defined as $E(\mathbf{x}) = \sum_{\mathcal{C}\in\mathbb{C}} E_{\mathcal{C}}(\mathbf{x}_{\mathcal{C}})$. The joint probability distribution of all the random variables represented by the MRF is then given as $p(\mathbf{X} = \mathbf{x}) = \frac{1}{Z} \exp\left(-E(\mathbf{x})\right)$, where $Z = \sum_{\mathbf{x}} \exp\left(-E(\mathbf{x})\right)$ is a normalization constant called the partition function.

4.2 Model for a Block-based Fingerprinting Scheme

We model content fingerprints as a Markov Random Field where each fingerprint value is represented as a node in the MRF, and pairs of nodes that have dependencies are joined by edges [89, 91]. As a concrete example of our modeling approach, we describe a model for a representative fingerprinting scheme that partitions each video frame into blocks and extracts one bit from each block [53]. While we use a simple two-dimensional model for ease of illustration, the analysis can be extended to three-dimensional and more complex models.

Suppose that each video frame of size $PH \times QW$ is partitioned into PQ blocks of size $H \times W$ each and one bit of the fingerprint is extracted from each block. For example, the fingerprint bit could be obtained by thresholding the average luminance of a block. Due to underlying correlations among the blocks of the frame,

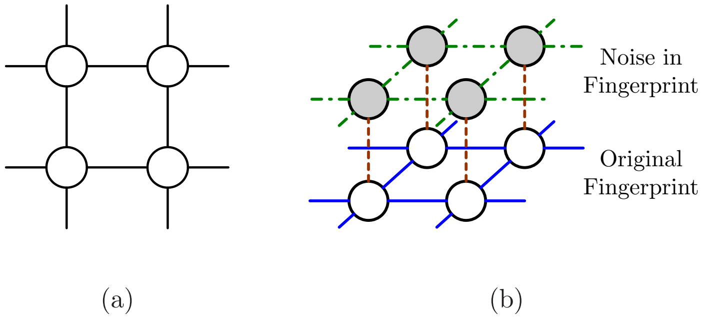


Figure 4.1: Markov Random Field model for a block-based fingerprinting scheme: (a) fingerprint components and (b) fingerprint and noise.

these bits are likely to be correlated. We denote the bit extracted from the $(i, j)^{\text{th}}$ block by $X'_{i,j}$. For notational convenience, we use a vector $\mathbf{X}$ to represent the bits $X'_{i,j}, 1 \leq i \leq P, 1 \leq j \leq Q$, which could be obtained by any consistent reordering, such as raster scanning. Specifically, let $X_{(i-1)Q+j} = X'_{i,j}$. The random variables X_k are represented as nodes in a graph and may take one of two values ± 1, with bit value '0' represented as $+1$ and bit value '1' represented as -1. Each node is connected to the four nearest neighbors, so that the overall graph $\mathcal{G}_0 = (\mathcal{V}_0, \mathcal{E}_0)$ satisfies 4-connectivity as shown in Figure 4.1(a). The set of nodes $\mathcal{V}_0 = \{1, 2, \ldots PQ\}$. The corresponding set of edges $\mathcal{E}_0$ contains pairs of the form $((i-1)Q+j, (i-1)Q+j+1)$, which are horizontally adjacent neighbors, or $((i-1)Q+j, iQ+j)$, which are vertically adjacent neighbors.

As described in Section 4.1, the joint probability distribution of the fingerprint can be specified by defining an energy function for the model. We use the energy function that has been commonly used for modeling binary images [9, Chapter 8]:

$$E_0(\mathbf{x}) = -\nu \sum_i x_i - \eta \sum_{(j,k)\in\mathcal{E}_0} x_j x_k. \tag{4.1}$$

This corresponds to the 2-D Ising model that has been widely used in statistical physics to model ferromagnetism arising out of interactions between individual spins. Here η controls the correlation between nodes that are connected and ν determines the marginal distribution of the individual bits. A higher value for η would increase the correlation among neighboring bits, and large ν would bias the bits to be $+1$. The joint distribution can then be written as $p_0(\mathbf{x}) = \frac{1}{Z_0}\exp(-E_0(\mathbf{x}))$, with Z_0 being the normalization constant to ensure that the distribution sums to 1.

The above model describes the fingerprint bits obtained from the original video frame. In many practical applications, fingerprints are extracted from possibly modified versions of the video and may be noisy. The noise components may be correlated and also dependent on the fingerprint bits. To accommodate such modifications, we propose a joint model for the noise bits and the fingerprint bits of the original unmodified video, which is shown in Figure 4.1(b). The filled circles represent the noise bits and the open circles represent the fingerprint bits. The solid edges capture the dependencies among the fingerprint components, while the dashed and dotted edges represent the local correlations among the noise bits. The dashed edges capture the dependence between the noise bits and the fingerprint bits. The noise may be causally dependent on the fingerprint of the original video, but the fingerprint bits of the original video should not be influenced by the noise. However, the addition of these undirected edges makes the graph symmetric with respect to the fingerprint and noise bits and does not accurately reflect the causal dependence. Factor graphs may be used to represent this dependence, and will be addressed in our future work.

We consider the case where the noise bits may be mutually dependent, but are independent of the fingerprint bits, implying that the dashed edges between the noise bits and the fingerprint bits are absent. In this example, the model for the noise bits $\mathbf{N}$ reduces to a 2-D Ising model with underlying graph $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ similar to that for the fingerprints. The energy function for a configuration $\mathbf{n}$ can be defined as:

$$E_1(\mathbf{n}) = -\alpha \sum_i n_i - \gamma \sum_{(j,k)\in\mathcal{E}_1} n_j n_k, \tag{4.2}$$

and the distribution is specified as $p_1(\mathbf{n}) = \frac{1}{Z_1}\exp(-E_1(\mathbf{n}))$. The parameters α and γ control the marginal distribution and the pairwise correlation among the noise bits, respectively.

The above MRF can be used to model block based binary video fingerprints computed on a frame by frame basis. For other fingerprinting schemes, different graphs can be used to capture the correlations among the fingerprint components.

4.2.1 Hypothesis Testing

As discussed in Section 3.1, content identification using fingerprints can be modeled as a multiple hypothesis test. Under the MRF model, since it is difficult to directly compute the error probabilities for the $(N+1)$-ary hypothesis test, we first consider a binary hypothesis test, where the detector compares the query fingerprint with a *given* fingerprint from the database.

Given a query video $\mathbf{Z}$ and a reference video $\mathbf{V}$ in its database, consider the problem where the detector has to decide whether $\mathbf{Z}$ is derived from $\mathbf{V}$ or whether the two videos are unrelated. To do so, the detector computes the fingerprints $\mathbf{y}$ and $\mathbf{x}$ from the videos $\mathbf{Z}$ and $\mathbf{V}$, respectively. The corresponding noise in the bit domain will then be given by the XOR of the two fingerprints. In the MRF model, as a logical value b is represented by $(-1)^b$, the XOR operation corresponds to an element-wise multiplication $\otimes$, so that the noise $\mathbf{n} = \mathbf{x} \otimes \mathbf{y}$. The detector then performs a binary hypothesis test with the null hypothesis H_0 that the two fingerprints are independent and the alternate hypothesis H_1 that the fingerprint $\mathbf{y}$ is a noisy version of $\mathbf{x}$:

$$\begin{aligned} H_0 &: (\mathbf{x}, \mathbf{y}) \sim p_0(\mathbf{x}) p_0(\mathbf{y}), \\ H_1 &: (\mathbf{x}, \mathbf{y}) \sim p_0(\mathbf{x}) p_1(\mathbf{n}), \end{aligned} \tag{4.3}$$

where $p_0(\cdot)$ is the distribution of the fingerprints and $p_1(\cdot)$ is the distribution of the noise.

We consider a Neyman-Pearson setting, where the detector seeks to maximize the probability of detection under the constraint that the probability of false alarm does not exceed ϵ. The optimal decision rule is to compare the log likelihood ratio (LLR) to a threshold:

$$LLR(\mathbf{x}, \mathbf{y}) = E_0(\mathbf{y}) - E_1(\mathbf{n}) \underset{H_0}{\overset{H_1}{\gtrless}} \tau, \tag{4.4}$$

where the constants have been absorbed into the threshold τ, which is chosen such that the probability of false alarm $= \epsilon$. In cases where the LLR is discrete, it may be necessary to incorporate randomization when the LLR equals the threshold.

For example, for the block-based binary fingerprinting scheme model described in Section 4.2, the LLR is given by:

$$LLR(\mathbf{x}, \mathbf{y}) = -\nu \sum_i y_i - \eta \sum_{\mathcal{E}_0} y_j y_k + \alpha \sum_i n_i + \gamma \sum_{\mathcal{E}_1} n_j n_k.$$

If the fingerprint bits are i.i.d. and equally likely to be ±1, corresponding to $\eta = \nu = 0$, and the noise bits are independent ($\gamma = 0$), the optimum decision rule reduces to a comparison of the Hamming distance between $\mathbf{x}$ and $\mathbf{y}$ to a threshold. However, when the bits are not independent, a decision rule that compares the Hamming distance to a threshold is *suboptimal*. The LLR can be interpreted as performing an implicit decorrelation by compensating the Hamming distance for the empirical correlations among the fingerprint components. We would like to quantify the accuracy using this optimal decision rule and the performance loss when the Hamming distance is used instead.

While we have adopted a Neyman-Pearson approach based on the likelihood ratio test to derive the optimal detector, another alternative may be to use a χ^2-test. When a large number of observations is available, it has been shown that under some conditions, the distribution of the log-likelihood ratio under the null hypothesis converges to a χ^2 distribution with the appropriate number of degrees of freedom [48, 100]. The threshold in the decision rule can then be chosen based on the χ^2 distribution, given the constraint on the probability of false alarm.

Under the Neyman-Pearson setting, define the probability of detection for this binary hypothesis test as $P_d^{(b)} = \Pr(LLR(\mathbf{x}, \mathbf{y}) > \tau | H_1)$ and probability of false alarm $P_f^{(b)} = \Pr(LLR(\mathbf{x}, \mathbf{y}) > \tau | H_0)$. One approach to estimate the error probabilities would be to draw samples from the MRF distribution using a traditional Markov Chain Monte Carlo (MCMC) technique and use these samples to estimate the probabilities [102]. However, a main challenge in accurately estimating the probabilities is that such error events have small probability of occurrence and are rarely observed in a typical MCMC simulation.

An alternative approach is to use the large deviations principle to obtain exponential bounds on the probability of making an error [21, Theorem 2.3.6 and 3.4.3]. Evaluating the exponent would then require the computation of a transform of the asymptotic limit of the cumulant generating function of the log likelihood ratio [21]. To the best of our knowledge, there is no closed form expression or simple technique to evaluate this rate function when the relevant distributions are Markov Random Fields, and we may have to resort to simulation-based techniques such as Markov Chain Monte Carlo as well. Instead, we take a different approach inspired by statistical physics to first estimate the so-called density of states and then utilize this information to estimate the probability of making an error.

4.2.2 Density of States

For ease of illustration, we again use the example of the binary fingerprint model described in Section 4.2. Suppose we define $M(\mathbf{x}) = \sum_i x_i$ and $E_{corr}(\mathbf{x}) = -\sum_{(j,k)\in\mathcal{E}_0} x_j x_k$, the LLR in Eqn. (4.4) can be written as $LLR(\mathbf{x}, \mathbf{y}) = -\nu M(\mathbf{y}) +$

$\eta E_{corr}(\mathbf{y}) + \alpha M(\mathbf{n}) - \gamma E_{corr}(\mathbf{n})$, since $\mathcal{E}_0 = \mathcal{E}_1$ in this model. Similarly, the energy for the fingerprint bits and the noise, $E_0(\mathbf{x})$ and $E_1(\mathbf{n})$, described in Eqns. (4.1) and (4.2) can be rewritten in terms of these parameters. Thus, the tuple

$$S(\mathbf{x}, \mathbf{y}) = (M(\mathbf{x}), E_{corr}(\mathbf{x}), M(\mathbf{y}), E_{corr}(\mathbf{y}), M(\mathbf{n}), E_{corr}(\mathbf{n})), \tag{4.5}$$

captures all necessary information regarding the configuration $(\mathbf{x}, \mathbf{y})$, and is a sufficient statistic [67] for the p_0 and p_1 distributions. Define $g(\mathbf{s}) = |\{(\mathbf{x}, \mathbf{y}) : S(\mathbf{x}, \mathbf{y}) = \mathbf{s}\}|$ as the number of configurations that have the same state $\mathbf{s}$. The function $g(\cdot)$ is referred to as the "density of states" in the physics literature and it depends only on the underlying graphical model and is independent of the parameters $(\nu, \eta, \alpha, \gamma)$ of the distributions. In some respects, the state of a vector is similar to the concept of "type" of a sequence in information theory. The main difference is that the method of types is typically used in conjunction with i.i.d. variables, whereas we consider correlated random variables in this section.

As all configurations $(\mathbf{x}, \mathbf{y})$ with the same state have the same LLR and probability of occurrence, the probability of detection $P_d^{(b)}$ can be rewritten as:

$$\begin{aligned} P_d^{(b)}(\tau) &= \sum_{(\mathbf{x},\mathbf{y})} \mathbb{1}\left(\{LLR(\mathbf{x}, \mathbf{y}) > \tau\}\right) p_0(\mathbf{x})\, p_1(\mathbf{n}) \\ &= \sum_{\mathbf{s}} g(\mathbf{s}) \mathbb{1}\left(\{LLR(\mathbf{s}) > \tau\}\right) p_{s,1}(\mathbf{s}), \end{aligned} \tag{4.6}$$

where the summation in Eqn. (4.6) is over all possible values of $\mathbf{s}$, $p_{s,1}(\mathbf{s})$ and $LLR(\mathbf{s})$ are the probability under H_1 and the LLR, respectively, of any configuration $(\mathbf{x}, \mathbf{y})$ with $S(\mathbf{x}, \mathbf{y}) = \mathbf{s}$, and $\mathbb{1}(\cdot)$ is an indicator function. Similarly,

$$P_f^{(b)}(\tau) = \sum_{\mathbf{s}} g(\mathbf{s}) \mathbb{1}\left(\{LLR(\mathbf{s}) > \tau\}\right) p_{s,0}(\mathbf{s}), \tag{4.7}$$

where $p_{s,0}(\mathbf{s})$ is the probability under H_0 of a configuration with state $\mathbf{s}$. As the LLR and the probabilities $p_1(\mathbf{n})$ and $p_0(\mathbf{x})$ depend only on $\mathbf{s}$, knowledge of $g(\mathbf{s})$ allows us to compute $P_d^{(b)}$ and $P_f^{(b)}$. We also note that the $M(\cdot)$ terms in the state vector in Eqn. (4.5) can only take $L+1$ values, whereas the $E_{corr}(\cdot)$ terms can take at most $\binom{L}{2} \sim \Theta(L^2)$ values. Thus, the number of states is a polynomial function of the number of bits and the summations in Eqns. (4.6) and (4.7) have manageable computational complexity. The problem of computing $P_d^{(b)}$ and $P_f^{(b)}$ has now been converted into one of estimating the density of states $g(\mathbf{s})$.

An algorithm to estimate the density of states was proposed by Wang and Landau in [96]. A summary of this algorithm and the key steps are provided as an appendix in Section 4.5. The main idea behind the algorithm is to construct a Markov chain that has $\frac{1}{g(\mathbf{s})}$ as its stationary distribution and use samples drawn from this distribution to estimate $g(\mathbf{s})$. An advantage of this "Wang-Landau" algorithm is that states with low probability of occurrence are also visited as often as high probability states, enabling us to estimate their probabilities accurately. We first use this algorithm to estimate the density of states $g(\mathbf{s})$ and then compute $P_d^{(b)}$ and $P_f^{(b)}$ using Eqns. (4.6) and (4.7). As the number of states is polynomial in the number of variables N, this approach is more efficient than an exhaustive evaluation of $P_f^{(b)}$ and $P_d^{(b)}$. However, the density of states estimation is limited by the amount of memory and computational resources available and cannot be used for arbitrarily large graphical models. In such situations, nested models may be used for modeling the fingerprints and will be examined in our future work.

4.2.3 Bounds on Error Probabilities for Overall Matching

Given the values of $P_d^{(b)}$ and $P_f^{(b)}$ obtained using the above technique, for completeness, we now derive bounds on the probability of correct identification for the overall matching process. We assume that the detector has no *a priori* knowledge of which video is more likely to be queried and compute the error probabilities assuming that the queries are equiprobable. If the detector has prior knowledge about the queries, these could be incorporated into the decision rule under a Bayesian setting.

Consider the probability of false alarm P_f, which can occur if the LLR of the query and *any* reference fingerprint exceeds the threshold. The false alarm probability can then be bounded by

$$P_f \leq 1 - (1 - P_f^{(b)})^N \leq N P_f^{(b)},$$

when $NP_f^{(b)} \ll 1$. Now suppose that the query video is actually a modified version of $\mathbf{V}^{(i)}$. A misclassification can occur if $LLR(\mathbf{X}^{(j)}, \mathbf{Y}) > \tau$ for any $j \neq i$. Thus, the probability of misclassification can be bounded as:

$$P_{mc} \leq 1 - (1 - P_f^{(b)})^{N-1} \leq (N-1) P_f^{(b)}.$$

An incorrect decision happens when either a misclassification occurs, or if $LLR(\mathbf{X}^{(j)}, \mathbf{Y}) < \tau$, so that

$$\begin{aligned} 1 - P_c &\leq 1 - P_d^{(b)} + P_{mc} \\ \Rightarrow P_c &\geq P_d^{(b)} - N P_f^{(b)}. \end{aligned}$$

Thus, given a desired overall probability of correct identification and false alarm, suitable values of $P_f^{(b)}$ and $P_d^{(b)}$ can be computed based on these bounds and used to choose the appropriate threshold in the binary hypothesis test. As these bounds have been derived by extending the results from the binary hypothesis test to the identification problem, the bounds may not be tight. Finding tighter bounds on the error probabilities for the identification problem when correlated fingerprints are used, is an open research problem that will be addressed in future work.

Another interesting question is how the error probabilities for correlated fingerprints scale as the number N and length L of the fingerprints increases. In the i.i.d. case, when N grows exponentially with the length as $N = 2^{LR}$, if the error exponents corresponding to the rate R as discussed in Section 3.2 are positive, the error probabilities reduce exponentially with the length L. It remains an open question whether this result would still hold for the case of correlated fingerprints, and if it does hold, how the error exponents would depend on the parameters of the MRF distribution and the resulting correlation.

4.3 Numerical Evaluation

We use the MRF model coupled with the technique for computing $P_d^{(b)}$ and $P_f^{(b)}$ described in the previous section to study the influence of correlation among the fingerprint components on the detection performance. We focus on binary fingerprinting schemes and provide numerical results for the model described in Section 4.2. We present results for the estimation of the density of states in Section 4.3.2, and compare the performance of the LLR and Hamming distance based detectors via numerical evaluations in Section 4.3.3. In Section 4.3.5, we validate the theoretical predictions by experiments using a database of images.

4.3.1 Correlation Structure among Fingerprint and Noise Components

We first analyze the correlation structure among the fingerprint components and noise under the MRF model. The correlation among the fingerprint components depends on the covariance $\mathbb{E}[\mathbf{X}\mathbf{X}^T]$. It is possible for two realizations $\mathbf{X}^{(i)}$ and $\mathbf{X}^{(j)}$ that have the same state $\mathbf{s}$ to have different outer products $\mathbf{X}^{(i)}\mathbf{X}^{(i)T}$ and $\mathbf{X}^{(j)}\mathbf{X}^{(j)T}$ depending on the exact arrangement of $+1$ and -1 within the vectors. Hence, the correlation cannot be directly obtained from the density of states. Instead, we draw 10^8 samples from the MRF distribution using the Metropolis-Hastings algorithm [32] by retaining only 1 out of 100 iterations to reduce the effect of correlations between successive iterates in the MCMC simulations. We then use these samples to estimate the correlation among the components. As the states with small probability do not significantly alter the correlation, this approach gives us an accurate estimate of the correlation.

We stack the fingerprint and noise samples into a single vector $[\mathbf{X}^T \ \mathbf{N}^T]^T$ and compute the correlation coefficient among its elements. Figure 4.2 shows the correlation coefficient among the fingerprint bits and noise for a 4×4 model, obtained by setting $\nu = 0, \eta = 0.3$, $\alpha = 0.3$, and $\gamma = 0.1$. For ease of visualization, Figure 4.3(a) shows the correlation between the $(1,1)^{\text{th}}$ bit (top left corner) and every other bit while Figure 4.3(b) and (c) show the same for the $(2,1)^{\text{th}}$ bit and the $(2,2)^{\text{th}}$ bit,

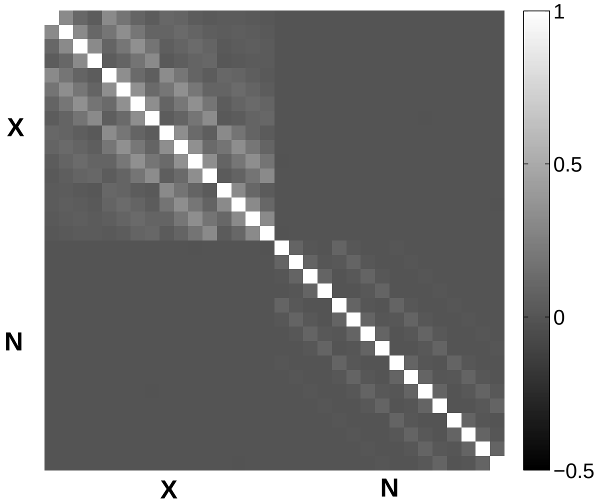
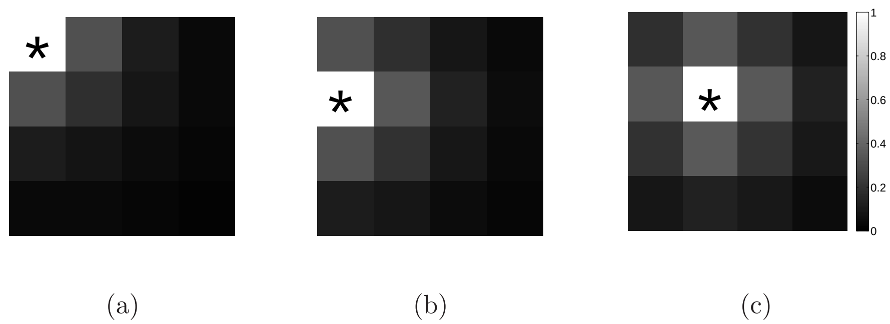


Figure 4.2: Correlation among the fingerprint and noise components under the MRF model for $\nu = 0, \eta = 0.3$, $\alpha = 0.3$, and $\gamma = 0.1$.

respectively. Due to symmetry, other bits in corresponding positions have similar correlations. We observe that the average correlation between each fingerprint bit and its nearest neighbor $\rho_x \approx 0.3$ and the correlation decays with distance. This is the typical correlation behavior observed in our model and reflects the correlation expected in practice - bits extracted from adjacent blocks are expected to be more correlated than bits extracted from blocks farther apart. The correlation among the noise bits has a similar structure, as the noise model is similar, while the noise bits are uncorrelated with the fingerprint bits.

4.3.2 Density of States Estimation

We evaluate the accuracy of the density of states estimation algorithm using known exact results for the density of energy states $g_I(E)$ for the 2-D Ising model



(a) (b) (c)

Figure 4.3: Typical correlation structure among the various fingerprinting bits. Correlation coefficients for the (a) $(1,1)^{\text{th}}$ bit, (b) $(2,1)^{\text{th}}$ bit, (c) $(2,2)^{\text{th}}$ bit and the remaining bits. The '*' denotes the bit under consideration.

with periodic boundary conditions [7]. To enable comparison, periodic boundary conditions are imposed on the graph $\mathcal{G}_0$ - the nodes $X'_{1,j}$ in the top row are connected to the corresponding nodes $X'_{M,j}$ in the bottom row, and the nodes in the first column are similarly connected to the nodes in the last column, so that every node is 4-connected. 4-connectivity is similarly achieved for the noise nodes $\mathbf{N}$. We then use the Wang-Landau algorithm to estimate the density of states $g(\mathbf{s}) = g(m_x, e_x, m_y, e_y, m_n, e_n)$ by performing a random walk in the 6-D parameter space [96] and use the obtained $g(\mathbf{s})$ to estimate the density of energy states $g_I(E)$ by summing over all other variables and normalization:

$$g_I(E) = \frac{1}{2^{PQ}} \sum_{(m_x, m_y, e_y, m_n, e_n)} g(m_x, E, m_y, e_y, m_n, e_n).$$

In our simulations, we use the parameters suggested in [96] and the maximum number of iterations is capped at 10^{10}.

We measure the accuracy of estimation by computing the relative error $\varepsilon(g_I(E))$

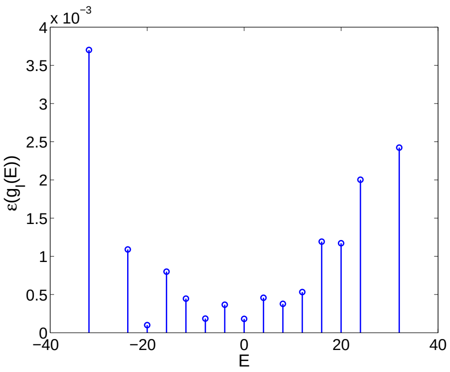


Figure 4.4: Relative error in the estimation of density of states for a 4x4 Ising model with periodic boundary conditions.

in the estimate of the density of states, defined as $\varepsilon(x) = \frac{|x - x_{est}|}{x}$. Figure 4.4 shows the relative error in the estimation of the density of states for a 2-D Ising model of size 4×4 with periodic boundary conditions. We observe from the figure that the maximum relative error is approximately 0.37%, and the mean relative error is 0.1%. These results demonstrate that accurate estimates of the density of states can be obtained using the Wang-Landau algorithm.

4.3.3 Detection Accuracy of Correlated Fingerprints

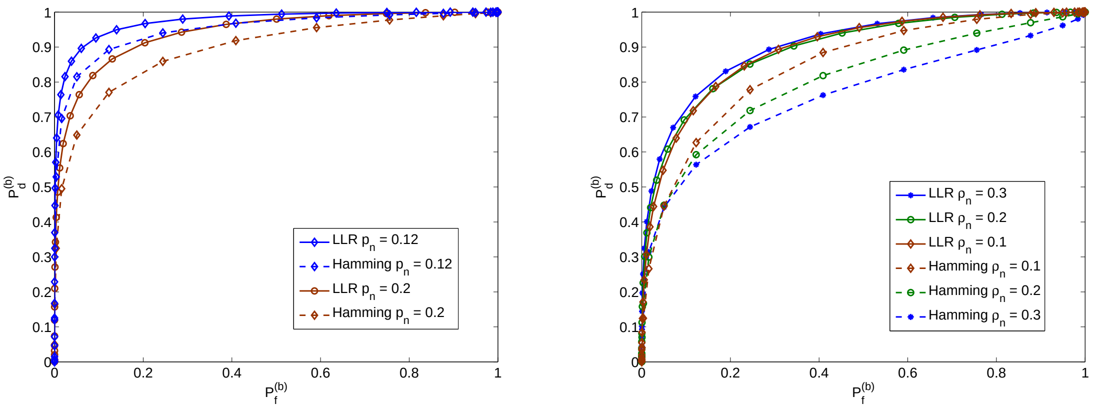
To examine the performance of correlated fingerprints, we use the model without periodic boundary conditions, as practical fingerprints are not expected to have such periodic relationships. The nodes at the corners are only connected to their two closest neighbors, the remaining nodes at the borders are connected to their three closest neighbors, and all the other nodes are 4-connected.

Using the estimated density of states, we compute the probabilities $P_d^{(b)}$ and

$P_f^{(b)}$ as described in Section 4.2.2 and study the effect of different parameters on the detection performance. Although errors in the estimation of the density of states will also affect the accuracy of the estimates of $P_d^{(b)}$ and $P_f^{(b)}$, as shown in Section 4.3.2, these errors are small, and the accuracy can be improved by obtaining a better estimate of the density of states.

We first examine the effect of the noise on the detection accuracy in Figure 4.5. We characterize the noise by the probability p_n of a noise bit being '-1' which is the equivalent of a binary '1' bit, and the average correlation among adjacent noise bits ρ_n, which are estimated from the MCMC trials. Figure 4.5(a) shows the ROC curves for a fingerprint of size 4×4 bits with correlation $\rho_x = 0.2$ under two different p_n and fixed $\rho_n = 0.2$, for detection using the LLR statistic and the Hamming distance statistic. We observe that for a given noise level, the LLR statistic gives $5 - 10\%$ higher $P_d^{(b)}$ at a given $P_f^{(b)}$ compared to the Hamming distance detector. As expected, the performance for any given detector is worse when there is a higher probability of the noise changing the fingerprint bits.

Fig 4.5(b) shows the influence of the noise correlation on the detection performance. The figure indicates that for a fixed correlation among the fingerprint bits $\rho_x = 0.2$ and a fixed marginal probability of the noise bits $p_n = 0.3$, detection using the LLR statistic is not significantly affected by the noise correlation. This is due to the fact that the LLR takes into account the correlation among the noise bits. On the other hand, using the Hamming distance leads to some degradation in the performance as the correlation increases. This can be explained by the fact that as the noise correlation increases, noise vectors with large Hamming weights become



(a) (b)

Figure 4.5: Influence of the noise distribution on the detection: (a) Impact of the noise marginal distribution p_n on the detection performance, $\rho_x = 0.2$ and $\rho_n = 0.2$, and (b) Impact of noise correlation ρ_n at fixed $p_n = 0.3$ and $\rho_x = 0.2$.

more probable, leading to higher missed detections.

Next, we examine the influence of the correlation among the fingerprint bits on the detection accuracy. Figure 4.6 shows the ROC curves for content identification using fingerprints of size 4×4 for different correlations, where the noise parameters $p_n = \rho_n = 0.2$. We again observe that detection using the LLR statistic, which compensates for the correlation among the fingerprint bits, is not significantly affected by the correlation. For the Hamming distance statistic, there is an increase in false alarms at a given $P_d^{(b)}$ as the correlation among the fingerprints increases, since similar configurations with smaller distances become more probable.

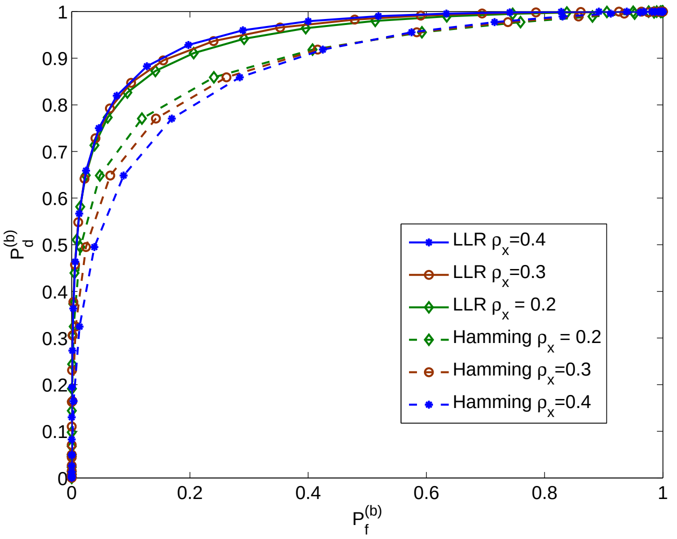
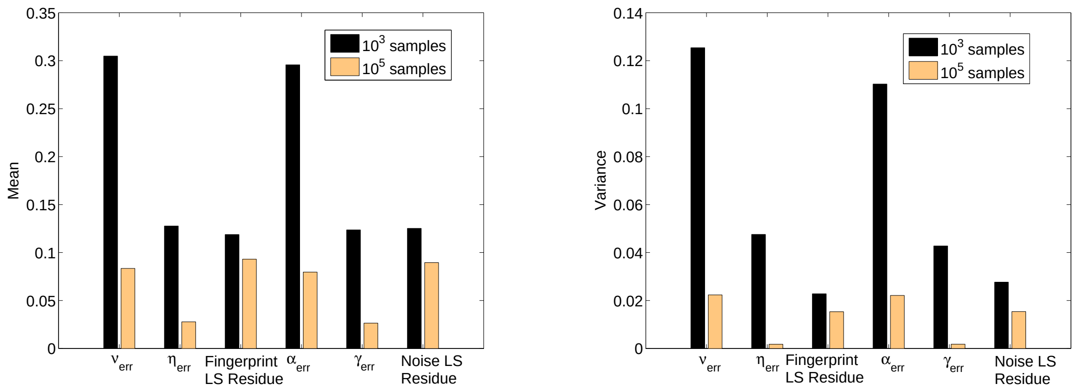


Figure 4.6: Influence of correlation of the fingerprint bits on the detection performance ($p_n = \rho_n = 0.2$).

4.3.4 LS Estimation of Parameters

Having examined the performance of the LLR and Hamming distance detectors for the MRF model through numerical simulations, we next validate our theoretical predictions through experiments on image databases. To predict the detection accuracy using a given database, we first need to obtain the parameters for the MRF model that best captures the distribution of the data. We use the Least Squares (LS) technique for estimating the parameters of the underlying MRF proposed in [23], which is equivalent to the coding method proposed in [8]. A summary of the estimation procedure is provided as appendix in Section 4.6. In this subsection, we present results on the accuracy of the LS parameter estimation.

To evaluate the accuracy of the LS parameter estimation technique, we independently choose parameters $(\nu, \eta, \alpha, \gamma)$ uniformly distributed in the interval $(0, 1)$



(a) (b)

Figure 4.7: Accuracy of the LS estimation of the MRF parameters: (a) Mean and (b) Variance of the estimation error over 1000 trials.

and generate samples from the MRF distribution with these parameters using the Metropolis-Hastings algorithm. These samples are then used to obtain estimates $(\hat{\nu}, \hat{\eta}, \hat{\alpha}, \hat{\gamma})$ of the parameters of the MRF distribution, and the error in estimating the parameters is determined. For example, $\nu_{err} = |\nu - \hat{\nu}|$, and $\eta_{err}, \alpha_{err}, \gamma_{err}$ are similarly defined. The residues in the LS estimation of the parameters of the fingerprint and noise distributions are also noted. Figure 4.7 shows the mean and variance of the error in estimating the parameters over 1000 trials. From the figure, we observe that the LS estimates for the η and γ parameters have approximately $3-4$ times less error on average, compared to the estimates for ν and α. When a larger number of samples is available, the estimation accuracy is significantly better for all the parameters. Further, we observe that the estimation accuracies of η and γ, and ν and α are similar, since these parameters play similar roles in the fingerprint and noise distributions.

4.3.5 Simulation Results using Image Database

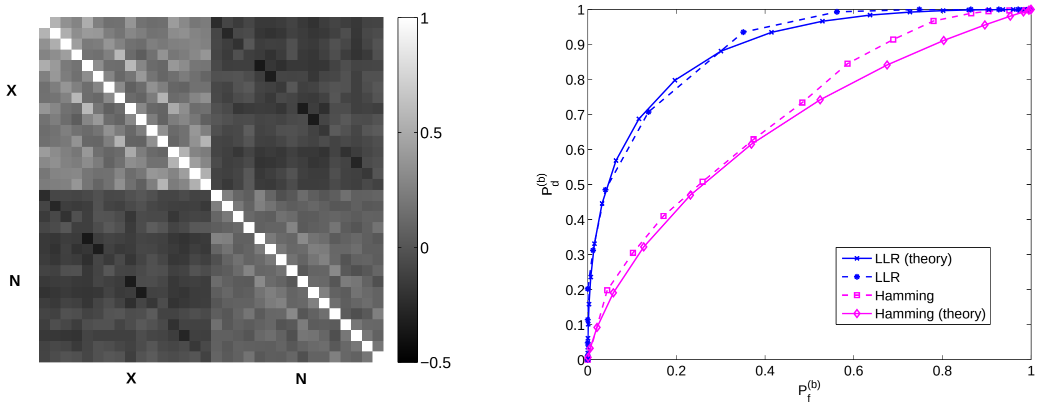
In this subsection, we compare the performance predicted by the theoretical analysis with simulation results obtained using two image databases. The first database consists of 1000 images downloaded from the Flickr photo hosting service by searching for the tag "panda"[1], and the second database is the Uncompressed Colour Image Database (UCID) [71]. For extracting the fingerprints, each image is divided into 16 blocks in a 4×4 grid and the average luminance within each block is computed. The average luminance is then quantized to one bit accuracy according to whether it is greater than or lesser than the grayscale value of 128, giving a 16-bit fingerprint for each image. To evaluate the appropriateness of our model, we require a distortion that results in the noise in different image blocks being correlated and significantly alters the fingerprint bits. We choose histogram equalization as the distortion and apply it to the luminance portion of the image and compute the noisy versions of the fingerprints. The hypothesis test described in Sec. 4.2.1 is then performed using the noisy fingerprints. Additionally, 1000 pairs of original fingerprints are randomly chosen and compared to each other to obtain an estimate of the false alarm probability. We also estimate the Ising model parameters for the fingerprints and the noise using the least squares method discussed in 4.3.4, and obtain the theoretical predictions for the ROC curves as described in Section 4.2.2.

Figs. 4.8 and 4.9 show the results obtained for the panda database and the UCID images, respectively. Figure 4.8(a) displays the correlation among the finger-

[1] Links to the images used in the experiments are available at www.ece.umd.edu/~varna/panda_database/

print and noise components for the fingerprints computed from the panda database. We observe that the correlation is similar to the correlation structure for the MRF model in Figure 4.2. The residue for the LS estimate of the hash parameters is 0.067 and for the noise parameters is 0.06 indicating that the MRF model is a good fit for this data. Figure 4.8(b) compares the ROC curves obtained from theory and simulation for the LLR detector and the Hamming distance based detector. As the data obtained from real images may not exactly follow the Ising model, we observe that there are some differences between the theoretical predictions and simulation results. For the LLR detector, the theory and simulation results agree well, but for the Hamming distance based detector and $P_f > 0.5$ region, there is some gap between the two curves due to the model mismatch.

For the UCID database, we find that the residues in the LS estimate for the hash and noise parameters are 0.24 and 0.253 respectively, which is around 4 times that for the panda database. Thus, we expect that the Ising model may not be a good fit for the fingerprints obtained from this particular database. Figure 4.9(b) shows that the theoretical predictions and the simulation results for the LLR detector have a similar trend. The Hamming distance based detector performs much better in the simulations compared to the theoretical predictions, and there is a large gap. From these results, we see that while the model mismatch can affect the accuracy of the theoretical predictions compared to the simulation results, the LLR detector derived through this analysis can improve the detection accuracy by approximately 5-20% over the simple Hamming distance based detector when the fingerprints and noise are correlated.

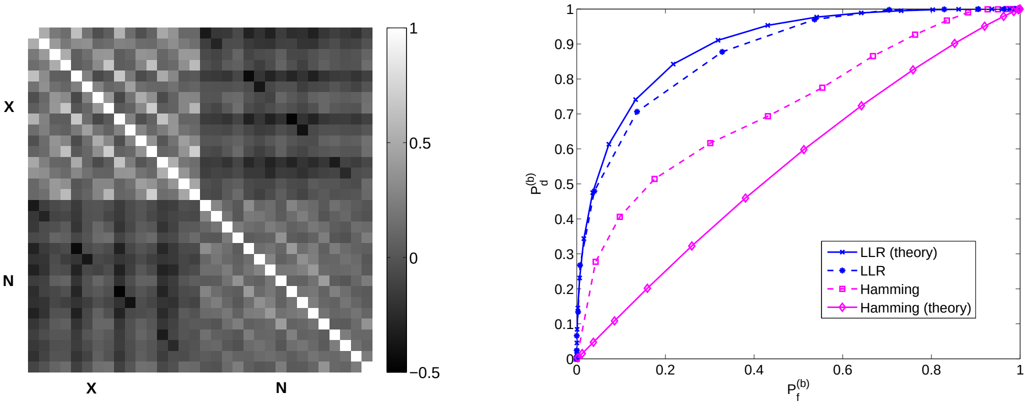


(a)

(b)

Figure 4.8: (a) Correlation among fingerprint and noise components, and (b) Comparison of theoretical and simulation results for a database consisting of 1000 images obtained from Flickr.

In general, the distribution of the fingerprints depends on the fingerprinting algorithm and the images in the database. Based on the particular algorithm, a suitable graphical model may be used for modeling the fingerprint and noise distribution. Given such a graphical model, the LLR detector can be derived and used for the fingerprint matching. A suitable set of parameters can be defined as the state and the Wang-Landau algorithm can be used to estimate the density of states. The density of states estimate can then be used to predict the detection accuracy that can be achieved using the fingerprinting scheme under this model.



(a)

(b)

Figure 4.9: (a) Correlation among fingerprint and noise components, and (b) Comparison of theoretical and simulation results for UCID images.

4.4 Chapter Summary

This chapter focuses on modeling correlated binary fingerprints and analyzing their performance under a hypothesis testing framework. We proposed a Markov Random Field model for the fingerprint and noise bits. Under this model, we examined fingerprint matching as a hypothesis testing problem and proposed a statistical physics inspired approach to compute the probabilities of detection and false alarm. Our analysis showed that Hamming distance based detection, which is commonly employed in many applications, is suboptimal in this setting and is susceptible to correlations among the fingerprint bits or the noise. The optimal log-likelihood ratio detector provides $5 - 20\%$ higher detection probability and the performance is relatively stable for different correlations among the fingerprint and noise components. Simulation results using image databases corroborate our theoretical analysis.

4.5 Appendix: Wang-Landau Sampling

In statistical physics, the density of states is an important quantity that enables the computation and characterization of various thermodynamic properties of a physical system. Given a system which may exist in different configurations, the density of states is defined as the number of configurations that have the same energy. This quantity is independent of the thermodynamic temperature and once determined, can be used to compute the properties of the system at any temperature. Traditional MCMC methods, which are used to estimate thermodynamic properties such as free energy, sample from the distribution over the set of configurations, and may not visit states with low probability often enough to allow for accurate estimation of the density of states. To address this problem, Wang and Landau [96] proposed a technique for estimating the density of states by performing a random walk in the energy space. We illustrate the algorithm using an example of a physical system with spins that can take values ±1.

Suppose we have a system with L spins $X_i \in \{-1, +1\}, i = 1, 2, \dots L$. Given a particular configuration of the spins $\mathbf{X} = [X_1\ X_2\ \dots\ X_L] = \mathbf{x} \in \{-1, +1\}^L$, let the energy of the system be given by $E_x(\mathbf{x})$. We are interested in determining the density of states $g(\cdot)$ defined as:

$$g(E) = |\{\mathbf{x} \in \{-1, +1\}^L : E_x(\mathbf{x}) = E\}|$$

where $|\cdot|$ denotes the cardinality of a set. The main idea behind the Wang-Landau algorithm is to construct a Markov Chain with stationary distribution proportional to $\frac{1}{g(E)}$. Samples $\mathbf{X}(1), \mathbf{X}(2), \dots, \mathbf{X}(t), \dots$ are then drawn from this Markov Chain and used to estimate $g(E)$.

The Markov Chain is constructed as follows. The initial value of the spins $\mathbf{X}(0)$ is chosen randomly from $\{-1, +1\}^L$ and an initial value is chosen for the density of states, e.g. $g(E) = 1, \forall E$. The number of times that a particular energy value has been encountered in the simulation is initialized to zero, i.e. $\text{count}(E) = 0, \forall E$, and the update factor for the density of states estimate χ is set to a moderately large value, e.g. $\chi = \exp(1)$. At the tth iteration, a value $J(t)$ uniformly distributed on $\{1, 2, \ldots, L\}$ is chosen and the sign of $X_{J(t)}$ is flipped with a certain probability q_t as described next.

Let $\text{flip}_j(\mathbf{x})$ be the function that flips the sign of the jth element of $\mathbf{x}$, i.e. $\text{flip}_j(\mathbf{x}) = [x_1\ x_2\ \ldots\ -x_j\ \ldots\ x_L]$. Define

$$\kappa_t = \frac{g(E_x(\mathbf{X}(t-1)))}{g(E_x(\text{flip}_{J(t)}(\mathbf{X}(t-1))))} \tag{4.8}$$

which is the ratio of (the current estimates of) the density of states for the energy of the current state $\mathbf{X}(t-1)$ and the energy of the state that would result if the $J(t)$th spin was flipped. Now let $q_t = \min(\kappa_t, 1)$ and

$$\mathbf{X}(t) = \begin{cases} \text{flip}_{J(t)}(\mathbf{X}(t-1)) & \text{with probability } q_t \\ \mathbf{X}(t-1) & \text{with probability } 1 - q_t. \end{cases}$$

It can be verified [96] that the samples $\mathbf{X}(t)$ thus obtained form a Markov Chain with stationary distribution proportional to $\frac{1}{g(E)}$. The count of the energies encountered and the estimate of the density of states are updated as:

$$\begin{aligned} \text{count}(E_x(\mathbf{X}(t))) &\leftarrow \text{count}(E_x(\mathbf{X}(t))) + 1 \\ g(E_x(\mathbf{X}(t))) &\leftarrow g(E_x(\mathbf{X}(t))) \times \chi \end{aligned}$$

Algorithm 1: Wang-Landau density of states estimation

```
Initialize: $X_i = \pm 1$ randomly, $i = 1, 2, \ldots, L$; $\chi = \exp(1)$.
$g(E) = 1, \text{count}(E) = 0, \forall E$.
while $\chi < \exp(10^{-8})$ do
    Choose $J$ uniformly at random from $\{1, 2, \ldots, L\}$
    $\kappa = \frac{g(E_x(\mathbf{X}))}{g(E_x(\text{flip}_J(\mathbf{X})))}$
    $q = \min(\kappa, 1)$
    Set $X_J \leftarrow \begin{cases} -X_J & \text{with probability } q \\ X_J & \text{with probability } 1 - q \end{cases}$
    $\text{count}(E_x(\mathbf{X})) \leftarrow \text{count}(E_x(\mathbf{X})) + 1$
    $g(E_x(\mathbf{X})) \leftarrow g(E_x(\mathbf{X})) \times \chi$
    if $\min_E \text{count}(E) > \mu \times \text{avg}(\text{count}(E))$ then
        $\text{count}(E) = 0, \forall E$
        $\chi \leftarrow \sqrt{\chi}$
    end
end
Normalize the density of states $g(E)$.
```

where χ is the update factor.

When all energy values have been encountered equally often, the estimate of the density of states has an accuracy proportional to $\ln(\chi)$. If a better accuracy is desired, the update factor χ is reduced, the counts are reset to zero, and the iterations are continued. Initially, the update factor χ is chosen to be large enough so that all the energy levels are visited quickly, and then it is progressively reduced to obtain a finer estimate. This process is continued until χ becomes small enough, e.g. $\chi < \exp(10^{-8})$. The $g(E)$ obtained after convergence is relative and is normalized to obtain an estimate of the density of states. The algorithm is summarized in Algorithm 1, where we have dropped the dependence of the variables on the iteration number t for notational convenience and presented in-place operations instead.

The above algorithm was described in the context of estimating the density of *energy* states. In some applications, multiple parameters of the system may be of interest, e.g. the energy $E_x(\mathbf{x})$ and the magnetization $M_x(\mathbf{x}) = \sum_i x_i$. The state may then be defined to consist of these multiple parameters: $\mathbf{s} = S(\mathbf{x}) = (E_x(\mathbf{x}), M_x(\mathbf{x}))$. The corresponding density of states $g(\mathbf{s})$ can be estimated by performing the random walk in the appropriate 2-D state space. The quantity κ_t would then be replaced by:

$$\kappa'_t = \frac{g(S(\mathbf{X}(t-1)))}{g(S(\text{flip}_{J(t)}(\mathbf{X}(t-1))))}.$$

For large systems, the parameter space can be divided into several regions and independent random walks can be performed over each of these regions for faster convergence. The overall density of states can then be reconstructed from these individual estimates by ensuring continuity at the boundaries. For further details regarding the algorithm, please see [96].

4.6 Appendix: Least Square Estimation of MRF Parameters

For the MRF model, as the partition function Z depends on the parameters in a complicated manner, direct Maximum Likelihood estimation of the parameters given sample data is typically difficult. Instead, various techniques such as pseudo-likelihood [8] and Least Square (LS) based estimation [23] are often used to estimate the parameters. For the Ising model, both these techniques turn out to be equivalent. In this appendix, we briefly summarize the Least Square(LS) estimation of the MRF model parameters described in [23], which we use in our experiments.

Consider a particular node X_i in the MRF and denote the set of neighbors of X_i by $\mathcal{N}_i$. Due to the Markov property of the MRF, the conditional distribution of X_i given all the remaining nodes depends only on the values of its neighbors. For the specific energy function defined in Eqn. (4.1), we have:

$$\Pr(X_i = x_i |\, \mathbf{X}_{\mathcal{N}_i}) \propto \exp\left(\nu x_i + \eta x_i \sum_{j \in \mathcal{N}_i} x_j\right)$$

so that

$$\frac{1}{2} \ln \frac{\Pr(X_i = +1 |\, \mathbf{X}_{\mathcal{N}_i})}{\Pr(X_i = -1 |\, \mathbf{X}_{\mathcal{N}_i})} = \nu + \eta \sum_{j \in \mathcal{N}_i} x_j. \tag{4.9}$$

The quantity on the left hand side of Eqn. (4.9) may be estimated from the samples of the MRF distribution by counting the number of occurrences of $X_i = +1$ and -1 for different values of the neighbors $\mathbf{X}_{\mathcal{N}_i}$. This yields a set of equations in ν and η that can be solved using the least squares technique to obtain an estimate of the parameters of the MRF model. The parameters α and γ for the noise can be estimated in a similar manner from the training data.

Chapter 5

Modeling Temporal Correlations in Content Fingerprints

In the previous two chapters, we have developed models for binary fingerprints and analyzed their performance. We first examined i.i.d. fingerprints with equally likely bits in Chapter 3. As practical algorithms generate fingerprints with correlated components, we proposed an MRF model for binary fingerprints in Chapter 4 that can capture these correlations. The MRF model was mainly described in the context of modeling the components of the fingerprint of a single frame, which can be considered to be spatial correlations. Practical fingerprints also exhibit correlations in the temporal direction. Fig. 5.1 shows the 512 successive fingerprints for a 100s long video sequence obtained using the algorithm described in [64]. Each column in the image represents the 32 bit fingerprint corresponding to one frame, with a white pixel representing the bit value '1' and a black value indicating '0'. From the figure,

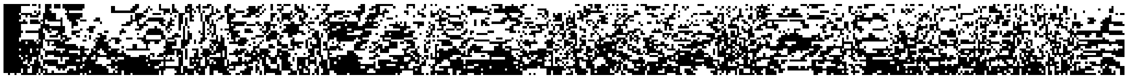


Figure 5.1: Example of temporal correlations among fingerprints from 512 successive frames for a 100s video using the algorithm in [64].

we observe that fingerprint values change slowly over time and exhibit correlations.

The MRF model proposed in the previous chapter can be used to model such correlations. As the number of states grows polynomially with the number of nodes in the model, the associated computational approach for estimating the detection accuracy may not be scale very well if the number of nodes becomes very large. To address this problem, in this chapter we examine various models for capturing such temporal correlations. Using detectors derived based on these models, we improve the accuracy of identifying matching content.

We model the temporal relations between fingerprints using a Markov chain in Section 5.1, and evaluate the suitability of this model using a database of videos in Section 5.2. As the experimental results indicate that the MC model is suitable only in a certain regime, we examine hybrid models for the fingerprints and corresponding detectors in Section 5.3.

5.1 Markov Chain based model for temporal correlations

In practical applications involving large databases, to reduce the complexity of the matching process, a coarse search of the database is first performed using the query fingerprint to identify likely matches. Approximate search techniques such as

Locality Sensitive Hashing [29] are typically used for this purpose. Once this reduced set of candidates is obtained, a more detailed matching is carried out to identify the most likely match. In this chapter, we focus only on the finer matching and describe a model for the matching process that explicitly accounts for correlations among fingerprint components.

In many practical schemes, fingerprints for a long multimedia sequence are obtained by concatenating the sub-fingerprints obtained from shorter sub-sequences [64]. We will refer to such a unit from which one sub-fingerprint is computed as a "frame". In some video fingerprinting schemes, this abstract frame may correspond to a single physical frame, whereas in others, it may correspond to a group of frames. Let $\mathbf{y}(j)$ represent the sub-fingerprint of the jth frame of the query and $\mathbf{Y} = [\mathbf{y}(1)\ \mathbf{y}(2)\ \ldots\ \mathbf{y}(L)]$ denote the overall fingerprint of the query. Similarly, let $\mathbf{X} = [\mathbf{x}(1)\ \mathbf{x}(2)\ \ldots\ \mathbf{x}(L)]$ be the fingerprint of a candidate video in the database.

We model the fingerprint matching as a hypothesis test [88] and consider the binary hypothesis test to determine whether the query fingerprint $\mathbf{Y}$ matches with fingerprint $\mathbf{X}$. The null hypothesis H_0 corresponds to the case where $\mathbf{X}$ and $\mathbf{Y}$ do not match, whereas the alternative hypothesis H_1 corresponds to the case where $\mathbf{X}$ is a match for $\mathbf{Y}$. The overall matching procedure may be considered as a sequence of such binary hypothesis tests whose results are combined to obtain the final decision. To characterize this hypothesis test, we require the joint distribution $p'_i(\mathbf{X}, \mathbf{Y}) = q_i(\mathbf{Y}|\mathbf{X})\ q'(\mathbf{X}), i = 0, 1$ under the two hypotheses. Here $q_i(\mathbf{Y}|\mathbf{X})$ represents the conditional distribution of the query $\mathbf{Y}$ given the reference $\mathbf{X}$ under H_i and $q'(\mathbf{X})$ is the marginal distribution of $\mathbf{X}$. However, as $\mathbf{X}$ and $\mathbf{Y}$ have high dimension, obtaining the joint distribution is not feasible in practice and alternative approaches are needed.

We first observe that the Likelihood Ratio (LR) for the hypothesis test, given by

$$LR'(\mathbf{X},\mathbf{Y}) = \frac{p'_1(\mathbf{X},\mathbf{Y})}{p'_0(\mathbf{X},\mathbf{Y})} = \frac{q_1(\mathbf{Y}|\mathbf{X})}{q_0(\mathbf{Y}|\mathbf{X})},$$

only depends on the conditional distributions q'_0, q'_1 of $\mathbf{Y}$ given $\mathbf{X}$ under the two hypotheses. Obtaining this conditional distribution still suffers from the problem of high dimensionality. To allow for practical modeling, we assume that the conditional distribution only depends on the *distance* between the fingerprints. This assumption is motivated by the use of simple distance based matching in practical applications and its good performance.

Let $\mathbf{d}(\mathbf{X},\mathbf{Y}) = [d(1)\ d(2)\ \ldots\ d(L)]^T$, be the vector of distances between the fingerprints $\mathbf{X}$ and $\mathbf{Y}$, where $d(j) = d(\mathbf{x}(j),\mathbf{y}(j))$ is the distance between the jth sub-fingerprint of the query and the reference obtained using a suitable distance metric. To capture the temporal correlations in the video frames that are reflected in the fingerprints, we model the sequence of distances $\{d(j)\}$ as following a Markov chain distribution with transition probability matrix $\mathbf{P}_i$ under hypothesis H_i [92]:

$$\begin{aligned} H_0 &: \ \mathbf{d} \sim \pi_0(d(1)) \prod_{j=2}^{L} P_0(d(j-1), d(j)) \\ H_1 &: \ \mathbf{d} \sim \pi_1(d(1)) \prod_{j=2}^{L} P_1(d(j-1), d(j)), \end{aligned}$$

where $P_i(k,l) = \Pr(d(j) = l \mid d(j-1) = k, H_i)$ represents the probability of transitioning from state k to state l for the Markov chain and π_i is the corresponding stationary distribution. The likelihood ratio test is given by:

$$LR^{MC}(\mathbf{X},\mathbf{Y}) = \frac{\pi_1(d(1))}{\pi_0(d(1))} \prod_{j=2}^{L} \frac{P_1(d(j-1),d(j))}{P_0(d(j-1),d(j))} \underset{H_0}{\overset{H_1}{\gtrless}} \tau^{MC}, \tag{5.1}$$

where τ is an appropriately chosen threshold. The above Markov chain based model may be applied to any fingerprinting scheme with the associated distance metric.

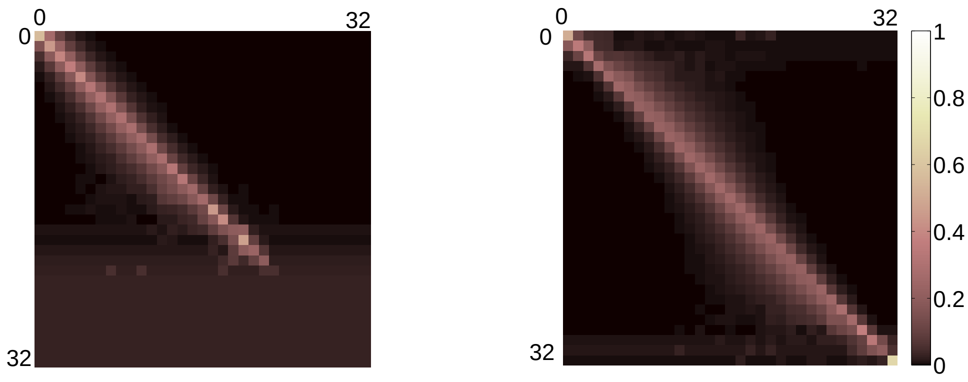
Another commonly used decision rule is to compare the average distance to a threshold:

$$\bar{d} = \frac{1}{L}\sum_{j=1}^{L} d(j) \underset{H_1}{\overset{H_0}{\gtrless}} \tau^{d}. \tag{5.2}$$

If all the fingerprint bits are i.i.d. and equally likely to be '0' or '1', and the noise operates in a similar manner, then the Hamming distance based detector is optimal, as shown in Section 3.2. In practice, this detector is usually preferred due to its low computational complexity and ease of implementation. In the next section, we compare the detection accuracy of these detectors using a practical fingerprinting scheme and a video database.

5.2 Experimental Results

In our experiments, we use the video fingerprinting scheme proposed in [64]. The frame rate of the video is normalized by downsampling to 5 fps and a 32-bit fingerprint is computed for every frame of the video. The MUSCLE-VCD-2007 database [47] is used for estimating the detection accuracy. The database consists of 101 videos with a total duration of approximately 100 hours. Half the videos are randomly selected for training and the remaining are used for testing. We use 10



(a) $\mathbf{P}_1$

(b) $\mathbf{P}_0$

Figure 5.2: Comparison of the transition matrices under the two hypotheses.

fold cross-validation [43] and average the identification results obtained from each of these individual runs.

5.2.1 Estimating the transition matrices

Histogram equalization is used as an example of the processing that may be encountered in a practical system. Other distortions can be treated similarly. The observed distances between the fingerprints of the distorted and reference videos are used to estimate the transition probability matrix $\mathbf{P}_1$ under hypothesis H_1. Similarly, the distances between the fingerprints of every pair of videos in the training set is used to estimate $\mathbf{P}_0$.

Figure 5.2 compares the transition matrices obtained when the first 50 movies are used for training. These transition matrices reveal the correlated nature of the distances between the sub-fingerprints of adjacent frames. The strong diagonal component in both the transition matrices indicates that the value of $d(j)$ is very

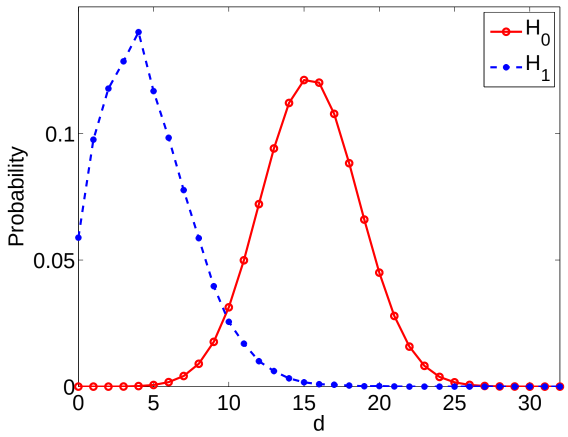
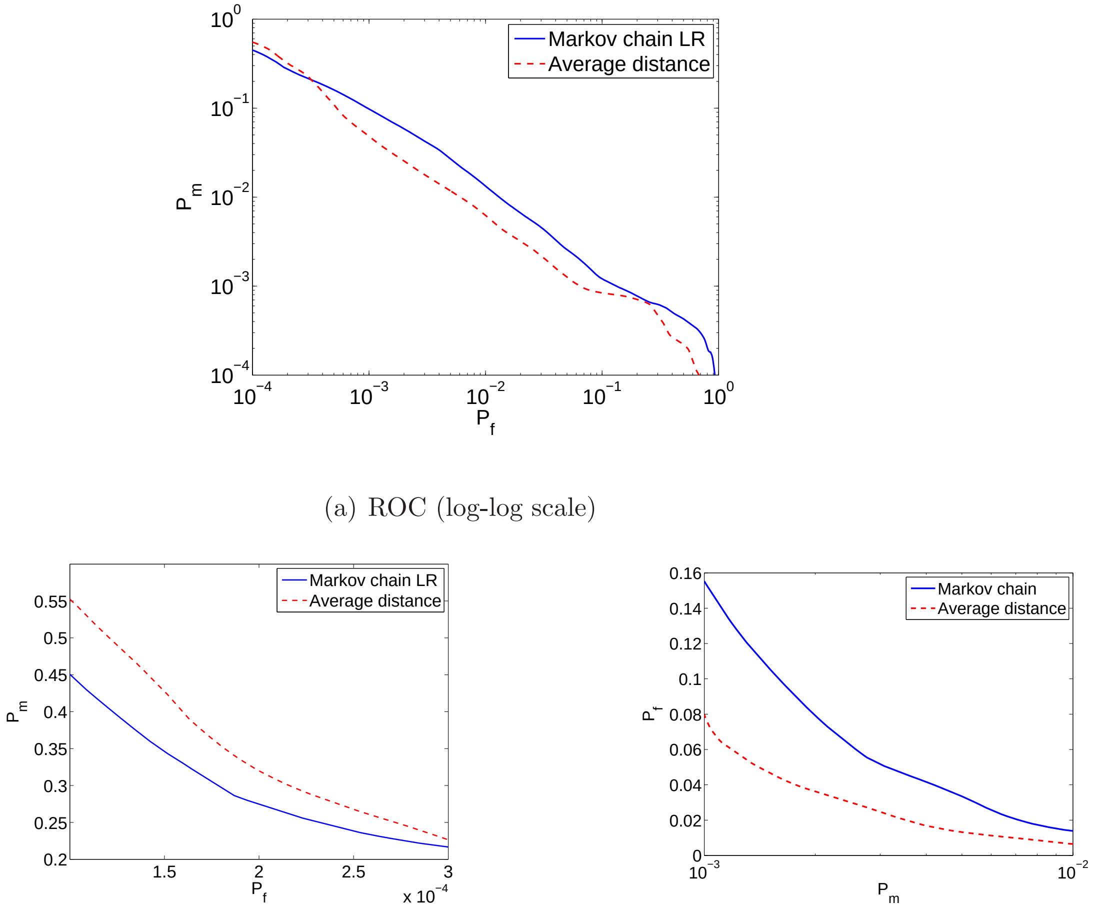


Figure 5.3: Stationary distribution of the distances under the two hypotheses.

likely to be close to $d(j-1)$. As large distances are not observed frequently under H_1, the probability of $d(j)$ given that $d(j-1)$ is a large value is approximately uniform, as seen by the last several rows of $\mathbf{P}_1$ in Figure 5.2(a). On the other hand, under H_0, large values are observed frequently enough to obtain an accurate estimate of the transition probability, as depicted in Figure 5.2(b).

Figure 5.3 shows the stationary distributions under the two hypotheses. From the figure, we see that under the null hypothesis H_0, the distribution resembles a binomial distribution centered at 16. Under H_1, the distances are clustered around $4-5$ and the distribution is non-binomial. This indicates that the noise within a given frame is possibly correlated. This is to be expected, since histogram equalization introduces correlated noise.



(a) ROC (log-log scale)

(b) Low P_f region

(c) Low P_m region

Figure 5.4: Comparison of the detection performance of the Markov Chain LLR detector and the average Hamming distance detector for query video size of 100 frames (a) Complete ROC curve in log-log scale (b) Low P_f region and (c) Low P_m region.

5.2.2 Detection Accuracy

We use the 50 videos in the testing set to compare the detection accuracy of the likelihood ratio and the average distance based detectors. Each video is distorted by applying histogram equalization and divided into distinct queries of 20s each with

100 sub-fingerprints. These are compared with the original undistorted fingerprints to estimate the probability of missed detection P_m. To estimate the probability of false alarm P_f, each video in the testing set is divided into queries of 100 sub-fingerprints each and compared with the fingerprints of every other video in the database. The P_m and P_f values obtained are averaged over the different iterations of the k-fold cross-validation.

Figure 5.4 compares the Receiver Operating Characteristic (ROC) curves for the Markov chain likelihood ratio and the average distance based detectors. Figure 5.4(a) shows P_m as a function of P_f for both detectors on a logarithmic scale. From the figure, we observe that the likelihood ratio based detector has lower P_m in the low P_f region, whereas the average distance based detector performs better in the low P_m region. Upon closer examination of the low P_f region, as shown in Figure 5.4(b), we observe that the LR detector has approximately $5 - 10\%$ lower P_m than the average distance based detector. In the low P_m region shown in Figure 5.4(c), the average distance based detector has a similar improvement.

5.3 Mixture and Adaptive Models

In the previous section, we have seen that the Markov chain is a better model for the distances in the low P_f regime, which corresponds to the case when the average distance is small. In the low P_m regime, an independent model for the distances is a better fit. This motivates us to explore the use of mixture models for the distances to achieve a better performance in both regimes. The first model we consider is a mixture model where the components of the mixture are a Markov chain like distribution and an independent model. Then, we consider a variation of this mixture model, where instead of assigning prior probabilities based on the training data, we adaptively choose the underlying mixture component based on the observation.

5.3.1 Mixture Model

In this section, we describe a mixture model [9, 57] for the distances between the query and reference fingerprint. The components of the mixture are chosen to be a Markov-Chain like distribution and an independent model. We introduce latent variables $\mathbf{z} = [z_1 \; z_2]^T \in \{[1 \; 0]^T, [0 \; 1]^T\}$ to denote the component from which the observation is drawn. Only one of z_1 and z_2 is equal to 1 and the other variable is 0, which corresponds to the 1-of-K coding scheme [9]. Under hypothesis H_i, the probability distribution of the distances $\mathbf{d}$ given the latent variable $\mathbf{z}^{(i)}$ is given by:

$$\begin{aligned}\Pr(\mathbf{d} \,|\, \mathbf{z}^{(i)} = [1 \; 0]^T, \mathbf{\Theta}^{(i)}, H_i) &= \pi_i(d(1)) \prod_{j=2}^{L} P_i(d(j-1), d(j)) \\ \Pr(\mathbf{d} \,|\, \mathbf{z}^{(i)} = [0 \; 1]^T, \mathbf{\Theta}^{(i)}, H_i) &= \prod_{j=1}^{L} q_i(d(j)),\end{aligned}$$

where in the second component, the distances are assumed to be i.i.d. with a common distribution q_i under H_i. In the above expressions, the parameters of the model $\mathbf{\Theta}^{(i)}$ consists of the transition matrix $\mathbf{P}_i$, the corresponding stationary distribution π_i and the common distribution q_i. The prior probabilities of each of the components is given as:

$$\begin{aligned}\Pr(\mathbf{z}^{(i)} = [1\ 0]^T \,|\, H_i) &= \alpha_1^{(i)} \\ \Pr(\mathbf{z}^{(i)} = [0\ 1]^T \,|\, H_i) &= \alpha_2^{(i)}\end{aligned}$$

Given the training data, the parameters of the models can be estimated using the Expectation-Maximization (EM) algorithm [22, 58]. The details of the EM algorithm for this model are provided as appendix in Section 5.5. Using the estimated parameters, the log-likelihood ratio corresponding to this mixture model is used to identify whether the query fingerprint is a match to the reference fingerprint or not.

Experimental Results

As before, we evaluate the appropriateness of this model using the MUSCLE-VCD database. 50 videos are randomly selected and used for training while the remaining are used for testing. This procedure is repeated 10 times and the results are averaged. Figure 5.5 compares the performance of the detector based on this mixture model with the detectors based on the Markov chain model and the average distance. From the figure, we observe that the performance of the mixture-model based detector very closely follows that of the Markov chain based detector. To understand this behavior, we examined the parameters estimated for each of the distributions during the training process. Under both hypotheses, we found that the prior probabilities of the Markov chain component $\alpha_1^{(i)} \geq 0.9$ implying that the contribution from the Markov chain component dominates the likelihood of the observations under this model and the contribution from the independent model is

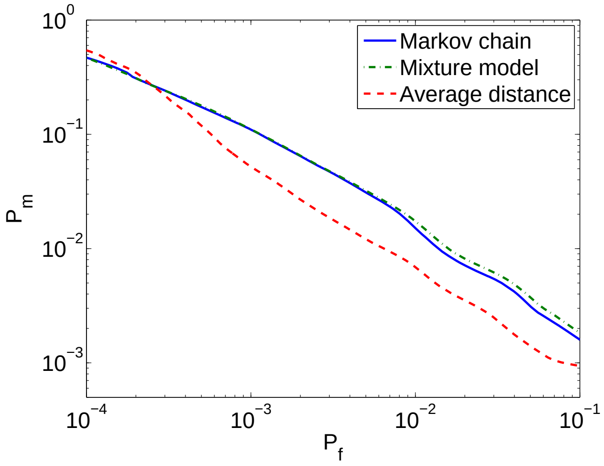


Figure 5.5: Comparison of the detectors based on the mixture model, Markov-Chain model, and the average distance.

very small. As a result, the performance of the mixture model is similar to that of the Markov chain model.

5.3.2 Adaptive Model

Based on the analysis and discussion in the previous subsection, we see that merely relying on the prior probabilities of the mixture components results in the likelihood being dominated by the Markov chain component. Further, the characteristics of the observation are not utilized in choosing the mixture component or equivalently, the values of the latent variables. To remedy this problem, we propose a different approach to estimate the value of the latent variable, and then utilize the estimated value to make a decision in the hypothesis test.

We use the average distance $\bar{d}$ as a parameter to guide us in choosing the appropriate model. If $\bar{d} \geq d_0$, we use the LLR detector based on the i.i.d. distribution, and the Markov chain based detector otherwise. This is equivalent to setting:

$$\mathbf{z}^{(i)} = \begin{cases} [0\ 1] & \text{if } \bar{d} \geq d_0 \\ \\ [1\ 0] & \text{if } \bar{d} < d_0 \end{cases}$$

Under this setting, the model for the distribution of the distances becomes:

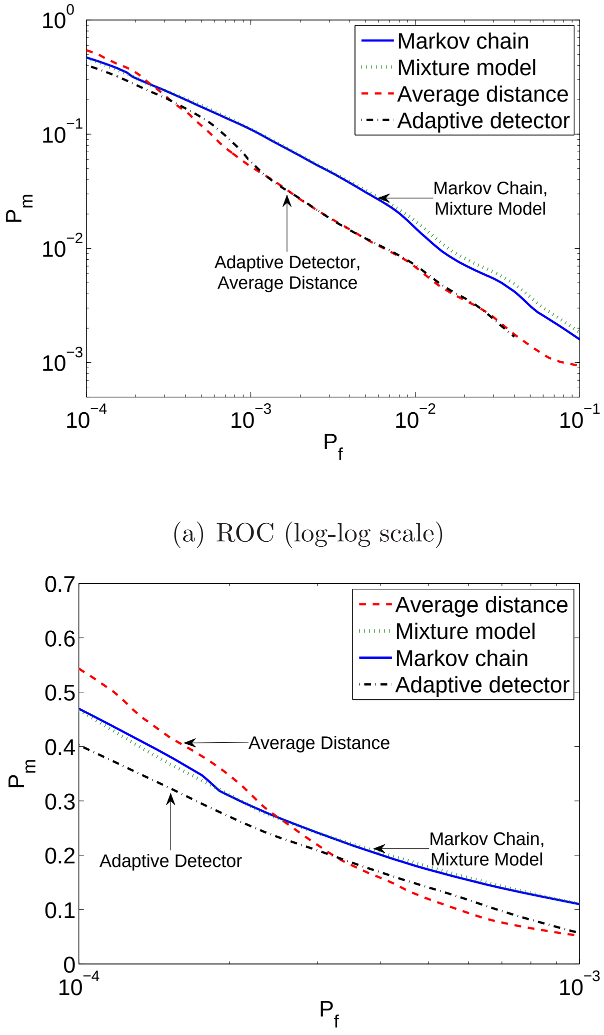
$$\begin{aligned} \Pr(\mathbf{d} \mid \bar{d} < d_0, H_i) &= \pi_i(d(1)) \prod_{j=2}^{L} P_i(d(j-1), d(j)) \\ \Pr(\mathbf{d} \mid \bar{d} \geq d_0, H_i) &= \prod_{j=1}^{L} q_i(d(j)), \end{aligned}$$

The corresponding decision rule is given as:

$$\begin{cases} LR(\mathbf{X}, \mathbf{Y}) = \underset{H_0}{\overset{H_1}{\gtrless}} \tau_1^A & \text{if } \bar{d} < d_0 \\ \prod_{j=1}^{L} \frac{q_1(d(j))}{q_0(d(j))} \underset{H_0}{\overset{H_1}{\gtrless}} \tau_2^A & \text{if } \bar{d} \geq d_0. \end{cases} \tag{5.3}$$

Thus, the detector adaptively chooses between the Markov chain and independent model based decision rules depending on the average distance of the observations. We will refer to the detector that utilizes this decision rule as the adaptive detector. The parameter d_0 can be estimated by evaluating the performance of the adaptive detector using the training set and choosing the value that results in the best performance. In practice, we found that the performance is not very sensitive to the value of d_0, and that a value $0.25\, d_{\max} \leq d_0 \leq 0.35\, d_{\max}$, where $d_{\max} = 32$ is the maximum possible distance between two sub-fingerprints provides a good detection accuracy. In our experiments, we use the value of $d_0 = 0.25\, d_{\max} = 8$.

As the decision thresholds (τ_1^A, τ_2^A) vary, different tradeoffs between P_m and P_f can be obtained. Figure 5.6 compares the ROC curves for the adaptive detector,



(a) ROC (log-log scale)

(b) Low P_f region

Figure 5.6: ROC curves for the various detectors.

the Markov chain LR, the mixture model based and the average distance based detectors on a logarithmic scale. As expected, we see that the adaptive detector performs better than or comparable to the other detectors for all values of P_m and P_f. In particular, the adaptive detector has approximately $5 - 10\%$ lower P_m for $P_f < 10^{-3}$.

In practical applications, the computational complexity of the detectors is also an important parameter. The average distance based detector requires $O(L)$ operations for computing the average distance, which is then compared to a threshold. The adaptive detector utilizes the average distance and the number of transitions between the various states, which can both be computed with one pass over the data requiring $O(L)$ operations. The individual likelihood ratios that appear in the decision rule, may be pre-computed and stored, so that computing the appropriate detection statistic requires at most an additional $O(L)$ operations. The overall complexity is still $O(L)$, implying that the adaptive detector based on this model is only slightly more expensive compared to the simple average distance based detector.

5.3.3 Model Evaluation using Information Criteria

In the previous section, we have seen that the adaptive model gives the best performance in both the low P_f and low P_m regimes. As the adaptive model is a more complicated model compared to the Markov chain model, there is a danger that the model is overfitting the available data. Typically, when various models are available for fitting the data distribution, various information criteria, such as the Akaike Information Criterion (AIC) [1] or the Bayesian Information Criterion (BIC) [38, 72] are used to evaluate the tradeoff between model complexity and the better fitting capability of a more complex model. In this subsection, we use these information criteria to compare the adaptive model and the Markov chain model.

The AIC is defined as:

$$AIC = \ln p(\mathbf{D}|\mathbf{\Theta}_{ML}) - N_p,$$

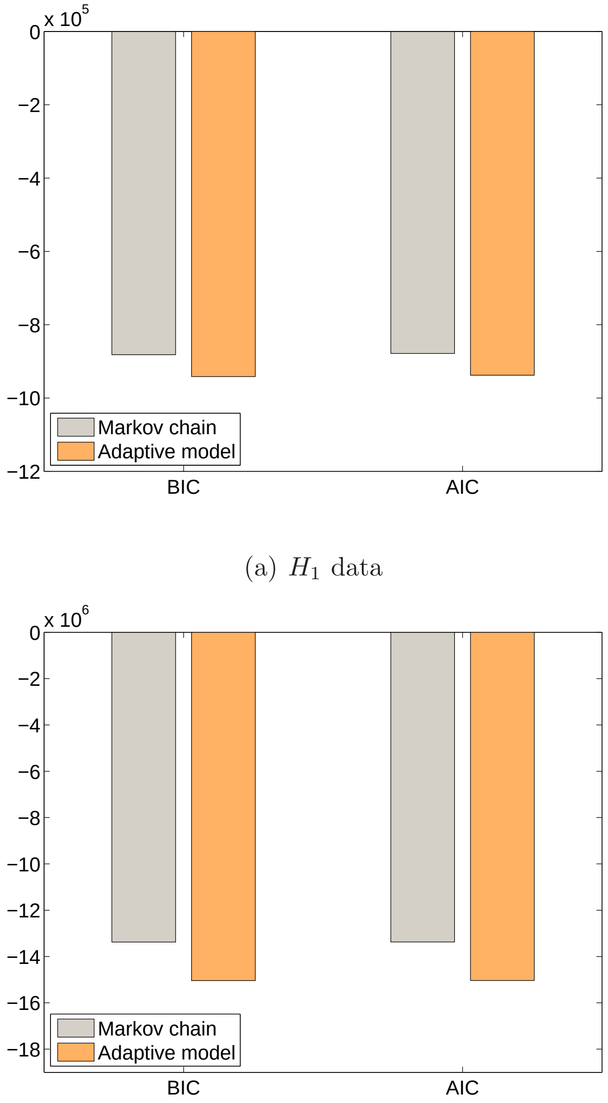
where $\mathbf{D}$ is the set of available observations, Θ_{ML} is the ML estimate of the model parameters, and N_p is the number of parameters in the model. A Bayesian approach was adopted in [37, 72] to derive a related information criterion called the BIC:

$$BIC = \ln p(\mathbf{D}|\mathbf{\Theta}_{ML}) - N_p \ln M,$$

where M is the number of training samples available. Compared to the AIC, the BIC has a higher penalty for the model complexity in terms of the number of parameters.

For the Markov chain model, we estimate the transition matrix from the training data. As the Hamming distance between any two sub-fingerprints can take values from 0 to $d_{max} = 32$, there are $(d_{max} + 1)$ states in the Markov chain and the transition matrix is of size $(d_{max} + 1) \times (d_{max} + 1)$. As the transition matrix is stochastic, implying that the sum of each row equals 1, only d_{max} entries in each row are independent. Thus, the number of parameters for the Markov chain model $N_p = (d_{max} + 1)d_{max}$. For the adaptive model, we have $(d_{max} + 1)d_{max}$ parameters for the transition matrix, d_{max} parameters for the independent distribution, and an additional parameter d_0, so that the total number of parameters $N_p = (d_{max} + 1)^2$.

Figure 5.7 compares the values of the AIC and BIC for the Markov Chain and adaptive models using the training data under the two hypotheses. From the figure, we observe that the Markov chain model has a slightly higher value for both the information criteria. The adaptive model has a 6% lower BIC under the H_1 hypothesis and 12% lower BIC under the H_0 hypothesis. This would imply that



(a) H_1 data

(b) H_0 data

Figure 5.7: Evaluation of AIC and BIC for Markov chain and adaptive models using training data under the two hypotheses.

from this perspective, the adaptive model is slightly worse compared to the Markov chain model. However, as shown via the experimental results in Section 5.3.2, the adaptive model is better at discriminating between the two hypothesis.

5.4 Chapter Summary

In this chapter, we developed models for temporal correlations in fingerprints. We first explored the use of a Markov chain to model the distribution of the distances between the query and reference fingerprints. Fingerprint matching was then considered as a hypothesis test and the optimal likelihood ratio based detector based on this Markov chain model was derived. Experimental results indicated that the Markov chain model is a good fit only for a certain part of the distribution, and an independent model may be a better fit in other regimes. Motivated by this observation, we proposed a hybrid model for the fingerprints and derived an adaptive detector that performs better than or comparable to the Markov chain and average distance based detectors. While this adaptive model has $5-10\%$ lower values for the model information criteria, the corresponding adaptive detector provides approximately $5-10\%$ lower P_m in the low P_f regime without significantly increasing the computational cost.

5.5 Appendix: EM Algorithm for the Mixture Model

In this appendix, we derive the Expectation-Maximization algorithm for estimating the parameters of the mixture model described in Section 5.3.1. The parameters for the model under H_1 and H_0 hypothesis are estimated separately using the corresponding sets of training data. As the model for the data under both hypotheses is the same, the steps involved in estimating the parameters are also identical. Below, we drop the explicit conditioning on the hypothesis H_i to simplify the notation.

Suppose that we have training data $\mathbf{D} = [\mathbf{d}_1 \ \mathbf{d}_2 \ \ldots \ \mathbf{d}_M]$, where each $\mathbf{d}_i$ consists of L components and M is the size of the training set so that $\mathbf{D}$ is an $L \times M$ matrix. Let the corresponding latent variables for each observation in the training data be represented by the matrix $\mathbf{Z} = [\mathbf{z}_1 \ \mathbf{z}_2 \ \ldots \ \mathbf{z}_M]$. As each $\mathbf{z}_i$ is a vector of size 2×1, $\mathbf{Z}$ has size $2 \times M$. Denote the parameters for each component in the mixture model by θ_k which are to be estimated from the training data and let $p(\mathbf{d} \,|\, z_k = 1, \theta_k) \triangleq p_k(\mathbf{d} \,|\, \theta_k)$.

The joint distribution of an observation and the corresponding latent variables can be written as:

$$p(\mathbf{d}, \mathbf{z} \,|\, \boldsymbol{\Theta}) = \prod_{k=1}^{2} \left(\alpha_k p_k(\mathbf{d} \,|\, \theta_k)\right)^{z_k},$$

where $\alpha_k = \Pr(z_k = 1)$ and $\boldsymbol{\Theta} = (\theta_1, \theta_2)$. For the overall training data we have:

$$p(\mathbf{D}, \mathbf{Z} \,|\, \boldsymbol{\Theta}) = \prod_{m=1}^{M} \prod_{k=1}^{2} \left(\alpha_k p_k(\mathbf{d}_m \,|\, \theta_k)\right)^{Z_{km}}, \tag{5.4}$$

5.5.1 E-step

For the E-step, we need to compute the expectation of $\ln p(\mathbf{D}, \mathbf{Z} \,|\, \mathbf{\Theta})$ under the distribution $p(\mathbf{Z} \,|\, \mathbf{D}, \mathbf{\Theta})$. Using Eqn. (5.4), we have:

$$\mathbb{E}_{p(\mathbf{Z} \,|\, \mathbf{D},\mathbf{\Theta})}[\ln p(\mathbf{D}, \mathbf{Z} \,|\, \mathbf{\Theta})] = \sum_{m=1}^{M} \sum_{k=1}^{2} \mathbb{E}_{p(\mathbf{Z} \,|\, \mathbf{D},\mathbf{\Theta})}[Z_{km}] \left(\ln \alpha_k + \ln p_k(\mathbf{d}_m \,|\, \theta_k)\right). \quad (5.5)$$

We now need to evaluate $\mathbb{E}_{p(\mathbf{Z} \,|\, \mathbf{D},\mathbf{\Theta})}[Z_{km}]$. As each column in $\mathbf{Z}$ is independent, we can consider each column individually. For the mth column,

$$\begin{aligned} \mathbb{E}_{p(\mathbf{z}_m \,|\, \mathbf{d}_m,\mathbf{\Theta})}[\mathbf{z}_m] &= \sum_{\mathbf{z}_m} \mathbf{z}_m p(\mathbf{z}_m \,|\, \mathbf{d}_m, \mathbf{\Theta}) \\ &= \sum_{\mathbf{z}_m} \mathbf{z}_m \frac{p(\mathbf{z}_m, \mathbf{d}_m \,|\, \mathbf{\Theta})}{p(\mathbf{d}_m \,|\, \mathbf{\Theta})} \\ &= \frac{\sum_{\mathbf{z}_m} \mathbf{z}_m p(\mathbf{z}_m, \mathbf{d}_m \,|\, \mathbf{\Theta})}{\sum_{\mathbf{z}_m} p(\mathbf{z}_m, \mathbf{d}_m \,|\, \mathbf{\Theta})}. \end{aligned}$$

Since $\mathbf{z}_m$ can take only two values $[1\ 0]^T$, $[0\ 1]^T$, we have:

$$\mathbb{E}_{p(\mathbf{Z} \,|\, \mathbf{D},\mathbf{\Theta})}[Z_{km}] = \frac{\alpha_k p_k(\mathbf{d} \,|\, \theta_k)}{\sum_{k=1}^{2} \alpha_k p_k(\mathbf{d} \,|\, \theta_k)} = \gamma(Z_{km}),$$

which allows us to compute the desired quantity in Eqn (5.5) and complete the E-step of the EM algorithm. The quantities $\gamma(Z_{km})$ are called the "responsibilities" as they indicate the responsibility for the mth observation taken by the kth component of the mixture model [9].

5.5.2 M-step

In the M-step of the algorithm, we choose the parameters $\mathbf{\Theta}$ that maximize the expectation $\mathbb{E}_{p(\mathbf{Z} \,|\, \mathbf{D},\mathbf{\Theta})}[\ln p(\mathbf{D}, \mathbf{Z} \,|\, \mathbf{\Theta})]$ obtained in the E-step of the algorithm and shown in Eqn (5.5).

First, consider the prior probabilities α_k. As the prior probabilities should sum to 1, we construct the Lagrangian:

$$\mathcal{L}^{\alpha}(\{\alpha_k\}) = \sum_{m=1}^{M} \sum_{k=1}^{2} \gamma(Z_{km}) \left(\ln \alpha_k + \ln p_k(\mathbf{d}_m \,|\, \theta_k)\right) + \lambda \left(1 - \sum_{k=1}^{2} \alpha_k\right).$$

Differentiating with respect to α_k and equating the derivative to 0, we obtain $\alpha_k = \frac{1}{\lambda} \sum_{m=1}^{M} \gamma(Z_{km})$. Using the condition that $\sum_{k=1}^{2} \alpha_k = 1$, we obtain $\lambda = \sum_{k=1}^{2} \sum_{m=1}^{M} \gamma(Z_{km}) = M$, so that

$$\alpha_k = \frac{\sum_{m=1}^{M} \gamma(Z_{km})}{M}.$$

Next, we estimate the parameters for each of the individual mixture components. For the Markov chain component, the parameter θ_1 to be estimated corresponds to the transition matrix $\mathbf{P}$. The log-likelihood under this component is given as $\ln p_1(\mathbf{d} \,|\, \theta_1) = \ln \pi(d(1)) + \sum_{j=2}^{L} \ln P(d(j-1), d(j))$. Let $N(i,j)$ denote the number of transitions from state i to j, so that the log-likelihood can be written in terms of the $N(i,j)$ as $\ln p_1(\mathbf{d} \,|\, \theta_1) = \ln \pi(d(1)) + \sum_{(i,j)} N(i,j) \ln P(i,j)$. While estimating the $P(i,j)$, we also have the constraint $\sum_j P(i,j) = 1, \forall i$. We construct the Lagrangian appropriately as:

$$\mathcal{L}^{1}(\mathbf{P}) = \sum_{m=1}^{M} \gamma(Z_{1m}) \left(\ln \alpha_1 + \ln p_1(\mathbf{d}_m \,|\, \theta_1)\right) + \sum_i \lambda_i \left(\sum_j P(i,j) - 1\right).$$

Substituting the expression of the log-likelihood into the above Lagrangian, differentiating with respect to $P(i,j)$, equating the derivative to 0, and utilizing the normalization constraint, we obtain:

$$P(i,j) = \frac{\sum_{m=1}^{M} \gamma(Z_{1m}) N(i,j)}{\sum_j \sum_{m=1}^{M} \gamma(Z_{1m}) N(i,j)}.$$

For the independent distribution component, the parameter θ_2 to be estimated corresponds to the common distribution q. The log-likelihood under this component is $\ln p_2(\mathbf{d} \,|\, \theta_2) = \sum_{i=1}^{L} \ln q(d(i))$. Let N_j denote the number of occurrences of the value j in $\mathbf{d}$, so that the log-likelihood can be expressed as $\ln p_2(\mathbf{d} \,|\, \theta_2) = \sum_j N_j \ln q(d(i))$. Using the constraint $\sum_j q(j) = 1$, the Lagrangian can be written as:

$$\mathcal{L}^2(\mathbf{q}) = \sum_{m=1}^{M} \gamma(Z_{2m}) \left(\ln \alpha_2 + \ln p_2(\mathbf{d}_m \,|\, \theta_2)\right) + \lambda'(\sum_j q(j) - 1).$$

Maximizing with respect to $q(j)$, we obtain

$$q(j) = \frac{\sum_{m=1}^{M} \gamma(Z_{2m}) N_j}{\sum_j \sum_{m=1}^{M} \gamma(Z_{2m}) N_j}.$$

The E-step and M-steps are performed alternately until the value of the objective function does not change much to obtain the final estimates of the parameters of the mixture model.

Chapter 6

Collusion-Resistant Fingerprinting for Compressed Multimedia

Content fingerprinting, which was studied in the previous chapters, relies on the intrinsic characteristics of multimedia to identify and filter them. It can be used to prevent the redistribution of multimedia via UGC websites and peer-to-peer networks. On the other hand, collusion-resistant fingerprinting is a proactive technique employed to deter multimedia piracy and prevent the leak of classified information. In each authorized copy of the multimedia, a unique signal is embedded that identifies the recipient. This *embedded fingerprint* can be extracted from a pirated copy and used to trace the user responsible for the leak.

As this fingerprint is unique to each recipient, a group of malicious users can collaborate to launch collusion attacks on the system. By comparing their individual fingerprinted copies, the colluders can attempt to identify the locations of the embedded fingerprint and remove them. Various collusion-resistant fingerprint designs have been proposed in the literature and are summarized in Section 6.1. Most of these techniques have been developed for protecting uncompressed multimedia. In many practical applications, multimedia is utilized in a compressed format, where it is necessary to embed fingerprints into compressed multimedia.

One representative application scenario is an online music/video store that wishes to deter illicit redistribution of the content purchased from the store. Primarily based on proprietary security protocols and data formats, most existing Digital Rights Management (DRM) [50] techniques are not interoperable between devices from different vendors and often restrict the freedom of the users to play the content on the device of their choice [42]. Further, if the protection provided by the DRM technique is circumvented, the user can redistribute the content without fear of being apprehended. Embedding imperceptible fingerprints, on the other hand, does not restrict content to be packaged in any proprietary format. It is thus interoperable and can be incorporated into existing systems to complement other protection techniques. For an online store to deploy fingerprinting to protect its multimedia content, the fingerprints should be embedded in the source audio or video files that are typically stored in compressed form to conserve storage space. When a user purchases a particular content, a unique fingerprint is embedded in the host audio or video signal and this fingerprinted signal is then transmitted to the user over the internet in compressed form to conserve bandwidth. As it is possible for users to gather multiple fingerprinted versions of the same content and apply collusion attacks, the embedded fingerprints should be resilient to collusions. This scenario

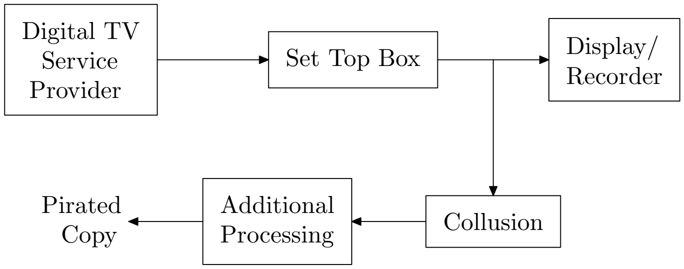


Figure 6.1: Digital TV distribution scenario where the host signal to be fingerprinted is compressed.

highlights the necessity of collusion resistant fingerprint design for compressed multimedia.

Another representative application scenario is shown in Figure 6.1. A digital TV service provider delivers compressed video to millions of subscribers. The video is compressed to meet bandwidth requirements and may be further encrypted to prevent unauthorized users from viewing the content. At the viewer's end, a set-top box decrypts, decompresses, and then displays the video stream. The video output of the set-top box may be intercepted by a malicious user who can then rebroadcast or resell the content for profit. Digital fingerprinting can be employed to deter and trace these adversaries [34]. If the fingerprint embedding is performed at the source (TV service provider), a unique stream would have to be transmitted to each user. The amount of bandwidth required for this scheme would be several orders of magnitude higher than broadcasting a single stream to all users. The bandwidth consumption can be reduced by employing coding techniques to reduce the number of different versions of the content that need to be transmitted [33,105], but would still be several times the amount of bandwidth required to transmit just one stream. An attractive alternative is to embed the fingerprints at the set-top box, which has already been secured and tamper-proofed for performing decryption. This post-distribution fingerprinting approach requires only a single transmission of the host video to all users. In this case, the set-top box would have to embed a fingerprint in the host stream that has been previously compressed. In order to combat adversaries who may store the video output of the set-top box and then collude to remove traces of their fingerprints before redistributing the content, the fingerprints embedded should be robust against collusion attacks.

6.1 Related Prior work

Collusion-resistant fingerprint design has been an active research topic for several years. A systematic binary fingerprint construction technique for generic data was proposed by Boneh and Shaw in [10] using an inner staircase code and a random outer code. The Boneh-Shaw code relied on the "marking assumption" that a group of colluders could only modify those parts of the content in which their copies differed. A fingerprinting code construction based on a relaxed version of the marking assumption and optimal in the code length was described in [79]. The marking assumption may not be valid for multimedia data, as the attackers can also modify the parts of the content containing undetectable bits, where their copies are the same, without causing significant perceptual degradation. These fingerprinting codes are usually adapted for multimedia fingerprinting by combining the codes with a fingerprint embedding layer. A fingerprinting scheme for multimedia data based on the Boneh-Shaw code and using a Direct Sequence Spread Spectrum embedding layer was proposed in [104] under a relaxed assumption that allowed modification of the undetectable bits. The code length of this scheme is high, limiting its feasibility in practical applications. Fingerprinting constructions based on q-ary Error-Correcting Codes (ECC) were proposed in [70] with tolerance towards erasures and cropping. This work did not explicitly consider embedding issues and used an abstract assumption to model the underlying embedding scheme. Other fingerprinting codes based on ECC and their properties were examined in [4]. Based on the robust spread-spectrum watermark embedding scheme by Cox et al. [17], a fingerprinting technique for multimedia content was proposed in [83] employing a combinatorial construction and orthogonal spreading sequences for modulation. Fingerprinting based on Quantization Index Modulation [11] was also explored in [77]. Multimedia fingerprint embedding techniques proposed in the literature so far, such as [17, 77], were primarily designed for fingerprinting uncompressed signals.

Information theoretical aspects of fingerprinting have been investigated by modelling fingerprinting as communications with side information [18]. Capacity expressions for fingerprinting signals with finite alphabets [75] and continuous alphabets have been derived [97]. Recently, a capacity achieving universal fingerprinting code has been proposed in [61], where the detector is not required to have knowledge of either the collusion attack or the number of colluders. These theoretical results guarantee the existence of good fingerprinting codes, but often require decoding schemes with high computational complexity and may not be suitable for practical implementations.

Another body of related literature addresses the problem of watermarking compressed signals. A few robust watermarking techniques for compressed signals have been proposed. Watermarks can be embedded in a compressed video stream by adding the Discrete Cosine Transform (DCT) coefficients of a watermark to the quantized DCT coefficients of the compressed host signal followed by re-encoding of the watermarked signal [31]. Another approach embeds watermarks by selectively discarding high-frequency DCT coefficients in certain regions of the image [45]. These techniques were not designed for fingerprinting applications and thus have limited collusion resistance.

Chapter Contributions

One of the reasons that fingerprinting compressed signals has not received much attention is perhaps the implicit belief in the robustness of the underlying embedding technique. Indeed, individual spread spectrum fingerprints embedded in uncompressed hosts are robust enough to survive strong compression [17]. However, as will be shown in this chapter, if the fingerprint is to be embedded in a compressed host signal and the fingerprinted result also has to be stored in compressed form, the corresponding fingerprint components for different users can only take values from a small, discrete set, making the system vulnerable to collusion. To address this problem, we describe a new technique called *Anti-Collusion Dithering* to overcome this constraint and resist collusion attacks [86]. Through experimental studies as well

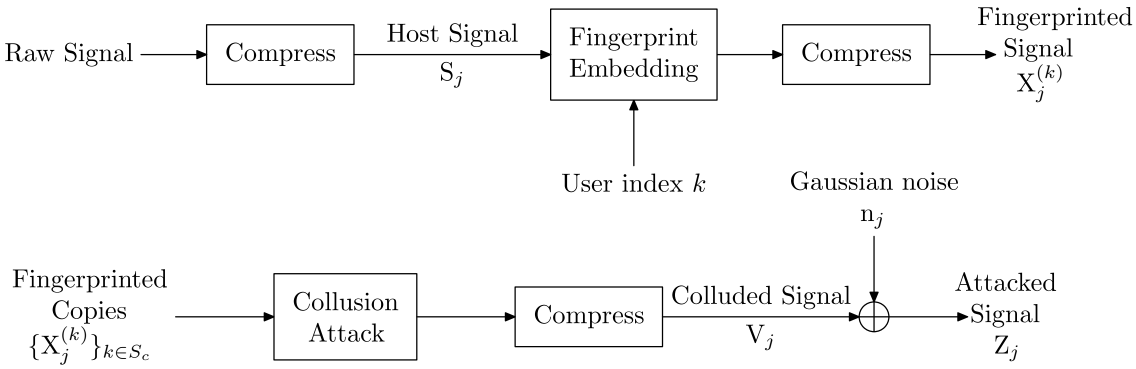


Figure 6.2: System model for compressed domain fingerprinting.

as theoretical analyses based on probability, estimation, and information theories, we will show that almost the same level of collusion resistance can be achieved when applying Anti-Collusion Dithered fingerprinting to compressed host signals as that obtained when fingerprinting uncompressed hosts.

Section 6.2 describes the system model for fingerprinting compressed signals. Section 6.3 examines the collusion resistance of quantized Gaussian fingerprints through simulations. Anti-Collusion Dithering technique is then proposed in Section 6.4 to improve the collusion resistance of the fingerprints. Theoretical analysis of fingerprinting techniques for compressed multimedia will be considered in Chapter 7.

6.2 System Model

Figure 6.2 depicts the system model for compressed domain fingerprinting. Let $\mathbf{S} = [S_1, S_2, \ldots, S_M]$ represent the compressed host signal of length M. For example, in the case of image or video fingerprinting, the host samples could correspond to the 8×8 block DCT coefficients that are widely adopted in image and video coding standards. For simplicity, we consider the vector $\mathbf{S}$ as comprising of elements from one frequency channel after compression, and model the compression of the host signal as a uniform quantization operation with step size Δ, so that $S_j = m\Delta$, where $m = 0, \pm 1, \pm 2, \ldots$ The analysis can be extended to the case of a host signal comprising of elements with different quantization step sizes by grouping samples with similar quantization as one channel.

A fingerprint is embedded in the compressed host signal $\mathbf{S}$ to obtain the fingerprinted signal for each of the N users. The fingerprinted signal for the k^{th} user, $\mathbf{X}^{(k)}$, is quantized with step size Δ_e, *i.e.* for each signal component, $X_j^{(k)} = m\Delta_e$. The value of Δ_e represents the compression of the fingerprinted signal and is chosen by the embedder to achieve a tradeoff between perceptual distortion and bandwidth consumption. If $\Delta_e < \Delta$, the bandwidth required to transmit the fingerprinted signal may increase compared to the host signal before fingerprinting. Alternatively, choosing $\Delta_e > \Delta$ may result in further perceptual distortion. Thus, a reasonable choice for the embedder is to set $\Delta_e = \Delta$. The fingerprinted signal for user k can be generated by additive embedding $\mathbf{X}^{(k)} = \mathbf{S} + \mathbf{W}^{(k)}$, where $\mathbf{W}^{(k)} = [W_1^{(k)}, W_2^{(k)}, \ldots, W_M^{(k)}]$ represents the k^{th} user's fingerprint. The energy of the fingerprint is chosen such that the distortion introduced by embedding does not cause visual artifacts. We quantify the distortion using the Mean Squared Error (MSE) and express the distortion constraint as:

$$\mathrm{E}[\|\mathbf{S} - \mathbf{X}^{(k)}\|^2] \leq M \cdot D(\Delta), \quad \forall k = 1, 2, \ldots, N, \tag{6.1}$$

where $D(\Delta)$ is the maximum allowed squared distortion per sample, given the quantization step size Δ.

6.2.1 Collusion Attacks

A group of K malicious users $\mathcal{U}_K$, may mount collusion attacks and attempt to create a copy $\mathbf{V}$ that does not contain traces of their fingerprints. The colluders can re-compress the attacked signal for ease of redistribution and to further remove traces of their fingerprints. Suppose the attackers compress the colluded signal by quantizing it with step size Δ_c so that $V_j = m\Delta_c$. The attackers' choice of Δ_c is affected by the value of Δ. Since the fingerprinted signal has previously been quantized with step size Δ, by choosing $\Delta_c < \Delta$ the colluders would not improve the perceptual quality of the attacked signal. Applying milder quantization would not only lead to increased bandwidth requirements for the colluded copy, but also favor the survival of fingerprints, resulting in a higher probability for at least one of the colluders to be caught. On the other hand, choosing $\Delta_c > \Delta$ would further reduce the perceptual quality of the colluded signal. A reasonable compromise for the attacker would be to choose $\Delta_c = \Delta$, which we will examine in the following section. The case of $\Delta_c \neq \Delta$ will be examined later in Section 6.4.3.

The j^{th} sample of the colluded version V_j is obtained as $V_j = g(\{X_j^{(k)}\}_{k\in\mathcal{U}_K})$, where $g(\cdot)$ is a suitable collusion function. Several linear and nonlinear collusion attacks have been studied in [106] for Gaussian based independent fingerprints for uncompressed host signals. We extend these attacks to compressed signals by adding quantization, and examine their effectiveness against the fingerprinting system. The attacks are defined as:

$$\begin{aligned}
\text{Average}: V_j^{\text{avg}} &= \text{round}\left(\frac{\sum_{k\in\mathcal{U}_K} X_j^{(k)}}{K\Delta_c}\right)\times\Delta_c,\\
\text{Median}: V_j^{\text{med}} &= \text{round}\left(\frac{\text{median}(\{X_j^{(k)}\}_{k\in\mathcal{U}_K})}{\Delta_c}\right)\times\Delta_c,\\
\text{Minimum}: V_j^{\text{min}} &= \min(\{X_j^{(k)}\}_{k\in\mathcal{U}_K}),\\
\text{Maximum}: V_j^{\text{max}} &= \max(\{X_j^{(k)}\}_{k\in\mathcal{U}_K}),\\
\text{Min-Max}: V_j^{\text{minmax}} &= \text{round}\left(\frac{V_j^{\text{min}}+V_j^{\text{max}}}{2\Delta_c}\right)\times\Delta_c,\\
\text{Modified Negative}: V_j^{\text{modneg}} &= V_j^{\text{min}}+V_j^{\text{max}}-V_j^{\text{med}},\\
\text{Randomized Min-Max}^1\text{:}\ V_j^{\text{randMinMax}} &= \begin{cases} V_j^{\text{min}} & \text{with probability } 0.5\\ V_j^{\text{max}} & \text{with probability } 0.5\end{cases}
\end{aligned}\tag{6.2}$$

where round(·) denotes rounding to the nearest integer. Further processing, such as addition of noise and filtering, may be applied to the colluded signal. For simplicity, we model these operations as additive white Gaussian noise $\mathbf{n}$, with zero mean and variance σ^2 to obtain $\mathbf{Z} = \mathbf{V} + \mathbf{n}$, as shown in Figure 3.1. It is also possible to consider the case where the noise $\mathbf{n}$ is quantized, but our experiments have shown that there is no significant difference in the results when the noise $\mathbf{n}$ is quantized, as the noise mainly serves to confuse the detector. Henceforth, we will restrict our attention to the case where $\mathbf{n}$ is a (continuous-valued) zero-mean white Gaussian noise vector.

Another attack that the colluders can mount is the random attack. The attackers first estimate the dynamic range of the fingerprinted signal, i.e. X_j^{min} and

[1] This attack corresponds to the randomized negative attack examined in [106].

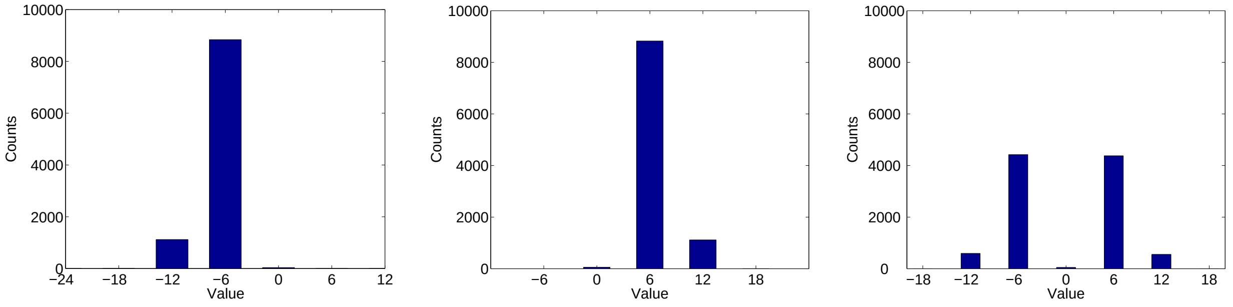
$X_j^{\max}$ such that, for all $k = 1, 2, 3, \ldots, N$,

$$X_j^{\min} \leq X_j^{(k)} \leq X_j^{\max}.$$

Since the values of $X_j^{\min}$ and $X_j^{\max}$ are typically chosen by the fingerprint embedder to ensure that no perceptual distortion is introduced by the embedding, the colluders can randomly choose $V_j = m\Delta_c$ from the interval $[X_j^{\min}, X_j^{\max}]$ without introducing perceptual distortion. This attack may be modelled as the Min-Max attack plus additional noise and hence will not be treated separately in this work.

6.2.2 Colluder Detection

For detection, we focus our attention on the problem of identifying one of the adversaries who have contributed to a colluded signal under question, known as the "Catch One" case [98]. The analysis can be extended to other cases such as "Catch More" or "Catch All", by properly adjusting the form of the detector and the corresponding threshold and evaluation criteria [98]. Since the host signal $\mathbf{S}$ is usually available to the detector in fingerprinting applications, the detector first performs registration and subsequently removes the interference from the host signal $\mathbf{S}$, by subtracting it from the attacked signal $\mathbf{Z}$ to obtain $\mathbf{Y} = \mathbf{Z} - \mathbf{S}$. The detector then applies preprocessing to remove any non-zero mean. We follow the method in [106] to preprocess the test signal and center the histogram of the test signal around zero. If a single non-zero sample mean is observed, such as that observed in the case of minimum or maximum attacks, shown in Figure 6.3(a) and (b), it is subtracted to obtain a zero mean signal, $Y_j' = Y_j - (\sum_{j=1}^{M} Y_j)/M$. If a



(a) (b) (c)

Figure 6.3: Histograms of extracted fingerprints $\mathbf{Y}$ for (a) Minimum, (b) Maximum, and (c) Randomized Min-Max Attacks.

bi-modal distribution is observed, as in the case of a randomized min-max attack (Figure 6.3 (c)), the fingerprint components are clustered into two distributions and the corresponding mean is subtracted for each distribution. Specifically, define $\mu_{\text{neg}} = (\sum_{j=1}^{M} Y_j.\mathbb{1}\{Y_j < 0\})/(\sum_{j=1}^{M} \mathbb{1}\{Y_j < 0\})$ and $\mu_{\text{pos}} = (\sum_{j=1}^{M} Y_j.\mathbb{1}\{Y_j > 0\})/(\sum_{j=1}^{M} \mathbb{1}\{Y_j > 0\})$, where $\mathbb{1}\{\cdot\}$ is an indicator function. The preprocessing applied can be expressed as:

$$Y_j' = \begin{cases} Y_j - \mu_{\text{neg}} & \text{if } Y_j < 0 \\ Y_j - \mu_{\text{pos}} & \text{if } Y_j > 0 \end{cases}.$$

The detector then correlates the test signal $\mathbf{Y}'$ with each of the fingerprints $\mathbf{W}^{(k)}$ in the database to obtain the detection statistic $T^{(k)}$ for each user k,

$$T^{(k)} = \frac{1}{M}\langle \mathbf{Y}', \mathbf{W}^{(k)} \rangle. \tag{6.3}$$

The user q whose fingerprint has the maximum correlation with the extracted test signal is declared guilty:

$$q = \arg \max_{k=1,2,\ldots,N} T^{(k)}. \tag{6.4}$$

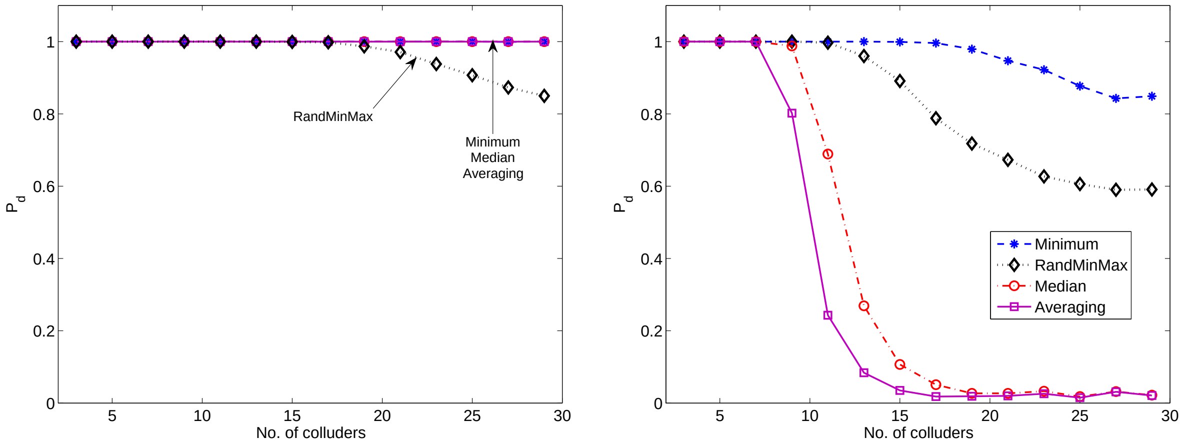
Gaussian based spread spectrum fingerprints have been shown to be effective against collusion attacks on uncompressed host signals [76,106] and have also served as an embedding layer for adapting systematic fingerprint construction techniques to multimedia fingerprinting [33, 83]. The Gaussian distribution has been shown to be the optimal distribution for uncompressed multimedia fingerprints under a wide variety of attacks [41]. Hence, we first examine the performance of Gaussian fingerprints when fingerprinting compressed multimedia signals in the next section.

6.3 Evaluation of Quantized Gaussian Fingerprints

In this section, we evaluate the collusion resistance when using Gaussian based independent signals as fingerprints for compressed host signals. In the embedding stage, watermark components $W_j^{(k)}$ are generated by quantizing independent and identically distributed (*i.i.d.*) samples from a zero-mean Gaussian distribution with step size Δ, $W_j^{(k)} = \text{round}\left(\frac{Q_j^{(k)}}{\Delta}\right) \times \Delta$, where $Q_j^{(k)}$ is a zero mean Gaussian random variable. These watermark sequences are then embedded into the host signal to obtain the fingerprinted signal. The variance of the Gaussian random variable $Q_j^{(k)}$ is chosen such that the fingerprinted signal satisfies the distortion constraint in Eqn. (6.1). For the attacks, we first concentrate on the case $\Delta_c = \Delta$, where the attackers use the original quantization step size of the host signal to compress the colluded signal, and consider the case $\Delta_c \neq \Delta$ in Section 6.4.3. Guilty users are identified using the correlation based detector in Eqn. (6.4).

In our experiments, we focus on one frequency channel in the block DCT domain, and the results can be extended to the multi-channel case. Since the host signal, fingerprint signal, and colluded signal are all quantized with the same quantization step size Δ, and the host signal is subtracted from the colluded signal before detection, the distribution of the detection statistics is independent of the host, as will be shown in Section 7.1. Thus, the simulation results obtained are independent of the host distribution. We consider a system with $N = 1024$ users, and choose the fingerprint length $M = 10^4$ which is the approximate number of coefficients in a 256×256 natural grayscale image that can be used for embedding the fingerprint. The maximum allowed squared distortion, $D(\Delta)$ is set to 15, which results in a Peak Signal to Noise Ratio (PSNR) of around 36dB if every DCT coefficient were to be used for embedding with the same maximum allowed distortion. We test the performance of the system for $\Delta = 6, 4$, and 1 which correspond to quantization step sizes for the AC_{11} band in the JPEG table for quality factors of 75, 85, and 95, respectively. A quality factor of 75 is the default in many applications as it generally provides a good tradeoff between signal quality and bit rate. Quality factors larger than 75 are typically used in applications that demand high quality and hence we investigate the performance under these settings.

Figure 6.4 shows the probability of catching one colluder, P_d, versus the number of colluders for various collusion attacks. In each case, the additive noise power is set to be comparable to the fingerprint power, *i.e.*, Watermark-to-Noise Ratio (WNR) = 0dB, but the overall distortion between the attacked image and the compressed host image would depend on the distortion introduced by the corresponding attack. Figure 6.4(a) shows P_d when $\Delta = 1$ under averaging, median, minimum



(a) $\Delta = 1$

(b) $\Delta = 6$

Figure 6.4: Probability of catching one colluder using quantized Gaussian fingerprints at WNR = 0dB, N = 1024 users, $M = 10^4$, $D(\Delta) = 15$ for (a) $\Delta = 1$ and (b) $\Delta = 6$.

and randomized min-max attacks, and Figure 6.4(b) shows the corresponding results for $\Delta = 6$. From Figure 6.4(a), we see that for $\Delta = 1$, we have approximately 100% detection against the examined attacks when the number of colluders is less than 30, except for the randomized min-max attack, under which P_d starts to drop moderately at 18 colluders. When $\Delta = 6$, averaging is the most effective attack and the fingerprinting system can resist only 7 colluders with $P_d \approx 1$, as shown in Figure 6.4(b). We also observe that the probability of detection does not degrade gracefully with the number of colluders for averaging and median attacks, and there is an abrupt drop around 10 colluders. By symmetry, the behavior of P_d under maximum attack is identical to that under minimum attack and is hence omitted from the figure. Under the Min-Max and modified negative attacks, $P_d = 1$ for up

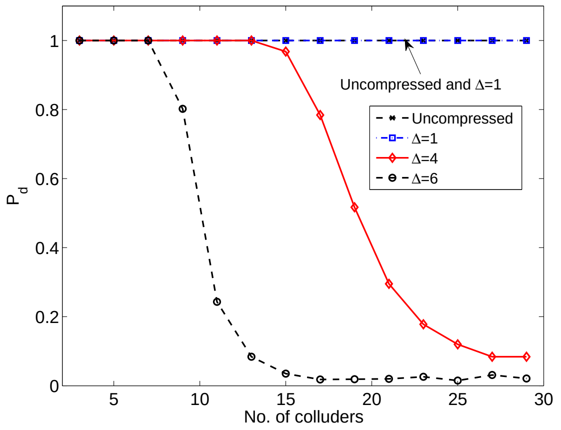
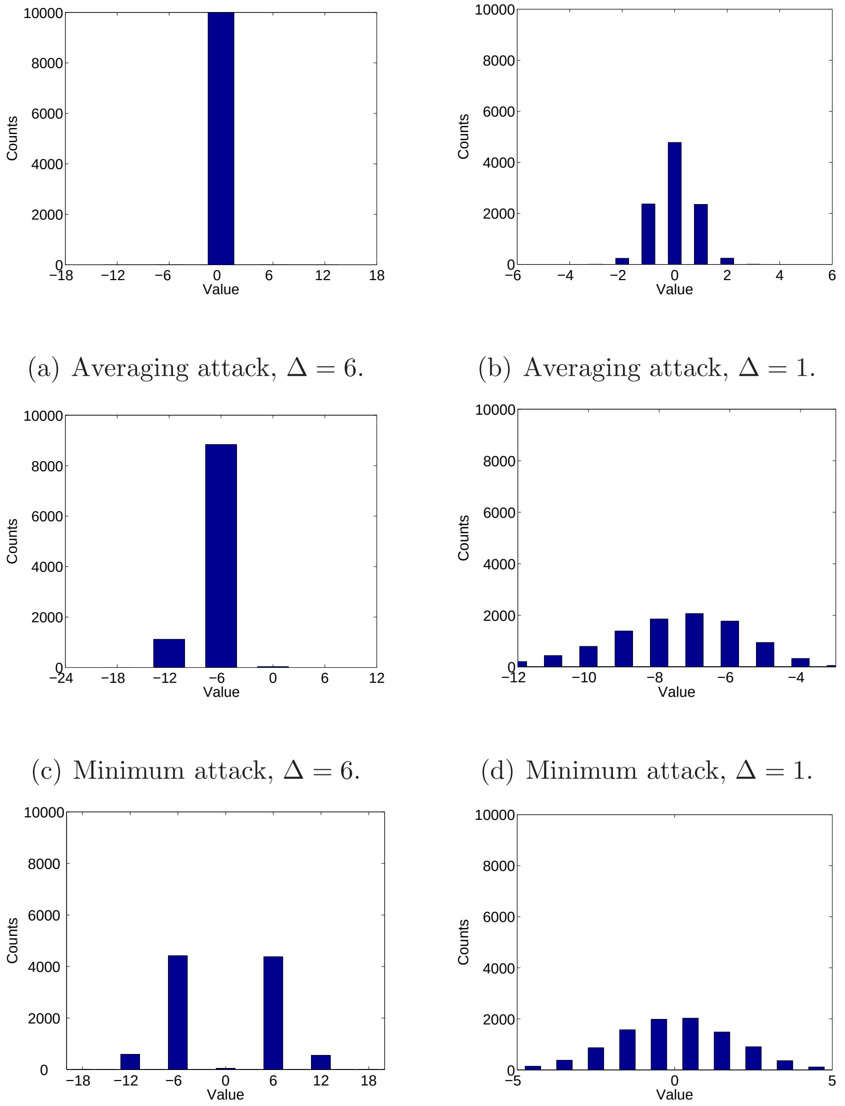


Figure 6.5: Probability of catching an attacker under averaging attack for different quantization step size Δ.

to 30 colluders in both scenarios and have been omitted from the figures for clarity.

From Figure 6.4(a) and (b), we observe that the value of the quantization step size has a significant influence on the detection performance. To understand the influence of the quantization step size Δ, in Figure 6.5 we compare the probability of detection under averaging attack for different quantization step sizes at constant $D(\Delta) = 15$. We observe that for weak quantization ($\Delta = 1$), the performance is similar to that obtained for uncompressed hosts. When stronger compression is applied and the host signal values become more discrete, the averaging attack becomes more powerful. When $\Delta = 6$, the fingerprinting system can be defeated by averaging just 10 copies. Further, as will be shown in Section 7.2, the averaging attack introduces the lowest distortion, making it a very powerful attack against fingerprint systems for compressed signals.

To gain insight into the reduced collusion resistance for compressed host sig-



(a) Averaging attack, $\Delta = 6$.

(b) Averaging attack, $\Delta = 1$.

(c) Minimum attack, $\Delta = 6$.

(d) Minimum attack, $\Delta = 1$.

(e) Randomized min-max attack, $\Delta = 6$.

(f) Randomized min-max attack, $\Delta = 1$.

Figure 6.6: Distribution of colluded fingerprint after averaging, minimum and randomized min-max attacks by 25 colluders.

nals, we examine in Figure 6.6 the histogram of the extracted fingerprint signal (without additive noise) for 25 users' collusion [87] using the same settings as before. Under averaging collusion for $\Delta = 6$, we see from Figure 6.6(a) that most of the fingerprint components are zero and completely removed from the host media after collusion, leading to a failure in identifying colluders. However, when $\Delta = 1$, Figure 6.6(b) shows that approximately half of the colluded fingerprint components remain non-zero under averaging collusion, which enables the detector to catch at least one of the 25 colluders with high probability. A similar trend is observed under the minimum and randomized min-max attacks for $\Delta = 6$ and $\Delta = 1$, as shown in Figs. 6.6(c) through (e). Comparing the histograms for averaging and minimum attacks under $\Delta = 6$, we can see that while averaging collusion removes almost all fingerprint traces (Figure 6.6(a)), the minimum attack still retains some fingerprint components and is thus less effective than averaging collusion (Figure 6.6(c)). Similar inferences can be drawn regarding the other attacks by studying their histograms, which explains the results reported in Figure 6.4(a) and (b) for the probability of catching one colluder.

6.4 Anti-collusion dither

In the previous section, we have shown that even at moderate compression, quantized Gaussian fingerprints may be removed by averaging a few different fingerprinted copies. The main reason that the traditional Gaussian based fingerprinting fails for compressed host signals is because of the discrete nature of the signals before and after fingerprint embedding. When the quantization step size becomes larger, *e.g.* $\Delta = 6$, we notice that the Gaussian distributed fingerprints are mostly quantized to 0, especially after multi-user collusion as shown in Figure 6.6(a). This does not happen for uncompressed host signals, because the relatively continuous nature of the host signal helps retain some of the fingerprint information even after the fingerprinted signal goes through compression. Inspired by this observation, we introduce a new fingerprinting technique that mimics the case of uncompressed host signals by adding a pseudo-random dither sequence to the compressed host signal before embedding fingerprints. We will show, through analysis and simulation that the proposed scheme has higher collusion resistance.

6.4.1 Fingerprint Embedding

We illustrate the proposed technique using an example. We model the probability distribution function (p.d.f.) of the host signal prior to quantization by a Laplacian distribution, which has been shown to be a good model for DCT coefficients of natural images [44,74]. Figure 6.7 shows the p.d.f. of the host signal before quantization, $f_{S_j^{(0)}}$, and after quantization, f_{S_j}. We make the quantized host signal appear more continuous to the fingerprint embedder by convolving it with a narrow rectangular distribution to approximate the distribution of the host signal before quantization.

Let $\mathbf{d} = [d_1, d_2, \ldots, d_M]$ denote *i.i.d.* random variables uniformly distributed over $[-\frac{\Delta}{2}, \frac{\Delta}{2})$, and let $S'_j = S_j + d_j$. The p.d.f. of S'_j is given as $f_{S'_j}(x) = f_{S_j}(x) \otimes$

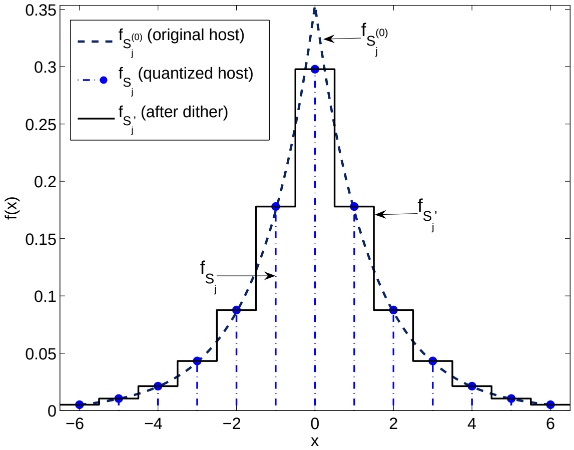


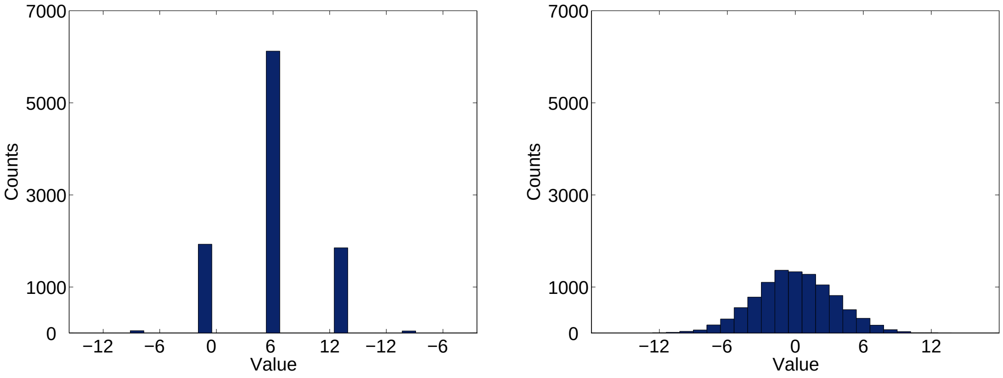
Figure 6.7: Distribution of the host signal before quantization, after quantization and after adding dither.

$f_{d_j}(x)$, where f_{d_j} is the p.d.f. of d_j and $\otimes$ denotes convolution. The p.d.f. $f_{S_j'}(x)$ is a staircase function that approximates the original host distribution as shown in Figure 6.7. We refer to the signal $\mathbf{d}$ as *Anti-Collusion Dither (ACD)*, and as will be shown subsequently, this dither signal that is added to the quantized host signal helps improve the collusion resistance of the system.

We construct the fingerprinted signal, $\mathbf{X'}^{(k)}$, by adding the ACD signal and the Gaussian fingerprint $Q_j^{(k)}$ to the quantized host signal, and then applying re-quantization:

$$X'^{(k)}_j = \text{round}\left(\frac{S_j + d_j + Q_j^{(k)}}{\Delta}\right) \times \Delta. \qquad (6.5)$$

As the S_j are multiples of Δ, the effective changes, $\mathbf{W_d}^{(k)}$, made on the signal sent to the k^{th} user is given by $W_{dj}^{(k)} = \text{round}\left(\frac{d_j + Q_j^{(k)}}{\Delta}\right) \times \Delta$, with its energy constrained by $E[\|\mathbf{W_d}^{(k)}\|^2] \leq M \cdot D(\Delta)$. Upon obtaining the attacked signal $\mathbf{Z}$, the detector



(a) Without ACD.

(b)With ACD.

Figure 6.8: Distribution of the effective fingerprint for a single user (a) without ACD and (b) with ACD for $\Delta = 6$.

extracts the fingerprint and declares user q to be guilty if

$$\begin{aligned} q &= \arg \max_{k=1,2,\ldots,N} \frac{1}{M} \langle h(\mathbf{Z} - \mathbf{S} - \mathbf{d}), \mathbf{W_d}^{(k)} - \mathbf{d} \rangle, \\ &= \arg \max_{k=1,2,\ldots,N} \frac{1}{M} \langle h(\mathbf{Z} - \mathbf{S} - \mathbf{d}), \mathbf{W_d}^{(k)} \rangle, \end{aligned} \tag{6.6}$$

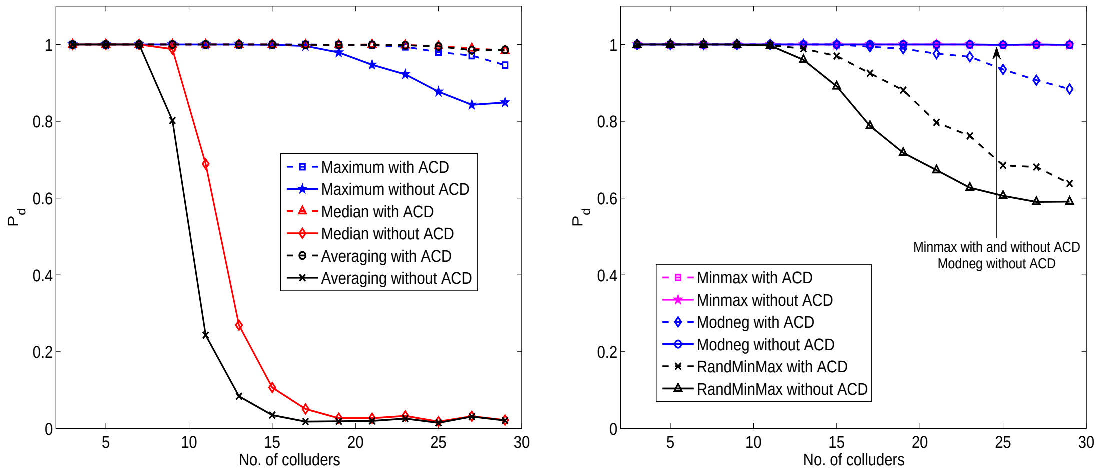
where $h(\cdot)$ is the preprocessing applied to make the histogram of the test signal symmetric around zero as explained in Section 6.2. The second equation is obtained from the fact that the dither term is independent of the user index k and hence does not affect the maximization.

6.4.2 Colluder Identification Accuracy

We test the fingerprinting system with the proposed ACD technique using the settings described in Section 6.3. Figure 6.8(a) and (b) show the histograms of the effective fingerprint for a single user without ACD ($\mathbf{W}$) and with ACD ($\mathbf{W_d} - \mathbf{d}$), respectively. We observe that the effective fingerprint $\mathbf{W_d}-\mathbf{d}$ is now more continuous in nature, and thus the collusion resistance is improved [87]. Figure 6.9(a) and (b) compare the probability of catching one colluder P_d, with and without ACD for $\Delta = 6$ at a WNR of 0dB. We observe that the probability of catching one colluder has increased significantly for fingerprinting with ACD as opposed to fingerprinting without ACD. The collusion resistance against averaging and median attacks is now quadrupled and the system with ACD can resist over 30 attackers' collusion compared to only 7 when without ACD. We observe that the performance against minimum and maximum attacks also improves and the P_d against Min-Max attack continues to be close to 1. However, the probability of detection under modified negative attack is reduced slightly.

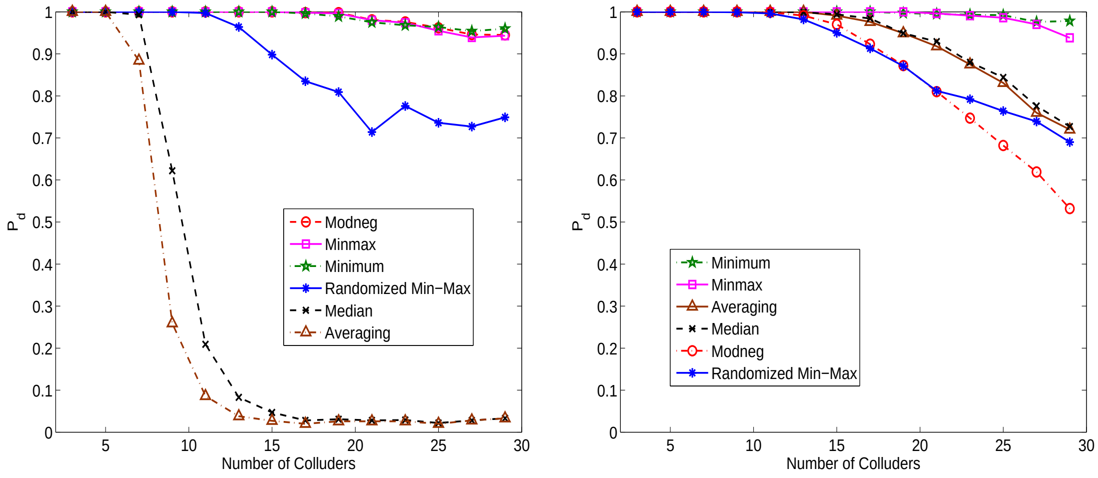
A similar performance improvement for ACD is observed at lower WNR levels. For example, when WNR = -5dB, the collusion resistance without dithering reduces further to only 7 colluders, whereas fingerprinting with ACD can resist around 13 colluders. For averaging attack with coalition size larger than 5, P_d without ACD reduces sharply to close to 0, whereas P_d with ACD degrades gracefully and is around 0.6 for 30 colluders.

We also compare the performance improvement of ACD under different attacks when the overall distortion introduced by the collusion attacks are kept the same. Using a similar evaluation framework as in [98], the power of the additional noise $\mathbf{n}$ added to the attacked signal after collusion is varied, such that the overall distortion introduced into the signal by the different attacks are approximately equal. Figure 6.10 shows the probability of detection for fingerprinting with and without



(a) (b)

Figure 6.9: Performance improvement for fingerprinting using ACD at WNR = 0dB.



(a) (b)

Figure 6.10: Comparison of P_d when the overall distortion introduced into the signal is approximately equal under different attacks for (a) Fingerprinting without ACD and (b) Fingerprinting with ACD.

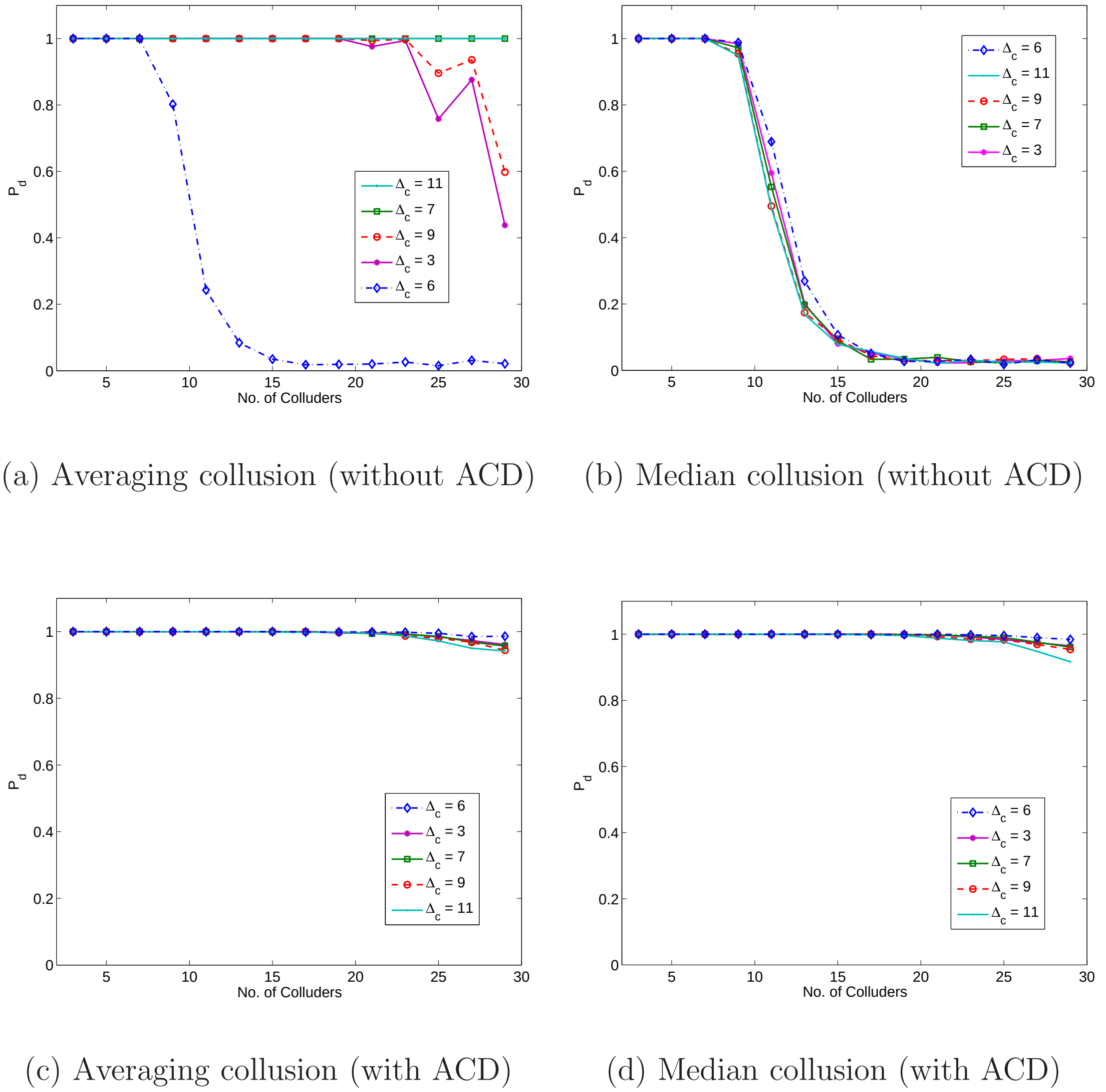
ACD when the overall distortion introduced into the signal is approximately 4 times the power of the watermark, or equivalently, the overall watermark to noise ratio is approximately -6dB. From Figure 6.10(a), we observe that the attacks follow the same trend under fingerprinting without ACD as when a constant amount of noise was added (Figure 6.4). Due to the higher noise power, the averaging attack leads to a lower probability of detection and the overall collusion resistance is now reduced to 5 colluders. Under fingerprinting with ACD, we observe that the overall collusion resistance is 13 colluders and the probability of detection is increased for all collusion attacks except the modified negative attack. As shall be seen from the theoretical analysis in Chapter 7, the modified negative attack leads to a higher variance in the detection statistics and hence a lower probability of detection. This suggests that the distortion constraints in higher fidelity applications may prevent the attackers from employing the modified negative attack, although if the overall allowable distortion in the attacked signal is large, modified negative would be the colluders' preferred attack against ACD-based fingerprinting systems.

6.4.3 Colluders' Choice of Quantization Step Size Δ_c

So far, we have only considered the case where the attackers use the same quantization step size to requantize the colluded signal as that chosen by the content owner, i.e. $\Delta_c = \Delta$. In this subsection, we examine the case where the attackers choose a different quantization step size $\Delta_c \neq \Delta$. We use the non-blind correlation detector to identify one guilty user. In our experiments, we set the fingerprint length

$M = 10000$ and the number of users is $N = 1024$. The host signal is modelled after the DCT coefficients of the AC$_{1,1}$ band in the Lena image, such that the host follows a Laplacian distribution with variance 61. The host is then quantized with step size $\Delta = 6$ and the fingerprint is embedded. The embedding power is chosen to satisfy the maximum distortion constraint $D(\Delta) = 15$.

Figure 6.11 shows the probability of catching at least one colluder for various choices of the quantization step size by the colluders at additive noise power of 0dB. Figure 6.11(a) depicts the results under averaging collusion for fingerprinting without ACD. From the figure, we observe that attackers' choice of $\Delta_c = \Delta = 6$ leads to the lowest probability of detection. Further, we see that the probability of detection for the cases when Δ_c and Δ are co-prime is higher than for other choices of Δ. Figure 6.11(b) shows the results under median collusion for fingerprinting without ACD. We observe that the results are similar regardless of the choice of Δ_c by the colluders. Figure 6.11(c) and (d) show the results for the case of fingerprinting using ACD under averaging and median collusion, respectively. The collusion resistance of the system improves significantly for some choices of Δ_c under averaging collusion and for all Δ_c under median attack. More importantly, the performance is now comparable for all Δ_c. Thus, from an attacker's perspective, the choice of $\Delta_c = \Delta$ is no worse than any other choice of Δ_c in terms of P_d and is preferable based on bandwidth and quality considerations, as discussed earlier. In view of these results, in the remainder of the chapter, we only consider the case of $\Delta_c = \Delta$.



(a) Averaging collusion (without ACD)

(b) Median collusion (without ACD)

(c) Averaging collusion (with ACD)

(d) Median collusion (with ACD)

Figure 6.11: Probability of successfully catching one colluder when $\Delta_c \neq \Delta$ and additive noise power is equal to the watermark power for (a) averaging and (b) median collusion for fingerprinting without ACD, and (c) averaging and (d) median collusion for fingerprinting with ACD.

6.5 Simulation Results on Images

So far, we have focused on the case of one channel, which may represent one feature or one frequency band. We now examine the performance of the proposed technique on actual images, where the fingerprint is embedded in a set of different frequency bands.

We use images of size 320 × 320 compressed with JPEG quality factor of 75 as host images, as this compression setting is the default in many applications to provide a good tradeoff between perceptual quality and bandwidth. The fingerprint is embedded in the middle frequency bands of the 8x8 block DCT domain, guided by the Human Visual System (HVS) model [66]. The embedding distortion is measured by the Peak Signal to Noise Ratio (PSNR) defined as $PSNR = 10 \log \frac{255^2}{\|I - I_w\|^2}$, where I is the original image, I_w is the fingerprinted image and $\|I - I_w\|^2 = \sum_{m,n}(I(m,n) - I_w(m,n))^2$. The embedding PSNR is set to 42dB and the fingerprint energy in each frequency band is allocated based on the corresponding quantization step size.

Figure 6.12 shows the fraction of the coefficients used for embedding that belong to a channel with a given quantization step size Δ for the Lena test image. From the figure, we observe that approximately 25% of the coefficients used for embedding have $\Delta = 6$. The other coefficients come from channels with larger quantization step sizes, or stronger compression, and hence we expect the overall performance of both the fingerprinting schemes to be lower than that obtained for the one channel $\Delta = 6$ case presented in Sections 6.3 and 6.4.

Figure 6.13 shows parts of Lena and baboon images before and after fingerprint

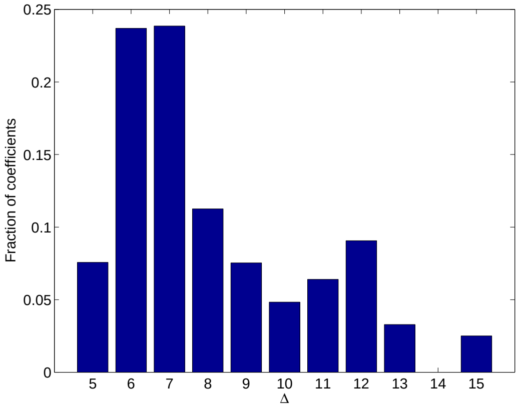
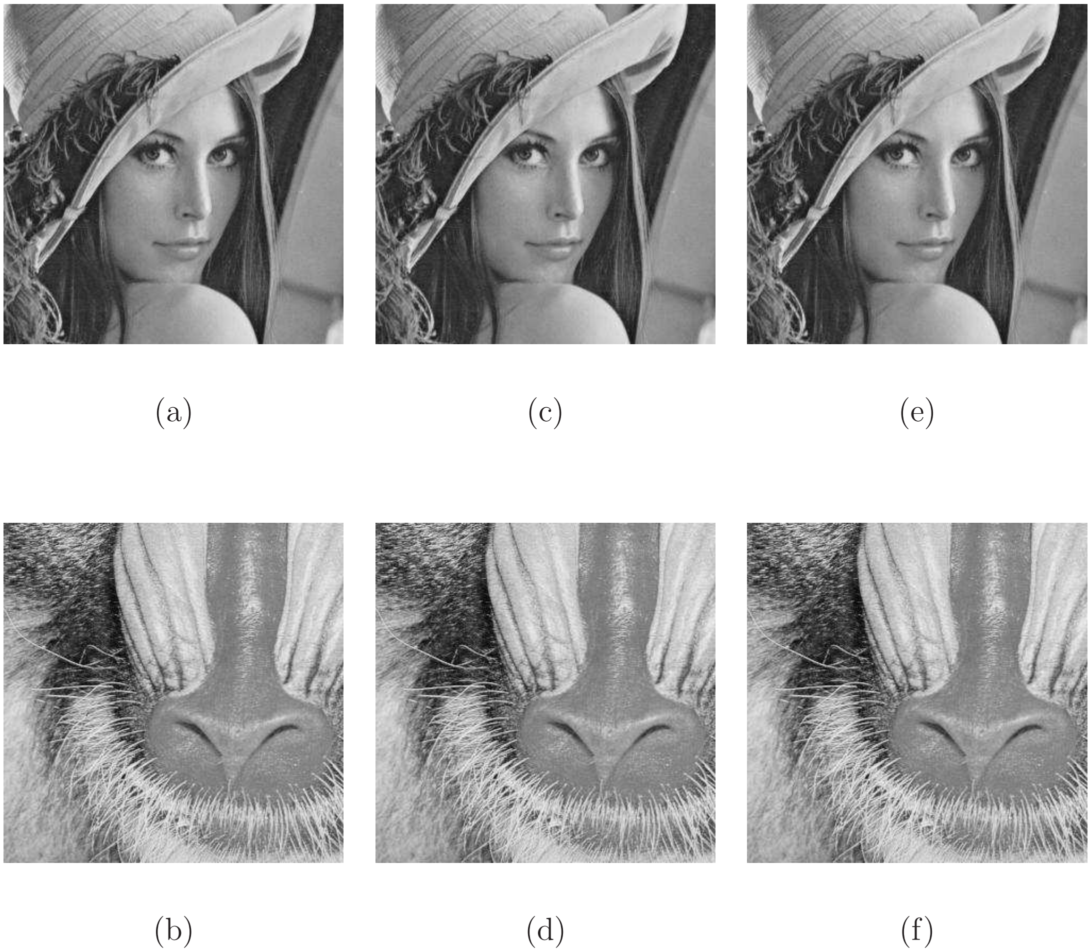


Figure 6.12: Fraction of the coefficients used for embedding belonging to channels with different quantization step sizes for the cropped Lena image.

embedding, with and without the proposed ACD technique. Figure 6.13(a) and (b) are the original JPEG compressed images, Figure 6.13(c) and (d) are images fingerprinted using quantized Gaussian fingerprints without ACD, and Figure 6.13(e) and (f) are images fingerprinted using the proposed ACD technique. We observe no perceptual difference between the original compressed image and the two fingerprinted images. Table 6.1 compares the file sizes of the resulting JPEG images. The numbers in parenthesis indicate the percentage change in the file size. From the table, we see that the file size of the JPEG compressed fingerprinted image is not significantly different from the original image, and thus the bandwidth required for transmission in our proposed scheme will be approximately the same as the requirement for transmitting the original host.

Figure 6.14 shows the average probability of catching at least one colluder for the various attacks described in Eqn. (6.2) when the additive noise power is equal to the watermark power. The simulation results are similar to the results



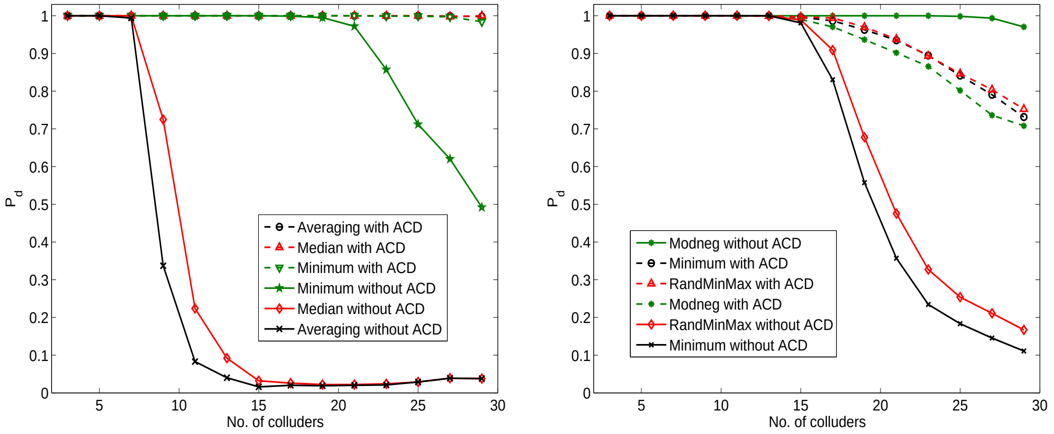
(a) (c) (e)

(b) (d) (f)

Figure 6.13: Parts of Lena and baboon images before and after fingerprint embedding. (a) and (b) Original compressed images at a JPEG quality factor of 75. (c) and (d) Fingerprinted images without ACD and (e) and (f) fingerprinted images with ACD. (Embedding PSNR = 42dB.)

Table 6.1: Comparison of file size (in Bytes) of original and fingerprinted JPEG images.

Image	Original Size	Fingerprinted image size (without ACD)	Fingerprinted image size (with ACD)
Lena	16,007	15,973 (-0.21%)	16,014 (+0.04%)
Baboon	24,592	24,631 (+0.15%)	24,588 (-0.02%)
Barbara	18,921	18,938 (+0.09%)	18,963 (+0.22%)



(a) (b)

Figure 6.14: Simulation results for images: average probability of catching at least one colluder for fingerprinting with and without ACD under various attacks for the Lena test image.

presented in Section 6.3 for the one channel case, but the exact probabilities seen in Figure 6.14 are slightly lower due to the different quantization step sizes of different frequency locations in the DCT of the image. From Figure 6.14, we observe that using ACD, the overall collusion resistance has approximately doubled from 7 to 13 colluders. The probability of detection has increased for all attacks, except the modified negative attack. Under the modified negative attack, the probability of detection reduces by around 20% for attack by 20 colluders. However, as discussed in Section 7.2, the distortion introduced by this attack under ACD is higher than without ACD. The increase in P_d for the randomized negative attack is also lower compared to the other attacks, due to the higher distortion introduced by this attack.

6.6 Chapter Summary

In this chapter, we examined the problem of collusion resistant fingerprint design for compressed multimedia. Our studies indicate that adding Gaussian spread spectrum fingerprints, that are effective for uncompressed signals, and recompressing the fingerprinted signal leads to a low collusion resistance. This can be mainly attributed to the discrete nature of the fingerprint. After multi-user collusion attacks, the fingerprint traces are completely removed from the media, making it difficult to identify users participating in the collusion. We found that the averaging and median attacks are particularly effective from the colluders' perspective, and the fingerprint can be removed by averaging around 10 copies.

To remedy this shortcoming, we propose an Anti-Collusion Dither technique to improve the collusion resistance. A pseudorandom dither signal is added to the compressed host before fingerprint embedding to make it appear more continuous from the embedder's perspective. The dither also makes the effective fingerprint more continuous, thereby improving its collusion resistance. Simulation results using JPEG compressed images demonstrate that the file-size of the image is not significantly altered by the fingerprinting and the perceptual quality is preserved. The proposed ACD technique can approximately quadruple the collusion resistance under the averaging and median attacks as compared to fingerprinting using quantized Gaussian fingerprints without ACD. Similar results have been obtained for the minimum, maximum, min-max and randomized min-max collusion attacks.

In large-scale applications involving compressed multimedia, the proposed technique can be combined with other coding schemes to create efficient and scalable fingerprinting systems with high collusion resistance. For example, for uncompressed hosts, the fingerprint construction and detection technique of [34] builds on Gaussian spreading sequences and employs ECC-based construction to support millions of users. Analogously, the proposed ACD-based fingerprinting can be used in conjunction with error correcting codes, Boneh-Shaw codes, or Tardos codes for implementing fingerprinting schemes for compressed multimedia.

Chapter 7

Theoretical Analysis of Fingerprinting Techniques for Compressed Multimedia

In the previous chapter, we have introduced the technique of ACD and demonstrated through simulations that the probability of identifying a colluder is higher for fingerprinting with ACD compared to without ACD. In this section, we provide a theoretical analysis of the two schemes for fingerprinting compressed multimedia from different perspectives. First, we derive expressions for the probability of detection for the two fingerprinting schemes. We then bring in an estimation viewpoint from the colluders' perspective, to compare the accuracy with which the attackers can estimate the host signal. We also evaluate fingerprinting with and without ACD from an information theoretic viewpoint in terms of the maximum number of distinct users that the system can accommodate, such that the asymptotic error probability goes to zero.

7.1 Probability of Detection

We now derive the probability mass function (p.m.f.) of the fingerprints and compute the probability of detecting one of the guilty users after a collusion attack [85]. Let $h(\cdot)$ denote the preprocessing applied to the test signal to make its distribution symmetric around zero, as described in Section 6.2. The test signal used by the detector can be represented as

$$h(\mathbf{Z} - \mathbf{S}) = h(\mathbf{V} + \mathbf{n} - \mathbf{S}) = h(g(\{\mathbf{W}^{(k)}\}_{k \in \mathcal{U}_K})) + \mathbf{n},$$

since the attacks satisfy $g(\{X_j^{(k)}\}_{k \in \mathcal{U}_K}) = S_j + g(\{W_j^{(k)}\}_{k \in \mathcal{U}_K})$. We have also used the approximation $h(\mathbf{n}) \approx \mathbf{n} - E[n]\mathbf{1} = \mathbf{n}$, where $\mathbf{1}$ represents a vector of all ones and the fact that the noise n has mean equal to zero. Denoting $g'(\cdot) = h(g(\cdot))$, we have the detection statistic for user α, $T^{(\alpha)} = \frac{1}{M} \sum_{j=1}^{M} (g'(\{W_j^{(k)}\}_{k \in \mathcal{U}_K}) + n_j) \times W_j^{(\alpha)}$. As the components $W_j^{(\alpha)}$ are *i.i.d.*, from the Central Limit Theorem, $T^{(\alpha)}$ approaches a Gaussian distribution when the fingerprint length M is large. Further, the mean and variance of the Gaussian distribution are independent of j due to the *i.i.d.* property, and depend only on whether α belongs to the set of colluders $\mathcal{U}_K$ or not. After dropping the subscript j to simplify the notation, the mean and variance of

$T^{(\alpha)}$ for $\alpha \notin \mathcal{U}_K$ are given by

$$\begin{aligned}
\text{mean: } \mu_0 &= E[g'(\{W^{(k)}\}_{k\in\mathcal{U}_K}) + n]E[W^{(\alpha)}] \quad = \quad 0, \\
\text{variance: } \sigma_0^2 &= \frac{1}{M}E[((g'(\{W^{(k)}\}_{k\in\mathcal{U}_K}) + n)W^{(\alpha)})^2], \\
&= \frac{1}{M}E[(g'(\{W^{(k)}\}_{k\in\mathcal{U}_K}))^2 + n^2]E[(W^{(\alpha)})^2].
\end{aligned}$$

Here, the equalities follow due to the independence assumption and that $W^{(\alpha)}$ has a zero mean. Similarly, for $\alpha \in \mathcal{U}_K$,

$$\begin{aligned}
\mu_1 &= E[g'(\{W^{(k)}\}_{k\in\mathcal{U}_K})W^{(\alpha)}] + E[n]E[W^{(\alpha)}], \\
&= E[g'(\{W^{(k)}\}_{k\in\mathcal{U}_K})W^{(\alpha)}],
\end{aligned}$$

as the noise n is independent of $W^{(\alpha)}$ and has zero mean. The variance of the detection statistic is given as

$$\begin{aligned}
\sigma_1^2 &= \frac{1}{M}\text{Var}([g'(\{W^{(k)}\}_{k\in\mathcal{U}_K}) + n]W^{(\alpha)}) \\
&= \frac{1}{M}\left(E[(\{g'(\{W^{(k)}\}_{k\in\mathcal{U}_K}) + n\}W^{(\alpha)})^2] - \mu_1^2\right) \\
&= \frac{1}{M}\left(E[(g'(\{W^{(k)}\}_{k\in\mathcal{U}_K})W^{(\alpha)})^2] + E[n^2]E[(W^{(\alpha)})^2] - \mu_1^2\right).
\end{aligned}$$

The quantities μ_1, σ_0^2, and σ_1^2 can be computed from the joint distribution of $g(\{W^{(k)}\}_{k\in\mathcal{U}_K})$ and $W^{(\alpha)}$, $\alpha \in \mathcal{U}_K$ and the distribution of $g(\{W^{(k)}\}_{k\in\mathcal{U}_K})$. The probability of successfully catching one colluder is then given by the probability that the detection statistic for one of the colluders is larger than the detection statistics of all the innocent users:

$$P_d = \Pr(\max_{k\in\mathcal{U}_K} T^{(k)} > \max_{\alpha\notin\mathcal{U}_K} T^{(\alpha)}).$$

7.1.1 Analysis of Quantized Gaussian Fingerprints

Consider the scenario of quantized Gaussian fingerprints under the averaging attack. Let $W' = \frac{1}{K}\sum_{k\in\mathcal{U}_K} W^{(k)}$ and W^{avg}=round $\left(\frac{W'}{\Delta}\right)\times\Delta$. Then,

$$\Pr(W^{\text{avg}} = m\Delta) = \Pr(W' \in \mathcal{I}_m), \tag{7.1}$$

where $\mathcal{I}_m = \left[m\Delta - \frac{\Delta}{2}, m\Delta + \frac{\Delta}{2}\right)$. The characteristic function of W', $M'(t) = E[\exp(itW')]$ is related to the characteristic function of $W^{(\alpha)}$, $M(t)$, as $M'(t) = [M(\frac{t}{K})]^K$, where K is the number of colluders. The probability mass function (p.m.f.) of W' is then given as

$$\Pr\left(W'{=}m\frac{\Delta}{K}\right) = \frac{1}{2\pi K}\int_{-\pi K}^{\pi K} \exp\left(-\frac{itm\Delta}{K}\right)\left[M\left(\frac{t}{K}\right)\right]^K \mathrm{d}t.$$

The joint p.m.f. $\Pr(W^{\text{avg}} = m\Delta, W^{(\alpha)} = n\Delta), \alpha \in \mathcal{U}_K$, can be written as the product of the conditional distribution $\Pr(W^{\text{avg}} = m\Delta | W^{(\alpha)} = n\Delta)$ and the marginal distribution $\Pr(W^{(\alpha)} = n\Delta)$. Here, the conditional distribution can be computed as

$$\begin{aligned}\Pr(W^{\text{avg}} = m\Delta | W^{(\alpha)} = n\Delta) &= \Pr\left(W' \in \mathcal{I}_m | W^{(\alpha)} = n\Delta\right) \\ &= \Pr\left(\frac{1}{K}\sum_{k\in\mathcal{U}_K\backslash\{\alpha\}} W^{(k)} \in \mathcal{I}_{m,n}\right)\end{aligned}$$

where $\mathcal{I}_{m,n} = \left[m\Delta - \frac{\Delta}{2} - \frac{n\Delta}{K}, m\Delta + \frac{\Delta}{2} - \frac{n\Delta}{K}\right)$ and

$$\Pr\left(\frac{1}{K}\sum_{k\in\mathcal{U}_K\backslash\{\alpha\}} W^{(k)}{=}\frac{m\Delta}{K}\right) = \frac{1}{2\pi K}\int_{-\pi K}^{\pi K} \exp\left(-\frac{itm\Delta}{K}\right)\left[M\left(\frac{t}{K}\right)\right]^{K-1} \mathrm{d}t.$$

The p.m.f. of the colluded fingerprints under other attacks can be derived similarly. The detailed derivations are omitted here due to space constraints.

Based on these derivations, we compute the probability of detection P_d for the settings described in Section 6.3. We choose the fingerprint length $M = 10000$, the

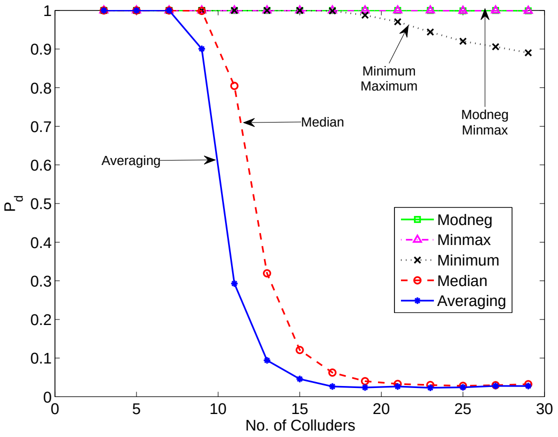


Figure 7.1: Probability of catching one colluder using traditional Gaussian based fingerprints at WNR = 0dB, 1024 users, $M = 10^4$, $D(\Delta) = 15$.

number of users $N = 1024$, and the distortion $D(\Delta) = 15$. To obtain numerical results, the integrals in the expressions for the probability distributions are evaluated using the trapezoidal rule. Figure 7.1 shows the probability of successfully catching one colluder (P_d) versus the number of users participating in the collusion for various attacks. The power of additive noise is set to be the same as the power of the watermark, *i.e.*, the Watermark-to-Noise Ratio (WNR) is set to 0 dB for each of the attacks. From the figure, we observe that the probability of catching a guilty user is the lowest for averaging attack and the system can resist only 7 colluders with $P_d \approx 1$. The median attack is also very effective at removing traces of the fingerprints. The minimum and maximum attacks are less effective, and the modified negative and the min-max attacks are the least effective attacks. These results agree very well with that obtained through simulations, presented in Section 6.3.

7.1.2 Performance Analysis under Anti-Collusion Dithering

With Anti-Collusion Dithering, we follow a similar approach to compute the probability of catching one colluder. We illustrate our analysis by deriving the p.m.f.s under the averaging attack. Let $W_d^{(\alpha)} = \mathrm{round}\left(\frac{d+Q^{(\alpha)}}{\Delta}\right)\times\Delta$, $W_d' = \frac{1}{K}\sum_{k\in\mathcal{U}_K} W_d^{(k)}$, and $W_d^{\mathrm{avg}} = \mathrm{round}\left(\frac{W_d'}{\Delta}\right)\times\Delta$, where we have dropped the subscript j due to the *i.i.d.* property and Q is a Gaussian random variable. For the averaging attack, we have:

$$\begin{aligned}
\Pr\left(W_d^{\mathrm{avg}} = m\Delta\right) &= \Pr\left(W_d' \in \mathcal{I}_m\right) \\
&= \frac{1}{\Delta}\int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} \Pr\left(W_d' \in \mathcal{I}_m | d = x\right) \mathrm{d}x \\
&= \frac{1}{\Delta}\int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} \Pr\left(\frac{1}{K}\sum_{k\in\mathcal{U}_K} \mathrm{round}\left(\frac{x+Q^{(k)}}{\Delta}\right) \in \mathcal{I}_m\right) \mathrm{d}x
\end{aligned} \tag{7.2}$$

The joint p.m.f. $\Pr(W_d^{\mathrm{avg}} = m\Delta, W_d^{(\alpha)} = n\Delta)$ can be computed by integrating the product $\Pr(W_d^{\mathrm{avg}} = m\Delta | W_d^{(\alpha)} = n\Delta, d = x)\Pr(W_d^{(\alpha)} = n\Delta | d = x) f_d(d = x)$ over the range $x \in [-\frac{\Delta}{2}, \frac{\Delta}{2}]$. The conditional distribution is computed as

$$\begin{aligned}
\Pr(W_d^{\mathrm{avg}} = m\Delta | W_d^{(\alpha)} = n\Delta, d = x) &= \Pr\left(W_d' \in \mathcal{I}_m | W_d^{(\alpha)} = n\Delta, d = x\right) \\
&= \Pr\left(\frac{1}{K}\sum_{k\in\mathcal{U}_K\setminus\{\alpha\}} \mathrm{round}\left(\frac{x+Q^{(k)}}{\Delta}\right) \in \mathcal{I}_{m,n}\right).
\end{aligned}$$

The last term can be computed by first obtaining the characteristic function and then computing the p.m.f. from the characteristic function as illustrated earlier. A similar analysis can be carried out for the other nonlinear attacks.

Figure 7.2 shows the probability of catching one colluder versus the number of colluders for fingerprinting using ACD. We observe that the collusion resistance

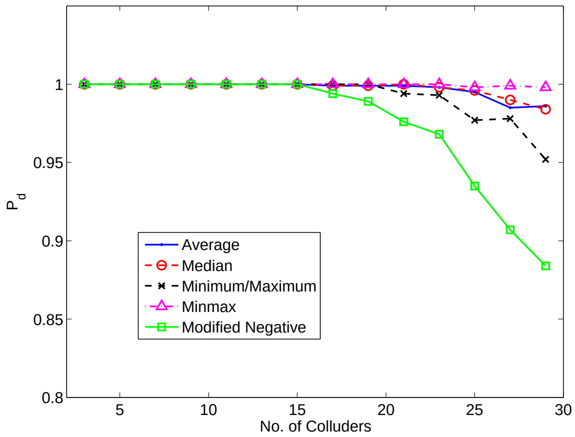


Figure 7.2: Probability of catching one colluder for fingerprinting with ACD at WNR $= 0$dB, 1024 users, $M = 10^4$, $D(\Delta) = 15$.

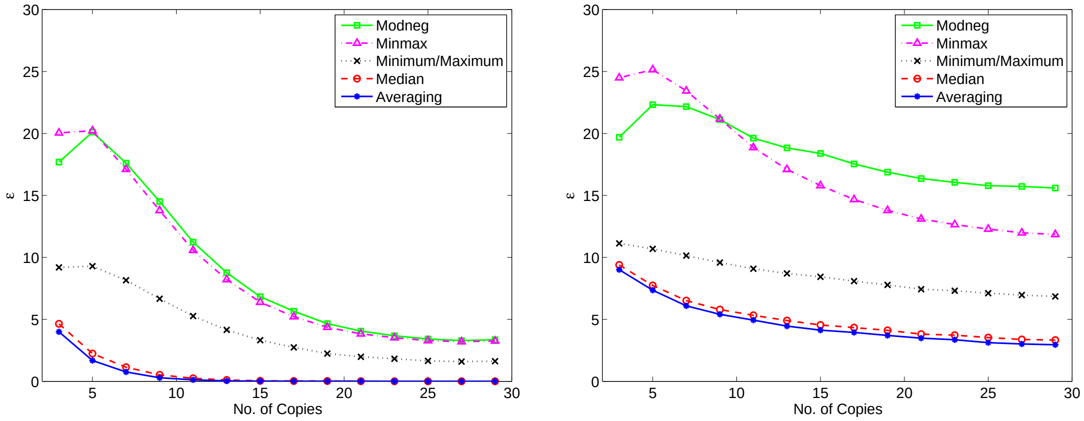
against averaging and median attacks has now quadrupled from 7 to approximately 30 colluders. The collusion resistance for the minimum and maximum attacks has also increased. As before, we observe that the probability of detection has slightly reduced for the modified negative attack. These results are consistent with the simulation results presented in Section 6.4.

7.2 Estimation Accuracy of Various Collusion Attacks

Collusion attacks to remove traces of the fingerprints can be formulated as the colluders' attempt to estimate the host signal given their fingerprinted versions. The accuracy with which attackers can estimate the host signal was suggested as one of the criteria for determining optimal collusion attacks for uncompressed domain fingerprinting in [41], but no explicit evaluation of the estimation accuracy was provided for different collusion attacks. In this subsection, we examine collusion from the attackers' perspective and evaluate the effectiveness of collusion attacks on compressed domain fingerprinting systems in terms of the accuracy of estimating the host signal. We provide an explicit evaluation and quantitative comparison of the estimation accuracy of different collusion attacks.

As before, denote the host signal sample by S_j, and the fingerprinted signal for user k by $X_j^{(k)}$. Let the estimate of the host signal be represented as $\hat{S}_j = G'(\{X_j^{(k)}\}_{k \in \mathcal{U}_K})$, where $G'(\cdot)$ is some suitable estimator. The accuracy of the estimate, or equivalently, the effectiveness of the collusion attack can be measured in terms of the Mean Squared Error (MSE), given by $\varepsilon = E[(S_j - \hat{S}_j)^2]$. The collusion attacks defined in (6.2) can be considered as estimators if we set $G'(\cdot) = h(g(\cdot))$ for the collusion attack $g(\cdot)$. These estimators satisfy $G'(\{X_j^{(k)}\}_{k \in \mathcal{U}_K}) = S_j + G'(\{W_j^{(k)}\}_{k \in \mathcal{U}_K})$. Thus, the MSE of estimation simply becomes the variance of the colluded fingerprint which can be computed using the distribution of the colluded signal as derived in Section 7.1.

Figure 7.3 shows the MSE of various estimators as a function of the number of colluders for the experimental setup described in Section 7.1. From Figure 7.3(a), we see that averaging collusion has the lowest MSE, followed by median, minimum, min-max, and modified negative attacks for fingerprinting using independent Gaussian based fingerprints. Figure 7.3(b) shows the corresponding MSEs under ACD fingerprinting. Comparing Figure 7.3(a) and Figure 7.3(b) we observe that the MSEs of all the estimators are significantly higher with ACD than without the dithering. The MSE of estimation for the averaging and median attacks, which are close to zero with just 11 fingerprinted copies for fingerprinting without ACD, remain non-



(a) Without ACD

(b) With ACD.

Figure 7.3: MSE (ε) of various estimators for fingerprinting (a) without ACD and (b) with ACD for $\Delta = 6$.

zero even with 30 colluders for fingerprinting with ACD. The MSE for other attacks approximately triples and thus the attacks are not effective at estimating the host signal for the ACD fingerprinting system.

The distortion introduced by the collusion attack, measured with respect to the host, is given by the second moment of the colluded fingerprint and is equal to the sum of the MSE and the square of the mean. For averaging, median, minmax and modified negative, the mean of the colluded fingerprint is zero and the distortion introduced is equal to the MSE. For the minimum and maximum attacks, the colluded fingerprint has non-zero mean and the overall distortion increases with the number of colluders.

From Figure 7.3(a), we observe that averaging introduces the lowest distortion for fingerprinting without ACD. The averaging attack also has the lowest probability of detection for a given distortion, as shown in Figure 6.10(a). Thus, the averaging attack is the best choice for the colluders under fingerprinting without ACD. When ACD is used, the colluders' strategy is not so simple. Although Figure 7.3(b) shows that the lowest distortion is again introduced by the averaging attack and that the modified negative attack has the highest distortion, we recall from Figure 6.10(b) that the modified negative attack has a lower probability of detection at a given level of distortion. This is because, unlike the averaging and median attacks which aim to estimate the host signal, the modified negative attack attempts to move the attacked signal in a direction opposite to that of a majority of the colluders' fingerprints. This results in a higher overall distortion, as shown in Figure 7.3. Despite this high distortion, for Gaussian fingerprints, due to the symmetry of the colluders' fingerprints, the resultant attacked signal may still have sufficient correlation with some colluder's fingerprint, enabling us to catch one of the colluders with high probability. However after applying ACD, the additive dither reduces the statistical symmetry among the colluders' effective fingerprint. This reduced symmetry makes the resulting direction of the modified negative attack less likely to be correlated with the colluders' fingerprint and reduces the probability of detection. Therefore, the adversaries' best strategy in an ACD fingerprinting system depends on the distortion allowable by their attacks. If the distortion introduced by the modified negative attack is within the attackers' distortion constraint, the modified negative attack would be more favorable from the colluders' perspective. On the other hand, if the allowable distortion is small, it is advantageous for the attackers to choose averaging collusion instead.

7.3 Comparison based on Mutual Information

In this subsection, we leverage the capacity results for fingerprinting finite alphabet from the literature to analyze the performance enhancement achieved by the proposed ACD techniques from an information theoretic standpoint.

For fingerprinting signals drawn from finite alphabets, the private fingerprinting game with non-blind detection was considered in [75], and the public fingerprinting game with blind detection was analyzed in [97]. In practice, due to bit rate and dynamic range limitations, compressed host signals can only take values from a finite set and the results from [75] are applicable in this case. The formulation in [75] considers a fingerprinting code of length M with N sequences such that $N = 2^{MR}$, where R is the rate of the code. The fingerprinted sequences are generated from the *i.i.d.* host signal $\mathbf{S}$ under a specified distortion constraint and the colluded version created by K users is also subject to a similar distortion constraint. A fingerprinting rate R is defined as "achievable" if the probability of not detecting any of the colluders tends to zero as the length of the fingerprint M approaches infinity.

The fingerprinting capacity as a function of the distortion constraints and the probability distribution of the host is defined to be the supremum of all achievable rates. Under memoryless collusion attacks by K colluders, and fingerprinting using constant composition codes, the capacity C_{fp} is shown to be related to the conditional mutual information as

$$C_{fp} = \max_{p(X|S)} \min_{p(Z|X_1,X_2,\ldots,X_K)} \frac{1}{K} I(Z; X_1, X_2, \ldots, X_K|S). \tag{7.3}$$

Here $p(X|S)$ is the conditional p.m.f. of the fingerprinted signal given the host signal, $p(Z|X_1, X_2, \ldots, X_K)$ is the conditional p.m.f. of the attacked signal given the fingerprints of the colluders, and $I(X;Y)$ denotes the mutual information between two random variables X and Y. The evaluation of the fingerprinting capacity (Eqn. (7.3)) is non-trivial even for simple settings. To the best of our knowledge, the capacity has not been determined for any host distribution with finite support, except for the case of binary hosts, and consequently capacity achieving fingerprint designs are unknown.

Inspired by connections drawn by the theoretical studies between the best achievable fingerprinting rate and the conditional mutual information, we propose to employ mutual information to guide the development and analysis of practical fingerprinting algorithms for quantized hosts. For a given fingerprinting scheme and a given attack, it can be shown that the rate $\frac{1}{K}I(Z; X_1, X_2, \ldots, X_K|S)$ provides a tight upper bound on the maximum number of users that the fingerprinting system can support [75]. This rate, denoted as R_{max}, is thus an indicator of the collusion resistance of the fingerprinting scheme under the given attack, as a higher value of R_{max} suggests that a larger number of users can be supported by the corresponding embedding scheme. We now use R_{max} to compare the two designs of fingerprinting codes for compressed hosts, namely, quantized gaussian fingerprints with and without ACD fingerprinting.

Consider the noise-free averaging collusion attack, which can be represented as

$$p(Z = z|X_1, X_2, \ldots, X_K)) = \delta(z - X^{\text{avg}}), \tag{7.4}$$

where

$$X^{\text{avg}} = \text{round}\left(\frac{X_1 + X_2 + \ldots + X_k}{K\Delta}\right) \times \Delta \quad \text{and} \quad \delta(z) = \begin{cases} 1 & z = 0 \\ 0 & z \neq 0 \end{cases}.$$

The maximum fingerprinting rate in terms of the conditional mutual information under this attack is given by

$$\begin{aligned} R^{\text{avg}}_{\text{max}} &= \frac{1}{K} I(Z; X_1, X_2, \ldots, X_K | S), \\ &= \frac{1}{K}[H(Z|S) - H(Z|X_1, X_2, \ldots, X_K, S)], \\ &= \frac{1}{K} H(Z|S), \end{aligned} \tag{7.5}$$

where $H(X)$ denotes the entropy of a random variable X, and the last equation follows from the fact that Z is a deterministic function of $X_1, X_2, \ldots, X_K$ and hence $H(Z|X_1, X_2, \ldots, X_K, S) = 0$. Noting that the averaging attack satisfies $g(X_1, X_2, \ldots, X_K) = S + g(W_1, W_2, \ldots, W_K) = S + W^{\text{avg}}$, Eqn. (7.5) can be further simplified as

$$\begin{aligned} R^{\text{avg}}_{\text{max}} &= H(S + W^{\text{avg}} \mid S), \\ &= H(W^{\text{avg}}), \end{aligned} \tag{7.6}$$

which can be evaluated numerically from the p.m.f. of the attacked fingerprint W^{avg} as derived in Section 7.1 (Eqn. (7.1) and (7.2)). This analysis can be carried out in a similar fashion for the remaining attacks.

We present results for the averaging and median attacks, which are shown to be the most effective attacks by our simulation results (Section 6.3) and analytical study (Section 7.1). Figure 7.4 shows the rate function R_{max} for averaging and

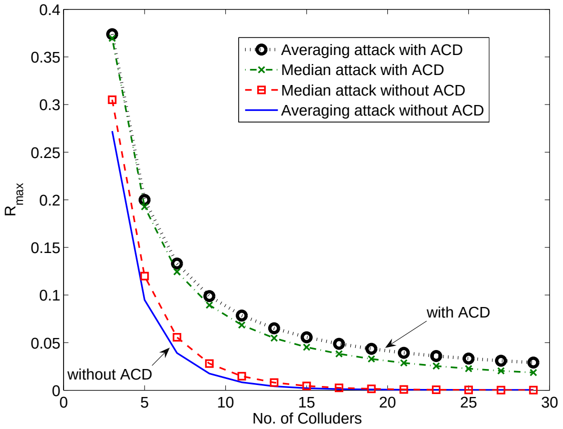


Figure 7.4: The rate function $R_{\max}$ for averaging and median attacks with and without ACD.

median attacks with and without ACD. We observe that the rate is higher for fingerprinting with ACD than without ACD, suggesting that fingerprinting with ACD is more resilient to averaging and median attacks.

7.4 Chapter Summary

In this chapter, we have performed a theoretical analysis of collusion-resistant fingerprinting techniques for compressed multimedia from various viewpoints, and demonstrated the significant advantages of the ACD technique. We first showed that ACD increases the probability of identifying a colluder using the correlation detector and the colluded copy, as compared to fingerprinting without dithering. ACD also reduces the accuracy with which attackers can estimate the host signal, and allows the fingerprinting system to accommodate a larger number of users under a given attack.

Chapter 8

Conclusions and Future Perspectives

In this dissertation, we examined two complementary approaches for multimedia protection using content and embedded fingerprints. Content fingerprints can be used to prevent the redistribution of multimedia through the internet, while embedded fingerprints can be used to trace individual copies and deter users from unauthorized redistribution.

This dissertation describes a framework for theoretical modeling and analysis of content fingerprints, whereby each individual module is studied to understand its influence on the identification accuracy. Under this framework, the impact of distortion of the content on the features, the resulting changes in the fingerprints computed, and the eventual effect on the matching process and identification accuracy can be examined separately. Accordingly, we studied how distortions in the features affect the fingerprints and identified the correlation between the features and the noise as an important factor. We then considered the problem of encoding the features into fingerprints and proposed an iterative algorithm to design the quantizer such that the identification performance is improved.

In the content identification problem, the fingerprint system designer and an adversary seeking to upload content and evade detection have conflicting objectives. We studied these interactions under a game-theoretic framework by modeling content identification as a two-player game. The designer and the adversary choose strategies to optimize their respective objective functions. Through this analysis, we showed that choosing the fingerprint bits to be i.i.d. and equally likely to take the values 0 or 1 is the optimal strategy for the designer.

Based on this result, we then studied the best identification performance achievable using fingerprints with i.i.d. equiprobable bits. We modeled content identification as a hypothesis testing problem and derived closed form expressions for the probability of making an error. For ease of use in practical applications, we derived bounds on the error probabilities and provided a lower bound on the length of the fingerprint needed to achieve a desired accuracy. This analysis also revealed the connections between content fingerprints and the problems of joint source-channel coding and errors and erasures decoding.

As practical fingerprinting schemes generate fingerprints with correlated components, we proposed a Markov Random Field model for the fingerprint and noise distributions. To evaluate the associated probability of making a detection error, we described a statistical physics inspired algorithm to estimate the density of states, and utilize the density of states to compute the probabilities of interest. We showed through experiments using image databases that the detector developed using this model can improve the detection accuracy.

The algorithm for estimating the density of states cannot be used with models containing a large number of random variables. This makes it difficult to use the MRF model for correlated fingerprints obtained from long sequences of multimedia. We proposed modeling the distribution of these fingerprints using a Markov chain model. Experiments indicated that the Markov chain model is a good fit only in certain regimes. We then proposed an adaptive model and an associated detector that provides the best detection accuracy over a wide range of operating points.

When an unauthorized copy of a multimedia document is detected using, for example, content fingerprints, embedded collusion-resistant fingerprints may be used to further identify the users responsible for the redistribution. Existing collusion resistant fingerprints were designed and tested using uncompressed multimedia, whereas most practical applications utilize compressed multimedia. Our study indicated that directly utilizing traditional schemes for embedding fingerprints in compressed signals leads to low collusion-resistance. To improve the collusion-resistance, we developed an Anti-Collusion Dithering technique for embedding fingerprints in compressed multimedia. The proposed technique approximately triples the number of colluders that can be resisted under many attacks. We also performed a theoretical analysis from different perspectives, and showed that the ACD technique increases the probability of detecting a colluder, reduces the accuracy with which colluders may estimate the host signal, and increases the fingerprinting capacity.

The analysis framework for content fingerprints described in this dissertation provides a foundation for further exploration and study. In particular, examining how distortions of the multimedia translate into changes in the features extracted is of considerable interest to the content fingerprinting community. Similarly, the robustness of different features to various processing can also be studied under the same framework. In this thesis, we adopted a model suitable for the study of features based on spatial or transform domain properties. Recently, many fingerprinting algorithms have advocated the use of local interest point based features such as SIFT [52], SURF [6] and other spatio-temporal features [46, 53]. Typically, these features are represented by vector quantization as "visual words" [63]. It would be interesting to study how distortion in these features would impact the fingerprints computed and the overall matching accuracy.

The game-theoretic modeling of content fingerprinting also holds promise and can potentially reveal interesting insights and guidelines for the design of fingerprinting schemes. The models developed for correlated fingerprints could be incorporated into the game framework to extend the analysis to non-i.i.d. fingerprints. In the game-theoretic study described in this dissertation, a simple model was adopted for the relation between the distortion in the fingerprints and the reduction in the adversary's payoff. As shown in Section 2.3, for many feature distributions of practical interest, it is possible to analyze this relation more carefully. The results of this analysis could be incorporated into the game-theoretic framework and create a more holistic model of the choices and payoffs for the designer and adversary. Another aspect of interest is to analyze how the behavior of the adversary and designer can evolve over time, as they learn from each other's strategies. Tools from evolutionary game theory [99] can be used to model such a time-varying behavior of the designer and adversary.

In content-fingerprinting applications, as the adversary typically has access to the results of the detector, he/she can repeatedly probe the detector with different inputs to better understand the algorithms and estimate the internal parameters. This knowledge could be exploited to design smart attacks for defeating the content identification system while minimizing the distortion introduced into the content. Such attacks have been studied under the name of "sensitivity attacks" in the watermarking literature [19], and various randomization techniques have been proposed for mitigating such attacks [49, 93]. Sensitivity attacks are also possible in the content fingerprinting context, but have not been systematically studied. As these attacks can be potentially devastating, it is of interest to study whether such attacks can be mitigated using randomization techniques, similar to those proposed for watermarking. A related question is whether introducing randomization techniques in any of the various stages of fingerprinting would improve the overall security of the system and benefit the fingerprint designer. The overall modeling framework and the game-theoretic approach could be used to obtain a better understanding of these issues.

The connections to joint source channel coding revealed by the study of the identification accuracy and quantizer design provide another avenue for further exploration. Error correcting codes have been used to quantize and improve the robustness of hashes designed for authentication in the robust image hashing literature [78]. We envision that similar concepts borrowed from the extensive literature in this research area could guide the design of future fingerprinting algorithms that achieve better performance.

Bibliography

[1] H. Akaike. A new look at statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, Dec. 1974.

[2] E. Ayanoglu. On optimal quantization of noisy sources. *IEEE Transactions on Information Theory*, 36(6):1450–1452, Nov. 1990.

[3] S. Baluja and M. Covell. Content fingerprinting using wavelets. In *Proceedings of the IET Conference on Multimedia*, London, Nov. 2006.

[4] A. Barg, G. R. Blakley, and G. Kabatiansky. Digital fingerprinting codes: Problem statements, constructions, identification of traitors. *IEEE Transactions on Information Theory*, 49(4):852–865, Apr. 2003.

[5] M. Barni and F. Bartolini. Data hiding for fighting piracy. *IEEE Signal Processing Magazine*, 21(2):28–39, Mar. 2004.

[6] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. SURF: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[7] P. D. Beale. Exact distribution of energies in the two-dimensional Ising model. *Physical Review Letters*, 76:78–81, 1996.

[8] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society*, 36(2):192–236, 1974.

[9] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[10] D. Boneh and J. Shaw. Collusion-secure fingerprinting for digital data. *IEEE Transactions on Information Theory*, 44(5):1897–1905, Sep. 1998.

[11] B. Chen and G. W. Wornell. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. *IEEE Transactions on Information Theory*, 47(4):1423–1443, May 2001.

[12] C.-W. Chen, R. Cook, M. Cremer, and P. DiMaria. Content identification in consumer applications. In *IEEE International Conference on Multimedia and Expo*, pages 1536–1539, Jul. 2009.

[13] W.-H. Chuang, A. L. Varna, and M. Wu. Performance impact of ordinal ranking on content fingerprinting. In *IEEE International Conference on Image Processing*, Oct. 2010.

[14] W.-H. Chuang, A. L. Varna, and Min Wu. Modeling and analysis of ordinal ranking in content fingerprinting. In *IEEE International Workshop on Information Forensics and Security*, pages 116–120, Dec. 2009.

[15] B. Coskun, B. Sankur, and N. Memon. Spatio-temporal transform based video hashing. *IEEE Transactions on Multimedia*, 8(6):1190–1208, Dec. 2006.

[16] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* Wiley Interscience, second edition, 2004.

[17] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687, Dec. 1997.

[18] I. J. Cox, M. L. Miller, and A. L. McKellips. Watermarking as communications with side information. *Proceedings of the IEEE*, 87(7):1127–1141, Jul. 1999.

[19] I.J. Cox and J.-P.M.G. Linnartz. Public watermarks and resistance to tampering. In *IEEE International Conference on Image Processing*, volume 3, Oct. 1997.

[20] H. A. David and H. N. Nagaraja. *Order Statistics.* Wiley, third edition, 2003.

[21] A. Dembo and O. Zeitouni. *Large Deviations Techniques and Applications.* Jones and Bartlett Publishers, 1993.

[22] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal Of The Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[23] H. Derin and H. Elliott. Modeling and segmentation of noisy and textured images using Gibbs random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):39–55, Jan. 1987.

[24] N. Farvardin and V. Vaishampayan. Optimal quantizer design for noisy channels: An approach to combined source - channel coding. *IEEE Transactions on Information Theory*, 33(6):827–838, Nov 1987.

[25] G. D. Forney. Exponential error bounds for erasure, list, and decision feedback systems. *IEEE Transactions on Information Theory*, 14(2):206–220, 1968.

[26] J. Fridrich and M. Goljan. Robust hash functions for digital watermarking. In *International Conference on Information Technology: Coding and Computing*, pages 178–183, 2000.

[27] R. G. Gallager. *Information Theory and Reliable Communication.* Wiley, 1968.

[28] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression.* Kluwer Academic Publishers, Norwell, Massachussetts, USA, 1992.

[29] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *Proceedings of the International Conference on Very Large Databases*, pages 518–529, 1999.

[30] J. Haitsma, T. Kalker, and J. Oostveen. Robust audio hashing for content identification. In *International Workshop on Content-Based Multimedia Indexing*, Brescia, Italy, Sep. 2001.

[31] F. Hartung and B. Girod. Watermarking of uncompressed and compressed video. *Signal Processing*, 66(3):283–302, May 1998.

[32] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[33] S. He and M. Wu. Joint coding and embedding techniques for multimedia fingerprinting. *IEEE Transactions on Information Forensics and Security*, 1(2):231–247, Jun. 2006.

[34] S. He and M. Wu. Collusion-resistant video fingerprinting for large user group. *IEEE Transactions on Information Forensics and Security*, 2(4):697–709, Dec. 2007.

[35] C. E. Jacobs, A. Finkelstein, and D. H. Salesin. Fast multiresolution image querying. In *Proceedings of Annual Conference on Computer Graphics and Interactive Techniques*, pages 277–286, New York, USA, 1995.

[36] A. K. Jain. *Fundamentals of Digital Image Processing.* Prentice-Hall, 1989.

[37] R. Kashyap. A Bayesian comparison of different classes of dynamic models using empirical data. *IEEE Transactions on Automatic Control*, 22(5):715 – 727, Oct. 1977.

[38] R. Kashyap. Inconsistency of the AIC rule for estimating the order of autoregressive models. *IEEE Transactions on Automatic Control*, 25(5):996 – 998, Oct. 1980.

[39] R. L. Kashyap. Univariate and multivariate random field models for images. In Azriel Rosenfeld, editor, *Image Modeling*, pages 245–258. Academic Press, 1981.

[40] R. Kinderman and J. L. Snell. *Markov Random Fields and their Applications.* American Mathematical Society, 1980.

[41] N. Kiyavash and P. Moulin. A framework for optimizing nonlinear collusion attacks on fingerprinting systems. In *Proceedings of Conferences on Information Sciences and Systems*, pages 1170–1175, Princeton, NJ, Mar. 2006.

[42] R. H. Koenen, J. Lacy, M. Mackay, and S. Mitchell. The long march to interoperable digital rights management. *Proceedings of the IEEE*, 92(6):883–897, Jun. 2004.

[43] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference On Artificial Intelligence*, pages 1137–1143, 1995.

[44] E. Y. Lam and J. W. Goodman. A mathematical analysis of the DCT coefficient distributions for images. *IEEE Transactions on Image Processing*, 9(10):1661–1666, Oct. 2000.

[45] G. C. Langelaar and R. L. Lagendijk. Optimal differential energy watermarking of DCT encoded images and video. *IEEE Transactions on Image Processing*, 10(1):148–158, Jan. 2001.

[46] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, and F. Stentiford. Video copy detection: a comparative study. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pages 371–378, New York, NY, USA, 2007. ACM.

[47] J. Law-To, A. Joly, and N. Boujemaa. Muscle-VCD-2007: A live benchmark for video copy detection, 2007. http://www-rocq.inria.fr/imedia/civr-bench/.

[48] E. L. Lehmann and J. P. Romano. *Testing Statistical Hypotheses*. Springer, third edition, 2005.

[49] J.-P. M. G. Linnartz and Marten Van Dijk. Analysis of the sensitivity attack against electronic watermarks in images. In *International Information Hiding Workshop and Lecture Notes on Computer Science*, pages 258–272, 1998.

[50] Q. Liu, R. Safavi-Naini, and N. P. Sheppard. Digital rights management for content distribution. In *Proceedings of the Australasian Information Security Workshop*, volume 21, pages 49–58, Adelaide, Australia, 2003.

[51] S. P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, Mar 1982.

[52] D. G. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision*, volume 2, pages 1150 –1157, 1999.

[53] J. Lu. Video fingerprinting for copy identification: From research to industry applications. In *SPIE/IS&T Media Forensics and Security*, San Jose, CA, Jan. 2009.

[54] S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.

[55] S. G. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, second edition, 1999.

[56] E. McCarthy, F. Balado, G. C. M. Silvestre, and N. J. Hurley. A framework for soft hashing and its application to robust image hashing. In *IEEE International Conference on Image Processing*, volume 1, pages 397–400, Oct. 2004.

[57] G. J. McLachlan and K. E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, 1988.

[58] G. J. McLachlan and T. Krishnan. *The EM Algorithm and its Extensions*. Wiley, 1997.

[59] M. K. Mihçak and R. Venkatesan. New iterative geometric methods for robust perceptual image hashing. In *ACM Workshop on Security and Privacy in Digital Rights Management*, 2001.

[60] R. Mohan. Video sequence matching. In *IEEE International Conference on Acoustic, Speech and Signal Processing*, volume 6, pages 3697–3700, May 1998.

[61] P. Moulin. Universal fingerprinting: Capacity and random-coding exponents. *preprint*, Jan. 2008.

[62] P. Moulin. Statistical modeling and analysis of content identification. In *Information Theory and Applications Workshop*, Feb. 2010.

[63] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[64] J. Oostveen, T. Kalker, and J. Haitsma. Feature extraction and a database strategy for video fingerprinting. In *Proceedings of the International Conference on Recent Advances in Visual Information Systems, Lecture Notes in Computer Science*, volume 2314, pages 117–128, 2002.

[65] M. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, first edition, 2001.

[66] C. I. Podilchuk and W. Zeng. Image-adaptive watermarking using visual models. *IEEE Journal on Selected Areas in Communications*, 16(4):525–539, May 1998.

[67] H. V. Poor. *An Introduction to Signal Detection and Estimation*. Springer, second edition, 1994.

[68] R. Radhakrishnan and C. Bauer. Video fingerprinting based on moment invariants capturing appearance and motion. In *IEEE International Conference on Multimedia and Expo*, pages 1532–1535, 2009.

[69] R. M. Roth. *Introduction to Coding Theory.* Cambridge University Press, 2006.

[70] R. Safavi-Naini and Y. Wang. Collusion secure q-ary fingerprinting for perceptual content. In *Proceedings of Security and Privacy in Digital Rights Management, Lecture Notes in Computer Science*, volume 2320, pages 57–75, Philadelphia, PA, Nov. 2001.

[71] G. Schaefer and M. Stich. UCID - An uncompressed colour image database. In *Proceedings SPIE, Storage and Retrieval Methods and Applications for Multimedia*, pages 472–480, 2004.

[72] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

[73] A. Shwartz and A. Weiss. *Large Deviations for Performance Analysis.* Chapman and Hall, 1995.

[74] S. R. Smoot and L. A. Rowe. Study of DCT coefficient distributions. In *Proceedings of the SPIE/IS&T Symposium on Electronic Imaging*, volume 2657, San Jose, CA, Jan. 1996.

[75] A. Somekh-Baruch and N. Merhav. On the capacity game of private fingerprinting systems under collusion attacks. *IEEE Transactions on Information Theory*, 51(3):884–899, Mar. 2005.

[76] J. K. Su, J. J. Eggers, and B. Girod. Capacity of digital watermarks subjected to an optimal collusion attack. In *Proceedings of the European Signal Processing Conference*, 2000.

[77] A. Swaminathan, S. He, and M. Wu. Exploring QIM-based anti-collusion fingerprinting for multimedia. In *Proceedings of SPIE/IS&T, Security, Steganography, and Watermarking of Multimedia Contents*, volume 6072, San Jose, CA, Jan. 2006.

[78] A. Swaminathan, Y. Mao, and M. Wu. Robust and secure image hashing. *IEEE Transactions on Information Forensics and Security*, 1(2):215–230, Jun. 2006.

[79] G. Tardos. Optimal probabilistic fingerprint codes. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 116–125, 2003.

[80] The Motion Picture Association of America. The cost of movie piracy. http://www.mpaa.org/press_releases/leksummarympa.pdf, 2006.

[81] The Wall Street Journal. YouTube removes 30,000 files amid Japanese copyright concerns. http://online.wsj.com/article/SB116133637777798831.html.

[82] The Wall Street Journal. Studios see big rise in estimates of losses to movie piracy. http://online.wsj.com/article/ SB114662361192442291.html, May 2006.

[83] W. Trappe, M. Wu, Z. J. Wang, and K. J. R. Liu. Anti-collusion fingerprinting for multimedia. *IEEE Transactions on Signal Processing*, 51(4):1069–1087, Apr. 2003.

[84] A. L. Varna, W.-H. Chuang, and M. Wu. A framework for theoretical analysis of content fingerprinting. In *Proceedings of SPIE Media Forensics and Security*, Jan. 2010.

[85] A. L. Varna, S. He, A. Swaminathan, and M. Wu. Analysis of nonlinear collusion attacks on fingerprinting systems for compressed multimedia. In *IEEE International Conference on Image Processing*, volume 2, pages 133–136, San Antonio, TX, Oct. 2007.

[86] A. L. Varna, S. He, A. Swaminathan, and M. Wu. Fingerprinting compressed multimedia signals. *IEEE Transactions on Information Forensics and Security*, 4(3):330–345, Sept. 2009.

[87] A. L. Varna, S. He, A. Swaminathan, M. Wu, H. Lu, and Z. Lu. Collusion-resistant fingerprinting for compressed multimedia signals. In *IEEE Conference on Acoustics, Speech and Signal Processing*, pages 165–168, Honolulu, HI, Apr. 2007.

[88] A. L. Varna, A. Swaminathan, and M. Wu. A decision-theoretic framework for analyzing binary hash-based content identification systems. In *ACM Workshop on Digital Rights Management*, pages 67–76, Oct. 2008.

[89] A. L. Varna and M. Wu. Modeling content fingerprints using Markov random fields. In *IEEE International Workshop on Information Forensics and Security*, Dec. 2009.

[90] A. L. Varna and M. Wu. Theoretical modeling and analysis of content identification. In *IEEE International Conference on Multimedia and Expo*, Jul. 2009.

[91] A. L. Varna and M. Wu. Modeling and analysis of correlated binary fingerprints for content identification. *to appear in IEEE Transactions on Information Forensics and Security*, 2011.

[92] A. L. Varna and M. Wu. Modeling temporal correlations in content fingerprints. In *IEEE International Conference on Acoustic, Speech and Signal Processing*, 2011. to appear.

[93] R. Venkatesan and M. H. Jakubowski. Randomized detection for spread-spectrum watermarking: Defending against sensitivity and other attacks. In *IEEE Conference on Acoustics, Speech and Signal Processing*, 2005.

[94] A. Vetro, S. C. Draper, S. Rane, and J. S. Yedidia. Securing biometric data. In Pier Luigi Dragotti and Michael Gastpar, editors, *Distributed Source Coding*, chapter 11, pages 293–323. Academic Press, Jan. 2009.

[95] S. Voloshynovskiy, O. Koval, F. Beekhof, and T. Pun. Robust perceptual hashing as classification problem: Decision-theoretic and practical considerations. In *IEEE Workshop on Multimedia Signal Processing*, pages 345–348, Oct. 2007.

[96] F. Wang and D. P. Landau. Efficient, multiple-range random walk algorithm to calculate the density of states. *Physical Review Letters*, 86(10):2050–2053, Mar. 2001.

[97] Y. Wang and P. Moulin. Capacity and random-coding error exponent for public fingerprinting games. In *Proceedings of IEEE International Symposium on Information Theory*, pages 1174–1178, Seattle, WA, Jul. 2006.

[98] Z. J. Wang, M. Wu, H. V. Zhao, W. Trappe, and K. J. R. Liu. Anti-collusion forensics of multimedia fingerprinting using orthogonal modulation. *IEEE Transactions on Image Processing*, 14(6):804–821, Jun. 2005.

[99] J. W. Wiebull. *Evolutionary Game Theory*. MIT Press, 1997.

[100] S. S. Wilks. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The Annals of Mathematical Statistics*, 9(1):60–62, 1938.

[101] F. Willems, T. Kalker, J. Goseling, and J.-P. Linnartz. On the capacity of a biometrical identification system. In *IEEE International Symposium on Information Theory*, page 82, Jun. 2003.

[102] G. Winkler. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods: A Mathematical Introduction*. Springer, second edition, 2003.

[103] M. Wu, W. Trappe, Z. J. Wang, and K. J. R. Liu. Collusion resistant fingerprinting for multimedia. *IEEE Signal Processing Magazine*, 21(2):15–27, Mar. 2004.

[104] Y. Yacobi. Improved Boneh-Shaw fingerprinting. In *Proceedings of Topics in Cryptology - CT-RSA, Lecture Notes in Computer Science*, volume 2020, pages 378–391, San Francisco, CA, Apr. 2001.

[105] H. V. Zhao and K. J. R. Liu. Fingerprint multicast for secure video streaming. *IEEE Transactions on Image Processing*, 15(1):12–29, Jan. 2006.

[106] H. V. Zhao, M. Wu, Z. J. Wang, and K. J. R. Liu. Forensic analysis of nonlinear collusion attacks for multimedia fingerprinting. *IEEE Transactions on Image Processing*, 14(5):646–661, May 2005.