WATERMARKING TECHNIQUES USING KNOWLEDGE OF HOSTS DATABASE

SUJOY ROY

NATIONAL UNIVERSITY OF SINGAPORE

2005

Name: Sujoy Roy

Degree: Doctor of Philosophy

Dept: Department of Computer Science

Thesis Title: Watermarking Techniques using Knowledge of Hosts Database

Abstract

Many watermarking applications deal with a database of hosts. When given a database of hosts to be watermarked, under the traditional watermarking approach, every host is independently watermarked. That is, encoding of one host does not use the knowledge of existence of other hosts in the database. If the encoder knows in advance about all the hosts in the database to be watermarked, intuitively, it has more information and hence can perform better. However it is not clear how to exploit this information and how significant is the improvement. In this thesis, we propose the notion of *knowledge of hosts database* and address this question: "*If the encoder has prior knowledge of the hosts database, and the detector has full or partial information of the hosts database, how to exploit this additional information to significantly enhance performance*".

To handle this question, a novel approach that demonstrates the efficacy of using *knowledge of hosts database* during the watermarking process is proposed. The proposed approach is generic and based on this, frameworks that address the problems associated with different applications can be designed. In this dissertation three different frameworks are proposed for three different applications, namely, copy detection, retrieval systems and database watermarking. In each case, novel methods are designed to implement each framework. Systematic theoretical formulation and practical experimental

evaluation is performed to validate the efficacy of the proposed frameworks.

Keywords: Copy Detection, Nearest Neighbor Search, Resolving Ambiguity, Retrieval, Watermarking, Database, Non-convex optimization.

WATERMARKING TECHNIQUES USING KNOWLEDGE OF HOSTS DATABASE

SUJOY ROY

(M. S by Research (Computer Science and Engg.), I. I. T. Delhi)

(B.E. (Computer Science), B. I. T. Mesra)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

NATIONAL UNIVERSITY OF SINGAPORE

2005

WATERMARKING TECHNIQUES USING KNOWLEDGE OF HOSTS DATABASE	**SUJOY ROY**	**2005**

Acknowledgements

I wish to express my sincere thanks to my advisor Dr. Chang Ee-Chien for introducing me to the field of Multimedia Security. His detailed comments on improving my technical writing, presentations and frequent and long discussions on research ideas were very helpful. Thanks are also due to my internal committee members Dr. Mohan Kankanhalli and Dr. Huang Zhiyong.

Satisfied with a software engineer position that was to be my next step after final year of B.E., I had practically no interest in pursuing higher studies. Naturally it amazes me to see myself standing on the verge of completing my PhD. I feel very happy to express my sincere gratitude to my teachers, Dr. and Mrs. Krishnan whose association and detailed guidance has been the transforming factor in my life. I owe the motivation and hard work in completing this thesis to them and their wonderful teachings. Their precept and example in living a principle centered life gave me the purpose and the courage to face up to the challenges and unpredictability of research career boldly and happily.

I would like to express my sincere thanks to Dr. Ankush Mittal, for his steadfast friendship and for setting the right example of how to work.

Thanks are also due to my friend Dr. Karthik Natarajan (Dept. of Maths, NUS) and Dr. Alec Stephenson (Dept. of Statistics, NUS) for the helpful discussions on optimization theory and multivariate statistics, respectively.

Special thanks are due to my friend Vipin for teaching me the intricacies of MFC programming and for giving his valuable time in reading my papers and hearing to my research ideas. I also wish to thank my friends Akshay, Sumit, Pankaj, Amit, Girish,

and Sumeet for their steadfast support and help in difficult times.

Thanks are due to my colleagues Qiming, Yu Hang, Vu Thanh and the members of the "Center for System Security" for their cooperation in adding to the proper environment for doing research.

I would like to thank, Singapore Millennium Foundation (SMF) for supporting my Ph.D study in NUS for the last two years.

Last but not the least, I would like to thank my parents and brother for their personal sacrifices in supporting me all these years.

Sujoy

December, 2005

Contents

Summary

Many watermarking applications deal with a database of hosts. When given a database of hosts to be watermarked, under the traditional watermarking approach, every host is independently watermarked. That is, encoding of one host does not use the knowledge of existence of other hosts in the database. If the encoder knows in advance about all the hosts in the database to be watermarked, intuitively, it has more information and hence can perform better. However it is not clear how to exploit this information and how significant is the improvement. This thesis proposes the notion of *knowledge of hosts database* and address the following question: "*If the encoder has prior knowledge of the hosts database, and the detector has full or partial information of the hosts database, how to exploit this additional information to significantly enhance performance*".

To handle this question, a novel approach that demonstrates the efficacy of using *knowledge of hosts database* during the watermarking process is proposed. The proposed approach is generic and based on this, frameworks that address the problems associated with different applications can be designed. In this dissertation three different frameworks are proposed for three different applications, namely, copy detection, retrieval systems and database watermarking.

In the first work, a unified framework, which can be viewed as a combination of watermarking and retrieval systems, is proposed to address the ambiguity problem in copy detection. Here, both the encoder and the detector have access to the hosts database and the size of the database is fixed. The encoder uses the knowledge of the original hosts database to generate a modified database. Given a query, the detector performs a

linear search in the modified database to retrieve the candidate original. Performance is measured by the tradeoff achieved between distortion (average or maximum distortion) undergone by the hosts in the database, and the robustness of the framework to noise.

In the previous work, the detector performs a linear search in the hosts database. For a large database, this is rather inefficient. This is because, for encoding many messages, high dimensions are required and searching in high dimensions is known to be difficult. Hence, in the second work, another unified framework is proposed where besides trading off distortion and robustness, the encoder generates an index tree to facilitate fast searching during detection. The additional facility in this framework is that it allows fast search in high dimensions during detection. The size of the original hosts database is fixed.

Under the first and second framework, the encoder and detector both have access to the whole database. In the third work, a novel framework is proposed where the detector only has partial information of the database, which may be required to be communicated from the encoder to the detector through a secure side channel. The proposed framework is shown to enhance watermarking performance compared to a traditional approach. Furthermore, an online scenario is investigated where more hosts are added into the database incrementally, i.e., size of the database is not fixed.

In each of the above works systematic theoretical formulation and practical experimental evaluations are performed to validate the efficacy of the proposed solutions.

List of Tables

List of Figures

List of Symbols

Acronyms

2-D	Two-dimensional
3-D	Three-dimensional
AFMT	Analytical Fourier-Mellin Transform
AWGN	Additive white Gaussian noise
BER	Bit error rate
CBIR	Content Based Image Retrieval
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DM	Dither Modulation
DWR	Document-to-watermark power ratio
dB	Decibel
JPEG	Joint Photographic Expert Group
MSE	Mean square error
MMSE	Minimum mean square error
PDF	Probability density function
PSNR	Peak signal-to-noise ratio
QIM	Quantization Index Modulation
RV	Random variable
SIFT	Scale Invariant Feature Transform
SS	Spread Spectrum

ST	Spread Transform
TTP	Trusted Third Party
WNR	Watermark-to-noise ratio

General

$\mathbf{R}$	Set of real numbers
$\mathbf{R}^+$	Set of positive real numbers

Stochastic

Prob$(\cdot)$	Probability
$p_x(x)$	PDF of random variable x
$p_x(x\|s=s)$	PDF of random variable x conditioned on $s=s$
$\mathcal{N}(\mu,\sigma^2)$	Gaussian PDF with mean μ and variance σ^2
$E[\cdot]$	Expected value
$var(\cdot)$	variance
$std(\cdot)$	standard deviation
σ_x^2	variance of random variable x

Information Theory

$h(x)$	Differential entropy of x
$h(x\|y)$	Conditional differential entropy of x for given y
$I(x;y)$	Mutual Information between x and y
C	Watermark capacity
$C_{Embedding}^{Attack}$	Watermark capacity for embedding scheme "Embedding" and attack "Attack"

Watermarking Related

x_n, x, $\mathbf{x}$, x	host sequence (nth element, RV, vector, vector RV)

$\tilde{x}_n$, $\tilde{\mathrm{x}}$, $\tilde{\mathbf{x}}$, $\tilde{\mathsf{x}}$	watermarked sequence (nth element, RV, vector, vector RV)
x'_n, x', $\mathbf{x}'$, x'	received sequence (nth element, RV, vector, vector RV)
w_n, w, $\mathbf{w}$, w	watermark sequence (nth element, RV, vector, vector RV)
v_n, v, $\mathbf{v}$, v	noise sequence (nth element, RV, vector, vector RV)
$\mathbf{I}$	Original (unmarked) image
$\tilde{\mathbf{I}}$	Modified (watermarked) image
$\mathbf{I}'$	Received Image
$\mathcal{I}$	Original image database
$\widetilde{\mathcal{I}}$	Modified image database
$\mathcal{A}(\mathcal{I})$ or $\mathcal{F}$	Set of $AFMT$ feature representations of images in $\mathcal{I}$
K	Key
L_x	Length of host sequence $\mathbf{x}$
τ	Spreading factor for ST watermarking
$\mathbf{t}$	Spreading vector for ST watermarking

Chapter 1

Introduction

The recent explosion of digital media in an entire range of everyday life, and the ability to store, manipulate and easily transmit them through fast and inexpensive data communication networks, has aroused serious concerns regarding its illegal copying and distribution. Watermarking technology has been proposed as a viable solution that attempts to address such concerns. Watermarking a media content entails hiding some information into the content in an imperceptible way, so that even if an adversary may be aware of the presence of the hidden information, he cannot remove it without seriously damaging the usability of the content. The hidden information, also called a message, mostly concerns the identity of the content. This is different from associating some identity information as header with a media content, as it can be easily removed by an adversary.

The process of hiding some information into the media content involves minimally distorting the content so that it carries the identifying information. From the definition of watermarking above, it is noted that the idea of minimally distorting the content so that it carries identifying information, does not mean that the actual information in the form of a message need to be always embedded into the content. For example, the modification introduced into the content can be an index into a database that stores all information related to all contents [2]. Also, this definition does not specify what is

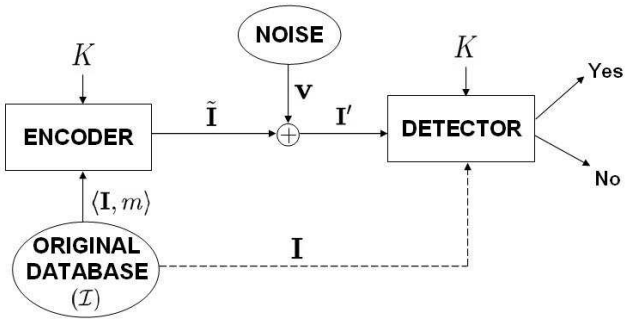


Figure 1.1: Traditional Watermarking Approach.

the associated information. For example, the information can be a confirmation about the membership of the content in a database, as in copy detection systems. Hence, in this dissertation this broader perspective is considered to look at watermarking as an information hiding technique used in applications which allow the data to be modified and the modification is associated with some information that is relevant for the specific application.

Apart from the criterion of minimizing distortion, the criterion of preventing an adversary from removing the hidden information while maintaining usability of the content entails the watermarking method to be robust to manipulations. All these criterion determine the performance of a watermarking systems.

In the next section, a traditional watermarking approach and the associated performance measures are discussed.

1.1 A Traditional Watermarking Approach

Figure 1.1 depicts a traditional watermarking approach. It consists of an *encoder* and a *detector*. The encoder takes as input a host sequence-message pair $\langle \mathbf{I}, m \rangle$[1], from a database of hosts $\mathcal{I}$ and a secret key K to generate a watermarked host $\tilde{\mathbf{I}}$. The key K modulates the message m to generate a watermark w which is an i.i.d[2] sequence

[1]In this dissertation, the host sequence mainly refers to an image, although it can refer to any other data type. So the notation $\mathbf{x}$ for host sequence is interchangeably denoted by $\mathbf{I}$.

[2]Independent identically distributed

and is embedded into the host sequence $\mathbf{I}$ to give the watermarked sequence $\tilde{\mathbf{I}}$, such that the distortion from $\mathbf{I}$ to $\tilde{\mathbf{I}}$ is small. Encoding may use knowledge of the hosts distribution. Embedding of w into $\mathbf{I}$ could be performed on the actual sequence $\mathbf{I}$ represented by a vector of pixels of an image, speech samples etc., or a transformed sequence represented by coefficients from a linear or non-linear transform of the host sequence $\mathbf{I}$. The transform function could be a linear transform, viz., Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), or a non-linear transform. The encoder uses the watermarked sequence and the original host $\mathbf{I}$, to reconstruct the watermarked host $\tilde{\mathbf{I}}$. The encoding of every host is independent of the encoding of other hosts in the database. The encoded hosts are distributed to the public. In the public domain these hosts can undergo some manipulations, denoted by noise $\mathbf{v}$, to generate $\mathbf{I}'$. The noise denotes both un-intentional and intentional manipulations that $\tilde{\mathbf{I}}$ may have to undergo.

To verify the presence of a watermark, the detector takes the sequence $\mathbf{I}'$, and *detects* whether $\mathbf{I}'$ is watermarked or not (Yes/No) using the key K and the original host $\mathbf{I}$. The process of verification of the presence of a watermark here is done by *detection* and not *decoding*. The difference between *decoding* and *detection* at the detector needs some clarification. In decoding, a message is extracted from $\mathbf{I}'$, whereas in detection, the detector just confirms the presence of the watermark by a Yes/No decision.

The dotted arrow in Figure 1.1 indicates that the original host sequence $\mathbf{I}$ can be either made available or not available during detection. When the detection unit has access to the host sequence $\mathbf{I}$ during the detection process, $\mathbf{I}$ could be subtracted from $\mathbf{I}'$ and their difference used for detecting. Such a scenario is called *non-blind* watermarking. When the detector does not have access to the host, it is called *blind* watermarking.

The performance of watermarking systems is measured by some performance measures. Some of the important performance measures are as follows:

- Distortion: It is a measure of the difference between $\mathbf{I}$ and $\tilde{\mathbf{I}}$. For example, the distortion between $\mathbf{I}$ and $\tilde{\mathbf{I}}$ can be measured by $\epsilon = \|\tilde{\mathbf{I}} - \mathbf{I}\|_2$ where $\|\cdot\|_2$ denotes

ℓ_2 norm distance metric.

- Robustness: Robustness refers to the ability to detect the watermark after common signal processing operations (intentional or unintentional) with high probability. The minimal required robustness is highly application-dependent. For signal processing operations like addition of $AWGN$, one may take the variance of the noise as the measure of robustness.

- False Alarm: It indicates the probability that a randomly chosen sequence will be declared as watermarked by the detector.

- Security: Watermarking security has been recently defined from a cryptanalytic point of view. The main idea is that information about the secret key leaks from the observations, for instance watermarked pieces of content, available to the opponent. Tools from information theory (Shannons mutual information and Fishers information matrix) have been shown [16] to be able to measure this leakage of information. The security level is then defined as the number of observations the attacker needs to successfully estimate the secret key. This dissertation does not address this concern.

The relative importance of these measures vary from application to application.

Applications of Watermarking Watermarking was primarily proposed as a technology for alleviating security concerns in applications like broadcast monitoring, owner identification, proof of ownership, transaction tracking, authentication, copy control, device control etc. The idea of associating the identity of the content with the content broadens the scope of use of watermarking technology, beyond security related applications. Examples of non-security applications are content identification, information embedding, database annotation etc.

1.2 Issues in Traditional Watermarking Approach

The concerns in a traditional watermarking approach can be identified as follows:

Trading off performance measures: The performance measures discussed are interdependent and hence a trade-off between them is required to be achieved. For example, if the content is distorted with a high watermark strength, one can expect that even after manipulations some portion of the watermark will be left over, good enough to verify its presence. But this may be at the cost of distorting the content such that the watermark becomes perceptible. Similarly, during detection, taking a low value of threshold increases detection performance but in turn increases false alarm.

Application dependent: Every application has an associated problem and there are some characteristics of the problem that make watermarking a suitable solution. Along with that, each application will have its own requirements on how to incorporate the watermarking process.

Detector side concerns For non-blind watermarking, the detector knows the original hosts corresponding to every $\mathbf{I}'$. This assumption is not practical, as in reality, for most applications where non-blind detection is applicable, the detector has access to the hosts database, but may not know which is the original host in it. For blind watermarking, a fixed detection routine, i.e., the presence of the watermark at the detector, a fixed detection boundary and known detection algorithm, makes the system vulnerable to attacks like sensitivity attacks [23].

Encoder side concerns The encoding process may use knowledge of the hosts distribution to do the watermarking, which requires an assumption to be made about the nature of the distribution. For unique identification, every host is associated with a unique message; all of which are required by the detector at the point of detection. For applications that deal with a large database of hosts, this approach is not practical.

The above issues motivate the need for a more generic approach to watermarking.

Need for another approach The traditional watermarking approach basically deals with watermarking a single host. Extending this approach to a database of hosts means, encoding of one host is independent of other hosts in the database. That is, in encoding a host, knowledge of encoding of other hosts in the database is not exploited. The observation made here is that, if the encoder knows in advance about all the hosts in the database to be watermarked, intuitively, it has more information and hence this information can be used to enhance performance. However it is not clear how to exploit this information and how significant is the improvement. Moreover, in the traditional approach with non-blind watermarking, the assumption is that the detector knows the original host. For most watermarking applications where the detector has access to the database of hosts, i.e., non-blind detection is possible, this assumption is not practical. Note that, although the detector may not know the original host, intuitively, the availability of full or partial information of the hosts database at the detector, not only provides information about the original but more. It is not clear, how this information can be used to enhance performance. This dissertation proposes the notion of *knowledge of hosts database* and addresses the following question:

"*If the encoder has prior knowledge of the hosts database, and the detector has full or partial information of the hosts database, how to exploit this additional information to significantly enhance performance*".

To handle this question, a novel approach that demonstrates the efficacy of using *knowledge of the hosts database* during the watermarking process is proposed in this dissertation. The proposed approach is generic and based on this, frameworks can be designed to address the problems associated with applications.

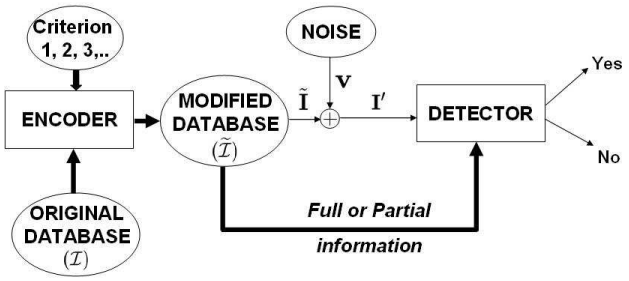


Figure 1.2: Proposed Watermarking Approach.

1.3 Proposed Approach

The proposed approach is depicted in Figure 1.2. The encoder takes the whole database $\mathcal{I}$ and based on some criterion which is decided according to the requirements of the application, generates a modified database $\widetilde{\mathcal{I}}$. Note that here the encoding of each host uses the knowledge of encoding of other hosts in the database. Hosts in the modified database are released to the public, denoted by $\tilde{\mathbf{I}}$, which may undergo some manipulations or noise to form $\mathbf{I}'$. During detection, the detector takes the sequence $\mathbf{I}'$ and uses either full or partial knowledge of the modified database to detect the identity of the host.

Noticeable differences of the proposed approach with a traditional watermarking approach are as follows:

1. Unlike in the traditional approach where knowledge of the hosts distribution may be used during encoding, here knowledge of the actual hosts in the database is used during encoding.

2. Encoding of one host is dependent on encoding of other hosts in $\mathcal{I}$. In the traditional approach, encoding of each host is independent of encoding of other hosts.

3. In the proposed approach either partial or full information of the *modified* hosts database is made available to the detector. However, given a watermarked host,

the detector does not know which host in the database is the original. In traditional blind watermarking, the detector has no information about the original host. In traditional non-blind watermarking, given a watermarked host, the detector indeed knows the original host.

4. In the proposed approach communication between the encoder and detector may be required. The bandwidth of the communication channel decides the amount of information exchange between them. This makes encoding and detection process dependent and adaptive.

5. In the traditional approach, to uniquely identify a host, every host is associated with a unique message. In the scenario where a large database of host need to be watermarked, the number of messages will be large. In the proposed approach using knowledge of the database, it can be shown that the number of messages required to uniquely identify every host need not be large, as discussed in Chapter 4.

Note that the proposed approach is generic and depending on the problem associated with an application, the criterion supplied to the encoder is varied to design an appropriate encoding routine. Similarly, depending on the application, the detector will have access to either partial or full information about the database. For example, a retrieval system can be seen as a special case of the proposed approach; where the criterions could be, (1) not to introduce distortions while generating the modified database and (2) to generate an index structure carrying information about the organization of the database. During detection, the detector uses the index structure to search into the database.

Hence the essential contribution of this dissertation is to show the efficacy of using *knowledge of the hosts database* during encoding and partial or full knowledge of the database during detection. This also explains the title of this dissertation.

1.4 Thesis Organization and Contributions

A background of some of the fundamental methods and theory in the field of traditional watermarking is provided in Chapter 2. These methods can form the underlying methods in implementing the frameworks that follow the proposed approach. In the rest of the dissertation three frameworks are proposed for three different applications, namely copy detection, retrieval systems, and database watermarking. A brief discussion of the existing literature on solving the problem associated with these applications is provided in the individual chapters.

Chapter 3 proposes a unified framework, which is a combination of retrieval and watermarking systems, to resolve ambiguity in copy detection. In the proposed framework, the encoder and detector have access to the original and modified hosts database respectively. The fundamental reason for ambiguity is identified to be the lack of sufficient separation between the hosts in feature space. Hence the solution lies in increasing this separation. The encoder uses knowledge of the original hosts database to increase this mutual separation and achieves a tradeoff between distortion and robustness. The detector searches for the nearest neighbor of a suspected copy (query) in the modified database by doing a linear search and returns it as the candidate original. Here size of the database is fixed. Experiments are performed to analyze the efficacy of the proposed framework under ambiguity attacks.

In the framework proposed in the previous chapter the detector performs a linear search. For a large database, this is rather inefficient. This is because, for encoding many messages, high dimensions are required and searching in high dimensions is known to be difficult. Chapter 4 proposes another framework that uses a combination of retrieval and watermarking systems to reduce nearest neighborhood search complexity to logarithmic order. The encoder and detector have access to the original and modified databases. Here the encoder generates an index tree to facilitate fast search. Encoding is based on a method called active clustering. The database in this scenario is also considered fixed.

In the previous chapters, the detector had access to the full database. In Chapter

5 a framework for database watermarking is proposed where the detector has access to partial information of the database, which is made available to the detector through a secure channel between the encoder and detector. The encoder uses knowledge of the original hosts database to generate the modified database and a partial description of the database. The size of the partial description is influenced by bandwidth limitations of the channel. The proposed framework demonstrates the efficacy of using knowledge of hosts database in improving watermarking performance measures over a traditional approach that uses i.i.d sequences as watermarks. An online setting where the database is not fixed and new hosts are incrementally added into the database is also investigated.

Each chapter in this dissertation discusses how, based on the proposed approach, frameworks can be designed to solve the problems associated with an application. Depending on the application, appropriate criterions are set that guide the design of the frameworks. This leaves scope for investigating how the proposed approach (using knowledge of hosts database) can be used to solve problems associated with other applications, thus opening up a wide spectrum of topics for future research. This is the point of discussion in the chapter on conclusions and future work (chapter 6).

Chapter 2

Background

In this chapter, a brief review of some of the fundamental concepts of watermarking theory that is also relevant to the contents of the work in this thesis, is presented. These concepts could form the building blocks for incorporating the novel approach proposed in this thesis, in solving the problems associated with several applications. Hence this chapter will provide the background to more clearly appreciate this work. However, not reading this chapter would not impede in understanding and appreciating the contents of this thesis. This dissertation dedicates specific chapters to specific applications, to demonstrate the efficacy of using knowledge of hosts database in those applications. So existing work related to a specific application is discussed in the individual chapters to put the motivation in context. Hence a reader familiar with these concepts may chose to move forward to the later chapters.

Section 2.1 discusses the fundamental watermarking methods from a theoretical point of view. Most of the watermarking methods proposed in the existing literature can be considered to be variations of these fundamental methods. In most of the works the encoder and detector are taken as separated units and there is no communication between the two during the detection process. Recent works have shown the efficacy of communication between the encoding and detection unit in improving watermarking performance. Section 2.2 discusses works that use communication between the encoding

and detection unit in the watermarking process.

2.1 Watermarking Methods

This section discusses existing watermarking methods. Watermarking is essentially a form of communication, where one wishes to communicate a message from the watermark encoder to the watermark detector. The question about how to communicate the message, brings in several considerations and hence several methods.

2.1.1 Host Interference Non-Rejecting Methods

A large number of watermark embedding algorithms are designed based on the premise that the host sequence is like a source of noise or interference. This view arises when the knowledge of the host sequence is not used for watermarking. Embedding methods in this class are often referred to as spread spectrum methods.

Spread Spectrum Watermarking (SS)

In spread-spectrum watermarking, one watermark bit is spread over many samples of the host sequence with the help of a modulated pseudo-random spreading sequence that is added to the host sequence. Ideally, the bandwidth of the spreading sequence covers the entire bandwidth of the host sequence. The term"spread spectrum watermarking" particularly refers to simple additive embedding of a watermark sequence $\mathbf{w}$ chosen independently of the original sequence $\mathbf{x}$, into $\mathbf{x}$, to generate the watermarked sequence $\tilde{\mathbf{x}}$, given by

$$\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{w}(m).$$

The number of watermarking techniques using this method is too large to give a complete survey of all of them. A few significant references are as follows.

Some of the earliest examples of these methods are given by Tirkel et. al [90, 89], Bender et. al [9], Cox et. al [28, 27], Smith and Cominsky [82] etc. A lot of watermarking

methods are in fact very similar and differ only in parts or single aspects of the three stages: sequence design, encoding, and recovery. The watermark sequence is often designed as a white [89, 90], or colored pseudo-random sequence with e.g., Gaussian [27] uniform or bipolar [62, 64, 89], probability density function (pdf). In order to avoid visibility of the embedded watermark an implicit or explicit spatial [56, 87, 95] or spectral [56, 72, 87, 86, 95] shaping is often applied with the goal to attenuate the watermark in areas of the image where it would become visible. The resulting watermark sequence is sometimes sparse and leaves image pixels unchanged [65], but mostly it is dense and alters all pixels of the image to be watermarked. The watermark sequence is often designed in the spatial domain, but sometimes also in a transform domain like the full-image discrete cosine transform (DCT) domain [27] or block-wise DCT domain [61].

Although additive watermarking is very attractive because of its simplicity, in many cases non-additive (multiplicative) watermarking is adopted, either to achieve image dependent watermarking[29], or to better exploit the characteristics of the human visual system (HVS). The watermark $\mathbf{w}$ in this case can be seen as a function of both the message m and the host sequence $\mathbf{x}$.

The watermark detection is usually done by some sort of correlation method. Since the watermark sequence in these schemes, is designed without knowledge of the host sequence, cross talk between watermark sequence and host sequence is a common problem. In order to suppress the crosstalk, many proposed schemes require the original host sequence in order to subtract it before watermark extraction (*non-blind* detection). Other proposed methods apply a pre-filter [35, 65, 90] instead of subtracting the original. Yet, other methods do not suppress the crosstalk [72]. Some researchers propose to use more sophisticated detectors than just simple correlation detectors, e.g., maximum *a-posteriori* (MAP) detectors [7]. Also a correlation based detection is adopted for simple additive watermarking based on the assumption that the host sequence follows a Gaussian pdf. Such a strategy is an optimum one, in that it permits to minimize the

error probability[73]. If the host does not follow Gaussian pdf and if the embedding is not additive, a correlation-based detector is not optimum. [8] proposes a decoder for optimum recovery of non-additive watermarks. SS watermarking is robust to interfering noise, as the amount of distortion that has to be added to the watermarked sequence to erase the watermark can be very high. In fact, the host sequence itself is seen as a source of interference.

Performance of SS Watermarking A theoretical realization about the maximum rate of spread spectrum watermarking can be determined under the assumption that the host sequence $x \sim \mathcal{N}(0, \sigma_x^2)$ and the noise v is constrained to AWGN where $v \sim \mathcal{N}(0, \sigma_v^2)$. $\mathcal{N}(\mu_x, \sigma_x^2)$ denotes a Gaussian random variable x with linear mean μ_x and variance σ_x^2. A Gaussian watermark $\mathbf{w}$ with power σ_w^2 is also assumed. Under these conditions, the maximum watermark rate is given by the capacity of an AWGN channel, which is

$$C_{non-blind}^{AWGN} = \frac{1}{2} log_2(1 + \frac{\sigma_w^2}{\sigma_v^2}), \tag{2.1}$$

for non-blind SS watermarking and

$$C_{blind}^{AWGN} = \frac{1}{2} log_2(1 + \frac{\sigma_w^2}{\sigma_x^2 + \sigma_v^2}), \tag{2.2}$$

for blind SS watermarking. Note that the bound (2.2) is applicable for any type of watermarking scheme, not just SS watermarking, that treats the host sequence $\mathbf{x}$ as interference. Also $\sigma_x^2 \gg \sigma_w^2, \sigma_v^2$, to satisfy the quality constraints for watermark embedding and attacks, respectively. Thus, the performance of non-blind SS watermarking facing AWGN attack is completely independent of the characteristics of the host sequence $\mathbf{x}$. The performance of non-blind SS watermarking depends solely on the watermark-to-noise power ratio $WNR = 10log_{10}(\sigma_w^2/\sigma_v^2)[dB]$. In contrast, performance of blind SS watermarking is determined by the document-to-watermark power ratio $DWR = 10log_{10}(\sigma_x^2/\sigma_w^2)[dB]$.

Improved Spread Spectrum Watermarking (ISS)

Malvar et al. [67] propose a watermark modulation technique, which when compared to traditional SS, the host sequence does not act as noise. This technique has shown competitive performance when compared to host interference rejecting methods to be discussed in Section 2.1.2, while maintaining the high robustness property of SS watermarking techniques. ISS works on the idea that, by using the encoder knowledge about the host sequence $\mathbf{x}$ (or more precisely, $\bar{\mathbf{x}}$, the projection of $\mathbf{x}$ on the watermark $\mathbf{w}$), performance can be enhanced by modulating the energy of the inserted watermark to compensate for the host interference. The embedding approach is a modification of SS embedding, where the amplitude of the watermark sequence is varied by a function $\mu(\mathbf{x}, b)$,

$$\tilde{\mathbf{x}} = \mathbf{x} + \mu(\bar{\mathbf{x}}, b)\mathbf{w}.$$

where, b denotes the message bit whose value is either $+1$ or -1, and $\bar{\mathbf{x}} = \langle \mathbf{x}, \mathbf{w} \rangle / \|\mathbf{w}\|$. Note that traditional SS is a particular case of ISS in which the function μ is made independent of $\bar{\mathbf{x}}$. They give an approximation of μ as a linear function, given by $\mu = \alpha b - \lambda \bar{\mathbf{x}}$, where parameters α and λ control the distortion level and removal of the carrier distortion on the detection statistic.

2.1.2 Host Interference Rejecting Methods

In the previous section a class of watermarking methods that consider the host sequence as interfering noise was discussed. During detection this interfering noise can be subtracted before decoding depending on whether it is a blind or non-blind watermarking. It can be naturally concluded that non-blind watermarking performs better than blind watermarking because of host interference. However, most practical watermarking scenario's require blind watermarking.

In this section blind watermarking techniques that reject interference of the host are reviewed. Such methods are commonly called "communication with side information

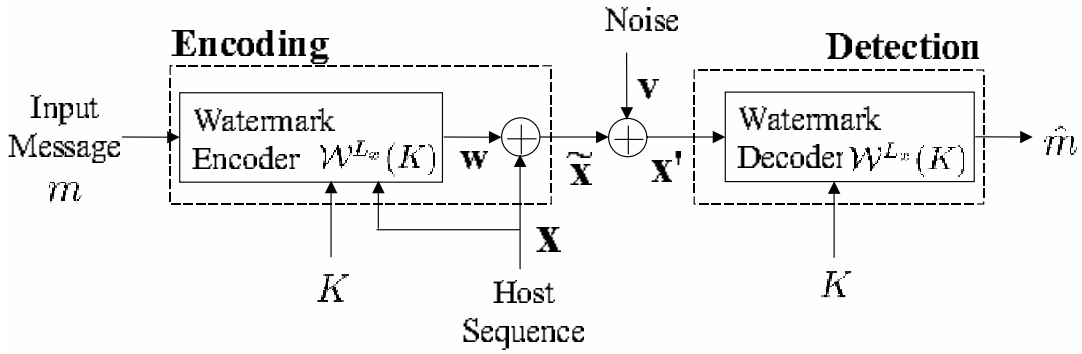


Figure 2.1: Communication with Side Information at the Encoder.

at the encoder" and were first published independently by Chen and Wornell [20] and Cox, Miller and McKellips [31]. The essential idea in such methods is that, although in blind watermarking the detector does not know the host sequence $\mathbf{x}$, the encoder can use its knowledge of $\mathbf{x}$ to reduce the influence of the host interference. [31] gives general concepts about the usefulness of side information at the encoder based on [80] but says little about how to optimally exploit side information and improve performance. Chen and Wornell introduced an almost forgotten paper by Costa [26], which is an extension of a work by Gel'fand and Pinsker [45] for continuous valued Gaussian sequences. Costa considered communication as side information at the encoder over a Gaussian channel and derived a scheme that performs as well as non-blind detection. Chen and Wornell also proposed watermarking schemes that can be considered as part of Costa's scheme, some of which perform as well as Costa's scheme. Costa's scheme is purely theoretical and hence several practical approaches to implement Costa's scheme have been proposed. In this section, first the precise communication perspective of the host interference rejecting methods for AWGN attacks is explained. Next, Costa's capacity achieving information embedding scheme for Gaussian IID host and AWGN attacks is discussed, followed by a brief explanation of some of the several practical implementations of Costa's scheme. Figure 2.1 gives an illustration of watermarking as communication with side information at the encoder.

Communication Perspective for AWGN attack

Figure 2.1 gives an illustration of the communication problem for watermarking as communication with side information at the encoder. The host sequence $\mathbf{x}$ is Gaussian IID of length L_x, $\mathbf{v}$ is AWGN, and the watermark message $m \in \{1, 2, \ldots, M\}$. The embedding process exploiting side information has two parts: first, an appropriate watermark sequence $\mathbf{w}$ representing m is selected and second, $\mathbf{w}$ is added to the host, $\mathbf{x}$. The mapping of m onto the sequence $\mathbf{w}$, also of length L_x, is determined by $\mathbf{x}$ and a codebook $\mathcal{W}^{L_x}(K)$, which is encrypted by the key K. Secrecy of the correspondence between m and $\mathbf{w}$ and the exact realization of all watermark sequences w in $\mathcal{W}^{L_x}(K)$ is obtained by a pseudo-random selection of all entries in $\mathcal{W}^{L_x}(K)$. Watermark sequences w are zero mean and IID. Note that (2.1) is an upper bound on the watermark capacity in case of AWGN attacks and is the desired capacity for host rejecting methods also.

Costa's Scheme

Gel'fand and Pinsker [45] show that for a discrete memoryless channel, for communication with side information at the encoder, the capacity is given by

$$C = \max_{p(\mathbf{u},\mathbf{w}|\mathbf{x})} (I(u; x') - I(u; x)), \tag{2.3}$$

where u is a finite alphabet auxiliary random variable and where the maximum is over all joint distributions of the form $p(\mathbf{x})$,$p(\mathbf{u}, \mathbf{w}|\mathbf{x})$, $p(\mathbf{x}'|\mathbf{w}, \mathbf{x})$. $I(u; x')$ and $I(u; x)$ denote the mutual information between u and the random variable x and the mutual information between u and the random variable x'. Costa considers x as additive channel "noise" which is side information to the encoder. At the encoder, the sequence to be transmitted is chosen depending on the message m, realizations $\mathbf{u}$ of u and the side information $\mathbf{x}$ available at the encoder. Appropriate realization $\mathbf{u}$ of u for all possible m and $\mathbf{x}$ are listed in a codebook $\mathcal{U}$, which must be known to the encoder and decoder.

The main ingredients of Costa's solution to the communication problem depicted

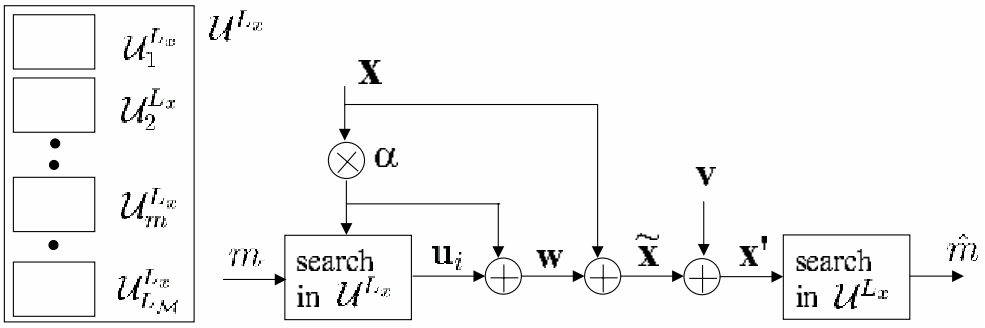


Figure 2.2: Codebook structure and encoding, decoding process for Costa's scheme.

in Figure 2.1 are (1) the design of a specific L_x dimensional codebook $\mathcal{U}^{L_x}$ and (2) an appropriate encoding process. The codebook $\mathcal{U}^{L_x}$ is given by

$$\mathcal{U}^{L_x} = \{\mathbf{u}_l = \mathbf{w}_l + \alpha \mathbf{x}_l | l \in \{1, 2, \ldots, L_\mathcal{U}\}, \mathbf{w} \sim \mathcal{N}(0, \sigma_w^2 I_{L_x}), \mathbf{x} \sim \mathcal{N}(0, \sigma_x^2 I_{L_x})\}, \quad (2.4)$$

where $\mathbf{w}$ and $\mathbf{x}$ are realizations of two L_x-dimensional independent random processes x and w with Gaussian PDF, I_{L_x} is an L_x-dimensional identity matrix, and $0 < \alpha < 1$ is a codebook parameter. The size (number of entries) of the codebook is given by $L_\mathcal{U} = \lceil 2^{(L_x \cdot I(u;x') - \epsilon)} \rceil$, where ϵ is an arbitrary small positive number. The codebook is partitioned into $L_\mathcal{M}$ disjoint sub-codebooks in such a way that each sub-codebook $\mathcal{U}_m^{L_x}$ contains the same number of sequences. Thus the total codebook is defined by $\mathcal{U}^{L_x} = \mathcal{U}_1^{L_x} \cup \mathcal{U}_2^{L_x} \cup \mathcal{U}_3^{L_x} \cup \ldots \cup \mathcal{U}_m^{L_x} \cup \ldots \cup \mathcal{U}_{L_\mathcal{M}}^{L_x}$. This codebook is available at the encoder and the decoder. The structure of $\mathcal{U}^{L_x}$ and the encoding and decoding process is depicted in Figure 2.2.

The encoding is defined as follows. Given, the host sequence $\mathbf{x}$ and the watermark message m, first a pair $(\mathbf{u}_i, \mathbf{x})$, also called the joint typical sequence, in the sub-codebook $\mathcal{U}_m^{L_x}$ is found. Searching for a joint typical sequence is equivalent to looking for a codebook entry $\mathbf{u}_i$ in $\mathcal{U}_m^{L_x}$ such that $\mathbf{w} = \mathbf{u}_i - \alpha \mathbf{x}$ is nearly orthogonal to $\mathbf{x}$ (in Euclidean space). If no such sequence is found, the encoder declares an error. however, the probability of finding no suitable sequence $\mathbf{u}_i$ vanishes exponentially as $L_x \to \infty$. The watermarked sequence $\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{w}$ is transmitted over the channel.

The decoder receives $\mathbf{x}' = \mathbf{w}+\mathbf{x}+\mathbf{v}$, searches the entire codebook $\mathcal{U}^{L_x}$ for a sequence $\mathbf{u}$ such that the pair $(\mathbf{u}, \mathbf{x}')$ is jointly typical. An error is declared if more than one sequence or no sequence is found, the probability of which is low. The index $\hat{m}$ of the sub-codebook $\mathcal{U}_m^{L_x}$ containing $\mathbf{u}$ is the decoded watermark message. The probability of error averaged over a random choice of codes goes to zero exponentially as $L_x \to \infty$.

Costa showed that for the codebook (2.4) with

$$\alpha = \alpha^* = \frac{\sigma_w^2}{\sigma_w^2 + \sigma_v^2} = \frac{1}{1 + 10^{-WNR[dB]/10}}, \tag{2.5}$$

the capacity is

$$C_{costa}^{AWGN} = \frac{1}{2} log_2(1 + \frac{\sigma_w^2}{\sigma_v^2}), \tag{2.6}$$

which is equal to $C_{non-blind}^{AWGN}$. (2.6) shows that not knowing the host sequence at the decoder does not decrease capacity, which is completely determined by the $WNR = 10log_{10}(\sigma_w^2/\sigma_v^2)[dB]$,. The proof of optimality of Costa's scheme is in the sense that there exists a randomly chosen codebook $\mathcal{U}^{L_x}$ so that the capacity can be achieved for $L_x \to \infty$. However, this is not practical since the size $L_{\mathcal{U}}$ of $\mathcal{U}^{L_x}$ can become very large, even for modest value of L_x and size of watermark alphabet $\mathcal{M}$. Besides this, searching in such a codebook is not practical due to its random structure and huge size. So some practical approaches to implement Costa's scheme have been proposed which are suboptimal in performance.

Quantization Index Modulation (QIM)

This technique was proposed by Chen and Wornell [20, 19, 18] where the host sequence $\mathbf{x}$ is quantized depending on the watermark information to be embedded. A quantizer can be uniquely described by a set of reconstruction points $\mathfrak{Q}$ in an L-dimensional space and a rule for assigning a length-L input sequence to one of the points defined in $\mathfrak{Q}$. The minimum distance rule is used for selecting the appropriate points and different quantizers are characterized by their reconstruction points $\mathfrak{Q}$. The number

of quantizers in the ensemble determine the information embedding rate. The size and shape of the quantization cells determine the embedding-induced distortions, all of which arises from quantization error. The minimum distance between the set of reconstruction points of different quantizers in the ensemble effectively determines the robustness of the embedding.

"QIM" involves first defining a set of quantizers $\{\mathfrak{Q}_1, \mathfrak{Q}_2, \ldots, \mathfrak{Q}_{L_\mathcal{M}}\}$ to the set of indices, $\mathcal{M} = \{1, 2, \ldots, L_\mathcal{M}\}$, representing the $L_\mathcal{M}$ possible watermark messages. Then for embedding message m, the host sequence $\mathbf{x}$ is quantized using $\mathfrak{Q}_m$ to obtain the watermarked sequence $\tilde{\mathbf{x}}$. For decoding, the decoder quantizes the received sequence $\mathbf{x}'$ by the union of all the quantizers $\{\mathfrak{Q}_1, \mathfrak{Q}_2, \ldots, \mathfrak{Q}_{L_\mathcal{M}}\}$. For equiprobable watermark messages m, the decoder determines the index of the quantizer containing the reconstruction point closest to the received sequence. Hence QIM can be seen as a special case of Costa's scheme for $\alpha = 1$, where $\mathbf{w}$ is equal to the quantization noise.

A key aspect behind the practical implementation of QIM systems involves the choice of practical quantizer ensembles. A convenient structure considered in [18] is the so-called dithered-quantizers, which has the property that the quantization cells and reconstruction points of any given quantizer in the ensemble are shifted versions of the quantization cells and reconstruction points of any other quantizer in the ensemble[1]. In non-watermarking contexts, the shifts typically correspond to pseudorandom vectors called dither vectors. For information embedding purposes, the dither vector can be modulated with the embedded sequence, i.e., each possible embedded sequence maps uniquely onto a different dither vector. The host sequence is quantized with the resulting dithered quantizer to form the composite sequence. Chen and Wornell call this type of embedding as "dither modulation"(DM) and the QIM technique can be called dither-QIM. Chen and Wornell discuss two QIM schemes, namely (a) binary dither modulation using a shifted uniform scalar quantizer and (b) spread transform dither modulation where the information embedding in a spread transform watermarking scheme is done

[1]The quantization noise in a dither quantizers closely emulates an IID sequence of uniformly distributed variables and is uncorrelated with the input sequence[48].

using dither-QIM. This technique is a suboptimal approach.

Spread Transform Watermarking (ST)

Chen et al [20] also proposed a particular QIM strategy, which is a more general approach to spreading watermark information over many host sequence elements and called it spread transform (ST) watermarking. Here, the watermark is not directly embedded into the host $\mathbf{x}$, but into the projection $\mathbf{x}^{ST}$ of $\mathbf{x}$ onto a random sequence $\mathbf{t}$. Thus l-th element of the projection $\mathbf{x}^{ST}$ is given by

$$x_l^{ST} = \sum_{n=\tau l}^{\tau l+\tau-1} x_n t_n,$$

where, τ denotes the spreading factor, meaning the number of host sequence elements, that belong to one element of $\mathbf{x}^{ST}$. This indicates that the information to be embedded in an element of $\mathbf{x}^{ST}$ will be spread over τ elements in $\mathbf{x}$, by the inverse ST. The watermarked sequence is computed by the inverse spread transform

$$\tilde{x}_n = x^n - x_l^{ST} t_n + \tilde{x}_l^{ST} t_n = x_n + w_l^{ST} t_n,$$

where $l = \lceil n/\tau \rceil$. For watermark detection, the received sequence $\mathbf{x}'$ has to be projected onto $\mathbf{t}$. Thus extraction and decoding of the watermarked information has to be performed on the transformed data $\mathbf{x}'^{ST}$, where

$$x'^{ST}_l = \sum_{n=\tau l}^{\tau l+\tau-1} x'_n t_n.$$

The basic idea of ST watermarking is that any component of the channel noise $\mathbf{v}$, that is orthogonal to the spreading vector $\mathbf{t}$ does not impair watermark detection. An attacker not knowing $\mathbf{t}$, has to introduce much larger distortions to impair a ST watermark as strong as a watermark embedded directly into $\mathbf{x}$. Under AWGN attack, the performance of ST watermarking is given by $WNR_\tau = WNR_1 + 10log_{10}\tau[dB]$. Thus increasing the

spreading length gives an additional power advantage over SS watermarking.

Chen and Wornell later proposed a class of watermarking schemes known as "Distortion-compensated QIM", which build directly on the scheme introduced by Costa. The scheme is based on large random codebooks. Watermark embedding boils down to replacing a vector of samples by a nearby codeword that corresponds to the to-be-embedded symbol. In order to optimize the rate of the watermark channel, an additional parameter α ($0 \leq \alpha \leq 1$) is introduced, referred to by Chen and Wornell as the distortion compensation parameter. The interpretation of this parameter is that, rather than replacing a sample $\mathbf{x}$ by the nearby codeword $\mathbf{c}$, an intermediate point is chosen. That is, $\mathbf{x}$ is replaced by $\mathbf{x} + \alpha(\mathbf{c} - \mathbf{x})$. Note that, setting $\alpha = 1$ equals embedding without distortion compensation, i.e., simply embedding using a scaled quantizer. The advantage of using a scaled quantizer is that the embedding robustness is increased by a factor of $1/\alpha^2$, and the distortion is also increased by a factor of $1/\alpha^2$. At the other extreme, setting $\alpha = 0$ is equivalent to complete distortion compensation, i.e., no embedding at all. Hence distortion is minimum. Both Costa and Chen et al. derive an expression for the rate maximizing value α^* of the distortion compensation parameter α, given by (2.6). In practice, both are impractical, as both of them are based on large random codebooks.

Suboptimal Schemes

QIM has been shown to fail to guarantee reliable communication, in cases where the watermark-to-noise ratio is negative, since the quantizer cells are too small. Also, although optimal it is not practical. Recently a few suboptimal approaches to implement Costa's scheme have been proposed[38, 75]. The scheme proposed in [38], named Scalar Costa scheme (SCS), replaces the use of large codebooks by simple structured coodbooks consisting of sample-wise uniform quantizers. For this, simpler scheme (which from the perspective of capacity is sub-optimal) a different value of α will maximize the rate. A numerical approach was taken to derive an expression for the rate-maximizing value

α^*. Ramkumar [75] proposed a watermarking scheme based on the idea of continuous periodic functions for self noise suppression (CP-SNS). The periodicity is related to the cell size in Costa's scheme using lattice codebooks. CP-SNS with thresholding is almost equal to SCS, except that the weighted embedding of the quantization error in SCS, is replaced by thresholding each quantization error sample to a given maximum absolute value.

Although lattice codes are appealing for their computational simplicity, they have some inherent limitations. First, simple orthogonal lattices do not pack code words together very efficiently. For example in 2-dimensions it is better to use a hexagonal lattice. This lattice ensures the same minimum distance between code words as the orthogonal lattice, but packs more code words onto the same area. Several good lattices for higher-dimensional spaces are also known [24]. However, the payload that is carried with simple orthogonal lattice systems using side information is vast compared to those that do not use side information. Still, it is far from the theoretical bound. So this is still a widely researched area. The second problem with lattice codes is that they are inherently weak against valumetric scaling, such as contrast changes in images. This is a serious concern for some applications. So marks invariant to scaling need to be found. In general, handling synchronization attacks or more general transformations in Costa's framework is still an open area of research. Presently only methods in which the host is transformed to some linear or non-linear transform domain have been proposed to tackle this issue. Some ideas based on error correction coding for coding with side information has been proposed as an alternative to lattice structure codes [22]. This is in contrast to error correction coding that is used by the detector to search for valid code words, this coding takes place at the encoder.

The above models for watermarking propose some practical ways for using side information in a watermarking system. Recently a lot of research has been directed towards improving these models. The frameworks proposed in this dissertation can use any of these host interference rejecting or non-rejecting models with suitable modifications.

The above review also sheds light on why there is a need to look into the prospects of using knowledge of the hosts database.

2.1.3 Methods using Partial Information at Detector

In section 2.1.1 methods that consider the host as noise were discussed. These methods primarily perform well under non-blind detection. In section 2.1.2 methods that do not consider the host as noise was discussed. It was shown how by using host knowledge at the encoder, theoretically, the watermarking rate achieved under blind detection can be made the same as that under non-blind detection. A natural follow up from these two scenario's would be methods that use partial information of the host at the detector.

A contemporary work by Cannon's and Moulin [14], proposes a system that uses a secure channel between the encoder and detector to send over a hash of the host to the detector. The hash conveys partial information about the host to the detector. They demonstrate that when combined with a statistically optimal detection test, the hash function can be designed to dramatically enhance detection performance, and in particular offer-host sequence rejection capabilities. A recent work by Voloshynovskiy et al. [92] proposes an extension of a traditional robust data-hiding set-up with host state at the encoder to a case when partial side information about the host statistics is also available at the decoder. They demonstrate that the knowledge of the host statistics at the decoder can relax the critical requirements of the random codebook based methods with assumption of knowledge of attack channel statistics at the encoder. This is an interesting result as in the above discussed host interference rejecting methods, an assumption about the knowledge of the noise variance is made at the encoder, although the actual attack parameters are not available, which can only be estimated at the decoder.

It is to be noted that the appropriate use of partial information at the detector and the corresponding development of relevant systems is still an open problem. This dissertation proposes a framework that gives some idea about how partial information

at the detector can be helpful.

2.2 Watermarking using Communication

One particular problem with state-of-the-art watermarking systems is that they are symmetric. This means that, the key K, which is a very important parameter defining the security of the system, and which is necessary for watermark encoding and detection is identical for both the encoding and detection unit. For watermark detectors that are in public domain, knowing all critical parameters of the watermarking scheme, makes them prone to attacks. Thus symmetric watermarking systems pose security risks as the detector has to know the private key. To counter this some public key watermarking methods [52, 42, 43, 44, 91, 39] have been proposed that do not give enough information during detection, to impair the embedded watermark. Note that public key watermarking methods are different from public watermarking methods. Public watermarking methods refers to blind watermarking methods, whereas public key watermarking methods refer to the fact that the encoding and detection keys are different, wherein the decoding key is publicly known and the encoding key is private. But there are some general doubts about how good the public key methods are, because they have been proved to be prone to attacks too. This behooves a careful look at watermarking systems themselves. One interesting observation is that although watermarking systems have been widely accepted to be analogous to communication systems, very few techniques actually take advantage of the possibility of communication between the encoder and detector.

From the inherent problems in private and public watermarking systems it is to be noted that the solution seems to be in giving in as much less information as possible at the detection unit. This brings in the concept of interactive watermarking, wherein very less information is provided at the beginning and detection is carried out through interactions between the server and the detector over the communication channel. This may bring in other issues of interest, concerning channel coding and so on. Recently

a solution to interactive watermarking based on the theory of cryptographic protocols has been proposed [3]. It uses zero-knowledge protocols and commitment schemes to establish proof of ownership of a watermarked host. Before this, several other works had also attempted to use zero-knowledge proofs for proving ownership [59, 47, 32]. Although theoretically, zero-knowledge proofs provide the best level of security, the communication cost involved in exchanging fairly big cryptographic keys between the encoder and detector, makes them somewhat impractical. Nevertheless, it does give a clue that communication between the encoder and detector can be exploited for more secure detection. It is also interesting to note that a commercial product - Digimarc's MediaBridge [5, 2] uses communication to extract extra information about a watermarked host. The extra information is stored in a server. This introduces a platform where the server and the detector in a watermarking system can communicate with each other and thus establish the presence of a watermark.

Works by Voyatzis and Pitas [93] and Cannons and Moulin [14] also propose the use of a secure communication channel between the encoder and detector. Voyatzis et al. use it to explain a content verification system, wherein a customer sends the to-be-verified data, to a Content verification system (CVS) for verification. The result of detection is relayed back to the customer. This method can use public key methods to do the verification but it is not the same as public watermarking. They do not mention the use of partial information at the detector. Cannons et al. also show the use of a secure side channel to share a hash of the host with the detector, which enhances detection performance. The hash gives partial information about the host to the detector as discussed before.

Chapter 3

Resolving Ambiguity in Copy Detection

3.1 Introduction

In Chapter 1 a generic watermarking approach that uses knowledge of hosts database was proposed. In this chapter, the proposed approach is treated as a combination of retrieval and watermarking framework to reduce the ambiguity problem in copy detection systems.

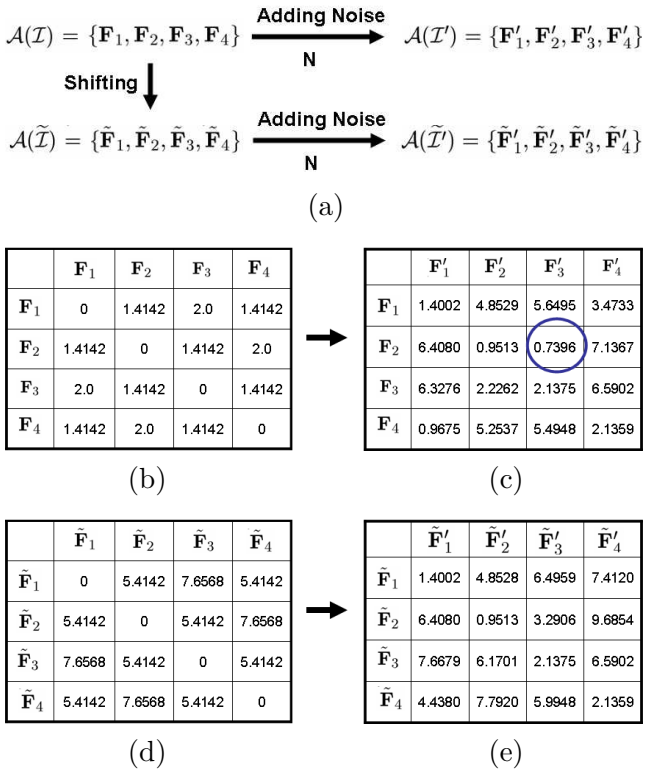
Several applications like near-replica detection[68, 58], copy detection[51, 17], sub-image retrieval[58], content based image retrieval[6, 81, 40, 15] etc., use retrieval systems as the underlying framework. Typically, in a retrieval system, a feature space is chosen and the distance of the features of the query from the features of the hosts in the database is measured based on a metric. The host or a set of hosts near to the query in feature space, is returned as search result. To be robust against permissible manipulations or inevitable noise, an effective retrieval system typically chooses a feature space and a metric, such that any two hosts in the database are well separated from each other. If two hosts are close to each other, detection error might occur, i.e., using a slightly manipulated version of one of them as query may lead to the wrong host.

Finding such a good feature representation and metric is not easy and is an active research area. Instead of refining a known feature representation and distance metric, an alternative approach to improve the effectiveness of retrieval systems is proposed herein. The original hosts are slightly modified to increase their mutual separation in feature space above a threshold, such that, the perceptual difference between the original and the modified host is minimized. The possibility of modifying the original derives its inspiration from the field of watermarking, where the encoder embeds information into the host by modifying the original. If the feature representation of the hosts is considered as data points in high dimensional space, the above proposal can be expressed by the following problem statement:

Given a set of multi-dimensional data points, how to minimally shift them so that their mutual separation is above a threshold.

Note that, under a practical setting, for increasing mutual separation, not all the hosts in the database need to be modified because of the natural separation of the hosts in feature space. This decreases average distortion. The use of knowledge of the mutual separation between the hosts in modifying them can be seen as a way of using "knowledge of the hosts database" during encoding.

An application that can benefit from a solution to the above problem is a copy detection system, where the emphasis is on exact detection as opposed to inexact detection (as is common in CBIR systems). In a typical copy detection scenario, an owner owns a large database of images that is made available to the public for viewing only. The owner may wish to know whether there are illegal copies of his images in the web. He could employ a web-robot, which randomly picks an image from the web, and checks whether this image is a copy of an image in his database. If a copy is found, the owner will decide about what action to carry out. Note that the illegal copy could be a modified, for example a lossy compressed, cropped, rotated, or even a maliciously altered version,



	$\mathbf{F}_1$	$\mathbf{F}_2$	$\mathbf{F}_3$	$\mathbf{F}_4$
$\mathbf{F}_1$	0	1.4142	2.0	1.4142
$\mathbf{F}_2$	1.4142	0	1.4142	2.0
$\mathbf{F}_3$	2.0	1.4142	0	1.4142
$\mathbf{F}_4$	1.4142	2.0	1.4142	0

	$\mathbf{F}'_1$	$\mathbf{F}'_2$	$\mathbf{F}'_3$	$\mathbf{F}'_4$
$\mathbf{F}_1$	1.4002	4.8529	5.6495	3.4733
$\mathbf{F}_2$	6.4080	0.9513	0.7396	7.1367
$\mathbf{F}_3$	6.3276	2.2262	2.1375	6.5902
$\mathbf{F}_4$	0.9675	5.2537	5.4948	2.1359

	$\tilde{\mathbf{F}}_1$	$\tilde{\mathbf{F}}_2$	$\tilde{\mathbf{F}}_3$	$\tilde{\mathbf{F}}_4$
$\tilde{\mathbf{F}}_1$	0	5.4142	7.6568	5.4142
$\tilde{\mathbf{F}}_2$	5.4142	0	5.4142	7.6568
$\tilde{\mathbf{F}}_3$	7.6568	5.4142	0	5.4142
$\tilde{\mathbf{F}}_4$	5.4142	7.6568	5.4142	0

	$\tilde{\mathbf{F}}'_1$	$\tilde{\mathbf{F}}'_2$	$\tilde{\mathbf{F}}'_3$	$\tilde{\mathbf{F}}'_4$
$\tilde{\mathbf{F}}_1$	1.4002	4.8528	6.4959	7.4120
$\tilde{\mathbf{F}}_2$	6.4080	0.9513	3.2906	9.6854
$\tilde{\mathbf{F}}_3$	7.6679	6.1701	2.1375	6.5902
$\tilde{\mathbf{F}}_4$	4.4380	7.7920	5.9948	2.1359

Figure 3.1: Illustration of the ambiguity problem and its solution using the proposed framework. Here $\mathcal{A}(\mathcal{I}) = \{\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3, \mathbf{F}_4\} = \{(3,2),(4,3),(3,4),(2,3)\}$, is the original feature database, $\mathcal{A}(\widetilde{\mathcal{I}}) = \{\tilde{\mathbf{F}}_1, \tilde{\mathbf{F}}_2, \tilde{\mathbf{F}}_3, \tilde{\mathbf{F}}_4\} = \{(3,-0.83),(6.83,3),(3,6.83),(-0.83,3)\}$ is the modified feature database and $\mathbf{N} = \{(-1.40, 0.12),(0.08,-0.95),(1.64,-1.37),(-2.0, 0.75)\}$, is the noise (intensional or un-intensional manipulations) that both $\mathcal{A}(\mathcal{I})$ and $\mathcal{A}(\widetilde{\mathcal{I}})$ encounters to generate $\mathcal{A}(\mathcal{I}') = \{\mathbf{F}'_1, \mathbf{F}'_2, \mathbf{F}'_3, \mathbf{F}'_4\}$ and $\mathcal{A}(\widetilde{\mathcal{I}}') = \{\tilde{\mathbf{F}}'_1, \tilde{\mathbf{F}}'_2, \tilde{\mathbf{F}}'_3, \tilde{\mathbf{F}}'_4\}$ respectively.

modified by an attacker who is aware of the detection mechanism. Furthermore, in a scenario where the owner had sold two copies of the same image in his database to two different customers, he may want to identify each copy individually. This is equivalent to having multiple copies (duplicates) of the same image in the database.

The problem of missed detection, which arises due to lack of separation between the hosts in feature space, is referred to herein as the *ambiguity problem*. Figure 3.1 gives an illustration of the ambiguity problem and also demonstrates the main idea. Given a feature database $\mathcal{A}(\mathcal{I})$, it is modified by shifting the features away from each other to generate a database $\mathcal{A}(\widetilde{\mathcal{I}})$. This is depicted in Figure 3.1(a). Table (b) in Figure 3.1

gives the pairwise distance between the elements in $\mathcal{A}(\mathcal{I})$. By adding noise $\mathbf{N}$ to $\mathcal{A}(\mathcal{I})$, $\mathcal{A}(\mathcal{I}')$ is obtained (refer Figure 3.1(a)). Table (b) gives the pairwise distance between $\mathcal{A}(\mathcal{I})$ and $\mathcal{A}(\mathcal{I}')$. The 3rd column in Table (b) shows that F_2 is the nearest to F_3'. Hence adding noise creates ambiguity which will lead to wrong detection when the query is F_3'. On the other hand, Table (c) gives the pairwise distance between elements in $\mathcal{A}(\widetilde{\mathcal{I}})$. Note that the mutual separation between the features has increased. By adding noise $\mathbf{N}$ to $\mathcal{A}(\widetilde{\mathcal{I}})$, $\mathcal{A}(\widetilde{\mathcal{I}}')$ is obtained (refer Figure 3.1(a)). Table (d) gives the pairwise distance between $\mathcal{A}(\widetilde{\mathcal{I}})$ and $\mathcal{A}(\widetilde{\mathcal{I}}')$. Note that there is no ambiguity problem as the features are still well separated even under noise $\mathbf{N}$.

In this work, the goal is to reduce ambiguity in copy detection systems and a unified framework combining retrieval and watermarking is proposed to achieve this goal. The next section explains the motivation behind looking into a unified framework which addresses some of the limitations of existing frameworks. In this chapter, a host sequence primarily refers to an image.

3.2 Related Work and Motivation

Recently there has been growing interest in copy detection for copyright protection of images [2, 51, 17, 68, 11, 58]. Most of the works highlight the importance of exact detection as opposed to inexact detection (as is common in CBIR systems). Existing copy detection systems can be classified into two categories or frameworks: retrieval based [68, 11, 58, 17, 51] and watermarking [2] based framework.

Retrieval Framework In a retrieval framework, detection is done by computing the feature representation of the query and using some similarity matching technique to search through the database of pre-computed features [50, 74]. Retrieval systems form an integral part of systems that organize multimedia content libraries [79, 88, 69] and retrieve multimedia data based on content [96, 55, 81, 40, 15, 6]. The emphasis here is on finding a good feature representation and distance metric. Some improvements

in distance metrics are proposed in [68]. With regard to finding an appropriate feature representation, it is noted that images can be represented either by global or local features. Recent works [66, 58, 70] highlight the efficacy of local features in improving accuracy. In [66, 58] robust scale, rotation invariant descriptors are proposed. Such features however create ambiguity problem when an image has multiple similar regions or when the database consists of images of the same scene taken from different poses. In [70] a solution to this is proposed which augments SIFT [66] descriptors with a global context vector that adds curvilinear shape information for a much larger neighborhood. If the database consists of duplicate copies of the same image this solution would also fail to resolve ambiguity. Another limitation of retrieval systems is that, they are inefficient. This is because searching and maintaining the high dimensional hosts are computationally intensive. In particular, performance of retrieval systems relies heavily on the nearest-neighbor search, which can be stated as follows: given a set of points in d-dimensions, with preprocessing allowed, how quickly can a nearest neighbor of a given query point q be found. Nearest-neighbor search is an important operation in retrieval systems and many algorithms have been proposed, such as R-tree[49], PMR quadtree [53], k-d-trees[10] and their variants [63, 94, 41]. The computing resources required by these algorithm are measured by the size of the index tree and the search time. In most algorithms, the required resources increase rapidly as the dimension of search space increases. This phenomenon is generally referred to as the dimensionality curse and is usually avoided by reducing the dimensionality of the search space. Reducing the dimensionality for fast retrieval leads to information loss and hence is not a good option, because it may lead to wrong retrieval.

Watermarking Framework In a watermarking framework, a unique identifying mark or message is embedded into every host. For example, hosts in database $\mathcal{I}$ are associated with messages $\langle(\mathbf{I}_1, m_1), (\mathbf{I}_2, m_2), \ldots, (\mathbf{I}_n, m_n)\rangle$, which are embedded into the hosts. Detection is done by extracting the message from the host and comparing against the messages associated with the hosts in the database. Once an associated message is

reliably extracted, searching is fast and hence efficient. Digimarc's MediaBridge Reader [5] is one of the few recent efforts towards using watermarking for multimedia content identification. It uses the concept of "smart images" wherein the watermarked message includes pointers to some knowledge structure on a local database or on the Internet. The watermark (message) detector extracts the message from the host and subsequently extracts more information about the host from the database. The issues in watermarking are that, distortions have to be introduced and a tradeoff between distortion and robustness has to be achieved. In watermarking, detection is fast and can be done in an off-line setting as the detector does not need to access a database. However, a watermarking framework with fixed detection routine, is vulnerable to attacks.

Note that retrieval and watermarking framework both have their advantages and disadvantages. Although the retrieval method is computationally expensive and introduces ambiguity, it achieves zero distortion. This is in contrast to the watermarking solution, which generates undesirable distortions, but achieves fast retrieval and resolves ambiguity. Hence, the interesting question is, whether a combination of both techniques can be devised to strike the right tradeoff, achieving low distortion, enabling fast retrieval, and resolving ambiguity. Kalker et. al [4] brings out some relationship between watermarking and retrieval systems (using perceptual hashing), but no concrete technique to combine them is specified. These observations form the motivation for investigating a combination of these two frameworks that achieves a tradeoff between them.

Some interesting observations that provide some idea on how such a combination framework can be designed are as follows: (1) robustness of retrieval systems is dependent on the robustness of the feature representation. If the same feature representation is employed by a watermarking-based and retrieval-based system, both systems would achieve the same robustness. (2) Although high dimensionality in retrieval systems hinders search speed, note that high dimensionality means high capacity for information embedding and hence is an advantage from the watermarking perspective. (3) Watermarking based systems are less secure, as the detection routine is fixed and can not be

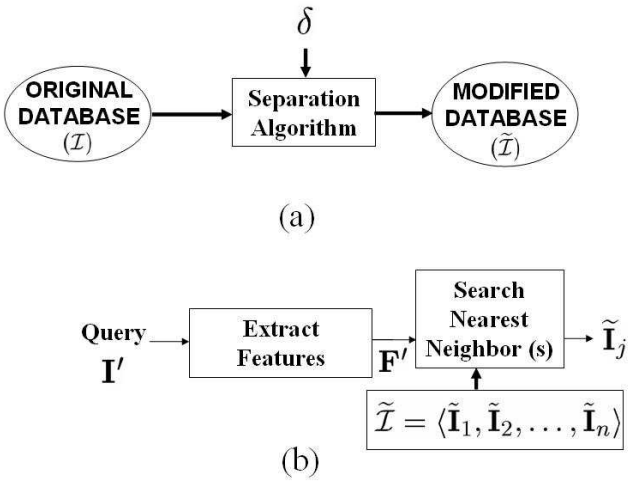


Figure 3.2: Illustrative block diagram of the proposed framework (a) Preprocessing stage (b) Detection stage.

changed after the images are watermarked, Hence, it is not easy to respond to subsequent attacks that target at the fixed watermarking method. A search based detector, as in retrieval systems, can enhance the security of such a system as the detection routine is not fixed.

Some of the above ideas are taken into consideration to propose a unified framework, which is formally presented in the next section.

3.3 Proposed Framework

The overall framework is depicted in Figure 3.2. The framework follows the approach modelled in Figure 1.2. The encoder is the routine named "separation algorithm" and the detector is the routine named "search nearest neighbor(s)". Note that the encoder has access to the original database and the detector has access to the modified database. The framework consists of a preprocessing and detection stage. The key component is the separation algorithm, which modifies the feature vectors such that they are well separated and yet minimized distortion. Section 3.4.1 will give a detailed discussion

of this algorithm. The reconstruction is also an interesting side issue but will not be elaborated upon.

Preprocessing Stage Given a database of images $\mathcal{I} = \{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_n\}$, $\mathcal{I}$ is preprocessed to get a modified database $\widetilde{\mathcal{I}} = \{\tilde{\mathbf{I}}_1, \tilde{\mathbf{I}}_2, \ldots, \tilde{\mathbf{I}}_n\}$. For this, first the feature representation of the images are extracted. Let $\mathcal{A}(\mathbf{I})$ be the feature representation of the image $\mathbf{I}$, and $\mathcal{F} = \{\mathcal{A}(\mathbf{I}_1), \mathcal{A}(\mathbf{I}_2), \ldots, \mathcal{A}(\mathbf{I}_n)\} = \{\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_n\}$, denote the set of features corresponding to the image database $\mathcal{I} = \{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_n\}$. Next, these feature vectors are significantly separated from each other using a separation algorithm to generate a modified set of features $\widetilde{\mathcal{F}} = \{\tilde{\mathbf{F}}_1, \tilde{\mathbf{F}}_2, \ldots, \tilde{\mathbf{F}}_n\}$.

Finally, the modified database $\widetilde{\mathcal{I}}$ is reconstructed from $\widetilde{\mathcal{F}}$. In other words, the reconstruction stage takes an image $\mathbf{I}$, its modified feature $\tilde{\mathbf{F}}$ and finds a $\tilde{\mathbf{I}}$ such that $\mathcal{A}(\tilde{\mathbf{I}}) = \tilde{\mathbf{F}}$ so that $\tilde{\mathbf{I}}$ is close to $\mathbf{I}$. The modified database $\widetilde{\mathcal{I}}$ is now ready to be released to the public.

Detection Stage Given a query image $\mathbf{I}'$, the detector extracts its feature representation $\mathbf{F}'$ and finds its nearest neighbor, in terms of ℓ_2 norm distance metric, in the modified database $\widetilde{\mathcal{I}}$. Based on the nearest neighbor(s), more elaborate tests can be conducted to determine whether the query is a copy, or we can simply decide whether it is a copy by comparing their distance with a threshold. For performance evaluation, the measure used in this work is to find out whether the system correctly output the nearest neighbor. In this work, the two main technical issues are: the reconstruction algorithm and the separation algorithm and the focus of this work is the separation algorithm.

3.3.1 Reconstruction

The reconstruction algorithm depends on the choice of feature representation. For certain representations, reconstructing the image is straightforward e.g., DCT, DFT coefficients. Besides the ease of reconstruction, the choice of feature representation

also depends on the type of noise to handle. Several global features, namely analytical Fourier-Mellin transform ($AFMT$) invariants [46], color histograms [85] etc, and local features, namely SIFT features [66, 57], that are robust to rotation-translation-scaling (RST), illumination variances, affine, and geometric transformations etc, have been proposed. As a proof of concept implementation, to achieve robustness against geometric distortions, $AFMT$ invariants [46] are chosen, which are robust to RST. One important assumption made in the analysis is that, distortion in the image space can be approximated by a proportional gaussian noise in feature space. The validity of this is experimentally verified in Section 3.5.1.

3.3.2 Separation

Given a set of feature vectors $\mathcal{F} = \{\mathbf{F}_1, \ldots, \mathbf{F}_i, \ldots, \mathbf{F}_n\}$ where each $\mathbf{F}_i$ is a vector in the d-dimensional space $\mathbf{R}^d$, and a parameter δ, the goal is to preprocess $\mathcal{F}$ to get a set of modified feature vectors $\widetilde{\mathcal{F}} = \{\tilde{\mathbf{F}}_1, \ldots, \tilde{\mathbf{F}}_i, \ldots, \tilde{\mathbf{F}}_n\}$, such that (1) the maximum distortion between $\mathcal{F}$ and $\widetilde{\mathcal{F}}$ is minimized, while (2) maintaining a minimum separation of δ between the elements in $\widetilde{\mathcal{F}}$. Specifically,

$$\begin{aligned} \textbf{minimize} \quad & \epsilon \\ \textbf{subject to} \quad & \|\tilde{\mathbf{F}}_i - \tilde{\mathbf{F}}_j\|_2 \geq \delta, \text{ for all } i \neq j \qquad (3.1) \\ & \epsilon \geq \|\tilde{\mathbf{F}}_i - \mathbf{F}_i\|_2, \text{ for all } i \qquad (3.2) \end{aligned}$$

Herein, ϵ denotes the *maximum distortion*, and δ the *separation*. By minimizing the maximum distortion, modification to each feature will be kept low. The constraint on separation ensures that the modified features are well separated.

Alternative objective function The usual practice in watermarking literature is to minimize average distortion instead of minimizing the maximum distortion. So an alternative formulation for constraint (3.2) would be $\epsilon \geq \sum_i (\|\widetilde{\mathbf{F}}_i - \mathbf{F}_i\|_2^2)/n$. Although

only maximum distortion is considered, the proposed algorithm and analysis can be adopted for average distortion also.

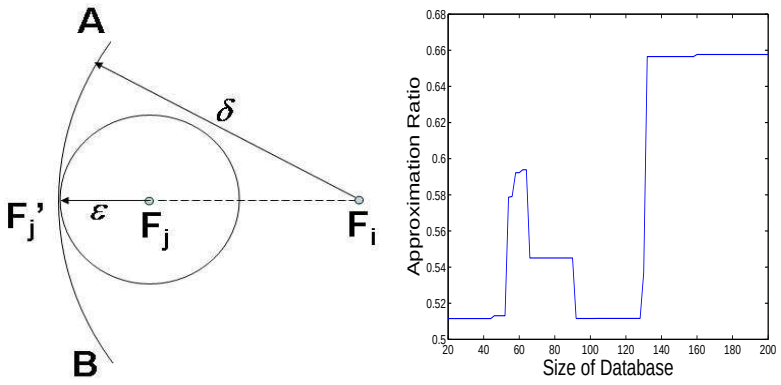
Choice of distance metric Herein, ℓ-2 norm distance metric is used to measure the separation in feature space. This is an assumption which is commonly adopted [58]. ℓ-2 norm distance metric is also used to measure the distortion during preprocessing. Note that, measuring the distortion by ℓ-2 norm does not necessarily translate to the same distortion in the image space. For example, an image $\mathbf{I}$ and its rotated version $\mathbf{I}'$ (say rotated by 45 degree) should be considered as copies of each other and thus their separation in feature space should be small, when the feature space is invariant to rotation. However, they are perceptually different and hence the distortion value should be large. In this case, the distortion value is the separation value, which is small. This happens because the feature representation is not based on perceptual considerations. Choosing a perceptual measure is a field of active research. Hence for the purpose of simplicity in representing the distortion as an optimization constraint (3.2), ℓ-2 norm is used.

3.4 Approximate Algorithm

The optimization problem constraint (3.1) is non-convex in the sense that the solution space defined by the constraints is non-convex. Such optimization, in general, is very difficult to solve. For example, by replacing the inequality in (3.2) to equality, it essentially becomes a map labelling problem which is NP-hard[83]. In this section, an efficient approximate algorithm by restricting the constraint is proposed. Two methods that help achieve further speedup are also proposed.

3.4.1 Restriction Method

The proposed restricted formulation is as follows,



(a) (b)

Figure 3.3: (a) Geometric explanation of restriction method. (b) Simulation of the behavior of approximation ratio with change in size of database. The database consists of randomly generated feature vectors of dimension 25.

$$\begin{aligned} &\textbf{minimize} && \epsilon \\ &\textbf{subject to} && \frac{(\mathbf{F}_i - \mathbf{F}_j)^T}{\|\mathbf{F}_i - \mathbf{F}_j\|_2}(\tilde{\mathbf{F}}_i - \tilde{\mathbf{F}}_j) \geq \delta, \text{ for all } i \neq j \\ & && \epsilon \geq \|\tilde{\mathbf{F}}_i - \mathbf{F}_i\|_2, \text{ for all } i \end{aligned} \tag{3.3}$$

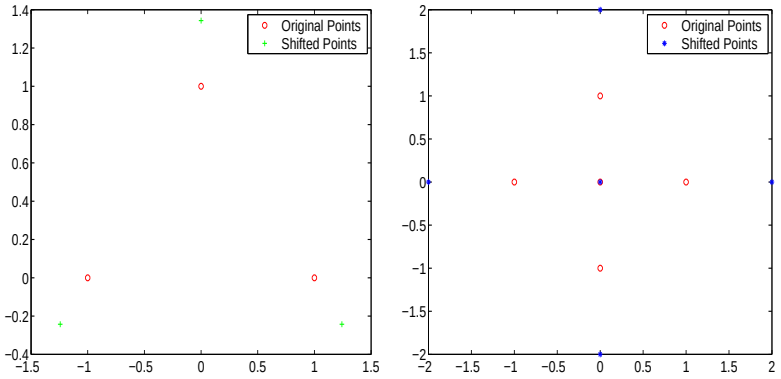
Each quadratic constraint in (3.1) has now been restricted to a linear constraint. The restriction is motivated by the following observation. By the Cauchy-Schwarz's inequality,

$$\|\mathbf{F}_i - \mathbf{F}_j\|_2 \ \|\tilde{\mathbf{F}}_i - \tilde{\mathbf{F}}_j\|_2 \geq (\mathbf{F}_i - \mathbf{F}_j)^T \cdot (\tilde{\mathbf{F}}_i - \tilde{\mathbf{F}}_j) \tag{3.4}$$

Putting (3.4) into (3.3), we have

$$\frac{(\mathbf{F}_i - \mathbf{F}_j)^T}{\|\mathbf{F}_i - \mathbf{F}_j\|_2} \cdot (\tilde{\mathbf{F}}_i - \tilde{\mathbf{F}}_j) \geq \delta \Rightarrow \|\tilde{\mathbf{F}}_i - \tilde{\mathbf{F}}_j\|_2 \geq \delta$$

Therefore the solution space of the restricted formulation is a convex subset of the



(a) (b)

Figure 3.4: Illustration of the Linear Constraint Restriction Method. All points are in $\mathbf{R}^2$ and consists of (a) 3 points (b) 5 points. The value of δ is taken as 2.

original solution space in (3.1). The restricted formulation (3.3) can be cast as a second order cone programming (SOCP) problem and has an efficient solver[84]. This kind of restriction for a hard non-convex constraint as (3.1) has been suggested in exercise 8.27 of [12]. For a *minimum distance constraint*, $\|\tilde{\mathbf{F}}_i - \tilde{\mathbf{F}}_j\|_2 \geq \delta$, the restriction is given as, $a_{ij}^T \cdot (\tilde{\mathbf{F}}_i - \tilde{\mathbf{F}}_j) \geq \delta$, where a_{ij} is any direction with $\|a_{ij}\|_2 = 1$. Note that this is equivalent to a projection of the vector between $\tilde{\mathbf{F}}_i$ and $\tilde{\mathbf{F}}_j$, onto a_{ij}. In the proposed formulation this direction is chosen to be the vector between the original data points. A geometric interpretation for this is given next.

Geometric Interpretation Figure 3.3(a) gives a geometric explanation of the restriction. Suppose that a feature vector $\mathbf{F}_i$ is fixed and another feature vector $\mathbf{F}_j$ is to be shifted to $\mathbf{F}'_j$ so that it is δ distance away from $\mathbf{F}_i$. The shifted feature vector $\mathbf{F}'_j$ must lie on or outside the arc AB of radius δ with $\mathbf{F}_i$ as the center (Figure 3.3). A point on the arc AB which is closest to the point $\mathbf{F}_j$ is the point $\mathbf{F}'_j$. Clearly, the minimum shift from $\mathbf{F}_j$ to $\mathbf{F}'_j$ is along the direction $(\mathbf{F}_j - \mathbf{F}_i)$.

Figure 3.4 (a) and (b) show two examples of the approximate algorithm implemented, for points in $\mathbf{R}^2$. In both examples, the solution for the restricted formulation is indeed the optimal solution of the original problem.

To investigate the accuracy of the approximate algorithm, an experiment was performed where a few hundred 25-dimensional vectors were randomly chosen from a multivariate Gaussian distribution where the covariance matrix is the identity. Note that a theoretical lower bound on the optimal maximum distortion (with respect to the original optimization problem) is $\epsilon_l = (\delta - d_{min})$ where d_{min} is the minimum distance between any two vectors in the data set. Figure 3.3 (b) show the ratio of ϵ_l/ϵ' where ϵ' is the maximum distortion obtained under the restricted formulation. It is observed that almost constant approximation is achieved in this experiment as the approximation ratio lies within 0.5 to 0.66. There are small ups and downs in the actual ratio value within in this range. The downs can be ascribed to the fact that, once a point is introduced that reduces d_{min}, the value of ϵ_l is increased, but proportionately more distortion is required to separate the points, thus increasing the value of ϵ' and reducing the value of the approximation ratio.

3.4.2 Improving Scalability

If the data set consists of n vectors, each in $\mathbf{R}^d$, then the number of constraints in the above formulation is in $\Theta(n^2)$, and the number of variables is dn. For a database of images where n and d are large, the number of constraints and variables are too large for existing SOCP solvers. For example, in the experiments performed for this work, the number of images is $n = 23000$ and $d = 400$. Hence the size of the input needs to be significantly reduced.

Constraint Pruning Many constraints are redundant as the features are already far apart from each other. Note that for a particular feature vector its interaction with all feature vectors which are within a ball of radius $\delta + 2\epsilon_u$ around needs to be considered, where ϵ_u is an upper bound of the maximum distortion ϵ. For the proposed application, a reasonable upper bound of ϵ is δ. This is because distortion of δ will give unacceptable perceptual distortion from the original image. Hence, feature vector pairs which are within a radius of 3δ could only be considered.

Although this method significantly prunes the number of constraints, it is still not good enough for very large databases.

Dividing into subproblems Another way of improving scalability would be to partition the feature $\mathcal{F}$ into well-separated subsets, so that the features in each subset can be independently modified. Given 2 subsets C_1 and C_2 of $\mathcal{F}$, define the distance between them as

$$d(C_1, C_2) = \min_{\mathbf{F}_1 \in C_1, \mathbf{F}_2 \in C_2} \{\|\mathbf{F}_1 - \mathbf{F}_2\|_2\}.$$

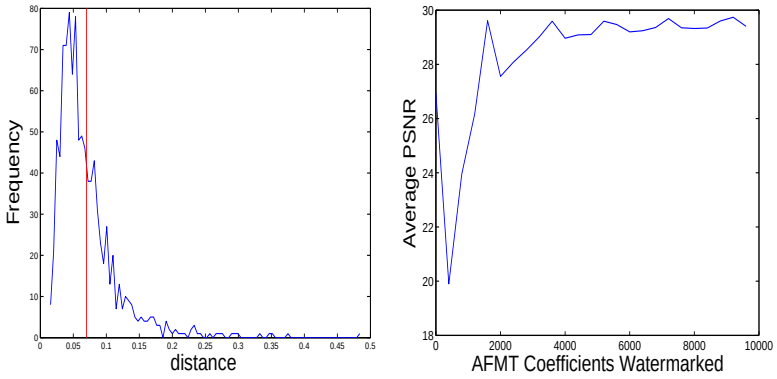
If $d(C_1, C_2) > \delta + 2\epsilon_{\mathrm{u}}$ where ϵ_{u} is an upper bound of ϵ, then there is no interaction between C_1 and C_2. Hence, optimization on C_1 and C_2 can be performed independently and yet the solution will still remain the same, as if they were considered together. Such partitioning can be easily found by scanning the pairwise distances among the features. Note that each feature can be viewed as a vertex in a graph, wherein, there is an edge between two features $\mathbf{F}_1$ and $\mathbf{F}_2$ if and only if $\|\mathbf{F}_1 - \mathbf{F}_2\|_2 < \delta + 2\epsilon_{\mathrm{u}}$. Then the partition corresponds to the different connected components in the graph.

In all experiments performed in this work, the combination of the above two methods is sufficient in reducing and dividing the optimization problem into manageable sub-problems. Note that pruning away constraints and dividing into sub-problems does not affect the optimality of the solution. That is, no approximation is being applied. Nevertheless, in cases where the above two methods fail to achieve manageable sub-problems, clustering algorithms can be applied on the feature set in that sub-problem. Features that lie on the boundary of the clusters are shifted so that the clusters are $\delta + 2\epsilon_{\mathrm{u}}$ apart. This is the technique applied in [77] and will be discussed in chapter 4. However, unlike the above two methods, this is an approximation and the speedup will affect the solution.

3.5 Implementation

A proof of concept system, RAM (**R**esolving **A**mbiguity by **M**odification) was developed and it follows the framework illustrated in Figure 3.2. Experiments were conducted on a database of colored images from two data sets, namely, (1) COIL-100 data set[71] (7200 images of average size 128×128) and (2) Corel Image database[25] (15949 images of average size 384×256 or 256×384). The Corel database consists of natural images, few of which are duplicates. Out of 15949 images there are 65 duplicates. The COIL-100 database consists of images of 100 objects taken from different poses. As noted by Ke et. al. [58] features robust to pose changes would not fair well for near-replica detection. So it is interesting to test how the ability to modify the features helps in near-replica detection of images of the same object at different poses. This is one of the primary motivations in choosing the COIL-100 database.

Feature Representation For invariance to color modification the Y-component of the YUV representation of the images were extracted and the $AFMT$ invariants of these representations were obtained. The fast algorithm as in [36] was employed to compute a two dimensional Fourier transform on the log-polar transformed image of the Y-component. Coefficients 1001 to 1400 of the $AFMT$ invariant vector were taken as the feature representation to form a set of feature vectors, $\mathcal{A}(\mathcal{I}) = \{\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_n\}$ and they correspond to the mid-frequency components. This choice was experimentally verified. 30 images were taken, whose blocks of 400 $AFMT$ coefficients were modified by adding a random sequence to them, starting with the first coefficient and then shifting it as a sliding window from the 1^{st} to the 10000^{th} coefficient. Figure 3.5(b) illustrates the perceptual measure (average $PSNR$) after reconstruction. For coefficients after the 1000^{th} coefficient, the $PSNR$ between the original and reconstructed images remain almost constant. Therefore the 1001^{th} to 1400^{th} coefficients were taken.



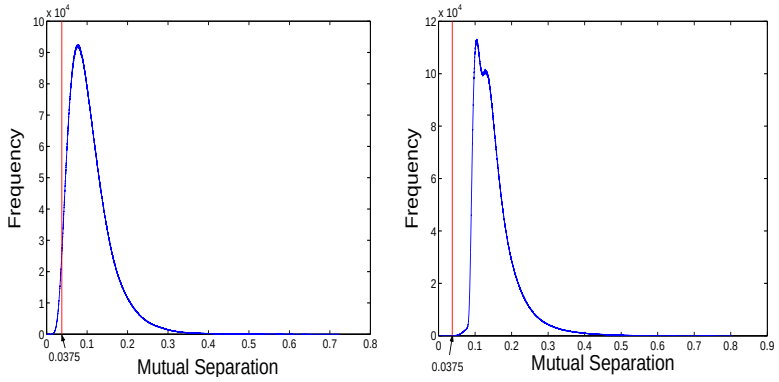
(a) (b)

Figure 3.5: (a) Distribution of the amount by which the feature representation $\mathcal{A}(\mathcal{I})$ gets shifted when their corresponding images in $\widetilde{\mathcal{I}}$ are manipulated (rotation, scaling, painting etc). The red line indicates the mean. (b) Change in perceptual distortion with change in blocks of $AFMT$ coefficients watermarked.

3.5.1 Estimating the Parameter δ

For our analysis, the manipulations on the images in the spatial domain (namely geometric transformations, cropping, painting, adding Gaussian noise, JPEG compression, brightness change, contrast change etc.) is modeled by AWGN noise in the feature domain. To verify this assumption, the distribution of $\|\mathcal{A}(\mathbf{I}) - \mathcal{A}(\mathbf{I}')\|_2$ (Squared Euclidean distance) was experimentally estimated, where $\mathbf{I}$ is a randomly chosen image from the database, and $\mathbf{I}'$ is obtained from $\mathbf{I}$ by a combination of a rotation of 10^o, cropping by removing 70%, scaling down by 4 times, painting of 4, and AWGN noise of variance 3. The manipulations were all performed in ImageMagick. Such an estimated distribution is shown in Figure 3.5. It is observed that this distribution of squared Euclidean distance emulates a χ^2 distribution. This intuitively supports the fact that the noise in the feature domain can be modeled as a Gaussian distribution. The variance of the noise due to various manipulations is used to estimate an appropriate value for δ, which makes the separation robust to manipulations.

In Figure 3.5 the variance of the distribution suggests the minimum mutual separation of the features so as to be robust against manipulations. Hence, the value of δ is



(a) (b)

Figure 3.6: (a) Histogram of the mutual separation between elements in $\mathcal{A}(\mathcal{I})$. (b) Histogram of the mutual separation between elements in $\mathcal{A}(\widetilde{\mathcal{I}})$. (For Corel database)

chosen as $\delta = 0.0375$.

3.5.2 Performance of proposed Framework

Preprocessing For the Corel database, the minimum and maximum separation between any two feature vectors in $\mathcal{A}(\mathcal{I})$ is 0 and 0.72483. For $\delta = 0.0375$, after preprocessing using the separation algorithm in Section 3.4, the minimum and maximum separation between the feature vectors in $\mathcal{A}(\widetilde{\mathcal{I}})$ is 0.0375 and 1.3338 respectively, with maximum distortion $\epsilon = 0.001$. For the COIL database, the minimum and maximum separation between any two feature representations in $\mathcal{A}(\mathcal{I})$ is 0.0007355 and 0.33369 before separation and 0.0375 and 1.2433 after separation with maximum distortion $\epsilon = 0.01855$. Figure 3.6 shows the histogram of the mutual separation between the elements in $\mathcal{A}(\mathcal{I})$ before and after preprocessing of the database $\mathcal{I}$. Note that, after preprocessing all the feature vectors are at least δ separated. An illustration of the reconstruction process is given in Figure 3.7. The availability of the original image in the inverse log-polar transformation stage (refer Figure 3.2(a)) helps to get an accurate reconstruction. Note that the original and the modified images are perceptually similar.

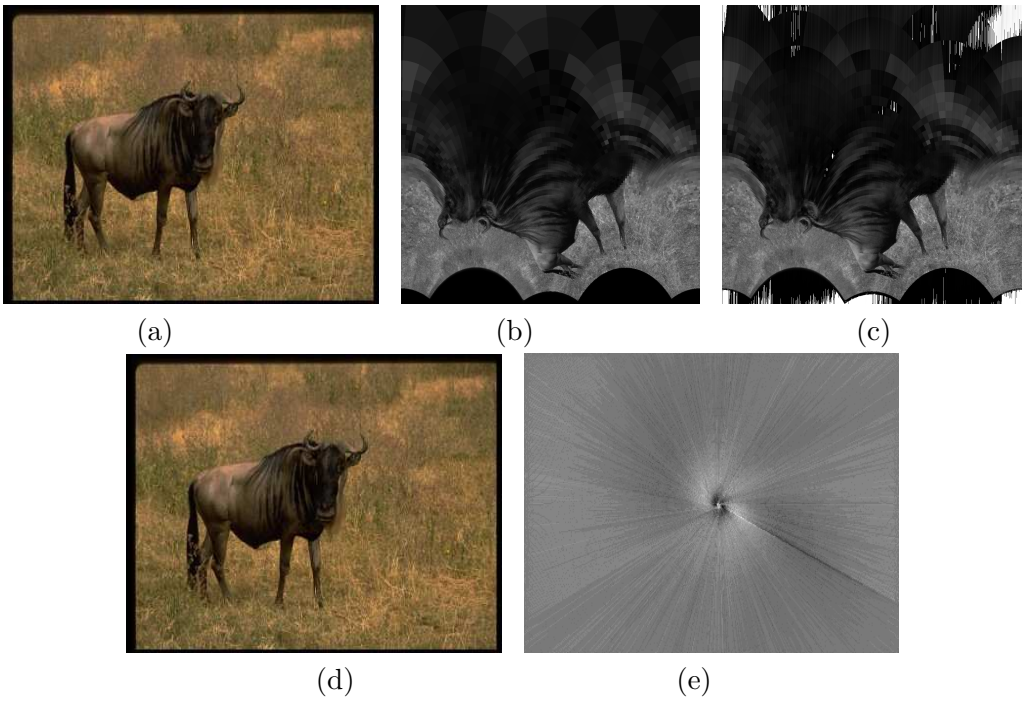


Figure 3.7: Reconstruction Process: (a) Original Image $\mathbf{I}$. (b) Log-polar transform of $\mathbf{I}$. (c) Reconstructed log-polar image of preprocessed $AFMT$ invariants $\mathcal{A}(\tilde{\mathbf{I}})$. (d) Reconstructed preprocessed image $\tilde{\mathbf{I}}$. (e) Difference between the luminance components of $\mathbf{I}$ and $\tilde{\mathbf{I}}$.

Detection To test the detection performance of RAM, 211 images in $\widetilde{\mathcal{I}}$ are randomly picked and each image is manipulated by rotating (45^o), cropping (removing 70% about center), scaling (down x4), adding Gaussian noise (strength 3), changing contrast (x2), changing brightness (150%), painting (x2), shearing (15% about x axis) and JPEG compressing (quality 20) them to generate 211 query images $\mathcal{I}' = \{\mathbf{I}'_1, \ldots, \mathbf{I}'_{211}\}$ for each category of manipulation, i.e, a total of 1899 images. The manipulations are performed by using ImageMagick. The manipulations performed are similar to the manipulations in [58, 68]. Next, for every query, their original is searched in $\widetilde{\mathcal{I}}$. Unlike [58, 68] that searches for the manipulated copies using the original as query, here search is done for the original in $\widetilde{\mathcal{I}}$ using the manipulated query and the nearest neighbors are returned. The results of the query are presented in the fifth column (titled "Preprocessed") of Table 3.1.

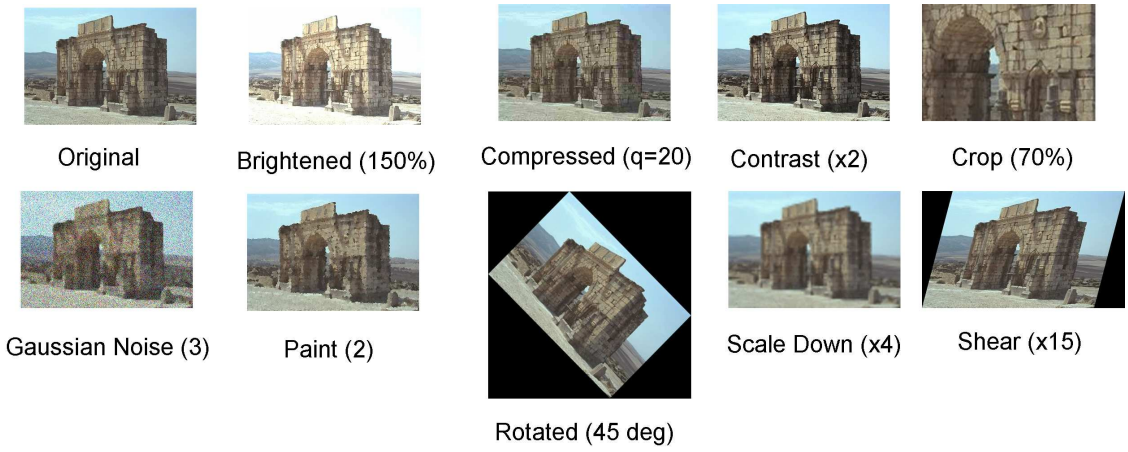


Figure 3.8: Examples of queries into the database. For all of them we can detect the correct original (for k=1).

3.5.3 Comparison with Existing Framework

For fair comparison of RAM with a retrieval framework that does not do preprocessing, a retrieval system that uses the 1001^{th} to 1400^{th} coefficients of the $AFMT$ invariants is considered as the feature representation. 211 images from $\mathcal{I}$ are taken and each image is manipulated using the image transforms described in Section 3.5.2 to generate 211 query images for each category of manipulation, i.e., a total of 1899 images. The manipulation are again performed using ImageMagick. Using each manipulated image as query, its original is searched for in $\mathcal{I}$ and the $k = 10$, $k = 5$ and $k = 1$ nearest neighbors are returned. A retrieval is considered correct if the correct copy is one of the k-nearest neighbors of the query. Columns 2, 3 and 4 of Table 3.1 give the detection accuracy obtained by the retrieval system by searching in the original database $\mathcal{I}$. Compared to the accuracy obtained using RAM (indicated in column 5 of Table 3.1), note that for a retrieval framework 100% detection accuracy is not achieved even for $k = 10$. In the proposed framework for most cases the nearest neighbor (i.e., $k = 1$) is the query.

Figure 3.8 gives examples of the different kinds of queries considered in the above experiments. For all these queries correct results were obtained. Since a global feature representation is used in RAM, the performance of RAM under manipulations like

	Nearest Neighbor Accuracy			
	Without Preprocessing(%)			Preprocessed(%)
	k = 10	k = 5	k = 1	k = 1
Rotation (45^o)	100	100	95.26	100
Scaling (down x4)	98.57	98.57	91.46	100
Cropping (remove 70%)	2.84	0.47	0	71.4
Gaussian Noise (3)	96.20	94.3	84.83	100
Paint (2)	95.73	95.73	88.15	100
JPEG Compression (20)	100	100	95.26	100
Contrast (x2)	53.08	47.39	31.75	100
Brightness (150%)	77.72	70.61	60.95	100
Shear (x15)	73.45	60.67	32.7	100

Table 3.1: Comparison of RAM with AFMT based retrieval systems without preprocessing (RF).

	RF(%)	SIFT(%)	RAM(%)
Rotation (45^o)	50	48.33	100
Scaling(down x4)	75	36.67	98.3
Cropping (remove 70%)	0	30	68.33
Gaussian Noise (3)	33.33	40	80
Painting (2)	76.67	38.33	90
JPEG Compression (20)	96.67	70	100
Contrast (x2)	16.67	81.67	98.3
Brightness (150%)	51.67	80	100
Shear (x15)	6.67	70	100
Original	100	90	100

Table 3.2: Performance comparison between (a) $AFMT$ based retrieval system without preprocessing (RF), (b) SIFT based systems and (c) RAM on the COIL database (for k = 1). Total 360 queries were used.

excessive cropping (say 90%) is less effective than a state-of-the-art retrieval system (for example [58]). However, the goal here is not to compare with a state-of-the-art system but to demonstrate the efficacy of the proposed framework by giving a proof of concept implementation.

Figure 3.9 depicts detection results for a query into the COIL database using RAM. The purpose of this test is to analyze how the problems due to pose invariant feature representations is solved by RAM. Table 3.2 gives a comparison of the proposed system with a SIFT based retrieval system and an $AFMT$ based retrieval system without preprocessing. The nearest neighbor is searched for all three cases, i.e., k = 1. The SIFT feature extraction and matching implementation used is the code made publicly available by David Lowe [66]. For the SIFT based system implementation, the image with the maximum number of "keypoint" matches is taken as the nearest neighbor. Lack of ability to resolve ambiguity in images of the same object taken from different

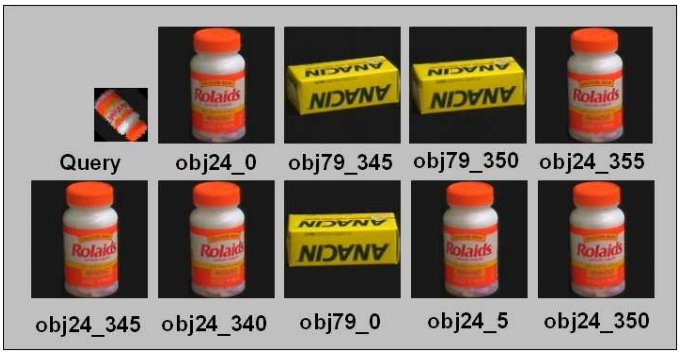


Figure 3.9: Result of search in COIL database: Nearest neighbors to the query arranged in decreasing order from left to right. Query is a rotated (130^o), cropped (8%), scaled down (2 times), and paint (strength 2) copy of the image **obj24_0**

poses by SIFT descriptor is indicated by the search results when the original is used as the query. $AFMT$ descriptor being a global descriptor has better discrimination ability. $AFMT$ features without preprocessing perform very poorly under cropping of 70% (removed). RAM clearly improves upon this. SIFT features do not seem to perform well under rotation, scaling and JPEG compression, for k=1. A possible explanation for this observation is that for COIL images, the number of keypoint's is less. This is mainly because the amount of texture in these images is less. This is also one of the known problems with local descriptors. Added to that, since many of the images in the database are of the same object taken from different poses, the descriptors are very close to each other and hence are not robust under such operations. From this it can be concluded that, for a copy detection system aiming at finding the nearest neighbor (k=1), this result is not good enough. Note that the COIL images are very much sensitive to Gaussian noise. Overall it is noted that RAM performs significantly better than existing systems on a database of images of same objects taken from different poses. This is true for both cases, when the query is the original or is a manipulated copy.

The proposed idea of selectively modifying some of the features is also an advantage over a watermarking based framework, where every image needs to be embedded with a message to identify it uniquely. In the proposed framework, the natural separation of the

images in feature space helps to perturb only those features which are close enough and are liable to create ambiguity problem. This can be seen as a method of "watermarking with knowledge of image database". This clearly demonstrates how knowledge of the image database helps improve the watermarking performance measures compared to a system that does not use knowledge of the database during the watermarking process.

3.6 Ambiguity Attacks

In this section attacks that try to create ambiguity by tampering the feature representation are analyzed. Firstly it is important to note that the notion of perceptual similarity is a subjective measure and there is no good existing measure for it. Nevertheless, any manipulation of the image that distorts the original semantics is likely to induce distortion in the feature domain. So with this assumption, the notion of security can be measured in terms of analyzing how much $\mathcal{A}(\tilde{\mathbf{I}})$ needs to be shifted in feature space to get $\mathcal{A}(\tilde{\mathbf{I}}')$, so that $\mathcal{A}(\tilde{\mathbf{I}}')$ is closer to the feature representation of another image in $\widetilde{\mathcal{I}}$.

The ability of an attacker to create ambiguities is dependent on his knowledge of the database itself. If the attacker has just one image from the database, he can add a random perturbation to its feature representation and try to create ambiguity. Herein the assumption made is that the attacker has full knowledge of the the database $\widetilde{\mathcal{I}}$. Hence, given any $\tilde{\mathbf{I}}$, the attacker is able to induce minimum distortion so that the distorted image will cause ambiguity.

For the concerned database, the average distance of an image in $\mathcal{A}(\widetilde{\mathcal{I}})$ to its nearest neighbor is 0.0726 and the distance between the closest pair is 0.0375. Thus, if an attacker has knowledge of the whole database $\widetilde{\mathcal{I}}$, given a randomly chosen image from $\widetilde{\mathcal{I}}$, he can create ambiguity by moving it towards its nearest neighbor, and the expected distortion is (0.0726/2). In the best case for the attacker, when the chosen image happens to be closest to its nearest neighbor, the attacker just has to distort the image by 0.0375/2. Figure 10(a),(b) illustrate the distortion required on a randomly chosen image, and 10(c), (b) illustrate the best case for the attacker. Note that the distortion

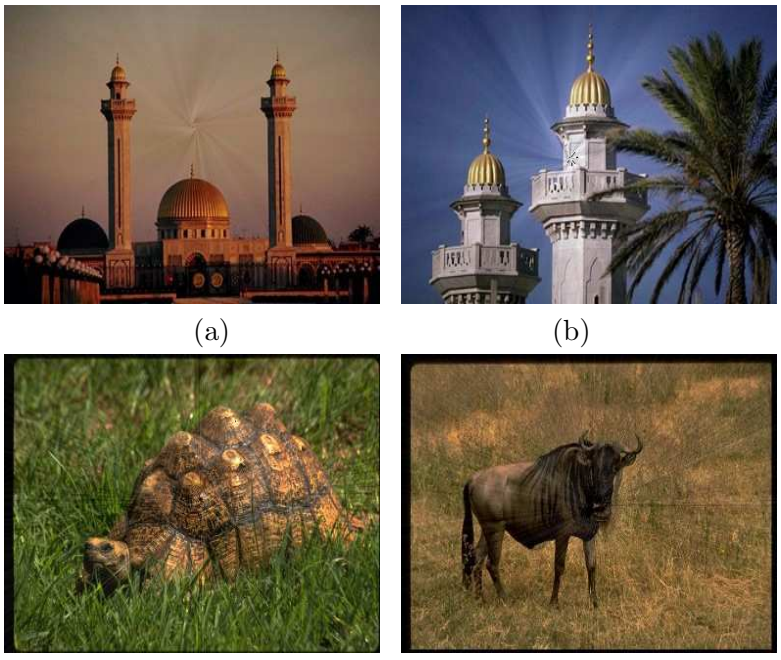


Figure 3.10: Attacked reconstructed images: (a)-(b) A randomly chosen image and its nearest neighbor, shifted towards each other by an amount half the distance between them. (c)-(d) The closest pairs in a database shifted towards each other by an amount half the distance between them.

is perceptually noticeable.

Figure 3.11 illustrates the nearest neighbor distance distribution for the full database, before and after preprocessing. This supports the fact that, for an attacker to create ambiguity by perturbing the feature representation of the images, it is much easier when the images are not preprocessed. Hence RAM is more secure to malicious attacks than a scheme that does not preprocess the database.

3.7 Discussions

A unified framework that resolves ambiguity by modifying the features which is applicable to any modifiable feature representation is presented. The efficacy of this framework in copy detection applications demonstrates how knowledge of hosts database can be effectively used. A proof of concept implementation, RAM, was proposed, that uses An-

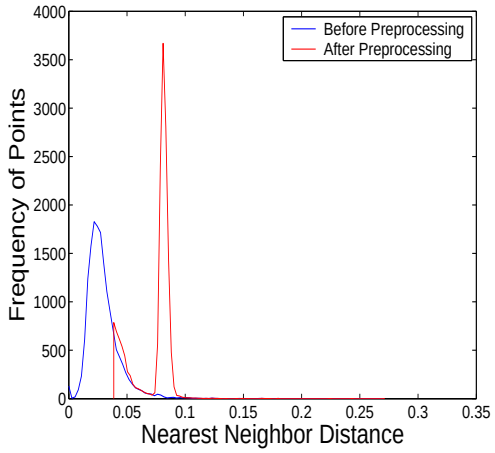


Figure 3.11: Comparison of nearest neighbor distance distribution before and after preprocessing.

alytical Fourier-Mellin ($AFMT$) invariants as features. Experiments and comparison with existing frameworks show promising results. The proposed framework does not attempt to present a new feature representation to resolve ambiguity. It is to be applied to existing feature representations to further reduce ambiguity. Hence, it complements existing methods. It does not try to improve upon a feature representation or giving an alternative method for watermarking.

Unfortunately the proposed framework inherits some of the limitations from watermarking and retrieval systems. (1) An explicit feature representation is needed. For certain feature representations it is not clear how the reconstruction can be achieved, for example, if the feature is derived from line and shape information in the images. (2) Access to the database during detection is required. (3) It is only possible in situations where modification of the database is allowed. On the other hand, unifying both retrieval and watermarking frameworks enhances performance: (1) It further separates the images and thus reduces the chances of ambiguity. (2) It is arguably more secure. (3) It introduces less distortions compared to a watermarking based approach. In view of the pro's and con's in existing frameworks, the proposed framework presents an alternative that complements currents methods.

The proposed framework is designed for a static database setting. For an on-line setting, it can be extended by adding constraints to the original optimization formulation. The added constraints retain the present separation between the data points and separate the added data in relation to it. Some studies on the effect on performance in the on-line setting can be found in [76] and in Chapter 5. This work appears in [78].

Chapter 4

Improving NN-Search Complexity in Retrieval Systems

4.1 Introduction

In the previous chapter, a unified framework for resolving ambiguity in copy detection was proposed. Given a query, the detector finds its nearest neighbor in the modified database by performing a linear search. Since the feature representation of the hosts is in high dimensions, for a large database of hosts this is not practical, because of the dimensionality curse. Also dimensionality reduction is not recommended, as it will lead to loss of information. Hence, the inefficiency of searching in high dimensions is a fundamental limitation of a search based detector, as in retrieval systems. On the other hand, if the system uses a watermarking framework as the underlying framework, and the detector has access to the full message-host association information, once the identifying information (message) is reliably extracted, retrieval is fast. However, reliable extraction of the message is affected by the distortion-robustness tradeoff problem. Note that here also, high dimensions are required for encoding more messages. Hence, the fundamental problem of designing a detector, that has access to the hosts database and employs a search based detection technique, lies in making nearest neighborhood

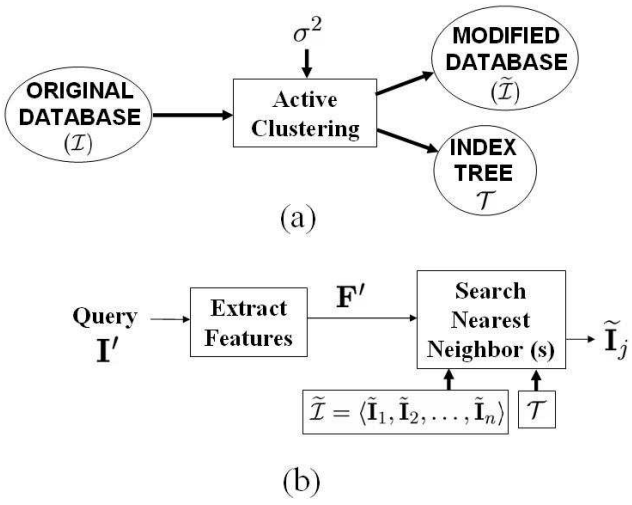


Figure 4.1: Illustrative block diagram of the proposed framework (a) Preprocessing stage (b) Detection stage.

search in high dimensions efficient. Since high dimensions are an inevitable requirement in host representation (as noted above), a solution to this problem gains importance.

In this chapter another unified framework, combining retrieval and watermarking systems is proposed. In the previous chapter the focus was on finding an encoding technique that reduces the ambiguity problem. Detection efficiency was not the primary concern as the framework was meant to address applications that focus on exact detection. In this chapter, apart from ensuring tradeoff of performance measures, there is another concern of making searching in high dimensions efficient. The encoder here ensures a proper tradeoff between distortion and robustness to generate a modified database and an index tree that is used to search into the database. The proposed method is based on active clustering. This idea is formally presented in the next section. In this chapter, the hosts database used is a collection of images.

4.2 Problem Formulation

The overall framework is depicted in Figure 4.1. Encoding as in Figure 1.2, is carried out by the "active clustering" algorithm. The framework consists of a preprocessing and detection stage. The key component is the active clustering algorithm, which uses knowledge of the hosts database to generate an index tree, which is used by the detector for searching. Section 4.3 will give a detailed discussion of this algorithm.

Given the database $\mathcal{I} = \langle \mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_n \rangle$, a distortion constraint ϵ and robustness σ^2, the goal is to preprocess $\mathcal{I}$ to obtain the watermarked $\widetilde{\mathcal{I}} = \langle \tilde{\mathbf{I}}_1, \tilde{\mathbf{I}}_2, \ldots, \tilde{\mathbf{I}}_n \rangle$ and an index tree. Each $\mathbf{I}_i = (a_1, a_2, a_3, \ldots, a_d)$ denotes a host, and is represented by a sequence of d coefficients. The watermarked $\widetilde{\mathcal{I}}$ satisfies the distortion constraint ϵ, that is,

$$\frac{1}{n} \sum_{i=1}^{n} \|\mathbf{I}_i - \tilde{\mathbf{I}}_i\|_2^2 < \epsilon. \tag{4.1}$$

The distortion should be small enough to meet the imperceptibility constraints of watermarking. In many perceptual models, imperceptibility is achieved by ensuring that the distortion (measured in 2-norm) is lower than a threshold. In the proposed model, average distortion is used as the measure of performance (as indicated in (4.1)). This can be also replaced by "maximum distortion", given by $\max_i\{\|\mathbf{I}_i - \tilde{\mathbf{I}}_i\|_2^2\}$. This ensures that the distortion of every image is lower than the required threshold ϵ.

The index tree facilitates searching such that given the query $\tilde{\mathbf{I}}_i$, the index i can be output efficiently. The searching is robust in the sense that if $\tilde{\mathbf{I}}_i$ is corrupted by additive white Gaussian noise (AWGN) with power σ^2, the output is correct with high probability. Specifically, suppose

$$\mathbf{I}' = \tilde{\mathbf{I}}_i + \mathbf{z},$$

where $\mathbf{z} = (z_1, z_2, \ldots, z_d)$ and each z_j is independently drawn from the normal distribution $N(0, \sigma^2/d)$. Then, taking $\mathbf{I}'$ as the query, the algorithm gives the correct output (which is i) with probability at least $(1 - 1/d)$. The error probability was chosen to be $1/d$ so that asymptotically, it goes to zero.

This formulation can be rephrased to an optimization problem. By fixing the distortion constraint, the objective is to find an index tree that maximizes the robustness σ^2, or vice versa, fixing σ^2 and minimizing the distortion.

In the above formulation, the messages associated to the hosts are actually its indices. This is different from the original description where the messages m_i could be a string. This difference is not critical because the actual message m_i can be easily looked up from a table.

Coding. A solution to the proposed problem has to address two issues. The first is regarding coding. If $\mathbf{I}_1 = \mathbf{I}_2 = \ldots = \mathbf{I}_n$ are identical, then the problem is the same as non-blind watermarking, that is, watermarking with original host available at the decoder. Because there is only one host, it can be used as the reference point. This reduces the problem to finding the watermarked $\tilde{\mathbf{I}}_1, \tilde{\mathbf{I}}_2, \ldots, \tilde{\mathbf{I}}_n$ that are far apart but subject to 4.1, the distortion constraint $\sum_i \|\tilde{\mathbf{I}}_i - \mathbf{I}_i\|_2^2 \leq n\epsilon$. This is essentially channel coding, where ϵ is the power constraint and σ^2 is the noise variance. Note that high dimensionality is required to encode large number of messages.

Searching. The other issue is on the computational aspect of searching. As it has been mentioned in chapter 3, the dimensionality curse prevents fast searching. Fortunately, a few differences of the proposed problem from the classical nearest-neighbor search can be exploited. The most notable difference is that, in the proposed problem, the data points can be slightly modified (watermarked) for better searching performance. In the extreme case, with unlimited distortion, the problem is trivially solved by aligning the watermarked hosts along a straight line. Since distortion is undesirable, the goal is to minimize the distortion while supporting fast retrieval.

4.3 Active Clustering by watermarking

In this section, an algorithm based on hierarchical clustering is proposed. This algorithm first finds a hyperplane that separates $\mathcal{I}$ into two balanced (within a constant factor) clusters. The hosts are then modified (watermarked) so that none of them are located

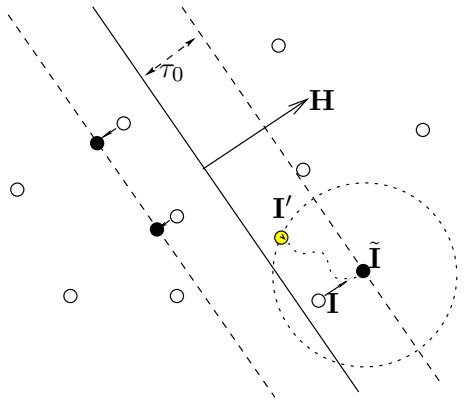


Figure 4.2: Each circle represents a host. Each filled circle represents the corresponding watermarked host, if it is different from the original. The region between the two dotted lines is the buffer zone, and its width is τ_0. The point $\mathbf{I}$ is an original host, $\tilde{\mathbf{I}}$ is the watermarked host and $\mathbf{I}'$ is a corrupted query. The normal of the separating hyperplane $\mathbf{H}$ can be viewed as the "watermark". Those hosts on the left halve contain the watermark $-\mathbf{H}$, while those on the right contain watermark $\mathbf{H}$.

near the hyperplane. The modification is done by pushing the host $\mathbf{I}$ in a direction $\mathbf{H}$, which is normal to the hyperplane. Finally, each cluster is recursively divided into sub-clusters. The slab (region between two parallel hyperplanes) that does not contain any watermarked host is called the *buffer zone*, and the distance of the hyperplane to the buffer zone's surface is called the buffer zone's *width*. Figure 4.2 shows that the modified host $\tilde{\mathbf{I}}$ is given by: $\tilde{\mathbf{I}} = \mathbf{I} + k \cdot \mathbf{H}$, where k is a constant, $\mathbf{H}$ is the watermark and $\tilde{\mathbf{I}}$ is the watermarked host in vector representation. This is similar to simple additive watermarking in spread spectrum watermarking method [28]. Note that the modifications are carried out to ensure that no hosts are located near the hyperplane. Since the hosts are modified to obtain better clusters, the proposed method is called *active clustering*. The modification can be viewed as a watermarking process.

The hierarchical clustering method discussed above gives an index tree for searching. The root and the internal nodes of this tree are the separating hyperplanes, and the leaves are the indexes of the only host in the corresponding cluster. Given a query, say the watermarked $\tilde{\mathbf{I}}_i$, it is easy to traverse the tree from the root down to the correct leaf (which is i). First of all, the inner product between the query and the root of the index

tree is found to determine the position (left or right of the hyperplane) of the query with respect to the root. Then the query is recursively compared with the internal nodes of the tree. This eventually leads to the leaf to which the query belongs.

Under influence of AWGN, the query become $\mathbf{I}' = \tilde{\mathbf{I}}_i + \mathbf{z}$ where $\mathbf{z}$ is the noise. This additive noise might lead to error. Recall that the hyperplane is surrounded by a thick buffer zone. The width of this buffer zone is analytically determined, so that the probability of $\mathbf{I}'$ crossing the hyperplane is extremely small. Thus, robustness is achieved. In Section 4.3.1, it will be quantified how large the buffer zone should be in order to achieve the required robustness.

Since the index tree contains at most n hyperplanes, and each hyperplane can be represented by its normal and a point on its surface, the total size of the index tree is linear with respect to the size of $\mathcal{I}$. Because the tree is balanced, the depth of the tree is $O(\log n)$. Thus, a compact data structure that facilitates searching in large data sets is obtained, overcoming the dimensionality curse. The proposed algorithm was tested on hosts generated from Gaussian source and natural images. In the experiments conducted, the index trees was always successfully built by the proposed algorithm.

Section 4.3.1 describes the single-level clustering algorithm. Section 4.3.3 describes how recursive clustering can be performed while achieving requirements on robustness and distortion.

4.3.1 Single level clustering

The single level clustering attempts to solve this sub-problem: Given $\mathcal{I} = \langle \mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_n \rangle$, a distortion requirement ϵ_0 and the buffer zone's width τ_0, the goal is to find a hyperplane (represented by its normal $\mathbf{H}_0$ and a point $\mathbf{C}_0$ on the plane), and a watermarked $\widetilde{\mathcal{I}} = \langle \tilde{\mathbf{I}}_1, \tilde{\mathbf{I}}_2, \ldots, \tilde{\mathbf{I}}_n \rangle$, such that:

1. The distortion is at most ϵ_0, that is $\sum_i \|\tilde{\mathbf{I}}_i - \mathbf{I}_i\|_2^2 \leq n\epsilon_0$.

2. For any watermarked $\tilde{\mathbf{I}}$, the distance of $\tilde{\mathbf{I}}$ from the hyperplane is at least τ_0 (that is, $|(\tilde{\mathbf{I}} - \mathbf{C}_0) \cdot \mathbf{H}_0| > \tau_0$).

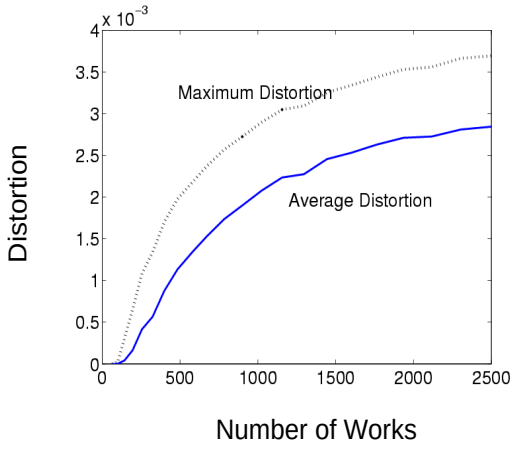


Figure 4.3: Performance of the single level clustering as the number of hosts increases. The dimension $d = 64^2$ and the width of buffer zone is $\tau_0 = 5/\sqrt{d}$. The upper graph gives the largest distortion among the n hosts. The lower graph gives the average distortion.

3. Furthermore, the hyperplane divides the watermarked hosts into two equal (within constant factor) halves. That is, suppose $\mathcal{I}_0$ is the set of watermarked $\tilde{\mathbf{I}}$ where $(\tilde{\mathbf{I}} - \mathbf{C}_0) \cdot \mathbf{H}_0 > 0$, then
$$\frac{1}{4} < |\mathcal{I}_0|/|\mathcal{I}| < \frac{3}{4}.$$

Figure 4.2 illustrates the result of a single level clustering in 2-dimensional space. This problem can be rephrased as an optimization problem by fixing the buffer zone's width τ_0 and minimizing the distortion, or vice versa.

Here is an approximation method, based on the 2-mean algorithm:

1. Compute the 2 means, $\mathbf{m}_0$ and $\mathbf{m}_1$ using the well-known iterative k-means method [37]. Let $\widehat{\mathbf{H}} = \mathbf{m}_0 - \mathbf{m}_1$ and $\widehat{\mathbf{C}} = (\mathbf{m}_0 + \mathbf{m}_1)/2$.

2. Partition $\mathcal{I}$ into two clusters $\mathcal{I}_0$ and $\mathcal{I}_1$, where $\mathcal{I}_0$ contains all the hosts in $\mathcal{I}$ that is nearer to $\mathbf{m}_0$, and $\mathcal{I}_1$ contains the remaining. Specifically, if $(\mathbf{I} - \widehat{\mathbf{C}}) \cdot \widehat{\mathbf{H}} > 0$, then $\mathbf{I}$ is in $\mathcal{I}_0$.

3. Find the hyperplane that separates $\mathcal{I}_0$ and $\mathcal{I}_1$. We want to find the hyperplane with the maximum distance from its nearest host. This criterion is the same as

the criterion on *Support Vector Machine*, which finds the separating hyperplane of two clusters with the largest margin. Here, the two clusters are $\mathcal{I}_0$ and $\mathcal{I}_1$, and the margin corresponds to the buffer width.

Support Vector Machine is an established technique, and the support vectors can be efficiently found using quadratic programming [21]. The theory of Support Vector Machines and its use in single level clustering is explained in Section 4.3.2. Let $\mathbf{H}_0$ and $\mathbf{C}_0$ be the normal and a point on this hyperplane respectively.

4. For all $\mathbf{I}$ in the buffer zone, watermark them by shifting them along the direction $\mathbf{H}_0$ and away from $\mathbf{C}_0$. They are shifted until they reach the surface of the buffer zone. Specifically, if $(\mathbf{I} - \mathbf{C}_0) \cdot \mathbf{H}_0 \geq 0$, then the watermarked $\tilde{\mathbf{I}}$ is

$$\tilde{\mathbf{I}} = \mathbf{I} + \max(0, \tau_0 - (\mathbf{I} - \mathbf{C}_0) \cdot \mathbf{H}_0)\mathbf{H}_0, \tag{4.2}$$

otherwise

$$\tilde{\mathbf{I}} = \mathbf{I} - \max(0, \tau_0 + (\mathbf{I} - \mathbf{C}_0) \cdot \mathbf{H}_0)\mathbf{H}_0,$$

Now, the relationship between τ_0 and the required robustness σ^2 is found. It can be claimed that, to achieve robustness σ^2, the buffer width τ_0 should be,

$$\tau_0 = A_d\sqrt{\sigma^2/d}, \tag{4.3}$$

where A_d is a slow-growing function, for example $\log d$. To see that, consider $\mathbf{I}'$ in Figure 4.2. The point $\mathbf{I}' = \tilde{\mathbf{I}} + \mathbf{z}$ is corrupted by noise $\mathbf{z}$. Error occurs during searching if the noise vector z, after being projected onto the one-dimensional normal $\mathbf{H}_0$, is more than τ_0 (or $-\tau_0$ depending on which side $\tilde{\mathbf{I}}$ is in). Because the noise is AWGN with variance σ^2, the distribution of the one-dimensional projected noise is also normally distributed but with variance σ^2/d. Since the probability of deviation from the standard deviation $\sqrt{\sigma^2/d}$ is small, τ_0 can be chosen to be $\tau_0 = A_d\sqrt{\sigma^2/d}$, where A_d is a slow-growing function, for example $\log d$. In the experimental studies (Section 4.4), instead of a slow-

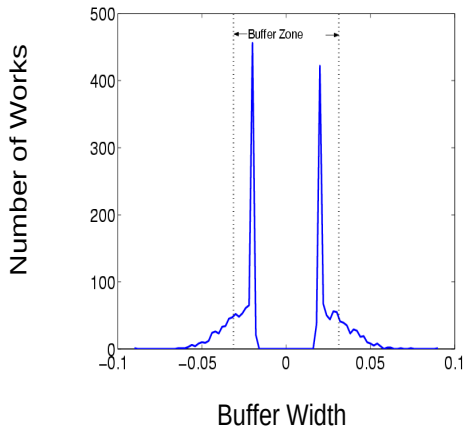


Figure 4.4: Histogram of the distances of original hosts from the hyperplane. The dimension $d = 64^2$, $n = 2048$. In-between the two vertical lines is the buffer zone with width $\tau_0 = 2/\sqrt{d}$.

growing function, A_d is chosen to be the constant 3. This gives the probability of error about 0.0015.

4.3.2 Remark on Support Vector Machines

Support Vector Machines have been deployed successfully in classification and regression tasks. In classification, Support Vector Machines are used to find an "optimal" separating hyperplane between two classes of data points. Among all separating hyperplanes, the optimal separating hyperplane is the one with the maximum distance from the nearest data points. Support vectors are those data points, nearest to the optimal hyperplane. The optimal separating hyperplane can be found by solving a quadratic programming problem for which efficient algorithms exist. (For details refer [33, 13]) In this implementation simplified support vector machine is used to find a separating hyperplane H_0. Although the separating hyperplane is not optimal, it serves as a good approximation for the intended purpose.

The method for finding the separating hyperplane is as follows. Initially, the width w i.e., the distance of the nearest point, is estimated based on the assumption that the data points are normally distributed. The separating hyperplane $\mathbf{H}_0$ is found out by an

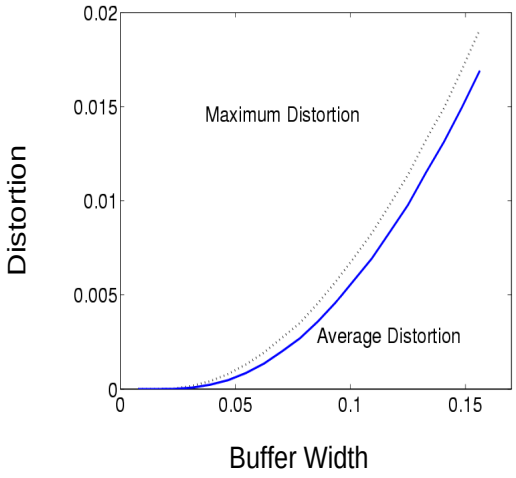


Figure 4.5: Distortion versus the buffer zone's width τ_0. The number of hosts is fixed at $n = 2048$ and the dimension is $d = 64^2$.

iterative algorithm. The initial guess for $\mathbf{H}_0$ is taken as the perpendicular bisector of the line joining the center of the two classes. A few data points nearest to the hyperplane are identified. The hyperplane is then updated so that its distance from the data points is at least $(w/2)$. This process is repeated for several iterations.

4.3.3 Extension to Multi-level

Extending the single level clustering to multi-level without special care might violate the robustness requirement. Recall that step 4 in Section 4.3.1 moves hosts out of the buffer zone. There are chances that the newly watermarked hosts re-enter the buffer zone created in previous levels. Geometrically, the buffer zone is an union of slabs, and the non-buffer zone is divided into disjoint polyhedrons. The task of watermarking is to move original $\mathbf{I}$ out of the buffer zone and to the nearest point in the non-buffer zone, which is on the surface of a polyhedra. For simplicity in implementation, instead of finding the nearest point on the polyhedrons, step 3 is iterated to ensure buffer zones in all levels are empty. This iteration might not give the nearest point. However, it converges fast and gives a good approximation.

In the optimization version (fixing overall distortion ϵ and maximizing overall ro-

bustness), the distortion can differ at different levels. Thus the allocation of "distortion" amounts to a resource allocation problem. Also, the allocation should be fair so that every level of clustering achieves same robustness. This is because a low robustness at any intermediate level ultimately determines the robustness of the final stage. Let $\epsilon_0, \epsilon_1, \ldots \epsilon_k$ be the distortion allocated to the k levels. Assuming that the hyperplanes at different levels are orthogonal, then the overall distortion is $\sum_i^k \epsilon_i = \epsilon$. The allocation of ϵ should be such that $\epsilon_i = \epsilon Dist(2^{-i}n)/B$, where $Dist(\cdot)$ is the average distortion as a function of the number of hosts, and B is a normalizing factor such that $\sum_i Dist(2^{-i}n) = B$. The expected number of hosts at each level is $2^{-i}n$. So the expected distortion at each level is $Dist(2^{-i}n)$. The allocation of ϵ is done such that every level of clustering achieves same robustness. $Dist(\cdot)$ can be obtained empirically.

4.4 Experimental Results

Two sets of experiments are conducted. In the first set, the hosts are generated from Gaussian source. In the second set, the database are natural images, using the down-sampled 64 by 64 gray-scaled values as the features (Figure 4.8).

Random hosts. In these experiments, hosts are generated from a Gaussian source, more specifically, it is a multivariate Gaussian random variable $\mathbf{I} = (x_1, x_2, \ldots, x_d)$ where each x_i is normally distributed with variance $1/d$. The embedding is performed directly on the x_i's, in other words, on the "pixel domain".

Figure 4.3, 4.4 and 4.5 illustrate the performance of the single level clustering. Figure 4.3 gives the average distortion as the number of hosts increases. When the number of hosts increases, the computed 2-means in Section 4.3.1 step 1 is closer to the overall average. Thus, the distortion should increase.

Figure 4.4 shows the distribution of the distance of the original hosts from the hyperplane. Note that these are the distances before watermarking. Observe that the center region is empty. This is because the hyperplane is derived from the support vectors. Thus, the slab enclosed by the support vectors is empty, even before watermarking. The

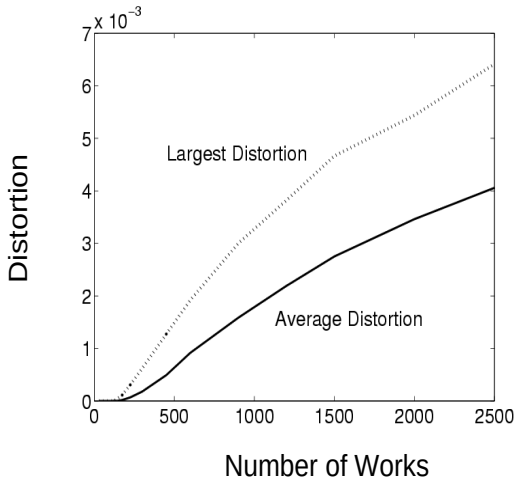


Figure 4.6: Distortion versus size of database.

two peaks in the histogram are side-effects of the proposed approximation algorithm. The two vertical lines in the figure indicate the buffer zone with $\tau_0 = 2/\sqrt{d}$. hosts that fall between these two lines have to be watermarked. Figure 4.5 shows how the width τ_0 affects distortion. Observe from the histogram that the hosts are concentrated around 0.025 and -0.025. Thus, for large τ_0, the distortion is approximately the square of the distance of τ_0 from 0.025. This observation is confirmed in Figure 4.5, where the distortion is approximately equal to $(\tau_0 - 0.025)^2$.

Figure 4.6 shows the overall distortion (generated by multi-level clustering) as the number of hosts increases. The width of buffer zones in all levels is kept at $\tau_0 = 3\sqrt{2/d}$. This value is chosen so that retrieval is robust under noise variance $\sigma^2 = 2$. That is, when the signal-to-noise ratio is at most 0.5. The distortion is generally very small. For example, at $n = 2048$, the distortion is 0.0027. This is much smaller than the energy of the hosts (which is 1). It is also significantly smaller than the noise variance $\sigma^2 = 2$. Figure 4.7 illustrates how distortion decreases as the dimensionality increases with number of hosts $n = 512$ and buffer width $\tau = 3\sqrt{2/d}$. Interestingly, performance improves as dimensionality increases. This is in contrast to general searching problems in high dimensional space.

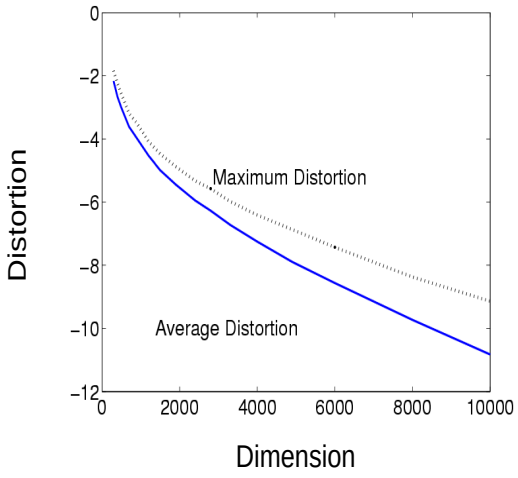
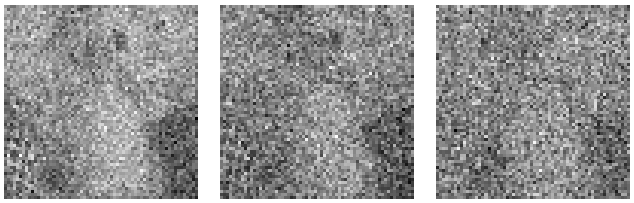


Figure 4.7: Distortion versus dimension. Logarithmic scale is used for the y-axis. The number of hosts $n = 512$ and width $\tau = 3\sqrt{2/d}$.

Natural Images. In this set of experiments, the database consists of 2048 natural images. The purpose of these experiments is to test the proposed idea on non-Gaussian source. The original size of each image is about 2000 by 1500 pixels. Although images are typically large, for watermarking purpose, their dimension is usually reduced to remove redundancies and coherence among the pixels. Because image representation is not a key issue here, the down-sampled 64 by 64 gray-scaled pixel domain is taken as the domain to work in. Thus $d = 64^2$. Other representations like wavelet coefficients, DCT coefficients, features etc. could also be used. Figure 4.8 shows samples from the database. Unlike the database of random hosts, some of the images are similar. Similar images are more difficult to handle, because they should be separated to resolve ambiguities.

The robustness σ^2 is chosen to be 2. This translates to the buffer zone's width of $\tau_0 = 3\sqrt{2/d}$. Figure 4.9 shows three instances of corrupted queries. The proposed algorithm successfully retrieves the correct index for (a) and (b), but not (c). The experiment is repeated for 1000 times, with same noise variance, but different noise instances. With noise variance of 1 and 2, the algorithm outputs the correct index for all instances. With noise variance of 4, it gives correct index in 655 instances. In the

Figure 4.8: Twelve sample images from the database.



(a) (b) (c)

Figure 4.9: Three watermarked corrupted queries. The noise variance is (a) 1, (b) 2 and (c) 4 respectively. The uncorrupted image is shown in the top-right corner of Figure 4.8. The proposed algorithm correctly retrieves the index for (a) and (b), but not (c).

implementation, the queries are normalized to unit energy before searching.

The average distortion generated is 8.5×10^{-4} and the maximum distortion among the images is 0.010. Figure 4.10 shows three watermarked images. The top row is the image with maximum distortion. Comparing to the random hosts (see Figure 4.6), the average distortion for the images database is much lower (0.0027 for random hosts) but the maximum distortion is higher (0.0052 for random hosts). Probably this is because natural images tend to form clusters, thus reducing the average distortion. On the other hand, a minority of the images might get too close, and require larger distortion for separation. This small cluster of images increases the maximum distortion.

Figure 4.11 shows selected nodes of the tree at the 1^{st}, 4^{th} and 8^{th} level. These

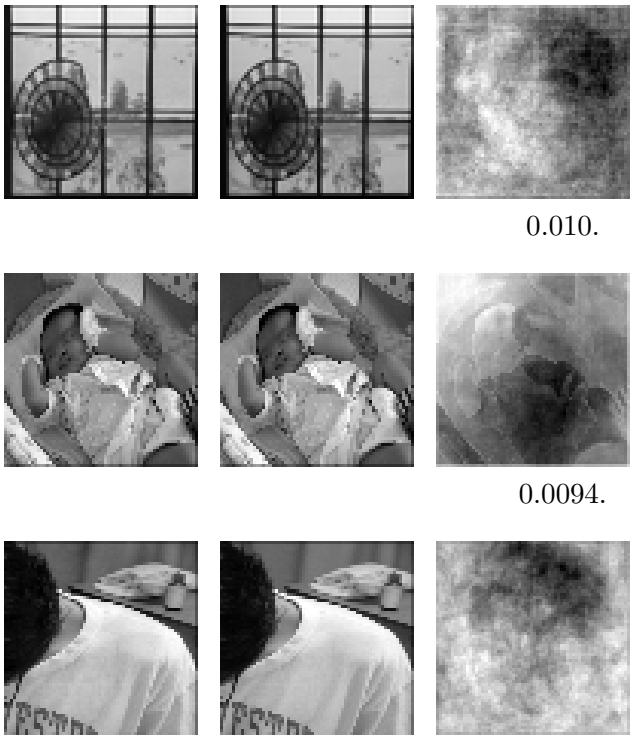


Figure 4.10: Images in column (a) are the original, (b) are the respective watermarked image and (c) are the difference (watermarked minus original). The images in (c) are enhanced (by scaling the intensity) for better printing quality. The values below the images are the distortion (that is, energy of (c)).

nodes are visited while searching for the top-right image in Figure 4.8. That is, the query image is first compared with $((a),(d))$, and finally compared with $((c),(f))$.

4.4.1 Comparison with watermarking

It is interesting to compare the performance of the proposed algorithm with methods based solely on watermarking. For the purpose of comparison, watermarking schemes which fall into the framework of Gaussian channel with side information are considered. Costa [26] showed that, the maximum achievable rate with distortion ϵ and robustness

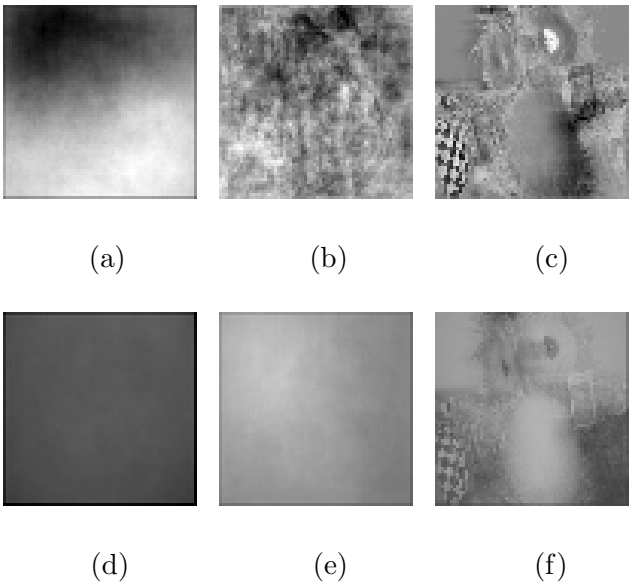


Figure 4.11: The normal **H** of the hyperplanes computed at the 1^{st}, 4^{th}, 8^{th} level are depicted as images (a), (b) and (c) . These normals can be viewed as the "watermarks". Image (d), (e) and (f) are the corresponding point C on the respective hyperplanes.

σ^2 is

$$C = \frac{d}{2} \log \left(1 + \frac{\epsilon}{\sigma^2}\right). \tag{4.4}$$

That is, the maximum number of messages that can be embedded is 2^C. If solely watermarking is employed to solve the identification problem, with the constraint on distortion and robustness, the size of the database is bounded above by $(1 + \epsilon/\sigma^2)^{d/2}$. From the experimental data in Section 4.4, with robustness $\sigma^2 = 2$, dimension $d = 64^2$ and distortion 0.0035, the proposed method can have 2048 hosts. In contrast, the theoretical maximum number achievable by watermarking is $(1 + 0.0035/2)^{d/2} < 36$.

4.4.2 Comparison with Retrieval systems

Nearest neighbor searching in high dimensions is both practically and theoretically difficult. The performance of all known and popular data structures for this problem degrades considerably for high dimensions. A brute force method would take $O(dn)$ time

complexity. The average case analysis of popular heuristics such as k-d trees reveals an exponential dependence of d in the query time. It is well known that the k-d tree and its variants perform poorly in dimensions higher than twenty. In such high dimensions, nearly the entire data structure is searched when it is queried for a near or nearest neighbor. Comparatively in high dimensions, the brute force method is more efficient.

There are many approximation algorithms, out of which [60] gives a theoretical construction of two algorithms which are close to the proposed method. The two algorithms perform the ϵ-approximate nearest neighbor search in a d-dimensional Euclidean space with a query time complexity of $O((d \log^2 d)(d + log\ n))$ and $O(n + d \log^3 n)$, where the $O(\cdot)$ notation suppresses terms that are quadratic in ϵ^{-1} . Also much work has been devoted to "dimension-reduction" techniques such as principle component analysis [54] and latent semantic indexing [34] for reducing the complexity of similarity matching in high dimensions, yet the dimensions are quite high and hence query time complexity is high. Moreover, the performance comparison results analyzed in such techniques deal mainly with specialized data sets. In contrast, experimental studies show that the proposed method, with fixed distortion, can successfully build the index tree, and searching in it involves a query time complexity $O(d \log n)$.

Chapter 5

Detection using Partial Information

5.1 Introduction

In this chapter a novel framework for database watermarking is presented. In chapters 3 and 4, two unified frameworks based on a combination of retrieval and watermarking systems were proposed. In both cases, the detector has access to the full hosts database. Some applications, do not have this facility, for example, blind watermarking, where the detector only has access to the messages associated with the hosts. Although most applications do assume that the messages (identifying information) should be available at the detector at all times, it need not be so. Note that the identifying information can be made available to the detector through a secure communication channel between the detector and encoder, which can also make the detector secure from some attacks. This facility brings in the additional consideration that the information to be communicated should adjust to the bandwidth constraints of the channel. This affects both encoding and detection.

Watermarking using communication has been proposed in some recent applications. For example, the detector in Digimarc MediaBridge Reader [5] makes use of the Internet

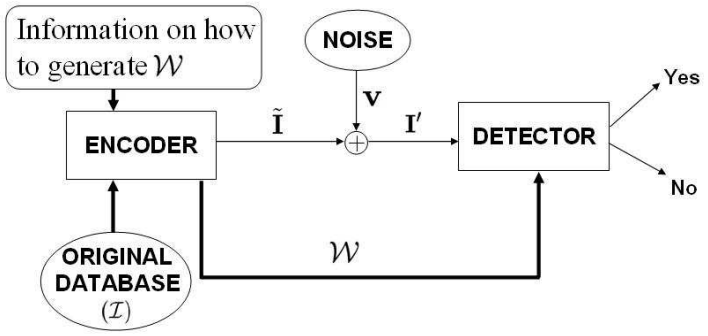


Figure 5.1: A schematic diagram of the proposed framework.

to lookup more information about an image from its server, which is part of the encoder, based on the message extracted from the image. Theoretical works like zero-knowledge detection [3] also exploit communication to enhance security.

In this chapter, a framework that exploits communication in another way is proposed. In the proposed framework, the encoder knows the actual hosts that are to be watermarked. Based on this knowledge, the encoder tailor-makes a set of watermarks and suitably watermarks the hosts in the database. The watermarks are then stored in the server. To determine whether a query host is watermarked, the detector requests the watermarks to the server and performs detection (Fig. 5.1). The discussion here primarily deals with images as hosts, and the main idea can be extended to other multimedia sources.

Unlike spread spectrum watermarking [30] and watermarking as communication with side information [26], where both the encoder and detector has knowledge of the host distribution and only the encoder has knowledge of the host state, in the proposed framework, the additional assumption that the encoder knows the **actual** hosts to be watermarked, and the detector knows a compact but partial description of the actual hosts (Fig. 5.1) is exploited. Note the fundamental difference between *a priori* knowledge of the distribution and knowledge of the actual database. The actual hosts are samples from the distribution, and can be used to estimate the distribution. On the other hand, knowing the host distribution is not sufficient to determine the actual hosts.

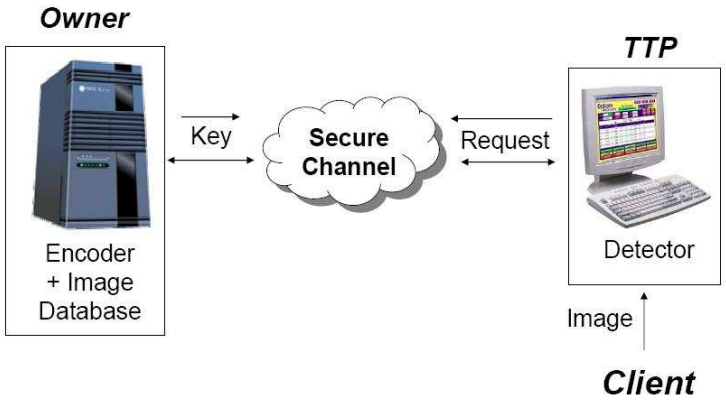


Figure 5.2: An application of our proposed framework.

Also note that in practice, the host distribution is usually assumed to be Gaussian, which is an over simplification for hosts like natural images. Here detection can be seen as semi-blind watermarking, as proposed in [14]. The partial or compact description in [14] is a hash of the original host. In the proposed framework the compact description is a partial description of the database. The considerations behind determining a compact description follows.

Compact description of database. Given the database of hosts, say $\mathcal{I}$, a possible but inefficient scheme takes the whole database as the watermarks, $\mathcal{W} = \mathcal{I}$. Thus nothing is done during encoding and zero distortion is achieved. To decide whether a query I' is watermarked, $\mathcal{W}$ is sent over to the detector, which searches for I' in $\mathcal{W}$. If it is within the proximity of a watermark in $\mathcal{W}$, then it is declared to be watermarked. This is similar to non-blind watermarking, with the additional work of the detector having to search for the appropriate watermark. Although this scheme achieves zero distortion, the number of watermarks is too large and thus inefficient. This brings forth the interesting issue of finding a trade-off between efficiency (number of watermarks) and performance (distortion, false alarm, and robustness).

Potential Applications A potential application of the proposed framework is as follows. An owner may posses a database of images and is concerned about protecting

his copyright over it. To address his concern, a practical setting, where a decision about the ownership over a disputed image has to be resolved, is conceived. The setting consists of three entities, the owner of the database, the client who has a query image and a trusted third party (TTP). The detector can be considered to be the trusted third party whom both the client and the owner trust. A secure channel exists between the owner and the trusted third party, using which, it has access to a compact description (partial information) of the database, as provided by the owner. The client trusts the TTP and submits the query image to it for any issue on validation of ownership. This security model prevents the client from giving away his image to the owner whom he does not trust. On the other hand it saves the owner from giving away the compact description of his database to any client whom he does not trust. Also the TTP need not access the whole database. Thus the detector acts like a proxy server. The client submits his image to the detector (TTP). The TTP contacts the owner for the compact description of the database using the secure channel and uses it to verify whether the image belongs to the owners database. Figure 5.2 gives an illustration of the above application setting.

Another possible application of this framework could be in relational database watermarking, where the whole database is watermarked. During detection it is required to verify whether a record or a set of records is watermarked.

5.2 Formulation

Let $\mathcal{I} = \langle \mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_m \rangle$ be a database of m images. Each image is a sequence of d coefficients which is generated from a source distribution. Given the image database $\mathcal{I}$, the encoder derives a compact description of the database $\mathcal{W}$. The compact description is organized as a set of ℓ watermarks $\mathcal{W} = \langle \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_i, \ldots, \mathbf{w}_\ell \rangle$, and is used to generate an encoded $\widetilde{\mathcal{I}} = \langle \tilde{\mathbf{I}}_1, \tilde{\mathbf{I}}_2, \ldots, \tilde{\mathbf{I}}_m \rangle$. The compact description is shared between the encoder and the detector using the secure channel between them. The size of $\mathcal{W}$, given by $|\mathcal{W}|$, is limited by the bandwidth of the secure channel. During detection,

given an image $\mathbf{I}'_i$ and the set of watermarks $\mathcal{W}$, the detector declares whether it is watermarked (output `Yes`) or not watermarked (output `No`).

$\mathcal{W}$ is organized as a set of watermarks given by $\mathcal{W} = \langle \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_i, \ldots, \mathbf{w}_\ell \rangle$. Each $\mathbf{w}_i$ is a sequence of d coefficients. For example, given a channel bandwidth of 524288 bits, $\mathcal{W}$ of size 524288 bits can be organized as one 256×256 sized image of pixel depth 8 (written as $[256 \times 256 \times 8]$) i.e., $\ell = 1$ and $d = 524288$) or two $[256 \times 256 \times 4]$ images (i.e., $\ell = 2$ and $d = 262144$) and so on, thus giving a set of watermarks.

The database of images can be realized in two setting, *static* and *dynamic* setting.

Static Setting In the static setting the image database remains unchanged, i.e., there are no addition or deletion of images from the database. The images in $\mathcal{I}$ are watermarked by $\mathcal{W}$ to generate the set of encoded $\widetilde{\mathcal{I}} = \langle \tilde{\mathbf{I}}_1, \tilde{\mathbf{I}}_2, \ldots, \tilde{\mathbf{I}}_m \rangle$. Since the size of the database does not change, the watermarks once determined remain unchanged throughout the process.

Dynamic Setting In the dynamic setting, the database changes as new images are added or deleted. Let $\mathcal{I}_t = \langle \mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_t \rangle$ be the database with the first t images, and $\widetilde{\mathcal{I}}_t$ be the corresponding set of encoded images. The encoding is done in an online manner, that is, the image $\mathbf{I}_t$ must be encoded before $\mathbf{I}_{t+1}$ arrives. Once an $\tilde{\mathbf{I}}_t$ is obtained it cannot be recalled for modification. Similar to the static setting, detection is done using the set of watermarks $\mathcal{W}$. However, because the database dynamically changes, $\mathcal{W}$ also changes. The watermarks are updated once a new image arrives, where, $\mathcal{W}_t = \langle \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_\ell \rangle$ denotes the set of watermarks after $\mathbf{I}_t$ has arrived and been encoded. An additional requirement is that the set $\mathcal{W}_t$ has to be *backward compatible*, i.e., the detector must be able to detect $\tilde{\mathbf{I}}_s$, for any $s \leq t$ based on $\mathcal{W}_t$.

Performance measures The performance measures for the proposed framework are: *false alarm*, *robustness* and *distortion*. Distortion is measured by finding the average

distance between an image $\mathbf{I}$ and its watermarked version $\tilde{\mathbf{I}}$ i.e.,

$$\frac{1}{m}\sum_{i=1}^{m}\|\mathbf{I}_i - \tilde{\mathbf{I}}_i\|_2^2.$$

The scheme is robust if, under AWGN attack, an encoded image is still declared to be watermarked with high probability (the actual probability is not a concern). The variance of AWGN is taken as the measure of robustness. The false alarm is the probability of a randomly chosen sequence (from the image distribution) to be declared as watermarked by the detector.

5.3 Watermarking Schemes

Herein, a variant of the well-known spread-spectrum method [30] is considered as the underlying watermarking method. However, the proposed framework is not restricted to this method, and can be extended to other methods. The watermarking method used can be considered to be a special case of the formulation for improved spread spectrum watermarking (ISS) proposed in [67].

This variant is parameterized by a watermark $\mathbf{w}$, a constant threshold T and a constant $\mathtt{K}$, which is representative of the detection strength. The encoding of $\mathbf{I}$ giving $\tilde{\mathbf{I}}$ is achieved by

$$\tilde{\mathbf{I}} = \mathbf{I} + \max(0, \mathtt{K} - \mathbf{I}\cdot\mathbf{w})\mathbf{w}. \tag{5.1}$$

The $\mathbf{w}$ is normalized so that $\|\mathbf{w}\|_2^2 = 1$. An $\mathbf{I}'$ is declared to be watermarked if

$$\mathbf{I}'\cdot\mathbf{w} > T. \tag{5.2}$$

The false alarm, robustness and distortion of this scheme can be obtained analytically. The threshold T is experimentally determined.

For the encoding method in (5.1), depending on how $\mathbf{w}$ is generated, watermark-

ing schemes under the static and dynamic database setting are proposed next. Their performance is compared with a traditional spread-spectrum method, where the watermark is an IID sequence. A traditional spread spectrum method can be viewed as an equivalent scheme with no knowledge of the image database. The false alarm and robustness is fixed in all schemes, which amount to the parameters K and T being fixed. For performance comparison of different schemes, by fixing false alarm and robustness, the distortion obtained in each scheme are compared.

5.3.1 Static with single watermark

This section gives two schemes *static* and *static iterative*. The encoder of *static* scheme, computes the normalized sum of the database and quantizes it to generate the compact description $\mathcal{W}$, that is,

$$\mathcal{W} = q\left(\sum_{\mathbf{I}\in\mathcal{I}} \mathbf{I} / \| \sum_{\mathbf{I}\in\mathcal{I}} \mathbf{I} \|\right), \tag{5.3}$$

where $q(\cdot)$ is the quantization function, and $\| \cdot \|$ indicates the norm. The amount of quantization is decided according to the bandwidth of the secure channel. For a static database with single watermark setting, $\mathcal{W}$ is organized to form one watermark i.e., $\mathcal{W}$ $= \langle \mathbf{w} \rangle$.

The encoding and detection is same as the method given in (5.1) and (5.2). Unlike the traditional spread spectrum (SS) method, where the watermark is randomly chosen, in this scheme, $\mathbf{w}$ is computed from the image database. Note that if both parameters K and T are fixed, then the false alarm and robustness of this scheme are the same as that of the traditional spread spectrum method. By fixing K and T, the goal is to know which scheme provides lower distortion.

The distortion of each image in (5.1) is a function of its correlation with the watermark. The difference between the detection strength K and the correlation gives a measure of the amount of distortion needed to be introduced. Distortion is zero when $\mathbf{I}\cdot\mathbf{w} > \mathtt{K}$. Otherwise it is proportional to the correlation $\mathbf{I}\cdot\mathbf{w}$. So for the whole database,

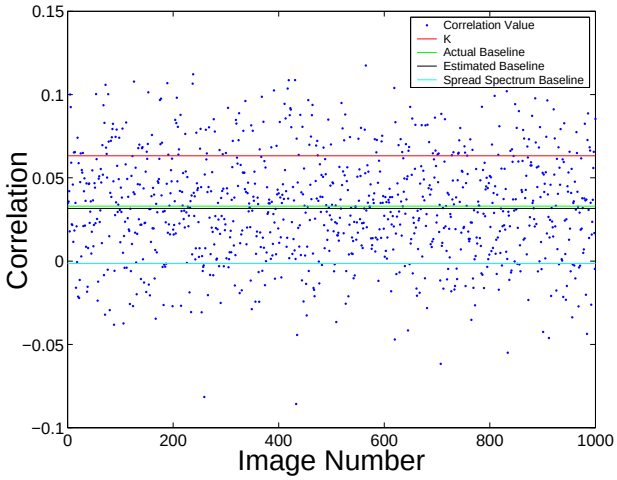


Figure 5.3: Correlation of Images in database with watermark for 1000 images. This figure illustrates the importance of $\mathcal{B}$ in defining the amount of distortion required for successful detection. The horizontal red line indicates `K`. The black and green horizontal lines indicate the estimated and actual average correlation values. The cyan line indicates the average correlation value for the spread spectrum scheme. Note that the distance of the cyan and green lines from the red line indicates the amount of distortion required in the spread spectrum and static scheme respectively to ensure successful detection.

the reduction in distortion can be analyzed by finding the average correlation value, $\mathcal{B}$, of $\mathbf{w}$ with the images in $\mathcal{I}$,

$$\mathcal{B} = \frac{1}{m} \sum_{\mathbf{I} \in \mathcal{I}} \mathbf{I} \cdot \mathbf{w}.$$

The average correlation is a good indicator of the reduction in distortion. The expected value of the baseline under both traditional SS and single-watermark static setting are as follows:

Traditional SS setting: For all $\mathbf{I}_i \in \mathcal{I}$, $\mathbf{I}_i = \{I_{i1}, \ldots, I_{ij}, \ldots, I_{id}\}$ with $I_{ij} \sim \mathcal{N}(0,1)$, and $\mathbf{w} = \{w_1, \ldots, w_j, \ldots, w_d\}$ with $w_j \sim \mathcal{N}(0,1)$, the expected baseline is given by $E(\mathcal{B}) = 1/m \sum_{\mathbf{I} \in \mathcal{I}} E(\mathbf{I} \cdot \mathbf{w}) = 0$.

Static single-watermark setting: For all $\mathbf{I}_i \in \mathcal{I}$, $\mathbf{I}_i = \{I_{i1}, \ldots, I_{id}\}$ and $I_{ij} \sim \mathcal{N}(0,1)$, then $\mathbf{w} = \{w_1, \ldots, w_j, \ldots, w_d\} = \frac{\sum_{i=1}^{m} \mathbf{I}_i}{\|\sum_{i=1}^{m} \mathbf{I}_i\|}$ can be represented as

$$w_j = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} I_{ij},$$

to ensure that $w_j \sim \mathcal{N}(0,1)$. Note that the definition of norm used here is,

$$\|\sum_{i=1}^{m} \mathbf{I}_i\| = \sqrt{var(\sum_{i=1}^{m} \mathbf{I}_i))} = \sqrt{\sum_{i=1}^{m} E(\mathbf{I}_i^2)}.$$

The expected baseline is given by

$$E[\mathcal{B}] = \frac{1}{m\sqrt{m}} \sum_{\mathbf{I} \in \mathcal{I}} (E[\mathbf{I} \cdot \mathbf{I}] + E[\mathbf{I} \cdot (\sum_{\mathbf{I}_j \neq \mathbf{I}, \mathbf{I}_j \in \mathcal{I}} \mathbf{I}_j)]) = \frac{d}{\sqrt{m}}.$$

Since $\mathcal{B}$ is raised to $d/\sqrt{m}$, the distortion required to "push" the image over the detection strength $\mathtt{K}$ is reduced. This is illustrated in Figure 5.3. Hence, theoretically, the static single-watermark setting achieves lower distortion than a traditional SS watermarking technique.

Figure 5.4 confirms the gain in performance achieved when the watermark $\mathbf{w}$ is obtained using (5.3) compared to a watermark whose components belong to a normal distribution of zero mean. Here, distortion is calculated for fixed robustness and false-alarm (that is $\mathtt{K}$ and T). Depending on the channel bandwidth, the size of the watermark will vary. For a static single-watermark setting, the compact representation $\mathcal{W}$ consists of a single watermark $\mathbf{w}$. For watermarks of fixed size, 256×256, to cater to varying channel bandwidth, the pixel depth is varied. Figure 5.4 compares the effect of such watermarks on the watermarking performance. Higher the pixel depth better is the performance. Note that for a given channel bandwidth as low as 131072 bits the distortion is significantly lower compared to a scheme using a traditional SS method. For watermarks with unit pixel depth, the distortion encountered by traditional SS methods is lesser.

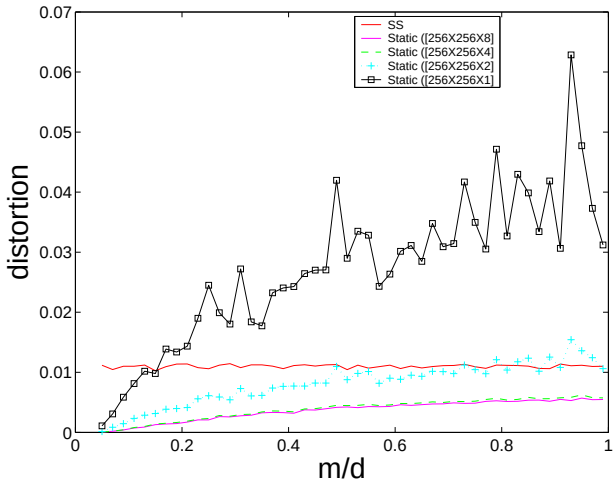


Figure 5.4: Distortion verses (m/d) for images from a Gaussian distribution for static single watermark setting with different bandwidths of secure channel. The number of coefficients is fixed at $d = 256 \times 256$, and $\mathtt{K} = 0.1$. The above graph shows that from bandwidth of $256 \times 256 \times 8$ bits till bandwidth of $256 \times 256 \times 2$ bits the watermarking performance in terms of distortion is better for the static single watermark scheme than a spread-spectrum scheme. For a bandwidth of $256 \times 256 \times 1$ bits, the spread-spectrum scheme performs better. Note that for the static setting as the size of the database increases, the distortion increases.

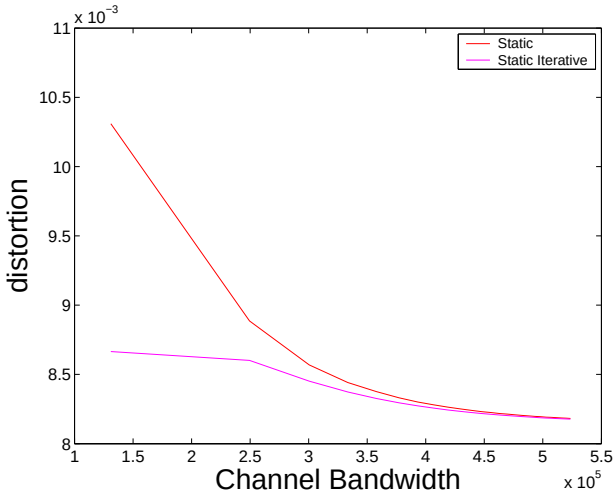


Figure 5.5: Comparison of distortion under various channel bandwidths between static and static iterative setting, for images from a natural image database. The database consists of 1000 images. The number of coefficients is fixed at $d = 256 \times 256$, and $\mathtt{K} = 0.1$. Note that a small improvement in performance is achieved using the static iterative setting.

The second encoder considered is a (*static iterative*) scheme, which tries to further improve the performance by searching for the watermark $\tilde{\mathbf{w}}$ which minimizes the average distortion. The minimization is done in an iterative manner, such that

$$\frac{1}{m}\sum_{\mathbf{I}\in\mathcal{I}}\max(0,\mathtt{K}-\mathbf{I}\cdot\tilde{\mathbf{w}}),$$

is minimized. The algorithm for the static iterative scheme is as follows:

Algorithm: Static Iterative Scheme	
Step1 :	Find $\tilde{\mathbf{w}} = q\left(\sum_{i=1}^{m}\mathbf{I}_i/\|\sum_{i=1}^{m}\mathbf{I}_i\|\right).$
Step2 :	Initialize $\mathbf{w}=\tilde{\mathbf{w}}$. For all $\mathbf{I}_i\in\mathcal{I},$ find $c_i=\mathbf{I}_i\cdot\tilde{\mathbf{w}}$.
Step3 :	For all $\mathbf{I}_i\in\mathcal{I}, \tilde{\mathbf{w}}=\tilde{\mathbf{w}}-\max(0,\mathtt{K}-c_i)\cdot\mathbf{I}_i$.
Step4 :	Until $\|\mathbf{w}-\tilde{\mathbf{w}}\|_2^2<0.001$ Go to **Step2**.

A performance comparison between the static and static iterative setting with varying channel bandwidth is illustrated in Figure 5.5. Note that for low channel bandwidths there is a marked difference in distortion between the two schemes, which gives an indication of the gain in performance that can be achieved by using the static iterative scheme. This shows that the *static iterative* scheme can be used to reduce the distortion encountered by a static scheme under low channel bandwidth, as noted before.

5.3.2 Static with Multiple watermarks

The multiple watermark scheme is particularly relevant when the size of the database increases. In this scheme, instead of organizing $\mathcal{W}$ as a single watermark, it is organized as a set of watermarks given by $\mathcal{W}=\langle\mathbf{w}_1,\mathbf{w}_2,\ldots,\mathbf{w}_k\rangle$. Thus the detector uses a set of watermarks as evidence during detection. An image $\mathbf{I}'$ is declared to be watermarked if there is an i such that $\mathbf{I}'\cdot\mathbf{w}_i>T$.

It is noted that if the size of the database is small, for a static single watermark setting the distortion should be small. This can be understood from the fact that for

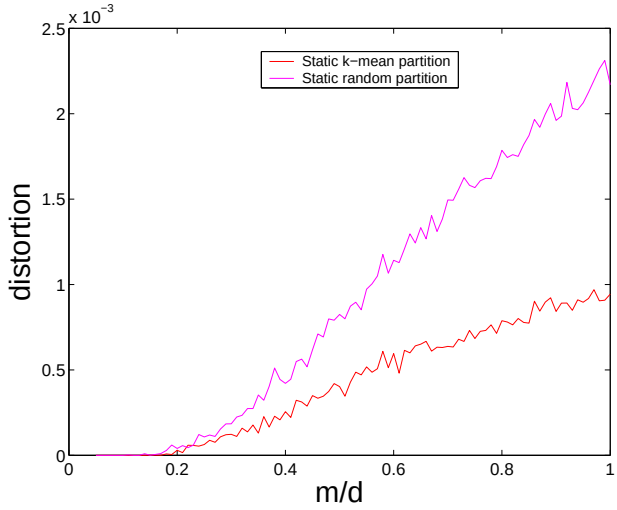


Figure 5.6: Illustration of the effect of partitioning algorithm on the distortion in a static multiple watermark setting, for images from a Gaussian distribution. The database consists of 1000 images. The number of coefficients is fixed at $d = 256 \times 256$, and $\mathtt{K} = 0.1$. Note that as the size of the database increases the efficacy of using a good partitioning algorithm becomes more prominent.

a database with a single image, the distortion is zero. Thus as the size of the database increases, the distortion increases. This was also illustrated in Figure 5.4. Thus for larger databases, if the database is divided into smaller subsets and the static single watermark scheme is applied on each subset, intuitively it will help reduce the distortion further. But the bottle neck in this option is that it would require more network resources to pass more watermarks to the detector. An extreme example as discussed in Section 5.1 considers every image in the database as a watermark. Hence for a secure channel with a limited bandwidth the number of watermarks cannot be significantly large. One proposition to solve this could be to reduce the pixel depth during the organization of $\mathcal{W}$ into watermarks. But as illustrated in Figure 5.4, as the bit depth decreases, the distortion increases significantly. Hence a proper trade-off is required to decide upon how to organize $\mathcal{W}$ as a set of watermarks.

To implement multiple watermark setting the database is partitioned into smaller subsets (say k) of images and the quantized normalized sum for each subset as in (5.3), is computed. This gives a set of k compact representations. Each of these compact

representations can be organized as a single watermark and thus a set of k watermarks is obtained. Note that the partitioning algorithm used to create the clusters also effects the distortion, for example, the database can be partitioned into k subsets using a random partitioning technique or a k-mean partitioning algorithm.

Figure 5.6 illustrates the effect of using a random partitioning, compared to using a 2-mean algorithm. For random partitioning, the database is randomly divided into two subsets of equal size. The watermark for each subset ($\mathbf{w}_1$ and $\mathbf{w}_2$) is generated using (5.3) by treating each subset as a single database. It is easy to verify that the average correlation value will improve by a factor of $1/\sqrt{2}$ and this implies that the distortion will also improve by approximately the same factor. On the other hand, the false alarm will increase by a factor of approximately 2. This is because in the 2-watermark situation, a sequence $\mathbf{I}$ is declared to be watermarked if either $(\mathbf{I} \cdot \mathbf{w}_1) \geq T$ or $(\mathbf{I} \cdot \mathbf{w}_2) \geq T$. However, constant factor growth of the false alarm is insignificant, because the false alarm decreases exponentially as the threshold T increases linearly. Thus, it is desirable to allow more watermarks, if efficiency is not a consideration. Alternatively, a good partition of the the database is found by using the well-known 2-mean algorithm. Figure 5.6 shows that the 2-means algorithm gives an improvement in distortion compared to randomly partitioning the database. Also note that as the size of the database increases, the effect of the partitioning algorithm on the distortion becomes prominent.

Several other inter-dependent parameters that affect the watermarking performance are, pixel depth, the number of watermarks and the channel bandwidth. These parameters are varied to analyze their effect on distortion. Figure 5.7 illustrates the performance of multiple watermarks setting for a fixed channel bandwidth. Note that the $1 \times [256 \times 256 \times 8]$ setting in Figure 5.7 is actually a single watermark static setting. For a fixed channel bandwidth as the number of watermarks increases the distortion decreases. Note that even for very low pixel depth, for example the $8 \times [256 \times 256 \times 1]$ setting, i.e., $\mathcal{W}$ is organized as 8 watermarks each quantized to 1 bit pixel depth, the

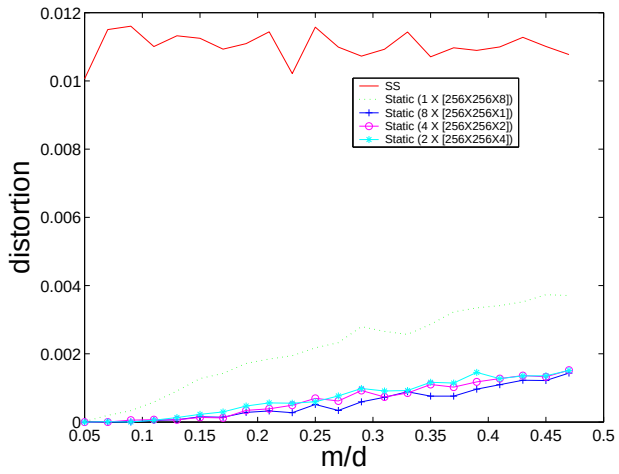


Figure 5.7: Distortion verses (m/d) for images from a Gaussian distribution for static multiple watermark setting with fixed bandwidth. The number of coefficients is fixed at $d = 256 \times 256$, and$\mathtt{K} = 0.1$. Note that the bandwidth of the side channel is fixed at 524288 bits. The above graph shows the performance of different sets of watermarks with varying database size and thus is an indicator of how to choose $\mathcal{W}$. Here $2\times[256\times256\times4]$ denotes 2 watermarks each of size 256×256 of pixel depth 4.

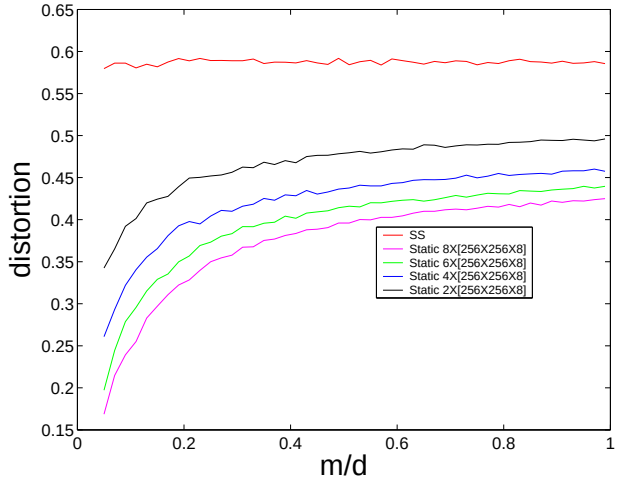


Figure 5.8: Distortion verses (m/d) for images from a Gaussian distribution for static multiple watermark setting with fixed pixel depth, i.e., for the same quantized amount. The number of coefficients is fixed at $d = 256 \times 256$, and $\mathtt{K} = 1$. The above graph shows the efficacy of choosing more number of watermarks, given that there is no constraint on bandwidth. Here $2 \times [256 \times 256 \times 8]$ denotes 2 watermarks each of size 256×256 of pixel depth 8.

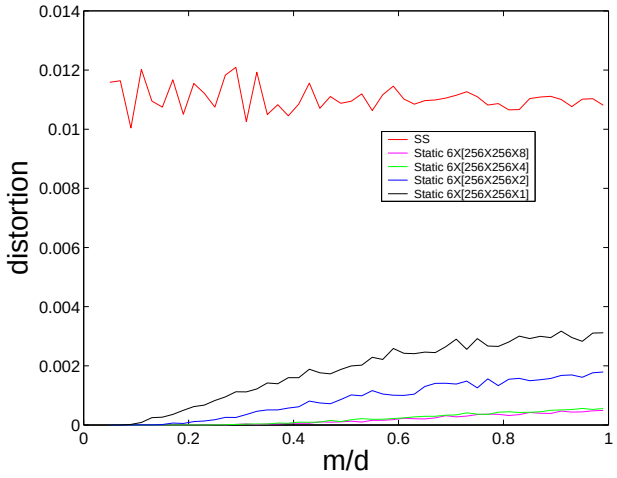


Figure 5.9: Distortion verses (m/d) for images from a Gaussian distribution for static multiple watermark setting with fixed number of multiple watermarks (6). The number of coefficients is fixed at $d = 256 \times 256$, and $\mathtt{K} = 0.1$. The above graph shows the efficacy of multiple watermark setting even with low pixel depth, i.e., if the channel bandwidth is low, the quantization can be high provided we have more watermarks, to get low distortion. Here $6 \times [256 \times 256 \times 2]$ denotes 6 watermarks each of size 256×256 of pixel depth 2.

distortion is lower compared to a traditional SS scheme. From Figure 5.7 it is noted that by increasing the number of watermarks, the effect of quantization (determined by the pixel depth) can be significantly traded-off. Figure 5.8 also shows the efficacy of using multiple watermarks for fixed quantization. Distortion decreases as the number of watermarks increases. This perfectly corroborates the discussion in Section 5.1. Figure 5.9 illustrates the effect on distortion for a fixed number of watermarks. This basically illustrates the effect of quantization on the watermarking performance. For low bandwidth the pixel depth should be increased. Even under high quantization the distortion is still low. This supports the previous conclusion that by having more watermarks the effect due to quantization can be reduced.

5.3.3 Dynamic with single watermark

In the dynamic setting, images arrive sequentially. Let $\mathbf{w}_1^t$ be the watermark computed after the arrival of $\mathbf{I}_t$. The encoding and detection is performed similar to (5.1) and

(5.2). The watermark satisfies the additional backward compatibility requirement, i.e., $\tilde{\mathbf{I}}_s \cdot \mathbf{w}_1^t \geq \mathtt{K}$, for any $s < t$ and $\tilde{\mathbf{I}}_s \in \widetilde{\mathcal{I}}$.

In this setting there are two interesting issues. The first issue is about how to enforce backward compatibility. Second, it is interesting to study the reduction in performance when information is available in an on-line manner, as opposed to the static setting where full knowledge of the database is available from the very beginning.

On arrival of the $t-$th image, the following iterative method searches for the new watermark $\mathbf{w}_1^t$.

Algorithm: Dynamic Single Key Generation

Step1 : Let $\mathbf{w}_1^t = \mathbf{w}_1^{t-1} + (1/\sqrt{t})\mathbf{I}_t$.

Step2 : If there is a $r < t$ such that $\mathbf{I}_r \cdot \mathbf{w}_1^t < \mathtt{K}$, then update $\mathbf{w}_1^t = \mathbf{w}_1^t + (\mathtt{K} - \mathbf{I}_r \cdot \mathbf{w}_1^t)\mathbf{I}_r$.

Step3 : Repeat **Step 2** until no such r is found.

It is important to choose the weighting function as $(1/\sqrt{t})$ in **Step 1**. This is done so that the estimated average correlation value still improves by $(1/\sqrt{t})$. For all $\mathbf{I}_t \in \mathcal{I}$, $\mathbf{I}_t = \{I_{t1}, \ldots, I_{tj}, \ldots, I_{td}\}$, if $I_{tj} \sim \mathcal{N}(0,1)$, the estimated value of $\mathbf{I}_t \cdot \mathbf{w}_1^t$ is

$$\begin{aligned} E[\mathbf{I}_t \cdot \mathbf{w}_1^t] &= E[\mathbf{I}_t \cdot \mathbf{w}_1^{t-1}] + \tfrac{1}{\sqrt{t}} E[\mathbf{I}_t \cdot \mathbf{I}_t] \\ &= 0 + \tfrac{d}{\sqrt{t}} \\ &= \tfrac{d}{\sqrt{t}} \end{aligned}$$

Similarly, for any other image $\mathbf{I}_s$ where $s < t$ and $I_{sj} \sim \mathcal{N}(0,1), 1 \leq j \leq d$, the estimated value of $\mathbf{I}_s \cdot \mathbf{w}_1^t$ is

$$\begin{aligned} E[\mathbf{I}_s \cdot \mathbf{w}_1^t] &= E[\mathbf{I}_s \cdot \mathbf{w}_1^{t-1}] + \tfrac{1}{\sqrt{t}} E[\mathbf{I}_s \cdot \mathbf{I}_t] \\ &= \tfrac{1}{\sqrt{t-1}} (E[\mathbf{I}_s \cdot \mathbf{I}_s] + E[\mathbf{I}_s \cdot (\textstyle\sum_{s \neq i} \mathbf{I}_i)]) + 0 \\ &= \tfrac{d}{\sqrt{t-1}}. \end{aligned}$$

Note that in a dynamic setting, as the size of the database increases incremen-

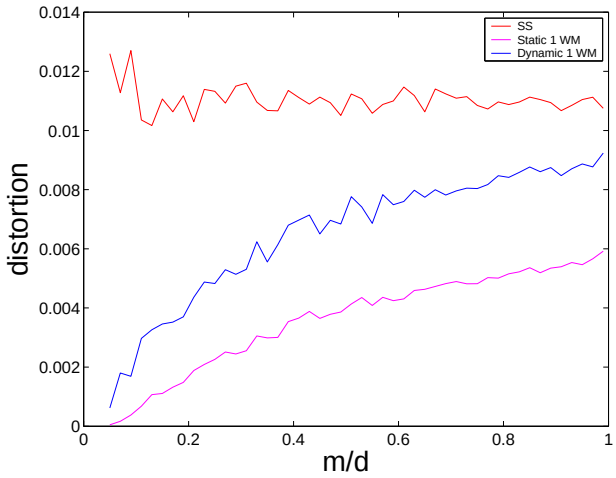


Figure 5.10: Distortion verses (m/d) for images from a Gaussian distribution for dynamic single watermark setting with fixed bandwidth of $256 \times 256 \times 8$ bits. The number of coefficients is fixed at $d = 256 \times 256$, and $\mathtt{K} = 0.1$. The above graph shows a comparison of performance between a spread spectrum, static and dynamic setting. Note that although the dynamic setting performs worse than a static setting it still improves upon a spread spectrum setting with random watermark.

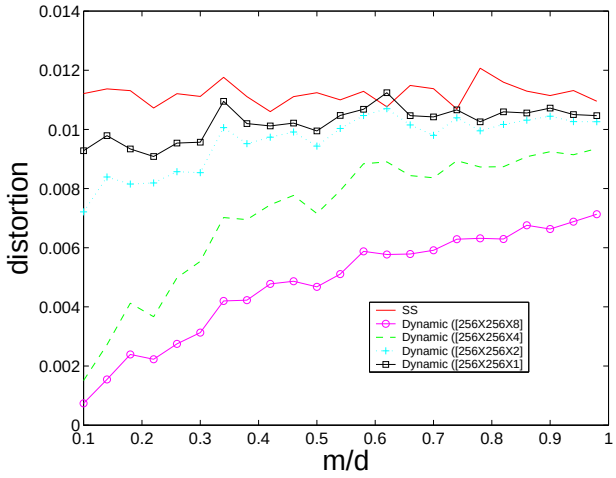


Figure 5.11: Distortion verses (m/d) for images from a Gaussian distribution for dynamic single watermark setting with varying channel bandwidth. The number of coefficients is fixed at $d = 256 \times 256$, and $\mathtt{K} = 0.1$. The above graph shows a comparison of performance of a dynamic setting under varying channel bandwidths. Note that for a low bandwidth of $256 \times 256 \times 1$ the distortion is still lower than a spread spectrum setting for most sizes of the evolving database. This is a promising sign of the efficacy of the dynamic setting in comparison to a spread spectrum scheme with a random watermark.

tally, the estimated average correlation value decreases sharply (note $d/\sqrt{t}$ is a sharply decreasing function). Intuitively, this explains the reduction in performance when information is available in an on-line manner. Hence compared to a static setting, for same number of images at the t-th time instant the distortion is expected to be more in a dynamic setting. This is illustrated in Figure 5.10, where the rise in distortion is sharper with increase in size of database. Note that still an improvement over a traditional spread spectrum setting is obtained. Backward compatibility of the watermark is implemented by modifying the watermark in an iterative way such that always a correct detection for any image in the database is obtained.

Figure 5.11 illustrates the performance of dynamic setting with single watermark in comparison to a spread spectrum setting. Note that for a channel bandwidth as low as $256 \times 256 \times 1$ (i.e., a single watermark with pixel depth 1), the distortion is less than the spread spectrum setting. As the allowed bandwidth increases the distortion decreases further. Still the sharp rise in distortion with increasing database size is evident. Thus a dynamic single watermark setting performs better than the spread spectrum setting using a random watermark, proving the efficacy of using prior knowledge of host database to generate the watermark, even under a dynamic database scenario.

5.3.4 Dynamic with multiple watermark

In this setting, more watermarks are allowed as is in the static multiple watermark setting (Section 5.3.2). The encoding unit employs a combination of the encoding used in the dynamic setting and the static multiple watermark setting. Here also a partitioning of the database is dynamically performed to generate the multiple watermarks. Note that watermarks in this setting evolve with time. On arrival of a new image $\mathbf{I}_t$, $\mathbf{I}_t \cdot \mathbf{w}_i$ is computed for all $\mathbf{w}_i \in \mathcal{W}$, to decide which subset it belongs to. Next the encoder uses an iterative technique (similar to dynamic single watermark setting) to ensure backward compatibility within its own subset of images. It was observed that in a dynamic setting as the size of the database increases incrementally, the distortion increases (i.e., average

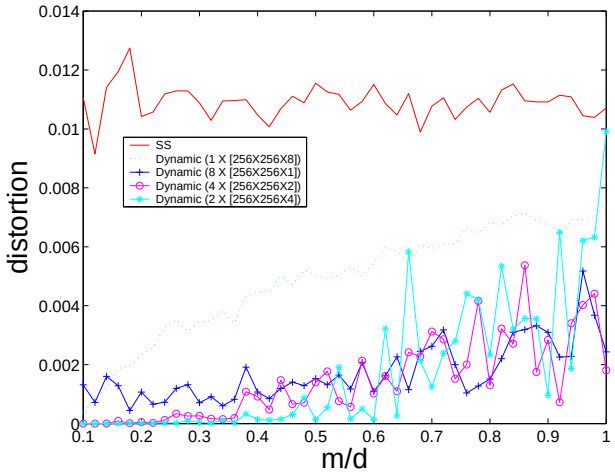


Figure 5.12: Distortion verses (m/d) for images from a Gaussian distribution for dynamic multiple watermark setting with fixed bandwidth. The number of coefficients is fixed at $d = 256 \times 256$, and $\mathtt{K} = 0.1$. Note that the bandwidth of the side channel is fixed at 524288 bits. The above graph shows the performance of different sets of watermarks with varying database size and thus in an indicator of how to choose $\mathcal{W}$. Herein $2 \times [256 \times 256 \times 4]$ denotes 2 watermarks each of size 256×256 of pixel depth 4.

correlation value given $d/\sqrt{t}$ decreases). Note that for a smaller value of t distortion can be lowered. By considering multiple watermarks, the database gets divided into smaller subsets and thus it is expected that distortion can be lowered.

Figure 5.12 gives a clue as to how $\mathcal{W}$ can be organized as a set of watermarks for fixed channel bandwidth in a dynamic setting. For fixed channel bandwidth, the amount of quantization and the number of watermarks need to be traded-off. Note that as the number of watermarks increases, distortion is less. For less number of watermarks even with relatively less quantization, the distortion is relatively higher, although less than in the traditional SS scheme. Figure 5.13 clearly illustrates the efficacy of using more watermarks. The distortion reduces significantly for more number of watermarks. Figure 5.14 compares the behavior of distortion to different amounts of quantization, denoted by pixel depth. Note that even under very low pixel depth the distortion is significantly less compared to traditional SS scheme.

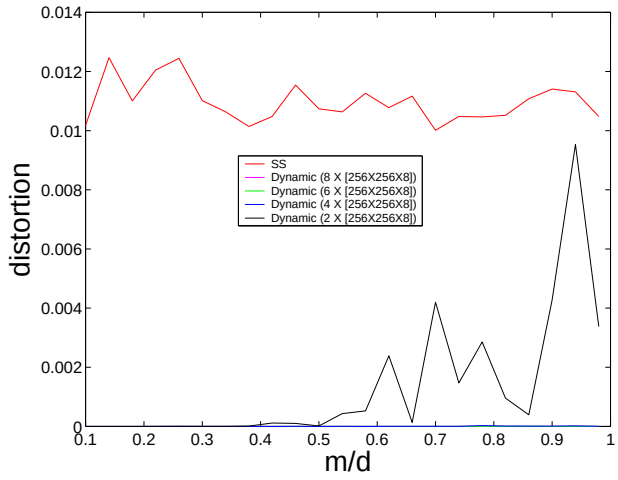


Figure 5.13: Distortion verses (m/d) for images from a Gaussian distribution for dynamic multiple watermark setting with fixed pixel depth. The number of coefficients is fixed at $d = 256 \times 256$, and $\mathtt{K} = 0.1$. For a fixed quantization it is very clear that increasing the number of watermarks reduces distortion significantly. Here $2 \times [256 \times 256 \times 8]$ denotes 2 watermarks each of size 256×256 of pixel depth 8.

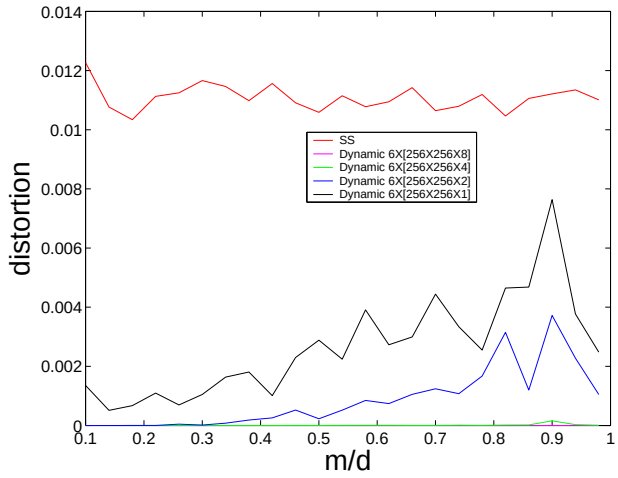


Figure 5.14: Distortion verses (m/d) for images from a Gaussian distribution for dynamic multiple watermark setting with fixed number of watermarks (6). The number of coefficients is fixed at $d = 256 \times 256$, and $\mathtt{K} = 0.1$. Note that even under high quantization an improvement over a spread spectrum scheme is obtained.

5.4 Experiment with Image Database

Experiments were conducted on a database of 1000 natural images from [1]. The main goal of this experiment was to test the efficacy of the proposed framework for non-Gaussian image distributions. Image representation is not the focus of this work, hence each image is represented by its re-scaled 256 by 256 gray-scale image. The efficacy of generating the watermark from the database is illustrated in Figure 5.15. Row 1 in Figure 5.15, shows five of the original images in the database. Row 2 in Figure 5.15 depicts the images in Row 1 watermarked using the static single-watermark scheme and Row 3 depicts images in Row 1 watermarking using a random watermark. The images are normalized to unit energy. Watermarking is carried out in the pixel domain. The images and the watermarks are all normalized and thus have energy 1. Hence it can seen that the chosen detection strength $\mathtt{K} = 1$ is high.

Under static single-watermark scheme, for $\mathtt{K} = 1$, and $\|\mathbf{w}\| = 1$, the actual baseline is 0.81056 and the average distortion is 0.042043. Note that the actual baseline is much lower than the theoretically estimated value of baseline under the assumption that the host coefficients are from a Normal distribution of zero mean and unit variance. This is due to the correlation between the images in the database. Under a traditional spread spectrum scheme, for $\mathtt{K} = 1$, and $\|\mathbf{w}\| = 1$, the actual baseline is 0.00011 and the average distortion is 0.9997. Compared to the static single-watermark setting, significant improvement in distortion is achieved. Intuitively, the significant improvement observed is due to strong coherence amongst the images in the database. Figure 5.16 depicts the watermark for the traditional SS scheme and the watermark for the static single-watermark scheme.

Experiments were also conducted to evaluate the performance of static and dynamic multiple watermark schemes for a natural image database of 1000 images. For implementing the static multiple watermark scheme, a 2-mean algorithm was used to partition the image database. A compact representation of the database with $\mathcal{W} = 1048576$ bits is organized as two watermarks of size $256 \times 256 \times 8$ each, which are the normalized

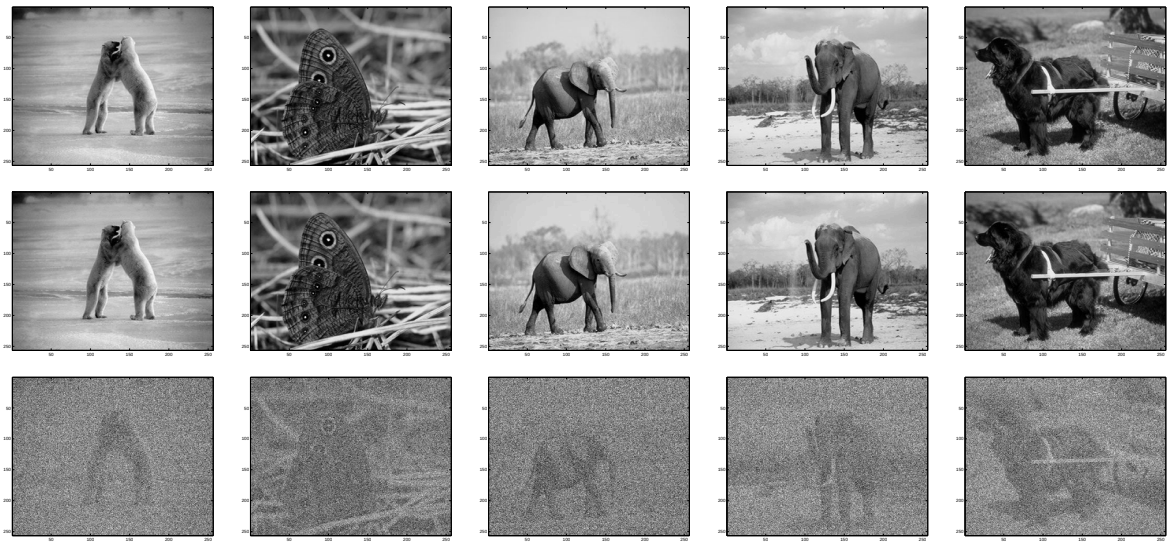
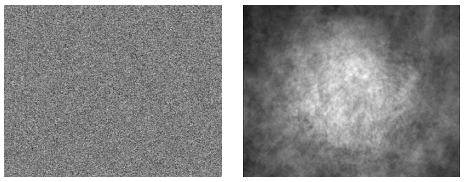


Figure 5.15: Comparison of distortion due to Spread Spectrum and Static Schemes on the actual database of 700 images. Row 1: Original Images Row 2: Images watermarked using static scheme with watermark generated using prior knowledge of image database. Row 3: Images watermarked using spread spectrum scheme with watermark generated randomly. For fair comparison the false alarm and robustness have been kept fixed for either watermarking methods. This is done by fixing the detection strength $\mathtt{K} = 1$ and threshold $T = 0.5$. The average correlation value, $\mathcal{B}$, for the spread spectrum scheme is 0.00011 and the average correlation value, $\mathcal{B}$, for the static scheme is 0.81056.

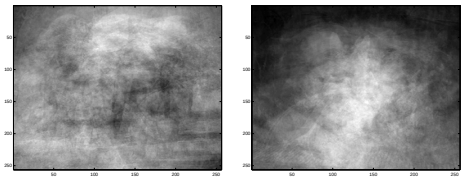
average of the two partitions. The two watermarks are depicted in Figure 5.17. The watermarks in Figure 5.17 are complementary in the sense that they point towards opposite directions if treated as vectors in the 256 by 256 dimensional space. This is characteristic of the 2-mean algorithm that was used to partition the database. Figure 5.18 depicts the evolving of 2 watermarks for a 2-watermark dynamic scheme. The watermarks in the dynamic 2-watermark scheme are updated as new images arrive. Note that, here also the watermarks are complementary because of the 2-mean algorithm used. The watermarks generated by the dynamic scheme are backward compatible.

Figure 5.19 illustrates the performance comparison of spread spectrum, static and dynamic settings for a natural image database of 1000 images. The conclusions are similar to a Gaussian database case except that the difference in distortion between a spread spectrum scheme and the proposed schemes is significantly higher, even more



(a) (b)

Figure 5.16: Watermarks for traditional SS scheme and static single-watermark scheme, for a natural database of 1000 images.



(a) (b)

Figure 5.17: Keys for the 2-watermark static setting, for the image database. Note that the 2 watermarks are almost complementary to each other.

than the theoretical findings. This is probably because of the strong coherence amongst the images in the database. It is interesting to note that the increase in distortion with increasing size of database is very slow. Figure 5.20 gives the performance of watermarking in terms of distortion with changing size of database under a static multiple watermark setting for images from a natural image database. Note that the distortion is very low and follows the trends as in the Gaussian case. Figure 5.21 illustrates the change in distortion with increase in size of database under a dynamic multiple watermark setting. Note that in this case also the difference in distortion from a spread spectrum scheme is significant.

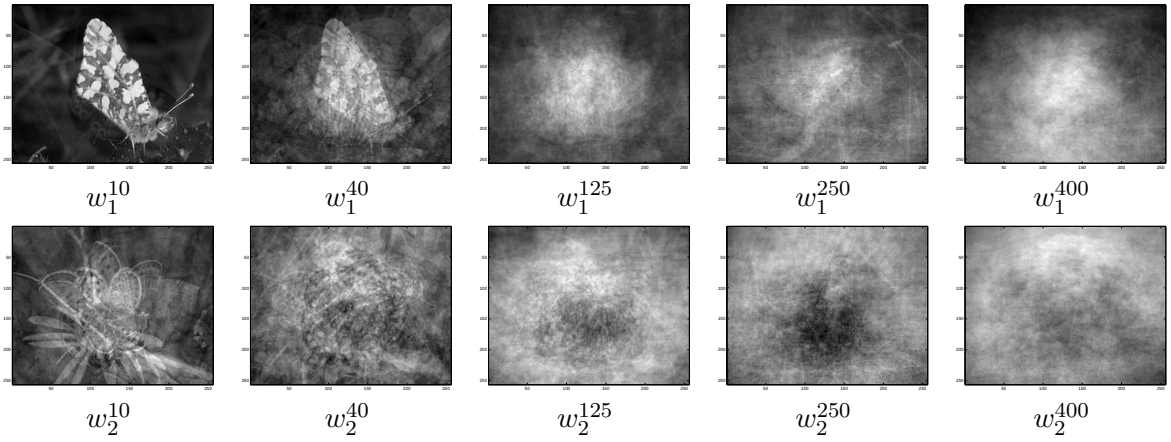


Figure 5.18: Evolving of watermarks in the dynamic 2-watermark case, for the image database. from left to right: First row depicts the evolving of the first watermark for the 10^{th}, 40^{th}, 125^{th}, 250^{th} and 400^{th} image. The second row depicts evolving of the corresponding second watermark. Note the complimentary nature of the watermarks which is indicative of the 2-mean algorithm used to divide the database. (Final size of database: 700 images)

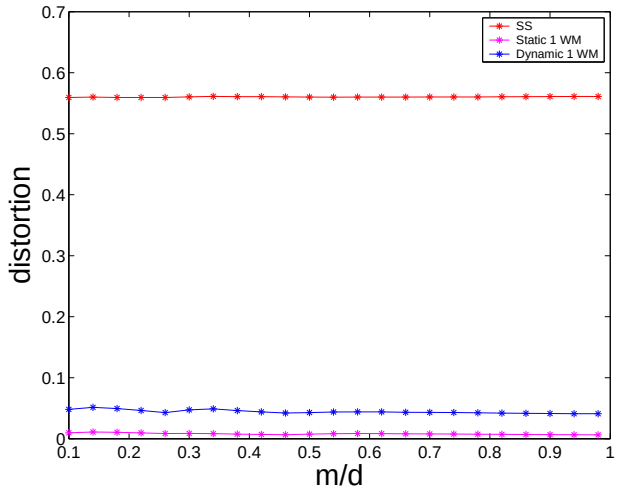


Figure 5.19: Distortion verses (m/d) for natural images from an image database for dynamic single watermark setting with fixed bandwidth of $256 \times 256 \times 8$ bits. The number of coefficients is fixed at $d = 256 \times 256$, and $\mathtt{K} = 1$. The above graph shows a comparison of performance between a spread spectrum, static and dynamic setting. Note that although the dynamic setting performs worse than a static setting it still improves upon a spread spectrum setting with random watermark.

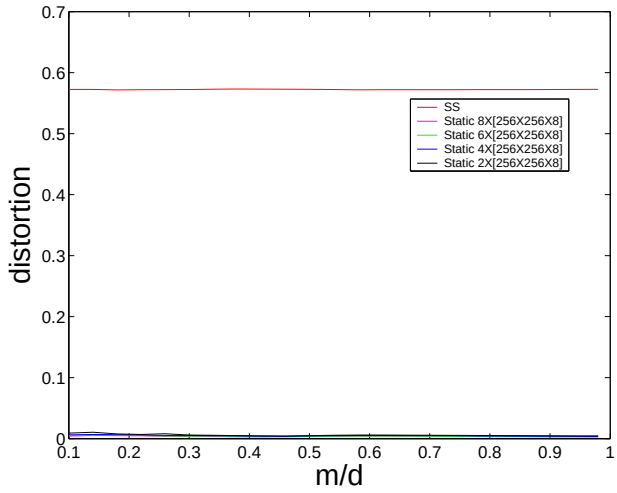


Figure 5.20: Distortion verses (m/d) for natural images from an image database for static multiple watermark setting with fixed bandwidth of $256 \times 256 \times 8$ bits. The number of coefficients is fixed at $d = 256 \times 256$, and $\mathtt{K} = 1$. As the number of watermarks increases, the distortion decreases. The distortion is very low.

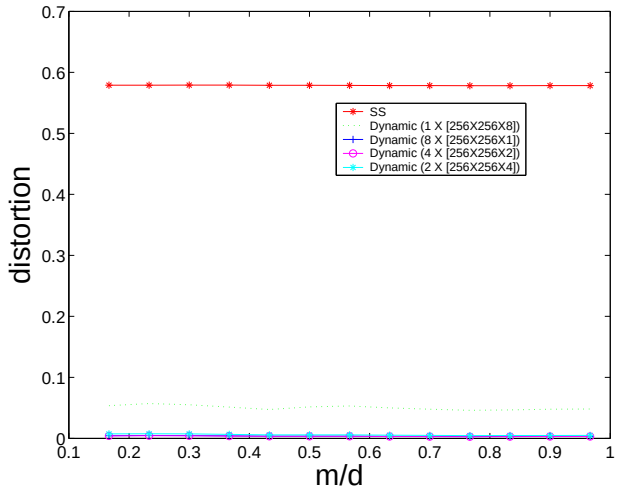


Figure 5.21: Distortion verses (m/d) for images from a natural image database for dynamic multiple watermark setting with fixed bandwidth. The number of coefficients is fixed at $d = 256 \times 256$, and $\mathtt{K} = 1$. Note that the bandwidth of the side channel is fixed at 524288 bits.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This dissertation highlights the efficacy of using *knowledge of the hosts database* in improving performance of watermarking based applications and presents a generic approach (Figure 1.2) that demonstrates this. In the proposed approach the encoder has access to the hosts database and the detector has access to either full or partial information about the database. This dissertation describes how the proposed approach can be effective in addressing the problems associated with applications that adopt traditional watermarking based solutions. In this work three applications are considered. All of them deal with a database of hosts, which can be either static or on-line.

In chapter 3 a unified framework, which is a combination of retrieval and watermarking systems, is proposed to reduce the ambiguity problem in copy detection systems. In this scenario the detector has access to the modified hosts database. It is shown that by increasing mutual separation of images in feature space ambiguity can be resolved. The separation problem is expressed as a solution to a non-convex optimization problem and a linear approximate algorithm to solve it is given. A prototype system, RAM is presented that shows the efficacy of this framework over state-of-the-art copy detection systems that use SIFT features, in resolving ambiguity. The intent is not to improve

upon a feature representation but to show the efficacy of the proposed framework on any modifiable feature representation. It is shown that the proposed system is arguably more secure than a simple watermarking based solution.

In chapter 4 another unified framework (combining retrieval and watermarking systems) for improving nearest neighborhood search speed in high dimensions in proposed. In this scenario the detector has access to the modified hosts database. The proposed framework is implemented using a novel method called *active clustering*, where the encoder takes as input the original database to generate a modified database and an index tree, that can be used to search in the database. The method is applied on images whose feature representation is taken as its spatial representation down-sampled to 4096 dimensions. Using the index tree the search complexity is logarithmic. A theoretical analysis of how the combination system improves upon both retrieval and watermarking systems is described.

In chapter 5 a framework for database watermarking is proposed. In this scenario the detector has access to a partial information of the hosts database, made available to it through a secure channel between the encoder and the detector. The encoder uses knowledge of the original hosts database to generate the modified database and a partial description of the database. The size of the partial description is influenced by bandwidth limitations of the channel. The proposed framework demonstrates the efficacy of using knowledge of hosts database in improving watermarking performance measures over a traditional approach that uses i.i.d sequences as watermarks. An online setting where the database is not fixed and new hosts are incrementally added into the database is also investigated.

In all the above three applications, the conflicting requirements of robustness and distortion are efficiently traded off. One common feature of all the three works is the use of adaptive modification. For this, note Eq. 5.1, Eq. 3.1 and 3.2, and Eq. 4.2. The natural separation of the hosts in their feature space or the knowledge of the detection strength is used to reduce average distortion and simultaneously improve robustness.

The proposed approach also shows how the watermarking of individual hosts can be made interdependent when a database of hosts is available at the encoder. The use of partial and full information at the detector, as proposed in this dissertation, is a more practical approach towards addressing the problems associated with applications that adopt a watermarking based solution. Note that the notion of detection by searching (a requirement in retrieval system) is inspired by the possibility of taking advantage of the natural separation of hosts in feature space.

6.2 Future Work

This dissertation proposes a novel watermarking approach that has opened up several directions of research. Availability of a database of hosts is a common occurrence in most practical applications. Hence one obvious direction of research is to show the efficacy of using knowledge of hosts database in other applications.

For the framework proposed in chapter 4, the database remains unchanged (*static*) throughout the encoding and query processing stages. It will be interesting to study the *dynamic* setting. In this setting, the database $\mathcal{I}$ starts from one host $\mathbf{I}_1$. New hosts can be added into $\mathcal{I}$, but once added, cannot be removed. Furthermore, the corresponding watermarked host must be computed before the arrival of new hosts. The watermarked host, once computed, can not be modified. The dynamic setting is motivated by applications where a stream of images are to be watermarked by a watermarking service provider before releasing to the public domain. The watermarking service provider does not know in advance the images to be watermarked, and the watermarked images, once released to the public domain, can not be recalled for modification. This is also applicable for the solution to resolving ambiguity presented in chapter 3. In a dynamic database, images can both be added and deleted from the database, so it will be interesting to investigate techniques to tackle the optimization problem in such a setting.

Another possible research direction in relation to the application in chapter 4 is to

study how the size of the index tree affects watermarking performance. Most watermarking formulations (for example, watermarking with side information) assume that the encoder and decoder know the distribution of the hosts, but not the actual hosts. In the proposed formulation, the encoder and decoder have access to the index tree and thus have full information of the actual database. In applications where the decoding is to be performed in the client-side, the index tree has to be sent over the network. This is practical only if the description of the database is small. Thus, it is useful to know how to obtain a compact description of the database, and how to tradeoff its size with other watermarking performance measures.

The framework proposed in chapter 5 opens up several research questions. The facility of communication between the detection and encoding unit in detecting the watermark behooves investigating what information and how much information must be actually embedded in the host, so that over a few interactions the detection results can be ascertained. This reduces the distortion further but adds to the communication cost. Thus a tradeoff needs to be found. Another research direction would be to investigate methods for generating the watermarks. Presently a simple k-mean or averaging technique is used. For large values of m, it may be required to organize the partial information appropriately to perform detection by searching.

Bibliography

[1] http://www.cs.washington.edu/research/imagedatabase.

[2] `http://www.digimarc.com/`.

[3] A. Adelsbach and A. Sadeghi. Zero-knowledge watermark detection. *4th Int. Workshop on Info. Hiding*, LNCS 2137:273–288, 2001.

[4] A.Kalker, J.Haitsma, and J. Oostveen. Issues with digital watermarking and perceptual hashing. In *SPIE Conf. on Multimedia Sys. & Appl.*, August 2001.

[5] A.M. Alattar. Briding printed media and the internet via digimarc's watermarking technology. *Multimedia and Security Workshop, ACM Multimedia*, 2000.

[6] A.W.M.Smeulders, M.Worring, S.Santini, A.Gupta, , and R.Jain. Content based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (12):1349–1380, 2000.

[7] M. Barni, F. Bartolini, V. Cappellini, and F. Rigacci. A m.a.p identification criterion for dct-based watermarking. In *European Signal Processing Conference (EUSIPCO)*, 1998.

[8] M. Barni, F. Bartolini, A. De Rosa, and A. Piva. A new decoder for the optimal recovery of nonadditive watermarks. *IEEE Transactions on Image Processing*, 10(5):755–766, 2001.

[9] W. Bender, D. Gruhl, N. Morimoto, and A. Lu. Techniques for data hiding. *IBM System Journal*, 35(3/4):313–336, 1996.

[10] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, September 1975.

[11] S. Berrani, L. Amsaleg, and P. Gros. Robust content based image searches for copyright protection. In *Proceedings of ACM Workshop on Multimedia Databases*, pages 70–77, November, 2003.

[12] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[13] C. Burges. A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining*, 2(2):121–167, 1998.

[14] J. Cannons and P. Moulin. Design and statistical analysis of a hash-aided image watermarking system. *IEEE Transactions on Image Processing*, October, 2004.

[15] M. L. Cascia and E. Ardizzone. JACOB: Just a content-based query system for video databases. In *IEEE Int. Conference on Acoustics, Speech and Signal Processing*, May, 1996.

[16] F. Cayre, C. Fontaine, and T. Furon. Watermarking security: Theory and practice. *IEEE Transactions on Signal Processing*, 53(10):3976–3987, 2005. special issue "Supplement on Secure Media III".

[17] E. Chang, J. Wang, C. Li, and G. Wiederhold. RIME: A replicated image detector for the world-wide web. In *SPIE Vol. 3527*, pages 68-67, 1998.

[18] B. Chen. *Design and Analysis of Digital Watermarking, Information Embedding, and Data Hiding Systems*. PhD thesis, MIT, USA, June 2000.

[19] B. Chen and G. Wornell. An information-theoretic approach to the design of robust digital watermarking systems. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 4:2061–2064, 1999.

[20] B. Chen and G.W. Wornell. Achievable performance of digital watermarking systems. *IEEE Int. Conf. on Multimedia Computing & Systems*, 1:13–18, 1999.

[21] V. Cherkassky and F. Mulier. *Learning from Data: Concepts, Theory, and Methods*. Wiley, New York, 1998.

[22] J. Chou, S.S. Pradhan, and K. Ramchandran. On the duality between distributed source coding and data hiding. *33rd Asilomar conference on Signals, System and Computers*, pages 1503–1507, 1999.

[23] Pedro Comesana, Luis Perez-Freire, and Fernando Perez-Gonzalez. The return of the sensitivity attack. In T. Kalker In M. Barni, I. Cox and editors H. J. Kim, editors, *Digital Watermarking: 4th International Workshop, IWDW 2005*, pages 260–274. volume 3710 of Lecture Notes in Computer Science, Springer-Verlag Heidelberg, September, 2005.

[24] J.H. Conway and N.J.A. Sloane. *Sphere Packings, Lattices and Groups*. Springer, 1998.

[25] Corel. `http://www.corel.com/`.

[26] M. Costa. Writing on dirty paper. *IEEE Trans. on Info. Theory*, 29(3):439–441, 1983.

[27] I. Cox, J. Killian, T. Leighton, and T. Shamoon. A secure robust watermark for multimedia. *Proc. of First International Workshop on Information Hiding*, pages 185–206, 1996.

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

[29] I. Cox and M. Miller. A review of watermarking and the importance of perceptual modeling. *Proceedings of the SPIE, Human Vision and Electronic Imaging II*, 3016:92–99, 1997.

[30] I. Cox, M.L. Miller, and J.A. Bloom. *Digital Watermarking*. Morgan Kaufmann, 2002.

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

[32] S. Craver. Zero knowledge watermark detection. *Third Internation Workshop on Information Hiding, LNCS*, 1768, 2000.

[33] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines (and other kernel-based learning methods)*. Cambridge University Press, 2000.

[34] S. Deerwester, S. Dumais, T. Landauer, G. Furnas, and R. Harshman. Indexing by latent semantic indexing. *J. Soc. Info. Sci.*, 41:391–407, 1990.

[35] G. Depovere, T. Kalker, and J.-P. Linnartz. Improved watermark detection reliability using filtering before correlation. In *ICIP*, 1998.

[36] S. Derrode and F. Ghorbel. Robust and efficient fourier-mellin transform approximations for gray-level image reconstruction and complete invariant description. *CVIU, Vol. 83(1)*, July 2001.

[37] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley-Interscience, 2001.

[38] J. Eggers, R. Bauml, R. Tzschoppe, and B. Girod. Scalar costa scheme for information embedding. *IEEE Transactions on Signal Processing (Special Issue on Signal Processing for Data Hiding in Digital Media and Secure Content Delivery*, 2002.

[39] J. Eggers, J. K. Su, and B. Girod. Public key watermarking by eigenvectors of linear transforms. In *Proc. of European Signal processing Conference*, April, 2000.

[40] M. Flickner et. al. Query by image and video content: the QBIC system. In *IEEE Computer*, pages 23–32, September, 1995.

[41] J.H. Friedman, J.L. Bentley, and R.A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. on Math Software (TOMS)*, (3):209–226, 1977.

[42] T. Furon and P. Duhamel. An asymmetric public key watermarking technique. *Third International Workshop on Information Hiding, LNCS 1768*, pages 88–100, 2000.

[43] T. Furon and P. Duhamel. Copy protection of distributed contents: An application of watermarking technique. *Workshop COST 254: Friendly Exchange through the net*, 2000.

[44] T. Furon, N. Moreau, and P. Duhamel. Audio asymmetric watermarking technique. In *Proc. of IEEE ICASSP*, June.

[45] S. I. Gel'fand and M. S. Pinsker. Conding for channel with random parameters. *Problems of Control and Information Theory*, 9(1), 1980.

[46] F. Ghorbel. A complete invariant description for gray level images by the harmonic analysis approach. *Pattern recognition letters, vol 15, pp 1043-1051*, October 1994.

[47] K. Gopalakrishnan, N. Memon, and P. Vora. Protocols for watermark verification. *Multimedia and Security Workshop at ACM Multimedia*, 1999.

[48] R. M. Gray and T. G. Stockman. Dithered quantizers. *IEEE Trans. on Information Theory*, 39(3), 1993.

[49] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 47–57, June 1984.

[50] J.A. Haitsma, J.C. Oostveen, and A. Kalker. Robust audio hashing for content identification. In *Content based multimedia Indexing (CBMI), Brescia Italy*, 2001.

[51] A. Hampapur and R. Bolle. Comparison of distance measures for video copy detection. In *IEEE ICME*, 2001.

[52] F. Hartung and B. Girod. Fast public-key watermarking of compressed video. In *IEEE International Conference on Image Processing*, volume 1, pages 528–531, 1997.

[53] G. R. Hjaltason and H. Samet. Ranking in spatial database. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 47–57, 1984.

[54] H. Hotelling. Analysis of a complex of statistical variables into principle components. *J. Educational Psychology*, 27:417–441, 1933.

[55] F. Idris and S. Panchanathan. Review of image and video indexing techniques. *Journal on Visual Communication and Image representation*, (2):146–166, 1997.

[56] M. S. Kankanhalli, Rajmohan, and K. R. Ramakrishnan. Content-based watermarking of images. In *Proc. ACM Multimedia*, September.

[57] Y. Ke and R. Suthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2004.

[58] Y. Ke, R. Suthankar, and L. Huston. Efficient near duplicate detection and sub image retrieval. In *ACM Multimedia*, October, 2004.

[59] H. Kinoshita. An image digital signature system with zkip for the graph isomorphism. In *IEEE International Conference on Image Processing*, volume 3, pages 247–250, 1996.

[60] J. M. Kleinberg. Two algorithms for nearest-neighbor search in high dimensions. In *Proc. of ACM STOC*, pages 599–608, 1997.

[61] E. Koch and J. Zhao. Towards robust and hidden image copyright labeling. In *IEEE Worshop on Nonlinear Signal and Image Processing*, 1995.

[62] D. Kundur and D. Hatzinalos. Digital watermarking using multiresolution wavelet decomposition. In *ICASSP*, volume 5, pages 2969–2972, 1998.

[63] R. Kurniawati, J.S. Jin, and J.A. Shepherd. An efficient nearest-neighbour search while varying euclidean metrics. *ACM Multimedia*, pages 411–418, 1998.

[64] M. Kutter. Watermarking resisting to translation, rotation, and scaling. In *Proc. of SPIE: Multimedia systems and applications*, volume 3528, pages 423–431, 1998.

[65] M. Kutter, F. Jordan, and F. Bossen. Digital signature of color images using amplitude modulation. *Journal of Elecronic Imaging*, 7(2):326–332, 1998.

[66] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[67] H. S. Malvar and A. F. Florêncio. Improved spread spectrum: A new modulation technique for robust watermarking. *IEEE Trans. on Signal Processing*, 51(4):898–905, April 2003.

[68] Y. Meng, E. Chang, and B. Li. Enhancing dpf for near-replica image recognition. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 416–423, 2003.

[69] P. Mitra, C. A. Murthy, and S. K. Pal. Data condensation in large databases by incremental learning with support vector machines. In *ICPR*, 2000.

[70] E. Mortensen, H. Deng, and L. Shapiro. A SIFT descriptor with global context. In *CVPR*, 2005.

[71] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (coil-100). In *Technical Report CUCS-006-96, February*, 1996.

[72] A. Piva, M. Barni, E. Bartoloni, and V. Capellini. Dct-based watermarking recovering without resorting to the uncorrupted original image. In *IEEE International Conference on Image Processing*, volume 1, page 520, 1997.

[73] J. G. Proakis. *Digital Communications*. McGraw-Hill, 1989.

[74] M. H. Jakubowski R. Venkatesan, S.-M. Koon and P. Moulin. Robust image hashing. In *Int. Conf. Image Processing, Vancouver, Canada.*, September, 2000.

[75] M. Ramkumar. *Data Hiding in Miltimedia: Theory and Applications*. PhD thesis, New Jersey Institute of Technology, Keany, NJ, USA, November 1999.

[76] S. Roy and E-C. Chang. Watermarking with knowledge of image database. In *Proc. of IEEE ICIP*, pages 471–474 ,year=.

[77] S. Roy and E-C. Chang. Watermarking with retrieval systems. *ACM Multimedia Systems Journal*, 9(5):433–440, 2004.

[78] S. Roy and E-C. Chang. A unified framework for resolving ambiguity in copy detection. In *Proc. of ACM Multimedia*, 2005.

[79] H. Schweitzer. Organizing image databases as visual content search trees. *Image and Vision Computing, Special Issue on Content-Based Image Indexing and Retrieval*, (7):531–540, 1999.

[80] C. E. Shannon. Channels with side information at the transmitter. *IBM Journal of Research and Development*, 2:289–293, October 1958.

[81] J. R. Smith and S. F. Chang. VisualSEEK: a fully automated content-based image query system. In *ACM Multimedia Conference*, November, 1996.

[82] J. R. Smith and B. O. Cominskey. Modulation and information hiding in images. *Proc. of First International Workshop on Information Hiding*, pages 207–226, 1996.

[83] T. Strijk and A. Wolff. Labeling points with circles. *International Journal of Computational Geometry and Applications*, 11(2):181–195, 2001.

[84] J. F. Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. In *Optimization Methods and Software, Special issue on Interior Point Methods 11-12*, pages 625–653, 1999.

[85] M. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–13, 1991.

[86] M. Swanson, B. Zhu, and A Tewfik. Transperant robust image watermarking. In *IEEE International Conference on Image Processing*, volume 3, pages 211–214, 1996.

[87] M. Swanson, B. Zhu, and A Tewfik. Multiresolution video watermarking using perceptual models and scene segmentation. In *IEEE International Conference on Image Processing*, volume 2, pages 558–561, 1997.

[88] D. L. Swets and J. Weng. Hierarchical discriminant analysis for image retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, (5):386–401, 1999.

[89] A. Z. Tirkel, G. A. Rankin, R. G. van Schyndel, W. J. Ho, N. R. A. Mee, and Charles F. Osborne. Electronic watermark. In *Proc. of Digital Image Computing Technology and Applications*, pages 666–672, 1993.

[90] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne. A digital watermark. In *IEEE International Conference on Image Processing*, volume 13-16, pages 89–90, 1994.

[91] R. G. van Schyndel, A. Z. Tirkel, and I. D. Svalbe. Key independent watermark detection. In *Proc. of IEEE Int. Conf. on Multimedia Compuing and Systems*, volume 1, 1999.

[92] S. Voloshynovskiy, M.Koval, F. P. Gonzalez, M. Mihchak, J. Vila-Forcen, and T. Pun. Data-hiding with partially available side information. In *Proc. of EUSIPCO*, 2005.

[93] G. Voyatzis and I. Pitas. The use of watermarks in the protection of digital multimedia. *Proceedings of the IEEE*, 87:1197–1207, 1999.

[94] D. A. White and Ramesh Jain. Similarity indexing with the ss-tree. In *Proc. 12th IEEE International Conference on Data Engineering*, Feburary 1996.

[95] R. B. Wolfgang, C. I. Podilchuk, and E. J. Delp. Perceptual watermarks for digital images and video. *Proceedings of the IEEE*, 87(7):1108–1126, 1999.

[96] A. Yoshitaka and T. Ichikawa. A survey on content based retreival for multimedia databases. *IEEE Trans. on Knowledge and Data Engineering*, (1):81–93, 1999.