DEVELOPMENT AND IMPLEMENTATION OF TEXT WATERMARKING AND INFORMATION HIDING TECHNIQUE

BY
GC HAMMAD JAVED (GP LDR)
GC KHAWIR MAHMOOD
CAPT AMEER AHMAD
PC SOHAIL NAWAZ

Submitted to the Faculty of Computer Science Department,
Military College of Signals, National University of Sciences and Technology,
Rawalpindi in partial fulfillment for the requirements of a B.E. Degree in Computer Software Engineering

MAY, 2005

ABSTRACT

The digital representation of text, audio, image and video documents has become very popular in the last decade. The success of digital technology is largely due to the capabilities of efficient transmission, storage, and perfect copying. However, especially the last feature leads to severe problems because unauthorized copying is also simplified. One approach to combat this problem is to mark a digital document such that a copyright can be proven or the distribution path be traced.

"Digital Watermarking" means embedding information into multimedia data that should be imperceptible but irremovable. Thus, in contrast to copyright information included in the header of the data streams, embedded watermarks remain in the document even after format conversions or D/A and A/D conversions.There are numerous applications of digital watermarking such as copyright protection, copy protection, content authentication, transaction tracking and broadcast monitoring.

Information hiding or steganography is related to watermarking but watermarking has the additional notion of being robust against attacks and having concern for not only the watermark but also the cover.

In this project a watermarking algorithm has been developed and implemented which slightly modifies inter-word spaces so that different lines across a text act as sampling points of a sine wave. The method not only embeds a watermark but also hides information through the watermark thus providing a robust and imperceptible way for steganography through text images. Preliminary experiments have shown promising results.

Dedicated to our Parents and our Instructors

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of award of qualification either at this institute or elsewhere.

ACKNOWLEDGMENTS

All praise is to ALMIGHTY ALLAH who enabled us to accomplish this great challenge fruitfully. Thanks to our parents for their prayers and support. Special thanks to Our Project Supervisor Lec Aihab for his cooperation and guidance in every field. We are also thankful to all those who have directly or indirectly provided their kind assistance in achieving this task.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

WATERMARKING OVERVIEW

1.1 Introduction

Multimedia watermarking technology has evolved very quickly during the last few years. A digital watermark is information that is imperceptibly and robustly embedded in the host data such that it cannot be removed. A watermark typically contains information about the origin, status, or recipient of the host data. Applications include copyright protection, data monitoring, and data tracking.

Multimedia production and distribution, as is seen it today, is all digital, from the authoring tools of content providers to the receivers. The advantages of digital processing and distribution, like noise-free transmission, software instead of hardware processing and improved re-configurability of systems, are all well known and obvious. Not so obvious are the disadvantages of digital media distribution. For example, from the viewpoint of media producers and content providers, the possibility for unlimited copying of digital data without loss of fidelity is undesirable because it may cause considerable financial loss. Digital copy protection or copy prevention mechanisms are only of limited value because access to clear text versions of protected data must at least be granted to paying recipients which can then produce and distribute illegal copies. Technical attempts to prevent copying have in reality always been circumvented.

One remaining method for the protection of intellectual property rights (IPR) is the embedding of digital water-marks into multimedia data. The watermark is a digital code irremovably, robustly, and imperceptibly embedded in the host data and typically contains information about origin, status, and/or destination of the data. Although not directly used for copy protection, it can at least help identifying source and destination of multimedia data and, as a “last line of defense,” enable appropriate follow-up actions in case of suspected copyright violations.

While copyright protection is the most prominent application of watermarking techniques, others exist, including data authentication by means of fragile watermarks which are impaired or destroyed by manipulations, embedded transmission of value added services within multimedia data, and embedded data labeling for other purposes than copyright protection, such as data monitoring and tracking. An example for a data-monitoring system is the automatic registration and monitoring of broadcasted radio programs such that royalties are automatically paid to the IPR owners of the broadcast data.

The development of watermarking methods involves several design tradeoffs [1]. Watermarks should be robust against standard data manipulations, including digital-to-analog conversion and digital format conversion. Security is a special concern, and watermarks should resist even attempted attacks by knowledgeable individuals. On the other hand, watermarks should be imperceptible and convey as much information as possible. In general, watermark embedding and retrieval should have low complexity because for various applications, real-time watermarking is desirable.

1.2 Information Hiding

Info Hiding is a general term encompassing many sub disciplines [2]. Two important sub disciplines are: steganography & watermarking as shown in Figure 1-1.

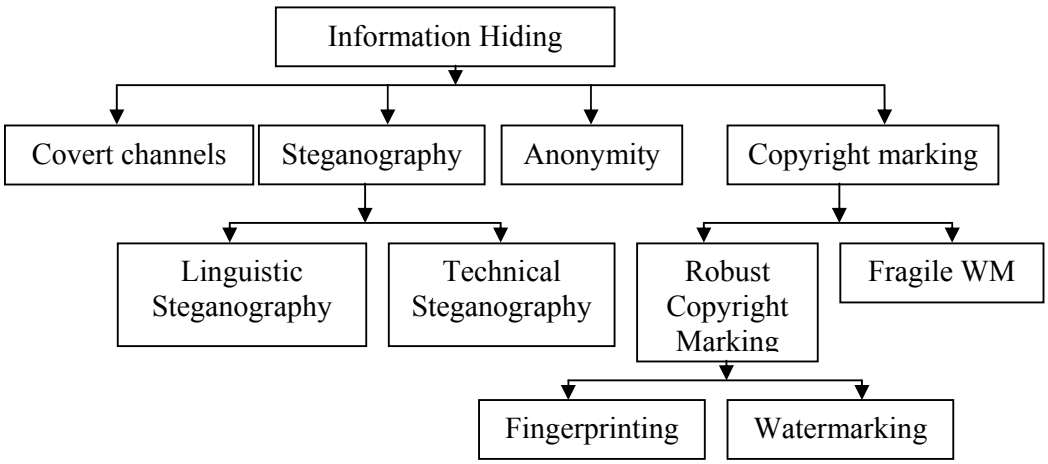

Figure 1-1 The Information Hiding Hierarchy

Steganography is hiding i.e. keeping the existence of the information secret whereas watermarking is making the information imperceptible**.** Information hiding is different than cryptography (cryptography is about protecting the content of messages).

1.3 Steganography

1.3.1 Definition and History

The word "steganography," which is still in use today, derives from the Greek derived from the Greek words steganos which means "covered" and graphia which means "writing", i.e. covered writing.

*Steganography s*tands for techniques in general that allow secret communication, usually by embedding or hiding the secret information in other, unsuspected data. Steganographic methods generally do rely on the assumption that the existence of the covert communication is unknown to third parties and are mainly used in secret point-to-point communication between trusting parties. As a result, steganographic methods are in general not robust, i.e., the hidden information cannot be recovered after data manipulation.

It is the art of concealed communication; the very existence of a message is secret. Examples of old steganography techniques are writing on shaved heads, invisible ink, microscopic images.

1.4 Watermarking

Watermarking: is the practice of imperceptibly altering a cover to embed a message about that cover. Watermarking as opposed to steganography, has the additional notion of robustness against attacks [2]. Even if the existence of the hidden information is known it is difficult—ideally impossible—for an attacker to destroy the embedded watermark, even if the algorithmic principle of the watermarking method is public. In cryptography, this is known as *Kerkhoffs law* [3]*:* a cryptosystem should be secure, even if an attacker knows the cryptographic principles and methods used but

does not have the appropriate key. A practical implication of the robustness requirement is that watermarking methods can typically embed much less information into host data than steganographic methods. Watermarks are inseparable from the cover in which they are embedded. *Unlike cryptography, watermarks can protect content even after they are decoded.*

1.4.1 History

More than 700 years ago [2], watermarks were used in Italy to indicate the paper brand and the mill that produced it. By the 18th century watermarks began to be used as anti-counterfeiting measures on money and other documents. The term watermark was introduced near the end of the 18th century. It was probably given because the marks resemble the effects of water on paper. The first example of a technology similar to digital watermarking is a patent filed in 1954 by Emil Hembrooke for identifying music works. In 1988, Komatsu and Tominaga appear to be the first to use the term "digital watermarking". About 1995, interest in digital watermarking began to mushroom

The analogy between paper watermarks, steganography, and digital watermarking is obvious, and in fact, paper watermarks in money bills or stamps actually inspired the first use of the term watermarking in the context of digital data. Digital watermarking has gained a lot of attention and has evolved very quickly, and while there are a lot of topics open for further research, practical working methods and systems have been developed.

1.4.2 Terminology

Today, digital communication is of concern. As in classical analog communication, also in digital communication there is interest for methods that allow the transmission of information hidden or embedded in other data. While such techniques often share similar principles and basic ideas, there are also important distinguishing features, mainly in terms of robustness against attacks. Several names have been coined for such techniques. However, the terms are often confused, and therefore it is necessary to clarify the differences [1].

1.4.2.1 Data hiding and data embedding

These methods are used in varying contexts, but they do typically denote either steganography or applications "between" steganography and watermarking, which means applications where the existence of the embedded data are publicly known, but there is no need to protect it. This is typically the case for the embedded transmission of auxiliary information or services that are publicly available and do not relate to copyright protection or conditional access functionalities.

1.4.2.2 Fingerprinting and labeling

Fingerprinting and labeling are terms that denote special applications of watermarking. They relate to copyright protection applications where information about originator and recipient of digital data is embedded as watermarks. The individual watermarks, which are unique codes out of a series of codes, are called "fingerprints" or "labels."

1.4.2.3 Bit-stream Watermarking

This method is sometimes used for data hiding or watermarking of compressed data, for example, compressed video.

1.4.2.4 Embedded Signatures

The term *embedded signatures* has been used instead of "watermarking" in early publications. Because it potentially leads to confusion with cryptographic digital signatures, it is usually not used anymore. Cryptographic signatures serve for authentication purposes. They are used to detect alterations of the signed data and to authenticate the sender. Watermarks, however, are only in special applications used for authentication and are usually designed to *resist* alterations and modifications.

1.4.2.5 Visible watermarks

*Visible watermark*s, as the name says, are visual patterns, like logos, which are inserted into or overlaid on images (or video), very similar to visible paper watermarks. However, the name is confusing since visible watermarks are not watermarks in the sense of this paper. Visible watermarks are mainly applied to images, for example, to visibly mark preview images available in image databases or

on the World Wide Web in order to prevent people from commercial use of such images.

1.5 Watermarking Applications

1.5.1 Copyright protection

This is the most prominent application. Information about the owner is embedded to prevent others from claiming copyright. It Require very high level of robustness.

1.5.2 Copy protection

Embed watermark to disallow unauthorized copying of the cover. For example, compliant DVD players will not playback or copy data that carry a “copy never” watermark.

1.5.3 Content Authentication

A watermark can be embedded to detect modifications to the cover. The watermark in this case has low robustness, “fragile”.

1.5.4 Transaction Tracking

A watermark can also be embedded to convey information about the legal recipient of the cover. This is useful to monitor or trace back illegally produced copies of the cover. This is usually referred to as “fingerprinting”.

1.5.5 Broadcast Monitoring

Embed a watermark in the cover and use automatic monitoring to verify whether cover was broadcasted as agreed

1.6 WM Requirements & Design Issues

The basic requirements in watermarking apply to all media and are very intuitive [3]. A watermark shall convey as much information as possible, which means the watermark data rate should be high. A watermark should in general be

secret and should only be accessible by authorized parties. This requirement is referred to as security of the watermark and is usually achieved by the use of cryptographic keys. A watermark should stay in the host data regardless of whatever happens to the host data, including all possible signal-processing that may occur, and including all hostile attacks that unauthorized parties may attempt. This requirement is referred to as robustness of the watermark. It is a key requirement for copy-right protection or conditional access applications, but less important for applications where the watermarks are not required being cryptographically secure, for example, for applications where watermarks convey public information. A watermark should, though being irremovable, be imperceptible.

Depending on the media to be watermarked and the application, this basic set of requirements may be supplemented by additional requirements. Watermark recovery may or may not be allowed to use the original, un-watermarked host data. Depending on the application, watermark embedding may be required in real time, e.g., for video fingerprinting. Real-time embedding again may, for complexity reasons, require compressed-domain embedding methods. Depending on the application, the watermark may be required to be able to convey arbitrary information. For other applications, only a few predefined watermarks may have to be embedded, and for the decoder it may be sufficient to check for the presence of one of the predefined watermarks (hypothesis testing).

1.6.1 Watermark Security and Keys

If security, i.e., secrecy of the embedded information, is required, one or several secret and cryptographically secure keys have to be used for the embedding and extraction process [4]. For example, in many schemes, pseudorandom signals are embedded as watermarks. In this case, the description and the seed of the pseudorandom number generator may be used as key. There are two levels of secrecy. In the first level, an unauthorized user can neither read nor decode an embedded watermark nor can he detect if a given set of data contains a watermark. The second level permits unauthorized users to detect if data are watermarked, however, the embedded information cannot be read without having the secret key. Such schemes can, for example, embed two watermarks, one with a public key and the other with a

secret key. Alternatively, schemes have been proposed which combine one or several public keys with a private key and embed one combined public/private watermark, rather than several watermarks. When designing an overall copyright protection system, issues like secret key generation, distribution, and management (possibly by trusted third parties), as well as other system integration aspects have to be considered.

1.6.2 Robustness

In the design of any watermarking scheme, watermark robustness is typically one of the main issues, since robustness against data distortions introduced through standard data processing and attacks is a major requirement. Standard data processing includes all data manipulation and modification that the data might undergo in the usual distribution chain, such as data editing, printing, enhancement, and format conversion. “Attack” denotes data manipulation with the purpose of impairing, destroying, or removing the embedded watermarks.

Although it is possible to design robust watermarking techniques, it should be noted that a watermark is only robust as long as it is not public, which means as long as it cannot be read by everyone. If watermark detector principle and key are public, and even if only a “black-box” watermark detector is public, the watermark is vulnerable to attacks. Hence, public watermarks, as sometimes proposed in the literature, are not robust unless every receiver uses a different key. This however is difficult in practice and gives rise to collusion attacks.

1.6.3 Imperceptibility

One of the main requirements for watermarking is the perceptual transparency. The data embedding process should not introduce any perceptible artifacts into the host data. On the other hand, for high robustness, it is desirable that the watermark amplitude is as high as possible. Thus, the design of a watermarking method always involves a tradeoff between imperceptibility and robustness. It would be optimal to embed a watermark just below the threshold of perception. However, this threshold is difficult to determine for real-world image, video and audio signals. Several measures to determine objectively perceived distortion and the threshold

of perception have been proposed for the mentioned media. However, most of them are still not perfect enough to replace human viewers or listeners who judge the visual or audio fidelity through blind tests. Thus, in the design of watermarking systems, it is usually necessary to do some testing with volunteers. The second problem occurs in combination with post watermarking processing, which might result in an amplification of the embedded watermark and make it perceptible. An example is zooming of watermarked images, which often makes the embedded watermarks visible, or contrast enhancement, which may amplify highly frequent watermark patterns that are otherwise invisible.

1.6.4 Watermark Recovery With / Without the Original Data

Watermark recovery is usually more robust if the original, un-watermarked data are available [5]. Further, availability of the original data set in the recovery process allows the detection and inversion of distortions which change the data geometry. This helps, for example, if a watermarked image has been rotated by an attacker. However, access to the original data is not possible in all cases, for example, in applications such as data monitoring or tracking. For other applications, like video watermarking, it may be impractical to use the original data because of the large data volume, even if it is available. It is, however, possible to design watermarking techniques that do not need the original for watermark extraction. Most watermarking techniques perform some kind of modulation in which the original data set is considered a distortion. If this distortion is known or can be modeled in the recovery process, explicitly designed techniques allow its suppression without knowledge of the original. In fact, most recent methods do not require the original for watermark recovery. Such techniques are sometimes called “blind” watermarking techniques.

1.6.5 Watermark Extraction / Verification

In the literature, two different types of watermarking systems can be found: systems that embed a specific information or pattern and check the existence of the (known) information later on in the watermark recovery, usually using some sort of hypothesis testing, and systems that embed arbitrary information

into the host data. The first type, verification of the presence of a known watermark, is sufficient for most copyright-protection applications.

The second type, embedding of arbitrary information, is, for example, useful for image tracking on the Internet with intelligent agents where it might not only be of interest to discover images, but also to classify them. In such cases, the embedded watermark can serve as an image identification number. Another example where arbitrary information has to be embedded are applications for video distribution where, e.g., the serial number of the receiver has to be embedded.

Although most presented methods or systems are designed for either watermark extraction or verification of presence for a given watermark, it should be noted that in fact both approaches are inherently equivalent. A scheme that allows watermark verification can be considered as a 1-bit watermark recovery scheme, which can easily be extended to any number of bits by embedding several consecutive "1-bit watermarks." The inverse is also true: a watermark recovery scheme can be considered as a watermark verification scheme assuming the embedded information is known.

1.7 Basic Watermarking Principles

The basic idea in watermarking is to add a watermark signal to the host data to be watermarked such that the watermark signal is unobtrusive and secure in the signal mixture but can partly or fully be recovered from the signal mixture later on if the correct cryptographically secure key needed for recovery is used.

To ensure imperceptibility of the modification caused by watermark embedding, a perceptibility criterion of some sort is used. This can be implicit or explicit, host data adaptive or fixed, but it is necessary. As a consequence of the required imperceptibility, the individual samples (e.g., pixels or transform coefficients) that are used for watermark embedding can only be modified by an amount relatively small to their average amplitude.

To ensure robustness despite the small allowed changes, the watermark information is usually redundantly distributed over many samples (e.g., pixels) of the host data, thus providing a "holographic" robustness, which means that the watermark can usually be recovered from a small fraction of the watermarked data, but the recovery is more robust if more of the watermarked data are available for recovery. Watermark systems do in general use one or more cryptographically secure keys to ensure security against manipulation and erasure of the watermark.

The first two issues, watermark signal design and water- mark signal embedding, are often regarded as one, specifically for methods where the embedded watermark is host signal adaptive.

Figures 1-2 and 1-3 illustrate the concept. Figure 1-2 shows the generic watermarking scheme for the embedding process. The input to the scheme is the watermark, the host data, and an optional public or secret key. The host data may, depending on the application, be uncompressed or compressed, however, most proposed methods work on uncompressed data. The watermark can be of any nature, such as a number, text, or an image. The secret or public key is used to enforce security. If the watermark is not to be read by unauthorized parties, a key can be used to protect the watermark. In combination with a secret or a public key, the watermarking techniques are usually referred to as secret and public watermarking techniques, respectively. The output of the watermarking scheme is the modified, i.e., watermarked data. The generic watermark recovery process is depicted in Figure. 1-3. Inputs to the scheme are the watermarked data, the secret or public key, and, depending on the method, the original data and the original watermark. The output of the watermark recovery process is either the recovered watermark or some kind of confidence measure indicating how likely it is for the given watermark at the input to be present in the data under inspection.

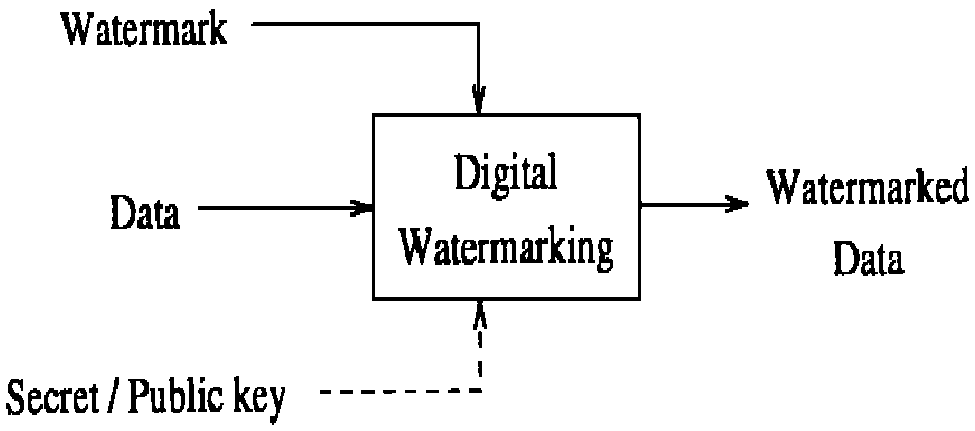

Figure 1-2 Generic Digital Watermarking Scheme

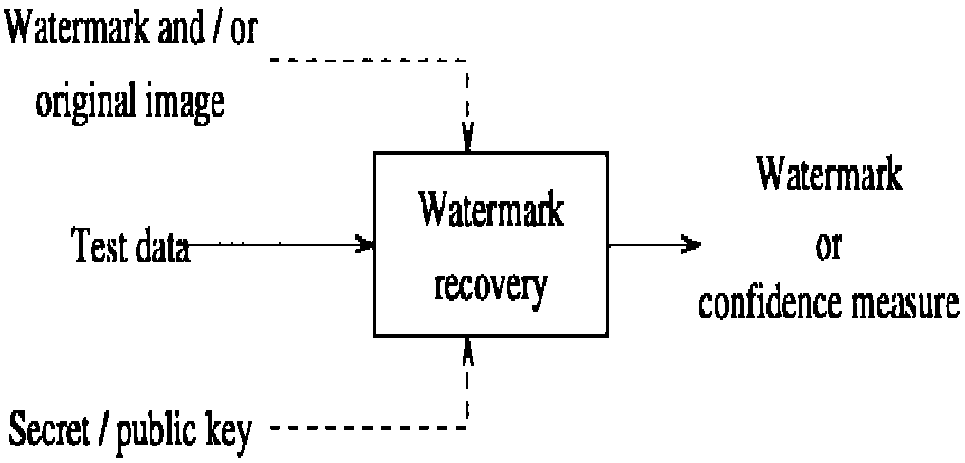

Figure 1-3 Generic Watermark Recovery Scheme

1.8 The Future of Digital Watermarking

The interest in watermarking technology is high, both from academia and industry. The interest from academia is reflected in the number of publications on watermarking and in the fact that conferences on watermarking and data hiding are being held. The interest from industry is evident in the number of companies in the field that have been founded within the past few years. Besides research activities in universities and industry, several international research projects

funded by the European Community have the goal to develop practical watermarking techniques.

TALISMAN [9] (ACTS project AC019, "Tracing Authors' rights by labeling image services and monitoring access network") aims to provide European Union service providers with a standard copyright mechanism to protect digital products against large scale commercial piracy and illegal copying. The expected output of TALISMAN is a system for protecting video sequences through labeling and watermarking.

OCTALIS [10] (ACTS project P119, "Offer of Content through Trusted Access Links") is the follow-up project of TALISMAN and OKAPI with the main goal of integrating a global approach to equitable conditional access and efficient copyright protection and to demonstrate its validity on large scale trials on the Internet and European Broadcasting Union (EBU) network.

International standardization consortia are also interested in watermarking techniques like ; the emerging video compression standard MPEG-4 (ISO/IEC 14 496), for example, provides a framework that allows the easy integration with encryption and watermarking, the DVD industry standard will contain copy control and copy protection mechanisms that use watermarking to signal the copy status of multimedia data, like "copy once" or "do not copy" flags.

Despite the many efforts that are underway to develop and establish watermarking technology, watermarking is still not a fully mature and understood technology, and a lot of questions are not answered yet. Also, the theoretical fundamentals are still weak, and most systems are designed heuristically.

Another drawback is that fair comparisons between watermarking systems are difficult. As long as methods and system implementations are not evaluated in a consistent manner using sophisticated benchmarking methods, the danger exists that weak and vulnerable systems and *de facto* standards are produced that result in spectacular failures and discredit the entire concept.

Thus, the expectations into watermarking should be realistic. It should always be kept in mind that every watermarking system involves a tradeoff between ***Robustness, Watermark data rate*** *(payload),* ***Imperceptibility***. Even when designed under realistic expectations, watermarks offer robustness against non-experts but may still be vulnerable to attacks by experts.

Although proof of ownership was the initial thrust for the technology, it seems that there is a long way to go before watermarking will be accepted as a proof in court, and it is likely enough that this may never happen. In copyright-related applications, watermarking must be combined with other mechanisms like encryption to offer reliable protection.

Still, there exist enough applications where watermarking can provide working and successful solutions. Specifically for audio and video it seems that watermarking technology will become widely deployed. The DVD industry standard, as an example, will use watermarking for the copy protection system. Similarly, there exist plans to use watermarking for copy protection for Internet audio distribution. Broadcast monitoring using watermarking is another application that will probably widely be deployed for both audio and video.

Whether the development of watermarking technology will become a success story or not is an interesting yet unclear question. Watermarking technology will evolve, but attacks on watermarks as well. Careful overall system design under realistic expectations is crucial for successful applications.

1.9 Conclusions

In this overview, the most important aspects, design requirements, system issues, and techniques for digital watermarking are reviewed. The historical roots of digital watermarking derive mainly from steganography, the art of data hiding. Although digital watermarking and steganography are in some sense similar, the main difference lies in the notion of robustness for digital watermarks. Watermark robustness is one of the major design issues, besides imperceptibility. It has been shown that the various digital watermarking applications, such as data tracking, data monitoring, and copyright protection, result in corresponding design

issues and algorithm requirements. Some schemes require the original data set in order to recover an embedded watermark and others do not. Further, in some publications methods are proposed that allow full watermark extraction, whereas in other publications techniques are presented which only allow verification if a given watermark is present in the data under investigation. However, these two approaches are inherently equivalent in that a watermark-extraction scheme can be transformed into a watermark-verification scheme and vice versa. Although often associated to still images, video, and audio, digital watermarking is also applicable to other digital data such as text, 3-D meshes, or face animation parameters.

Designing watermarking methods does not only have to consider robustness against standard data processing, but also robustness against malicious attacks. Several classes of attacks have been outlined, and remedies were given to make watermarks attack resistant. As a general statement, it can be said that watermarks should be sufficiently over designed and contain enough redundancy to ensure resilience against attacks. For copyright enforcement, additional aspects have to be considered. One problem is to prove who first watermarked data if several watermarks are present in the data. Solutions to this problem might consist of digital time stamping or watermark registration. Further, it has been shown that robustness is not sufficient to resolve rightful ownership, even if the original data are available. In addition, the used watermarking method needs to be noninvertible. Although working systems are already available, research in digital watermarking has to continue. There is a huge demand from content providers and IPR owners. The market is currently far from being saturated and many more companies are expected to be founded in the near future. The question whether digital watermarks will be used as legal proof in court is not yet decided and difficult to answer. There are, however, other applications, like multimedia copy protection systems and data broadcast monitoring, where watermarking can be seen in operation.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Methods for embedding information into text documents have been used for a long time by secret services. For text watermarking, methods that hide information in the semantics have to be distinguished, which means in the meaning and ordering of the words, and methods that hide information in the format, layout and the appearance.

The first class designs a text around the message to be hidden. In that sense, the information is not really embedded in existing information, but rather covered by misleading information. This class of techniques is outside the scope of this project. The project concentrates on the latter type of information-embedding methods which use an existing text document into which data are embedded. Formatted text is probably the medium where watermarking methods can be defeated most easily. If the watermark is in the format, then it can obviously be removed by "retyping" the whole text using a new character font and a new format where "retyping" can be either manual or automated using optical character recognition (OCR). OCR systems are still not perfect for many applications today and often need human supervision. Thus, removal of watermarks either yields bad results (single characters are wrong, due to OCR) or is expensive.

The goal is to make watermark removal more expensive than obtaining the right to copy from the copyright owner. If this goal is achieved, text watermarking makes sense, though it can be defeated [7]. Text watermarking has applications wherever copyrighted electronic documents are distributed. Important examples are virtual digital libraries where users may download copies of documents, for example, books, but are not allowed to further distribute them or to store them longer than for a certain predefined period. In this type of application, a requested document is watermarked with a requester specific watermark before releasing it for

download. If later on illegal copies are discovered, the embedded watermark can be used to determine the source.

2.2 Basic Text Watermarking & Detection

A mark can be placed in a *formatted* document by altering either the appearance or the position of a text element such as an individual character or word. A collection of marks within a document forms a *codeword*, and the document is encoded. In this section three distinct types of marks are introduced. Electronic copying does not alter the marks, but conventional copying techniques degrade the image and may make the marks unreadable. The challenge is to find imperceptible text alterations that can be reliably detected after documents have been printed, photocopied, or transmitted by facsimile.

2.2.1 Marking Techniques

The four marking techniques that will be discussed illustrate different approaches rather than form an exhaustive list of marking techniques [7]. They can be used either separately or jointly. Each technique enjoys certain advantages or applicability. For example line-shift coding is more robust to attacks as compared to word-shift coding but word-shift coding has a greater payload i.e. the amount of information that can be hidden as compared to line shift coding.

2.2.1.1 Line-Shift Coding

In this approach a mark is embedded on a page by vertically displacing an entire text line. In a typical implementation, a line is moved up or down by a small number of pixels in such a way that the movement is imperceptible. The lines immediately above or below (or both) are left unmoved. These unmoved adjacent lines serve as reference locations in the decoding process.Figure 2-1 shows two samples of text line images where in the second sample the second line of the text is slightly moved up.

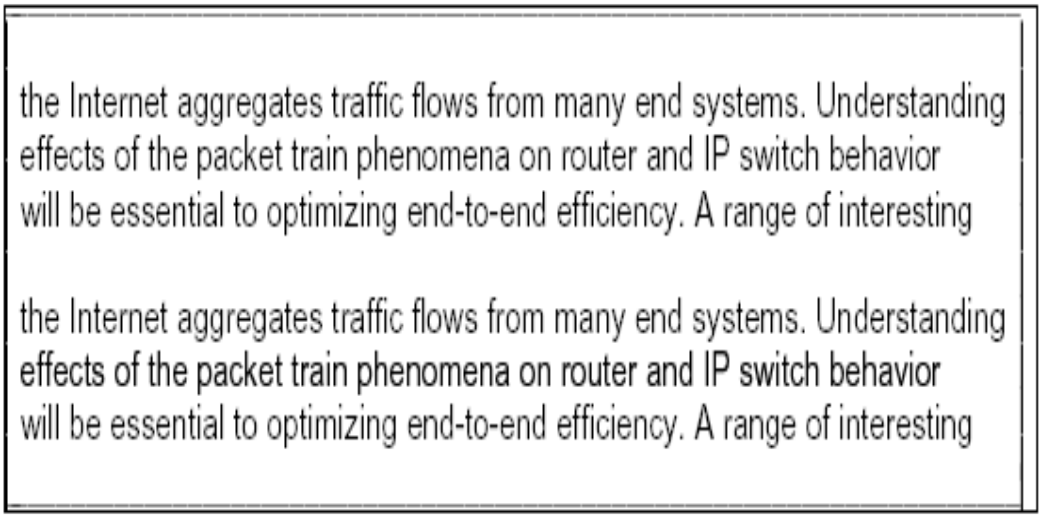

Figure 2-1 Example of Line-Shift Coding

Most documents are formatted with uniform spacing between adjacent lines within a paragraph. Though the human eye is particularly adept at noticing deviations from uniformity, experiments suggests that vertical line displacements of 1⁄300 inch and less go unnoticed by readers.

The principal advantage of this marking technique is found in decoding. Since a document's initial interline spacing is uniform, the presence or absence of a mark can be detected by analysis of the interline spacing of a recovered document, with no need for any additional information about the original, unmarked document. Therefore, anyone can read the information. This technique can also be used to include computer readable information in books or journals for cataloging and automatic identification.

2.2.1.2 Word-Shift Coding

In this technique a mark is embedded by horizontally shifting the location of a word within a text line, as shown in Figure 2-2. In a typical implementation, a word is displaced left or right, while the words immediately adjacent are left unmoved. These unmoved words can then serve as reference locations in the decoding process.

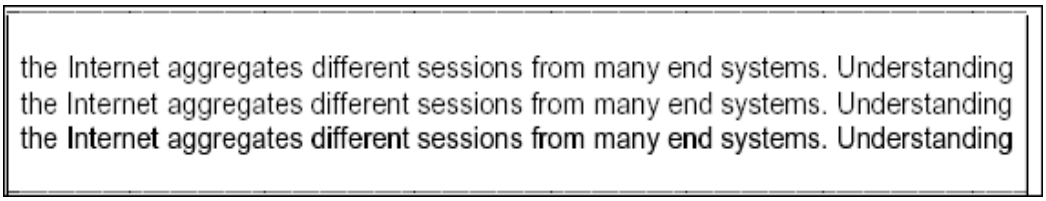

Figure 2-2 Illustration of Word-Shift Encoding

Formatted documents with justified text typically use variable spacing between words to distribute white space in a visually pleasing fashion. Readers accept a wide variation in text setting within a line; experiments suggest that horizontal word displacement of 1/150 inch and less readily go unnoticed. Since the word spacing in the original document is not uniform, detecting a word displacement requires knowledge of the original word spacing. Hence, the word positions in the unmarked document must be known in order to extract the hidden information. The hidden information can only be read by the organization that owns the original document or its agent.

2.2.1.3 Character Coding

Character coding is a class of techniques which embed a mark by altering a particular feature of an individual character, as shown in Figure 2-3. Examples of possible feature alterations include a change to an individual character's height, or its position relative to other characters (e.g. a "kerning" adjustment). Once again, some character features are left unchanged to facilitate decoding. For example, a detection algorithm might compare the height of a hypothetically altered character with that of another unmodified instance of the same character elsewhere on the page.

Figure 2-3 Example of Character Coding

Imperceptibly embedding a mark by character alteration often requires extremely careful attention to the context of the character to be altered. A reader is

more likely to notice a character alteration if an identical, unaltered character is immediately adjacent. Detecting the presence or absence of a mark might or might not require information from the original, unmarked image, depending on the marking technique and the rule for selecting the characters that are altered.

2.2.1.4 Comparison

These common watermarking techniques alter ever smaller textual elements to embed a mark; the size of the alterations is similar but the "signal level" is greater for the large elements. Since the larger text elements have a larger signal to noise ratio when subjected to the same distortions, line-shifting is expected to be the most robust marking technique[7]. For this reason, line-shifting is particularly well suited to marking documents to be distributed in paper form, where considerable degradation can be introduced by photocopying, or simply handling.

Word-shifting is expected to be less discernible to the reader than line-shifting, yet also more difficult to detect in the presence of noise. Character encoding has the advantage of a potentially large coding *density*; on a given page of text, many more marks can be inserted by altering characters than by altering lines or words. This property makes this technique attractive in applications requiring very wide distribution of uniquely marked electronic documents (e.g. distributing a large circulation, general interest magazine). The large coding density also allows redundancy to be added for error correction.

2.3 A Text Watermarking Algorithm Based On Word Classification & Inter-Word Space Statistics

2.3.1 Introduction

For an efficient inter-word space patterning, this algorithm [8] uses the novel concepts of word classification and inter-word space statistics. All the words in a text document are classified depending on some features. Then adjacent words comprise a segment and the segment is classified depending on the class labels of the words

within the segment. The same amount of information is inserted into each of the segment classes. The information is encoded by modifying statistics of inter-word spaces of the segments belonging to the same class. Since a large number of segments belonging to the same class would be available, the information can be encoded by adjusting statistical values of inter-word spaces [8].

The algorithm has a global property in the sense that it hides some information into a segment class and the segments in that class are distributed over the entire document. Due to this reason, this algorithm is tolerable to some kind of segmentation errors and reveals several advantages in terms of imperceptibility, robustness, and payload.

2.3.2 Word and Segment Classification

It is assumed that the input text document is already segmented into pages, lines, and words. The next step will be the classification of these pages, lines and words according to the size of the words and the classification criterion involved so that subsequently the inter-word space statistics can be determined, analyzed and modified to embed the information..

2.3.2.1 Word classification

The algorithm will be described assuming a line containing n words. The i-th word is denoted by wi and the width of word image of wi by $l(wi)$. The $l(wi)$ is measured by number of pixels. Let K the number of word classes and $class(wi)$ the class of wi. The widths of word images are used as a feature for the word classification task. Rather than using the absolute width values, the word classifier compares the widths of neighboring words. Table 2-1 illustrates a classification rule that generates two word classes, i.e. K=2. It classifies the word wi by comparing widths of the left (wi-1) and right (wi+1) words. Another classification rule compares five consecutive words and produces four classes, i.e. K=4. Table 2-2 shows the rule. The conditions are designed with the aim of keeping a balance of occurrence frequencies of word classes.

Table 2-1 Word Classification Rule (K=2)

Conditions	***Class* (*wi*)**
$l(wi\text{-}1)>l(wi+1)$	0
$l(wi\text{-}1)\leq l(wi+1)$	1

Table 2-2 Word Classification Rule (K=4)

Conditions*	***class*(*wi*)**
$a\geq b$ and $c\geq d$	00 (0)
$a\geq b$ and $c<d$	01 (1)
$a<b$ and $c\geq d$	10 (2)
$a<b$ and $c<d$	11 (3)

* $a=l(wi\text{-}2)+l(wi\text{-}1)$, $b=l(wi+1)+l(wi+2)$

$c=l(wi\text{-}1)+l(wi+1)$, $d=l(wi\text{-}2)+l(wi+2)$

For the classification of the first and last words (*w1* and *wn*) in the line, the word list is regarded to be circular. That is, the left word of *w1* is *wn* and the right word of *wn* is *w1*.

2.3.2.2 Segment classification

The segment is defined to be *s* consecutive words in a line. The first segment is the ordered list (*w1*, *w2*, … , *ws*). The next segment starts from the last word of previous segment, so the second segment is (*ws*, *ws+1*, … , *w2s-1*). Table 2-3 demonstrates a line having 9 words (i.e. n=9) that are classified into 2 classes (i.e. K=2). The segments constructed for s=3 are also shown.

Table 2-3 Word Classification (K=2) and Segment Classification (s=3 and L=8).

***L*(*wi*)** :	19	48	155	107	58	25	61	138	38
Word class	1	1	1	0	0	1	1	0	0
Seg class		111		100		011		100	

Segments are introduced so that the word-shifts do not interfere with each other. Since the first and last words in a segment are shared by neighboring segments, their locations are fixed. Only the inner words are allowed to move.

The class labeling of a segment is done by using the class labels of the words in the segment. For example, the second segment has three words having the class labels, 1, 0, and 0, respectively, so the segment has class label 100. The number of segment classes is denoted by L where L is K raised to the power s.

2.3.3 Insertion and detection of watermark signals

A fixed same amount of information is independently inserted into each of the segment classes. Assuming p bits of information for a segment class, the payload of the algorithm is $p*L$ bits where L represents the number of segment classes.

Simple encoding rules can be designed as:

Rule 1:

(s=3, L=64, $\Omega = \mu$, payload=64 bits)

if($\mu 1 \leq \mu 2$) signal 1,
otherwise signal 0.

Rule 2:

(s=3, L=64, $\Omega = (\mu, \sigma)$, payload=128 bits)

if ($\mu 1 \leq \mu 2$ and $\sigma 1 \leq \sigma 2$) signal 00 (0),

else if ($\mu 1 \leq \mu 2$ and $\sigma 1 > \sigma 2$) signal 01(1),
else if ($\mu 1 > \mu 2$ and $\sigma 1 \leq \sigma 2$) signal 10(2),
else if ($\mu 1 > \mu 2$ and $\sigma 1 > \sigma 2$) signal 11(3).

In rule 1, one bit is hidden in one segment class, so the total number of bits hidden in a document is 64 bits. In the rule 2, two bits are inserted, so 128 bits can be hidden in a document.

The information encoding requires specific statistical distributions of inter-word spaces. So the words in a segment need to be shifted left or right depending on the required distributions. This task can be easily accomplished since the adjacent segments share only one word at the boundaries. The position of the common words is fixed and only the remaining *s*-2 words in a segment are shifted.

The conditions in the encoding rules use not the individual spaces but the statistics of a number of spaces so if a segment class satisfies the condition, there is no need to shift any word. Otherwise the words are shifted one pixel at a time until the condition is met allowing some margin. This scheme allows information to be inserted with the least amount of word shift.

The signal detection process is straightforward. First apply line and word segmentation to the input document images, next classify the words and segments then construct the segments sets and compute the inter-word spaces, then compute the statistical distributions and finally decode the signals from the distributions.

2.4 Conclusion

Four techniques have been described for encoding information into text images; 2.2.1.1 line shift encoding, 2.2.1.2 word shift encoding and 2.2.1.3 character modification 2.3 Inter-word space modification. These techniques offer an increasing number of locations for placing information. However, the techniques with the greatest number of locations are least able to survive the distortion introduced by printing, copying and faxing.

Word Shift and line shift encoding can be decoded using the edges of features, the center of mass of features or the correlation between the profiles of distorted image and perfect copies of the various encodings. The three techniques require increasing amounts of processing. Correlation decoding is necessary for word shift encoding when distortion is present, while centroid decoding is adequate for line shift encoding. Decoding based upon edges or baselines is only appropriate for line shift encoding in a low noise environment. The last two decoding techniques require information from the publisher about the original positions of the centroids of lines or the profiles of words. However, because baselines are equally spaced before encoding, this technique has the unique characteristic that information can be extracted from a document without additional information from the publisher. Therefore, information can be extracted by anyone, not only the publisher's agent. Information encoded in line spaces may be extracted by both baseline decoding and centroid decoding depending upon who is extracting the information and how much the document is distorted. There are many instances where a publisher may want to send information along with the document (i.e. keywords for cataloguing or signatures for insuring that the electronic document has not been altered). Since these documents are likely to have remained in electronic form and do not have severe distortion, baseline decoding is appropriate for extracting this information.

CHAPTER 3

ALGORITHM & ITS IMPLEMENTATION

3.1 Foreword

Digital watermarking methods for text documents are limited because of the binary nature of text documents. A distinct feature of a text document is its space patterning. In this project an algorithm has been devised that slightly modifies the inter-word spaces of different text lines. After the modification, the average spaces of various lines have the characteristics of a sine wave and the wave constitutes a mark. Both private and public watermarking algorithms have been developed. Preliminary experiments have shown promising results. Experiments suggest space patterning of text documents can be a useful tool in digital watermarking.

3.2 Introduction

With the wide spread use of the Internet in the society, the distribution and access of information is greatly facilitated. However, without methods which prevent or discourage illicit redistribution and reproduction of information content, copyright can be easily infringed. Digital watermarking is widely believed to be a valid solution to the problem and currently there is intensive research in this area, in both academic and industrial communities.

Compared to the plurality of previously proposed methods in digital watermarking for picture and video images, digital watermarking methods for text documents are very limited. One reason for this difference is that text is a binary image and lacks rich grayscale information.

3.3 Development tool

The tool selected for implementation of the algorithm was MATLAB. The main reasons why MATLAB was selected are, Matlab is “the” tool for simulating

and implementing modulation techniques, the ability to treat the data of text images in the form of matrices, ability to perform complex mathematical operations that were a vital part of the algorithm, image processing ability of MATLAB.

3.4 Environment

The project can be run on Microsoft Windows 98, Microsoft Windows XP, Microsoft Windows 2000/NT with Matlab 7.0 and Microsoft Access 2003. It will not run in Linux based operating systems. The version of Matlab has to be 7.0 or above because of the use of GUIDE (GUI Development Environment) which generates .fig files that are not recognized by previous versions.

3.5 Loading & Digitizing the Image

The watermarking scheme has been developed for the images of text documents. That is images in the form of bmp, jpg, png that contained the text and white background were selected.

The loading of the images was the first step. The program developed successfully loads images in bmp, png, jpg format. The loaded image is converted into binary format. This process is known as digitizing the image and is done by first calculating a threshold value using the function greythresh and then according to the threshold value and whether there is a text or white space present the image, pixels are converted into 1's and 0's

As a result the places where there is data are stored as 0's and where there is the white background are stored as 1's in a matrix of dimensions similar to that of the image.

A text page in digital form can be represented by the function

$$f(x, y) \in [0,1], x = 0,1,...,W, y = 0,1,...,L$$

that presents black and white pixels. Here, W and L are the width and length of the page in pixels respectively.

3.6 CALCULATION OF IMAGE STATISTICS

3.6.1 *Calculating the Horizontal Profiles of the Image*

The horizontal profile of an image is the sum of all the pixel values from the left end to the right as shown in Figure 3-1.

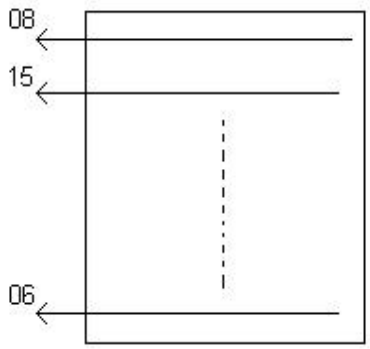

Figure 3-1 Horizontal Profile

This is done by taking the summation of the rows of the matrix in which the image is stored in binary format.

3.6.2 *Calculating the Vertical Profiles of Image*

Similarly the vertical profile of an image is the sum of all the pixel values from the top to the bottom as shown in Figure 3-2.

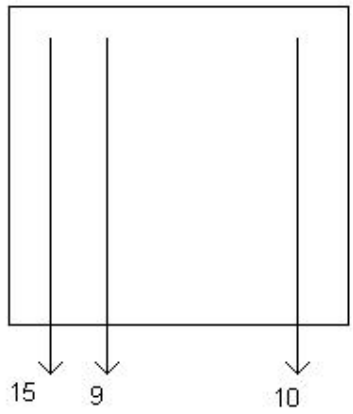

Figure 3-2 Vertical Profile

This is calculated by taking the summation of the columns of the matrix in which the image is stored in binary format.

3.6.3 Calculating the Vertical Start and End of a Line

With the help of the horizontal profile the vertical start and end of the text lines in an image can be found. If the horizontal profile value is equal to the width of the image this indicates that the row corresponding to this index contains all white pixels i.e. there is no text in this row from the left end to the right end. If the horizontal profile value is less than the width of the image this indicates that the row corresponding to this index contains some black pixels as well as white pixels i.e. there is text in this row.

Thus for consecutive rows of values equal to the width of the image, there is no text i.e. the line has not started yet. When this sequence is broken for the first time, it indicates that the lines have started from this row. Figure 3-3 shows the horizontal profile of a text image having 26 lines.

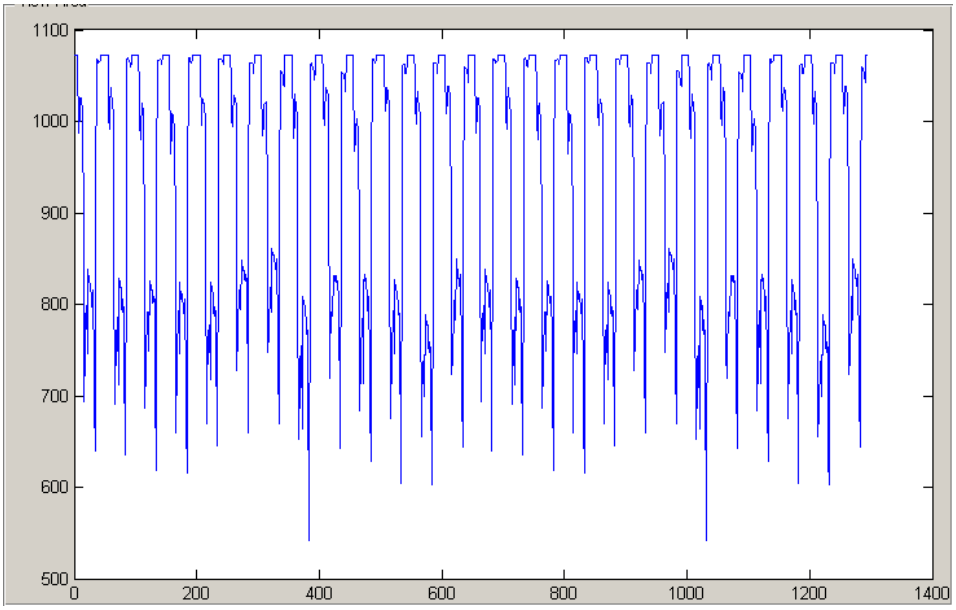

Figure 3-3 The Horizontal Profile of a Sample Text Document with 26 Lines

The depressions in the Figure 3-3 indicate text lines whereas the solid maximum values indicate the interline spaces

3.6.4 Calculating the Height of a Line

The height of the line is calculated by counting the number of successive horizontal profile values that are lesser than the width of the file. They represent horizontal regions in which there is nothing written from the left end of the text image to the right end.

3.6.5 Calculating the Number of Lines

After each line there is once again a set of horizontal profile values that are equal to the width of the file these are the inter-line spaces. After a consecutive number of values equal to the width of the document these values drop which indicates the vertical start of a line. Then the next few values will be less than the width of the text image until the line ends and there are once again values equal to the width of the document. The vertical starts of new lines can be added together to determine the number of lines. First the number of lines is set to zero and then every time the horizontal profile value drops after a series of maximum values the number of lines value is incremented by one thus when the document is exhausted in length the total number of lines can be obtained from this value. The vertical start and end values will be unique for each line because they represent the pixels where these lines are starting and ending in height.

3.6.6 Calculating the Horizontal Start and End of a Line

With the help of the vertical profile the horizontal start and end of the text lines in an image can be found. If the vertical profile value is equal to the height of the image this indicates that the column corresponding to this index contains all white pixels i.e. there is no text in this whole column from top to bottom. If the vertical profile value is less then the height of the image this indicates that the column corresponding to this index contains some black pixels as well as white pixels i.e. there is text in this column.

Thus for consecutive columns of values equal to the height of the image, there is no text i.e. the line has not started yet. When this sequence is broken for the first

time, it indicates that the lines have started from this column. Figure 3-4 shows the vertical profile of a line containing five words.

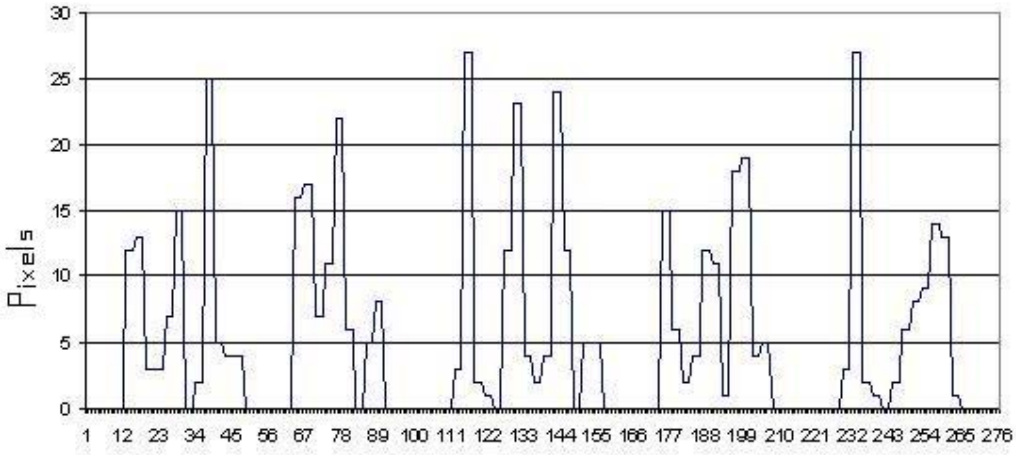

Figure 3-4 Vertical Profile of a Line Containing Five Words

3.6.7 Calculating the Length of a Line

The length of a line is calculated by counting the number of successive vertical profile values that are lesser than the height of the image. For left and right justified text, these values will be the same for each line.

3.6.8 Start / End of Each Word in a Line & Number of spaces

For each line, the number of words, the start and end of each word, the total no of spaces in the line is calculated. The start and end of the words are calculated with the help of the height of each line. The vertical profile of the whole line is taken and the values are assessed. If the values are equal to the height of the line it means that no word is written in it. If the value is less then the height it means that a word is written at this column.

The successive values of the vertical profile of the line are observed to find where a word starts and where it ends. A threshold which varies with font size is used

to differentiate between inter-word space and inter-character space. With these start and end values the length of the word is calculated. The total number of spaces S_t is found by subtracting the total length of all the words from the length of the line. The process is repeated for all the lines.

3.6.9 Calculating the Average Space of Each line

For a line with *d* words, an average space S_a is obtained as in Equation 3-1

$$S_a = S_t / (d-1), \qquad d \neq 1 \qquad \textit{(Equation 3-1)}$$

where S_t is the total inter-word space in a text line calculated in pixels.

Figure 3.5 shows a typical profile of S_a with respect to text lines in a paragraph. The average inter-word space in pixels is measured as a decimal to retain measurement accuracy.

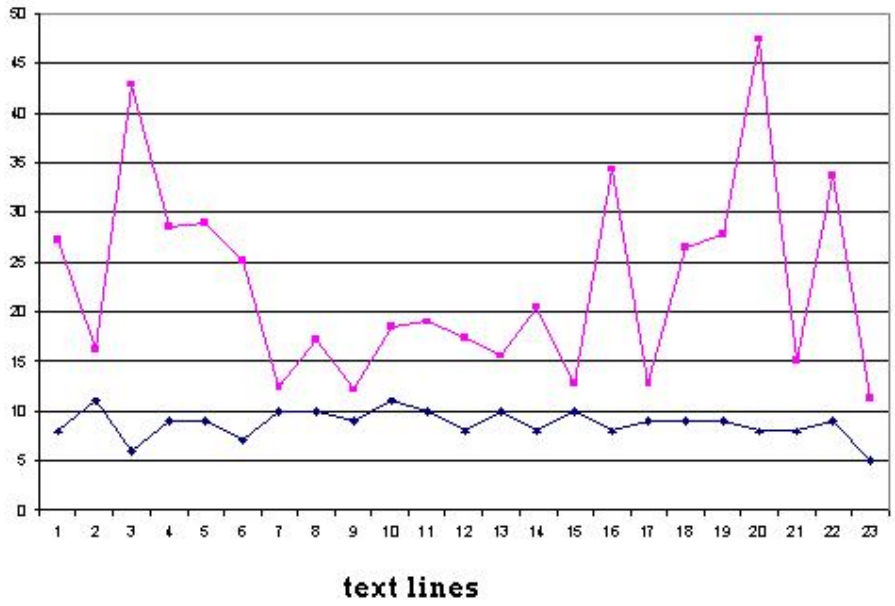

Figure 3-5 Profile of Average Space for Text Lines (resolution: 300pixels/inch)

The red line gives the average space in pixels of the text line and the blue line gives the number of words in the corresponding line.

It can be seen that S_a varies randomly across different lines. In this Figure 3-5 is that the greater the number of words in a text line the lesser is S_a which is logically correct. This process is also repeated for all the lines and so an array of S_a values is obtained.

3.7 SPACE MARKING

In the developed algorithm inter-word spaces have been identified as the key parameter. Due to the random nature of average spaces of text lines across a text document, the inter-word spaces can be considered as a discrete random variable *(rv)X(n)* as shown in the Equation 3-2.

$$\boldsymbol{X}(n) = S_{an}, n = 0,1,..., N-1 \qquad \textit{(Equation 3-2)}$$

where n represents the index of a text line in a text document with *N* lines. S_{an} represents S_a of line n.

The *(rv)X(n)* is both inherent in a text document and is invisible to human readers. Another view of the space marking method can be considered as marking on the *(rv)X(n)*. Thus, encoding can be considered as modifying *(rv)X(n)*, and decoding can be considered as reconstructing the modified *X(n)* and extracting information from it.

3.7.1 *Space Varying Sine Wave*

A sine wave which varies over a number of text lines, has some attractive characteristics for watermarking inter-word space i.e. a sine wave varies gradually so local variation may be unnoticed, a sine wave's amplitude, frequency and phase can be used to carry coding information and a sine wave's periodic symmetry may make decoding easy and reliable.

Thus, different text lines across a text document can be used to encode a sine wave. More specifically, the values of Sa for different text lines, or their variations, can act as sampling values of a sine wave.

In comparison to merely shifting words horizontally in word-shift coding, the modification of the interword space of a text line involves the words in this line being horizontally expanded or shrunk so that the required total interword space, and thus Sa of the line, is obtained.

3.7.2 *Phase of a Sine Wave*

The phase of a sine wave has been chosen primarily to carry information in space watermarking because for space watermarking to be unnoticeable, changes in interword spaces have to be kept to a minimum. On the other hand, there should be enough space modification so that marking can be correctly detected. These contradicting requirements suggest a narrow range of marking amplitude.

Moreover the Sampling Theorem suggests that the sampling frequency has to be at least twice that of the marking sine wave. There exist certain frequencies at which variations are most noticeable according to research in human visual perception [6] and it is reasonable to avoid marking in the neighborhood of these frequencies. So the capacity of a sine wave's amplitude and frequency in carrying watermarking information is limited.

3.8 Private Watermarking

It is the type of watermarking in which the original as well as the watermarked text image is required for information extraction

3.8.1 First, the mean of Sa is calculated as shown in Equation 3-3

$$a_1 = \frac{\sum_{n=p}^{q} S_{an}}{q - p + 1}, 0 \le p < q < N \qquad \textit{(Equation 3-3)}$$

Where p and q are the indices of the first and last text lines in the workplace between which a watermarking sine wave will reside.

3.8.2 Then for each line, a watermark component is determined by the sine wave shown in equation 3-4

$$W_n = C_1 a_1 \sin(\omega_1 (n - p) + \phi_1) \qquad \textit{(Equation 3-4)}$$

where variations to ϕ_1 carry the information to be embedded..

3.8.3 Next, Wn is added to Sa for line n, thus generating a new average space which is shown in Equation 3-5

$$S_{an}' = S_{an} + W_n \qquad \textit{(Equation3-5)}$$

3.8.4 This S_a will be used to modify the words in each text line as will be shown in section 4.3 of present document

The private method can be considered as adding a constant part to the original $rvX(n)$ thus generating a new $rvY(n)$ as shown in Equation 3-6

$$\boldsymbol{Y}(n) = \boldsymbol{X}(n) + W_n \qquad \textit{(Equation 3-6)}$$

3.9 Public Watermarking

It is the type of watermarking in which the original text image is not required for information extraction.

Unlike private watermarking, after which neighboring text lines still have random values of Sa, the values of Sa of the lines used in public watermarking should have a certain relationship so that they can directly act as the sampling values of a sine wave.

Experiments have shown that it is inappropriate to take all lines in a text image for public watermarking because of the degree of variation in Sa of the original text lines. Observation of the profiles of Sa suggests text lines with a larger number of words have closer values of Sa. This can be contributed to two reasons. First, in a text line with a larger number of words, an average word and its associated space is allocated a smaller number of pixels. Thus, the difference between Sa of similar lines is smaller. Second, a text line with a larger number of words is less likely to be justified, or is justified to a smaller degree.

3.9.1 Based on observations, first a key is chosen in a text so that all text lines whose total number of words, are larger than or equal to the key are watermarked.

3.9.2 Then a set Sw of text lines is selected from the text so that the number of words from each line in this set is not less than the selected key.

3.9.3 Then the mean of Sa of text lines in set Sw is calculated from Equation 3-7

$$a_2 = \frac{\sum_{m=u}^{v} S_{am}}{v-u+1}, 0 \le u < v < N \qquad \textit{(Equation 3-7)}$$

where u and v are similar to p and q in Equation 3-3, but u and v are the indices of the text lines within Sw, rather than the indices in the original text in the implementation; m is the index of a text line within Sw, and Sam represents Sa of line m.

3.9.4 For each text line in Sw, a watermark component is determined by a sine wave as shown in Equation 3-8.

$$W_m = C_2 a_2 \sin(\omega_2 (m-u) + \phi_2) \qquad \textit{(Equation 3-8)}$$

where variations to ϕ_1 carry the information to be embedded.

3.9.5 Next, for each line in Sw, Sa is replaced with the sum of a_2 and Wm, thus generating a new average space which is shown in Equation 3-9

$$S_{am}' = a_2 + W_m, \quad \text{if line } m \in S_w, \text{ otherwise unchanged} \quad \text{(Equation 3-9)}$$

3.9.6 This S_a will be used to modify the words in each text line as will be shown in section 4.3 of present document

Thus, for text lines belonging to Sw, a new $rvY(n)$ is obtained as shown in Equation 3-10

$$\mathbf{Y}(m) = a_2 + W_m, \quad \text{if line } m \in S_w, \text{ otherwise unchanged} \quad \text{(Equation 3-10)}$$

CHAPTER 4

CREATING, EMBEDDING & EXTRACTING INFORMATION

4.1 Creating the Watermark Information

For each line, a watermark component is determined by the sine wave as shown in equation 4-1

$$W_n = C_1 a_1 \sin(\omega_1 (n - p) + \phi_1) \qquad \textit{(Equation 4-1)}$$

where Wn represents the desired watermark component of a text line with an index of n ; ω_1 and ϕ_1 are the radian frequency and initial phase angle of the sine wave respectively. C_1 is a constant determining the amplitude of the sine wave. Experiments have shown that the preferred value of C_1 is 0.2, because for a font size of ten to fourteen $C_1 = 0.2$, makes the watermarking both unnoticeable and correctly recoverable.

PSK-16 has been used to embed the information in the sine wave. Thus each line in the text document is able to embed four bits of data. Each possible combination of these four bits adds a pre-defined phase shift to the sine wave thus changing the characteristic of the sine wave.

In other words it can be said that there are sixteen possible sine waves and each line is space marked with one of the possible sixteen sine waves. In a simple set-up where each successive combination changes the phase of the sine wave by 1 pi /8, the values obtained are shown in Table 4-1.

Table 4-1 Different Values for Phase in PSK-16

Value in Bits	ϕ_1
0000	1 * (pi / 8)
0001	2* (pi / 8)
0010	3* (pi / 8)
0011	4* (pi / 8)
0100	5* (pi / 8)
0101	6* (pi / 8)
0110	7* (pi / 8)
0111	8* (pi / 8)
1000	9* (pi / 8)
1001	10* (pi / 8)
1010	11* (pi / 8)
1011	12* (pi / 8)
1100	13* (pi / 8)
1101	14* (pi / 8)
1110	15* (pi / 8)
1111	16* (pi / 8)

There are sixteen possible values because the maximum value of the sine wave is 2pi. Then with the help of formula alpha Wn is calculated. After that the new average space of each line is calculated with the help of Wn in one of two possible ways

4.2 Adding Information to the Watermark

Any digital information can be added to the watermark. If a string, image or number is to be added it is first converted into bits and then the phase of the sine wave is manipulated to contain the information.

4.2.1 String Information

To embed the string information, the ASCII values of string characters are taken, these values are then converted into decimal form, after which the decimal form is converted into binary form.

Each alphabet letter comprises of 8 bits. Thus to embed it in the watermark, using the PSK-16 method, in the first go the first 4 bits of information are taken and according to their value the phase of the sine wave is altered with the Equation 4-1.

Table 4-2 shows the different binary values obtained from the alphabet characters and the corresponding phase variation.

Table 4-2 Showing the Different Values for Characters

Alpha-bets	ASCII Values	Binary Value	Info in 1st line	Phase Variation	Info in 2nd line	Phase Variation
A	65	01100101	0110	7* (pi / 8)	0101	6* (pi / 8)
B	66	01100110	0110	7* (pi / 8)	0110	7* (pi / 8)
C	67	01100111	0110	7* (pi / 8)	0111	8* (pi / 8)
D	68	01101000	0110	7* (pi / 8)	1000	9* (pi / 8)
E	69	01101001	0110	7* (pi / 8)	1001	10* (pi / 8)
F	70	01110000	0111	8* (pi / 8)	0000	1* (pi / 8)
G	71	01110001	0111	8* (pi / 8)	0001	2* (pi / 8)
H	72	01110010	0111	8* (pi / 8)	0010	3* (pi / 8)
I	73	01110011	0111	8* (pi / 8)	0011	4* (pi / 8)
J	74	01110100	0111	8* (pi / 8)	0100	5* (pi / 8)
K	75	01110101	0111	8* (pi / 8)	0101	6* (pi / 8)
L	76	01110110	0111	8* (pi / 8)	0110	7* (pi / 8)
M	77	01110111	0111	8* (pi / 8)	0111	8* (pi / 8)
N	78	01111000	0111	8* (pi / 8)	1000	9* (pi / 8)
O	79	01111001	0111	8* (pi / 8)	1001	10* (pi / 8)
P	80	10000000	1000	9* (pi / 8)	0000	1* (pi / 8)

Q	81	10000001	1000	9* (pi / 8)	0001	2* (pi / 8)
R	82	10000010	1000	9* (pi / 8)	0010	3* (pi / 8)
S	83	10000011	1000	9* (pi / 8)	0011	4* (pi / 8)
T	84	10000100	1000	9* (pi / 8)	0100	5* (pi / 8)
U	85	10000101	1000	9* (pi / 8)	0101	6* (pi / 8)
V	86	10000110	1000	9* (pi / 8)	0110	7* (pi / 8)
W	87	10000111	1000	9* (pi / 8)	0111	8* (pi / 8)
X	88	10001000	1000	9* (pi / 8)	1000	9* (pi / 8)
Y	89	10001001	1000	9* (pi / 8)	1001	10* (pi / 8)
Z	90	10010000	1001	9* (pi / 8)	0000	1* (pi / 8)

Thus for string coding strings two lines are needed to embed one alphabet letter.

4.2.2 Image Information

With sufficient payload, any image can be added into the watermark by first digitizing the image, next taking 4 consecutive bits of information and then using the group to change the phase of the sine wave thus modifying a line The process is repeated until the bits are exhausted.

4.3 Embedding the information

4.3.1 Change in total Space

The new average space S_a' is calculated by either the private or the public watermarking methods. If it is to be used after the modification of the inter-word space of a text line then, the change of the total inter-word space of this text line in pixels is calculated as shown in Equation 4-2.

$$S_{tc} = (S_a' - S_a)(d-1) \qquad \textit{(Equation 4-2)}$$

Where d is the number of words, and Sa is the original average space of the text line as in (Equation 4-1).

If $S_{tc} > 0$, then the total inter-word space of this text line will expand and the words in this text line will shrink. If $S_{tc} < 0$, the total inter-word space of this text line will shrink and the words in this text line will expand.

4.3.2 Distribution of New Space

For any word in this text line with an index i, suppose its width before the modification is Pxl_i in pixels, then the expansion or shrinkage of width distributed to this word is given by Equations 4-3 and 4-4.

$$ES_i = \left\lfloor \frac{S_{tc}}{\sum_{i=1}^{d} Pxl_i} Pxl_i \right\rfloor, \qquad if \quad S_{tc} \geq 0$$

(Equation 4-3)

or

$$ES_i = \left\lceil \frac{S_{tc}}{\sum_{i=1}^{d} Pxl_i} Pxl_i \right\rceil, \qquad if \quad S_{tc} < 0$$

(Equation 4-*4)*

4.3.3 Check

ES_i is rounded to an integer, since it presents a number of pixels. Therefore, a difference may exist between S_{tc} and the sum of ES_i, which is shown in Equation 4-5

$$S_d = S_{tc} - \sum_{i=1}^{d} ES_i \qquad \text{(Equation 4-5)}$$

The difference S_d is added to the largest ES_i i.e., the word with the largest width in the text line.

4.3.4 Expanding or Shrinking a Word

To expand or shrink a word, vertical lines of equal intervals in the word are duplicated or removed respectively. The interval is calculated as in Equation 4-6

$$Iv_i = \left\lfloor \frac{Pxl_i}{|ES_i|} \right\rfloor \qquad \text{(Equation 4-6)}$$

Again, the interval Iv_i is rounded to an integer. After the expansion or shrinkage of this word, it will have a new width in pixels, which is given by Equation 4-7

$$Pxl_i' = Pxl_i - ES_i \qquad \text{(Equation 4-7)}$$

The two sides of a text line are not changed while the line is being expanded or shrunk. To shrink or expand words, at the left half of this text line, the left side of each word is kept fixed and the word is shrunk or expanded, i.e. vertical lines of equal intervals are removed or duplicated; at the right half of this text line, the right side of each word is kept fixed and the word is shrunk or expanded. An alternative to this

implementation is to pre-calculate the optimum horizontal position of each word in a text line, and accordingly, shift the whole word while it is being expanded or shrunk. It is considered an error if a word is expanded to such an extent that the interword space becomes indistinguishable or neighboring words overlap each other after the expansion. Thus, a text line justified at both sides will remain justified after modification of interword space has been applied.

4.3.5 Work Place and Sampling Points in a Sine Wave

A single text page or several pages of a text document are considered to be a workplace. Relevant text lines in this workplace are considered as sampling points of the sine wave for watermarking. Phase information can be either, the absolute phase in this workplace or, the relative phase if several waves are involved.

4.4 Watermark Detection

4.4.1 Private Method

When a text has been watermarked using the private method, $rvY(n)$can be obtained by a reconstruction of Sa as in formula (1). With the original unmarked text, the watermark component Wn from Equation 4-7 will now be given as shown in Equation 4-8.

$$W_n = \mathbf{Y}(n) - \boldsymbol{X}(n) \qquad \textit{(Equation 4-8)}$$

where $Y(n)$ is the random variable that represents the average spaces in the watermarked image and $X(n)$ is the random variable that represents the average spaces in the original image. The difference between the two gives the watermark Wn added

4.4.2 Implementation

In order to carry out the extraction the whole process of finding the average spaces in each line is repeated for both the original and the watermarked image i.e.

first the horizontal profile of the image is calculated, next the vertical profile of the image is calculated, then the vertical start and end of a line is determined, then the height of a line is calculated, then the number of lines is calculated the horizontal start and end of a line is found, then calculation of the length of a line is done, then the start & end of each word in a line and the number of spaces is calculated and finally the average space of each line is calculated

Thus if $Y(n)$ represents the average spaces in the watermarked image and $X(n)$ represents the average spaces in the original image From this the watermark Wn is detected as shown by Equation 4-8.

Figure 4-1 shows a reconstructed profile of Sa and detected watermarking information of the text of Figure 2-2 after it has been subjected to private watermarking.

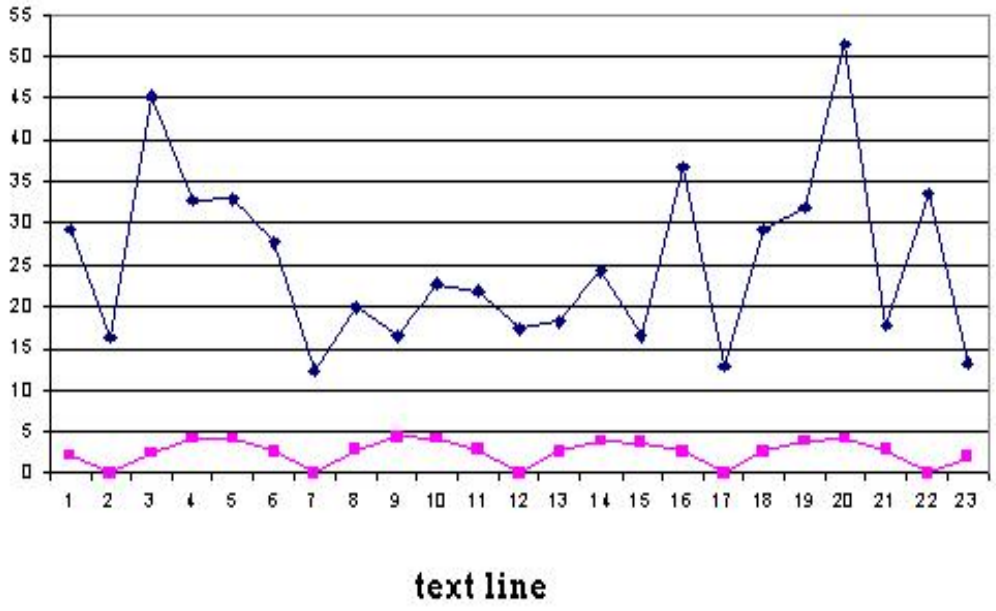

Figure 4-1 Reconstructed Profile of Average Space & Detected Watermark

The top line shows the reconstructed average space and the bottom line shows the watermark information extracted from each text line.

4.4.3 Public Method

When a text has been watermarked using the public method, and supposing the watermarking key is known, thus the set *Sw* of the text lines is reconstructed and the mean of the average spaces a_2 is calculated with Equation 4-9

$$a_2 = \frac{\sum_{m=u}^{v} S_{am}}{v - u + 1}, 0 \le u < v < N \qquad \textit{(Equation 4-9)}$$

, and the watermark component W_m is obtained from Equation 4-10

$$W_m = \mathbf{Y}(m) - a_2, \quad \textit{for text lines in Sw} \qquad \textit{(Equation4-10)}$$

4.5 Extracting Information from the Watermark

Next, the originally marked phase information can be detected by calculating the cross-correlation of a detecting sine wave with Wn or Wm as shown in Equation 4-11.

$$r(j) = \frac{1}{T}\sum_{n=0}^{T-1} W(n) A_d \sin(\omega_d n + j) \qquad \textit{(Equation 4-11)}$$

where W represents *Wn* or *Wm*; ω_d is the radian frequency of the detecting sine wave; and j represents a lag in the number of text lines and varies so as to detect the marked phase information. Through the j that produces an extreme value of $r(j)$, the original marked phase information can be recovered. A_d is the

amplitude of the detecting sine wave. T is the summation number, which depends on the number of items in Wn or Wm as well as ω_d [11].

4.5.1 Implementation

In order to extract the information, detecting sine waves are generated and then cross-correlated with Wn or Wm for each line.

If PSK-16 is used to embed the insert the information in the watermark then 16 unique waves are generated by varying i in Equation 4-12

$$W_n = C_1 a_1 \sin(\omega_1 (n - p) + i \phi_1 / 18)$$

(Equation 4-12)

These waves are then compared with the embedded watermark Wn or Wm and the index number of the wave which has minimum difference with each value of Wn or Wm is noted for that line. This index value is then referenced with Table 4-2 to get the corresponding information that was embedded in the watermark.

CHAPTER 5

TEXT WATERMARK SECURITY, ATTACKS AND THEIR REMEDIES

5.1 Introduction

In practice, a watermarked object may be altered either on purpose or accidentally, so the watermarking system should still be able to detect and extract the watermark. Obviously, the distortions are limited to those that do not produce excessive degradations, since otherwise the transformed object would be unusable.

5.2 Watermark Robustness

Robustness against attacks is a major watermarking requirement. Absolute robustness against all possible attacks and their combinations may be impossible to achieve. Thus, the practical requirement is that a successful attack must impair the host data to the point of significantly reducing its commercial value before the watermark is impaired so much that it cannot be recovered. In fact, with appropriate design, fairly high robustness can be achieved, but it should be pointed out that robustness always has to be traded against watermark data rate and imperceptibility, and the optimum tradeoff depends on the application.

5.3 Levels of Required Robustness

The requirements and the design constraints for text watermarking technologies strongly depend on the final application. For obvious reasons there is no "universal" watermarking method. Although watermarking methods have to be robust in general, different levels of required robustness can be identified depending on the specific application-driven requirements.

In authentication applications, the watermarks have to resist only to certain attacks. Among all possible watermarking applications, authentication watermarks require the lowest level of robustness. The purpose of such watermarks is to authenticate the data content. For example, data can be watermarked such that the watermark can accommodate lossy compression, but they are destroyed as soon as the data are manipulated in a different way.

Applications such as data monitoring and tracking require a higher level of robustness. The main purpose is to detect or identify stored or transmitted data. Examples are automatic monitoring of radio broadcast for billing purposes or identification of images on the World Wide Web with the help of web crawlers.

In fingerprinting applications, watermarks are embedded that identify the recipient of each individual distributed copy. The purpose is to have a means to trace back pirated copies to the recipient who pirated it. Fingerprinting applications require a very high level of robustness against data processing and malicious attacks.

Watermarking for copyright protection is used to resolve rightful ownership and requires the highest level of robustness. *There is a tradeoff between watermark robustness on one hand and watermark imperceptibility and watermark data rate on the other hand* as shown in Figure 5-1.

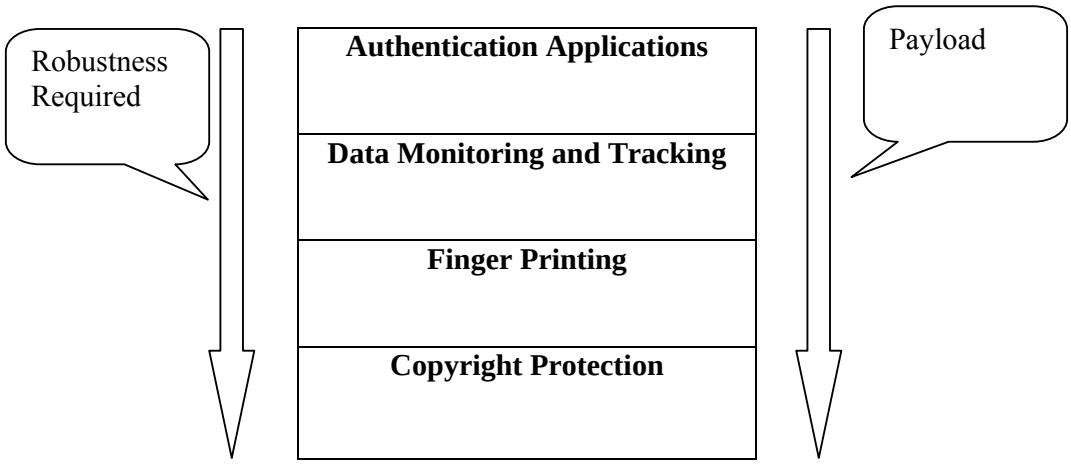

Figure 5-1 Tradeoff between Robustness and Imperceptibility

5.4 Classification of Attacks

Four different types of attacks can be identified.

5.4.1 Simple attacks

Also known as "noise attacks" are conceptually simple attacks that attempt to impair the embedded watermark by manipulations of the whole watermarked data (host data plus watermark) without an attempt to identify and isolate the watermark.

Examples include linear and general nonlinear filtering, waveform-based compression (JPEG, MPEG), addition of noise, quantization in the pixel domain, conversion to analog, and gamma correction.

5.4.2 Detection-disabling attacks

Also known as "synchronization attacks" are attacks that attempt to break the correlation and to make the recovery of the watermark impossible or infeasible for a watermark detector. The methods used are geometric distortion like zooming, shift in spatial or temporal (for video) direction, rotation, shear, cropping, pixel permutations, sub-sampling, removal or insertion of pixels or pixel clusters.

5.4.3 Ambiguity attacks

Also known as "deadlock attacks" or "inversion attacks" or "fake-watermark attacks", are attacks that attempt to confuse by producing fake original data or fake watermarked data.

An example is an inversion attack that attempts to discredit the authority of the watermark by embedding one or several additional watermarks such that it is unclear which was the first, authoritative watermark.

5.4.4 Removal attacks

These are attacks that attempt to analyze the watermarked data, estimate the watermark or the host data, separate the watermarked data into host data and watermark, and discard only the watermark. Examples are collusion attacks, de-noising, certain nonlinear filter operations, compression attacks using synthetic modeling of the image (e.g., using texture models or 3-D models). Also included in this group are attacks that are tailored to a specific watermarking scheme and combat it by exploiting conceptual cryptographic weaknesses of the scheme that make it vulnerable to a specific attack.

The transitions between the groups are sometimes fuzzy and that some attacks do not clearly belong to one group. Collusion attacks could be argued to be a group of its own, since they require, unlike the other attacks, more than one differently watermarked copy of the data. However, since they attempt to reconstruct the un-watermarked original host data, and thus remove the watermark(s), the classification as a "removal attack" holds. Remedies are given that make watermarks more robust against malicious attacks.

5.5 Remedies against Waveform-Based Attacks

As already mentioned, noise-like distortions, for example, due to lossy compression, result in a distorted watermark signal in the watermark recovery or verification process.

In the case of text documents the waveform-based attacks are not efficient in impairing the watermark because of the binary nature, block/line/word patterning, clear separation between foreground and background areas of text images. For the types of attacks that are a threat to text images there are two main remedies:

5.5.1 Increasing the embedding strength

Increasing the embedding strength is straightforward and efficient in many cases, especially if appropriate masking according to the properties of human perception is used to determine the maximum allowable embedding strength.

5.5.2 Applying redundant embedding

Redundant embedding can be performed in many ways. In the spatial domain it might consist of embedding a watermark many times and then taking a majority vote in the recovery process

5.6 Geometrical Distortions and Remedies

Watermarks are typically most vulnerable to geometrical distortions. The reason is that, for most proposed watermarking methods, the watermark detector has to know the exact position of the embedded watermark. Geometrical distortions tend to destroy the synchronization such that watermark embedding and watermark detection are misaligned and do not fit anymore.

The Text Watermarking algorithm has to be explicitly designed to withstand geometric attacks such as affine transforms, clipping, and cropping. In case of both Private and Public Watermarking Schemes developed, all images statistics are recalculated for the Watermarked image. The algorithm that calculates the image statistics of the Watermarked image is modified to cater for the challenge of finding the original watermark reference within the host data.

Remedies against the jittering attack which has been proven to be very efficient in removing watermarks for many algorithms are inherited in the basic watermarking scheme which uses average and rounded off values to represent inter-word spaces instead of absolute values or pixel positions in the document

5.7 Remedies Against Watermark Ambiguities

To resolve rightful ownership, it must be possible to determine the authoritative watermark in case several watermarks are present in a data set.

5.7.1 Timestamps

To determine who first signed a set of data, timestamps should be used. The timestamps should be provided by third parties. This is the best solution to resolve multiple watermark conflicts in text documents. The party which has an earlier date embedded through the watermark wins the case.

5.7.2 Noninvertible Watermarks

Certain watermarking methods are invertible and allow reverse engineering to produce a counterfeit original. This scenario creates an ownership deadlock because both the rightful owner as well as the attacker can show that their watermark is presents in the signed data and counterfeit original. Hence it is not possible to resolve rightful ownership since all claims from both parties are legally speaking equivalent.

The devised text watermark algorithm makes watermarks that are noninvertible because the hash table used to encode the phase changes in the sine wave is a one way hash function.

In this case, it is computationally infeasible for an attacker to create a counterfeit original because it depends on the watermark, which in turn depends on the counterfeit original which is not yet existing.

5.8 Robustness Test Utilities and Watermark-Removal Software

Similar to conditional access and copy-prevention mechanisms, the existence of watermarking technology and its potential possibilities have stimulated individuals

to come up with attempts to defeat watermarking. Examples are publicly available tools to test the robustness of image watermarking techniques [1].

5.8.1 Unzign

It is a utility that works for images in JPEG format. In version 1.1, Unzign introduces pixel jittering in combination with a slight image translation. For many proposed watermarking techniques, the embedded watermarks are efficiently destroyed. However, besides removing the watermark, Unzign version 1.1 introduces severe artifacts. An improved version 1.2 has been released. Although the artifacts were decreased, its watermark destruction capability decreased as well.

5.8.2 StirMark

It is a simple generic tool to test the robustness of image watermarking techniques. It simulates re-sampling to emulate a printing–scanning procedure and applies minor geometric distortions (stretching, shearing, shifting, and rotation) followed by resampling and bilinear or Nyquist interpolation. In addition, small and smoothly distributed errors are introduced into all sample values. Applying StirMark only once introduces a practically unnoticeable quality loss in the image. The author claims that his tool removes all current watermarks.

CHAPTER 6

TESTS AND ANALYSIS

According to [15], the functionalities of watermarking that have to be tested and evaluated to different levels are imperceptibility, reliability (robustness), capacity and speed.

6.1 Perceptibility

A major requirement of watermarking is imperceptibility. The text algorithm developed in this project is scaled as being of moderate-high imperceptibility by the *Strimark Benchmark* watermark evaluation software.

The reason for this high imperceptibility is that the algorithm works by modifying the inter-word spaces, the image modifications are imperceptible except for the case in which the original is compared with the watermarked image side by side. This reveals that certain spaces are modified but gives no insight whatsoever about the methodology behind the space modification

Table 6-1 shows perceptibility levels. Figure 6-2 shows the original text image (6-2a), the image watermarked by the public method (6-2b) and the image watermarked by the private method 6-2c).

Table 6-1 Summary of the Possible Perceptibility Assurance Levels

Level of assurance	Criteria
Low	-PSNR(when applicable) - Slightly perceptible but not annoying
Moderate	- Metric based on perceptual model - Not perceptible under domestic conditions, that is using mass market consumer equipment
Moderate high	Not perceptible in comparison with original under studio conditions
High	Evaluation by a large panel of persons under conditions

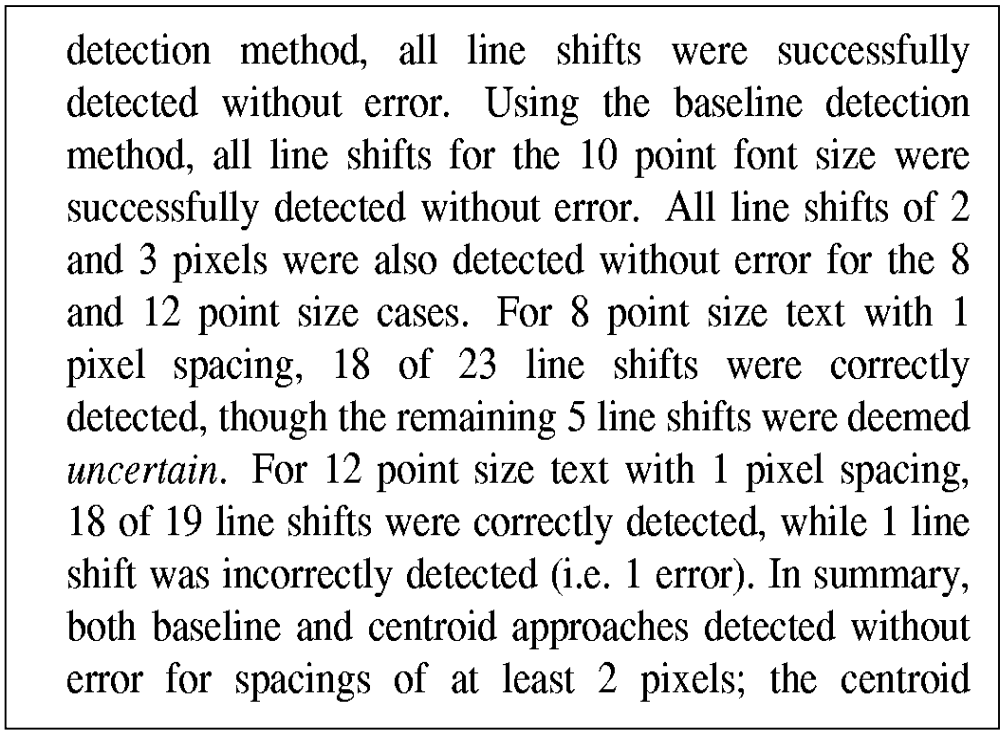
detection method, all line shifts were successfully detected without error. Using the baseline detection method, all line shifts for the 10 point font size were successfully detected without error. All line shifts of 2 and 3 pixels were also detected without error for the 8 and 12 point size cases. For 8 point size text with 1 pixel spacing, 18 of 23 line shifts were correctly detected, though the remaining 5 line shifts were deemed *uncertain*. For 12 point size text with 1 pixel spacing, 18 of 19 line shifts were correctly detected, while 1 line shift was incorrectly detected (i.e. 1 error). In summary, both baseline and centroid approaches detected without error for spacings of at least 2 pixels; the centroid

Figure 6-1a Original Text Image

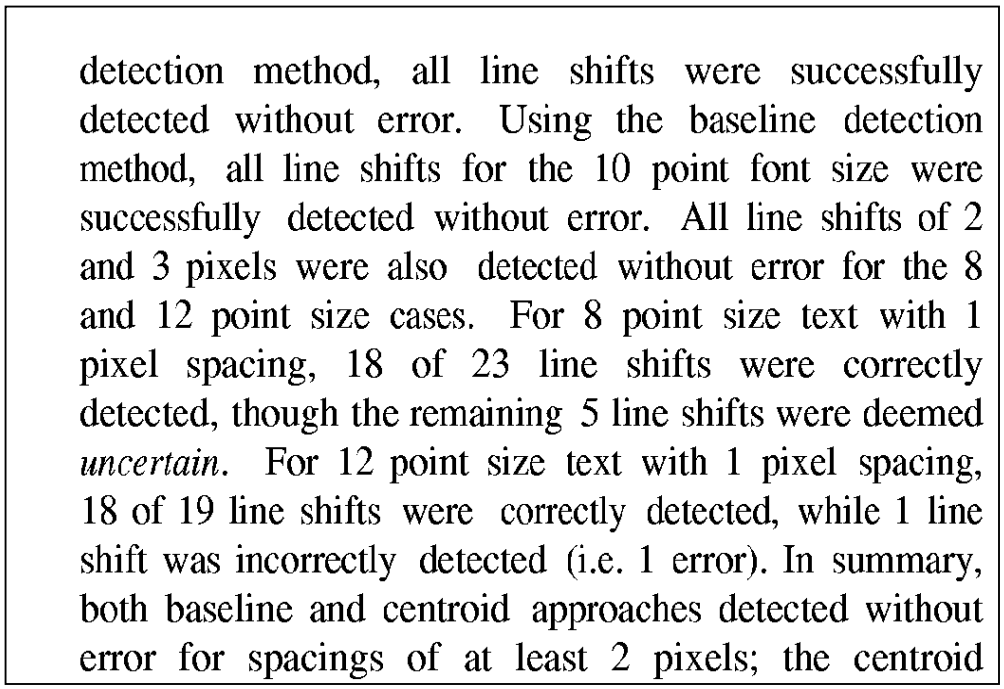
detection method, all line shifts were successfully detected without error. Using the baseline detection method, all line shifts for the 10 point font size were successfully detected without error. All line shifts of 2 and 3 pixels were also detected without error for the 8 and 12 point size cases. For 8 point size text with 1 pixel spacing, 18 of 23 line shifts were correctly detected, though the remaining 5 line shifts were deemed *uncertain*. For 12 point size text with 1 pixel spacing, 18 of 19 line shifts were correctly detected, while 1 line shift was incorrectly detected (i.e. 1 error). In summary, both baseline and centroid approaches detected without error for spacings of at least 2 pixels; the centroid

Figure 6-1b Text Image Watermarked by Private Method

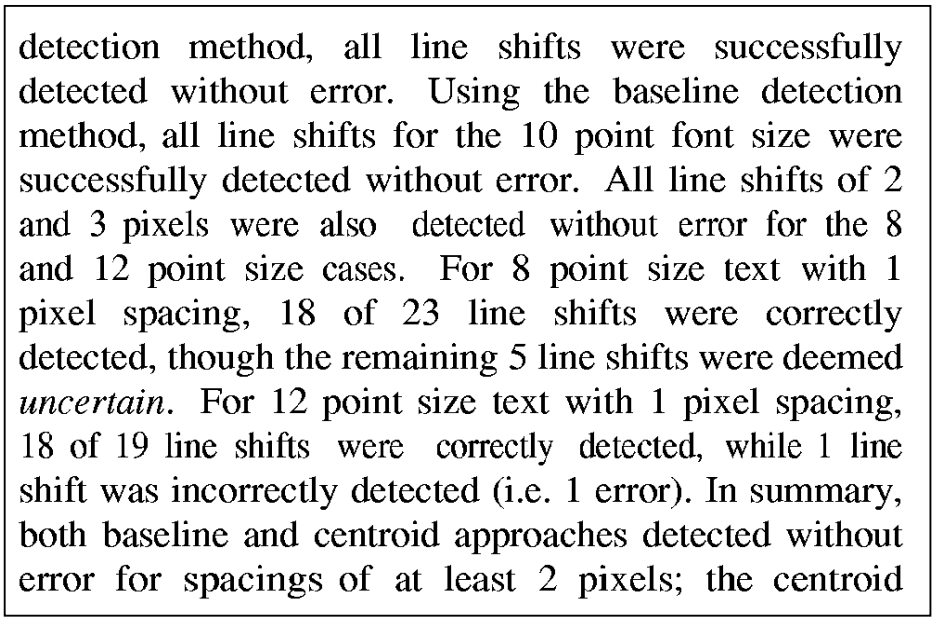
detection method, all line shifts were successfully detected without error. Using the baseline detection method, all line shifts for the 10 point font size were successfully detected without error. All line shifts of 2 and 3 pixels were also detected without error for the 8 and 12 point size cases. For 8 point size text with 1 pixel spacing, 18 of 23 line shifts were correctly detected, though the remaining 5 line shifts were deemed *uncertain*. For 12 point size text with 1 pixel spacing, 18 of 19 line shifts were correctly detected, while 1 line shift was incorrectly detected (i.e. 1 error). In summary, both baseline and centroid approaches detected without error for spacings of at least 2 pixels; the centroid

Figure 6-1c Text Image Watermarked by Public Method Using Key 9

6.2 Reliability/Robustness

Although robustness and capacity are linked in the sense that there is a tradeoff between capacity and robustness, still it is possible to evaluate them separately. Watermarking schemes are defined for a particular application and each application only requires a certain fixed payload so the concern is only with the robustness of the scheme for this given payload.

The robustness can be assessed by measuring the detection probability of the mark and the bit error rate after samples of watermarked images are subjected to different attacks.

6.2.1 Simple attacks

These attacks attempt to impair the embedded watermark by manipulations of the whole watermarked data (host data plus watermark) without an attempt to identify and isolate the watermark.

Waveform base compression and contrast enhancement attacks are not applicable to text images because text images have only two colors black and white.

Another form of simple attacks is of noise addition such as skewing. Test results with skewed images have revealed that it is possible to detect the watermark information from skewed images with skewing angle less than 7°. These tests are summarized in Table 6-2.

Table 6-2 Test Results of Watermark Detection for Different Skewing Angles

	Skewing angles				
	1 – 2 °	**3 – 4 °**	**5 – 6 °**	**7 °**	**8°**
Detection	100%	98%	95%	90%	85%

The reason for resistance of the algorithm to this kind of attacks is because the algorithm is not base on word shift or line shift modifications; rather it embeds the watermark by modifying the inter-word spaces. Figure 6-2 shows a text image on which skewing has been performed to 7°.

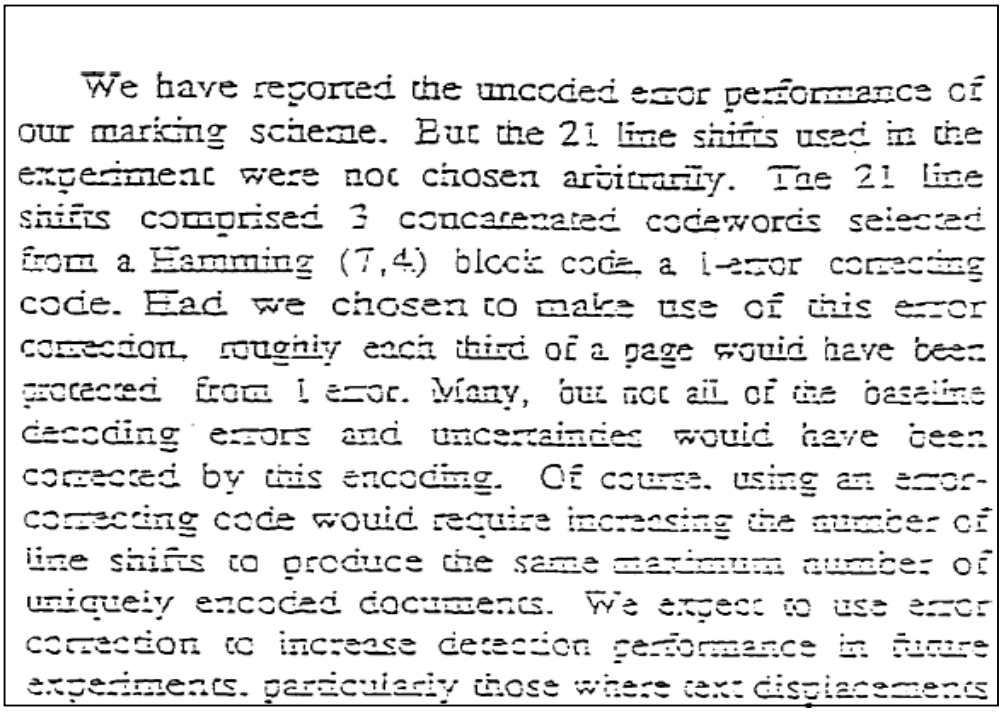
We have reported the uncoded error performance of our marking scheme. But the 21 line shifts used in the experiment were not chosen arbitrarily. The 21 line shifts comprised 3 concatenated codewords selected from a Hamming (7,4) block code, a 1-error correcting code. Had we chosen to make use of this error correction, roughly each third of a page would have been protected from 1 error. Many, but not all of the baseline decoding errors and uncertainties would have been corrected by this encoding. Of course, using an error-correcting code would require increasing the number of line shifts to produce the same maximum number of uniquely encoded documents. We expect to use error correction to increase detection performance in future experiments, particularly those where text displacements

Figure 6-2 A Text Image Skewed by 7°

6.2.2 Detection disabling attacks

These attacks attempt to break the correlation and to make the recovery of the watermark impossible or infeasible e.g. zooming, cropping, rotation.

Zooming the image is an unsuccessful attack when applied to the watermarking algorithm developed in this project because the algorithm works on average inter-word spaces in each line. Zooming the image maintains the same ratio of the spaces between the words and the number of words. Results are given in Table 6-3.

Table 6-3 Effects on Detection Due to Zooming

	Zooming factor			
	1.5x	**2x**	**5x**	**10x**
Detection	100%	100%	100%	100%

To resist rotation attacks the image detection algorithm is slightly modified. Instead of considering an inter-word space to exist only if the vertical profile of the line gives zero value, a line with value slightly more than zero is still accepted as an inter-word space. Figure 6-3 shows an image rotated by 15°.

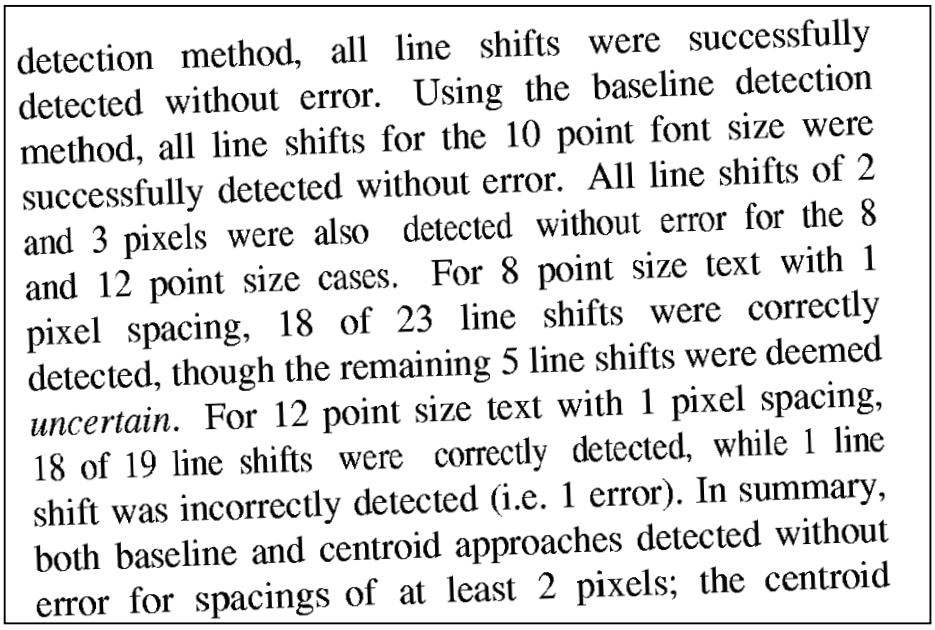
detection method, all line shifts were successfully detected without error. Using the baseline detection method, all line shifts for the 10 point font size were successfully detected without error. All line shifts of 2 and 3 pixels were also detected without error for the 8 and 12 point size cases. For 8 point size text with 1 pixel spacing, 18 of 23 line shifts were correctly detected, though the remaining 5 line shifts were deemed *uncertain*. For 12 point size text with 1 pixel spacing, 18 of 19 line shifts were correctly detected, while 1 line shift was incorrectly detected (i.e. 1 error). In summary, both baseline and centroid approaches detected without error for spacings of at least 2 pixels; the centroid

Figure 6-3 Publicly Marked Text Image Rotated by 15°

But rotation attacks only rotate the image by 1-2° so that the rotation is imperceptible. Watermark detection accuracy for images rotated to certain angles is summarized in Table 6-4

Table 6-4 Effect on Detection Due to Rotation of Text Images

	Rotation angles				
	1 – 2 °	**3 – 4 °**	**5 – 6 °**	**7 °**	**8°**
Detection	100%	94%	79%	60%	50%

When images are rotated by angles greater than 3° the attack becomes perceptible and is rendered useless

6.2.3 Ambiguity attacks

These attacks attempt to confuse by producing fake original data or fake watermarked data. An example is an inversion attack that attempts to discredit the authority of the watermark by embedding one or several additional watermarks such that it is unclear which was the first, authoritative watermark.

The algorithm developed in this project is non-invertible because it uses hash table to index values for the phase angle of the sine wave which is to be embedded across different lines. Thus the algorithm **cannot be reverse engineered**.

Another form of ambiguity attack is embedding **multiple watermarks**. These attacks can be avoided by time-stamping the watermark embedded. Thus in case of multiple marks, the time stamp would reveal the true data.

6.3 Capacity

A major constraint in text watermarking is the little amount of payload available as compared to normal image watermarking. The challenge is to embed more bits imperceptibly and robustly.

The algorithm developed in this project embeds information through a sine wave whose sampling values are represented by the inter-word spaces of a line. The number of possible sine waves or in other words the number of possible phases of the sine wave depends upon the modulation technique used. A summary of the number of bits and the phase angle difference for various versions of the PSK are shown in Table 6-5.

Table 6-5 Capacity of PSK Schemes

Modulation Technique	Successive angle change	No. of bits	No of combinations
PSK-2	360 / 2 = 180	1	2
PSK-4	360 / 4 = 90	2	4
PSK-8	360 / 8 = 45	3	8
PSK-16	360/ 16 = 22.5	4	16
PSK-32	360/ 32 = 11.25	5	32
PSK-64	360/ 64 = 5.625	6	64

But as the modulating level is increased e.g. from PSK-16 to PSK- 32, there is a tradeoff in detection accuracy. Experiments have revealed that for text image of font size 10-14 the images can be detected accurately if a maximum of PSK-16 is used.

6.4 Speed

Speed is very dependent on the type of implementation: software or hardware. Complexity is an important criteria and some application impose a limitation. For a software implementation is also depends very much on the hardware used to run it but comparing performance result obtained on the same platform (usually the typical platform of end users) provide a reliable measure.

In case of the watermarking scheme developed in this project, the speed of algorithm depends upon the image resolution, image format, the modulation scheme used (PSK-8, PSK -16 etc.), the number of bits to be encoded (due to indexing in the hash table), the public key (in case of public watermarking).

Tests on the developed watermarking scheme with existing watermarking techniques is shown for a text image of resolution 1072 x 1296, format PNG, have 26 lines with maximum of 12 words and a minimum of 7 words, in the graph of Figure 6-4

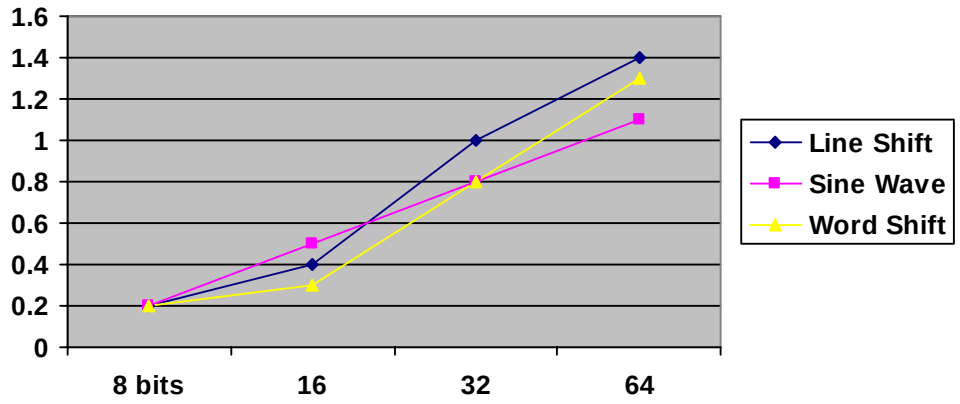

Figure 6-4 Comparison of Watermark Embedding Schemes

It can be seen from Figure 6-4 that the developed sine wave algorithm is more efficient when more number of bits are encoded.

Figure 6-5 shows a comparison of the developed scheme with existing algorithms performed on a text image of resolution 1072 x 1296, format PNG, have 26 lines with maximum of 12 words and a minimum of 7 words,

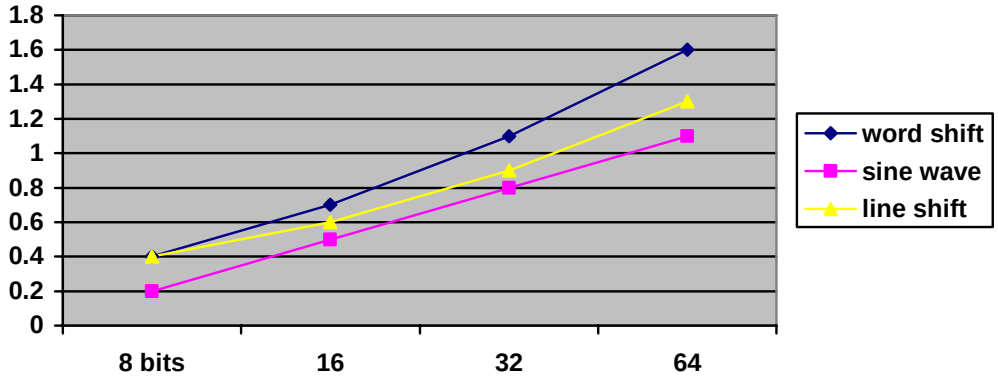

Figure 6-5 Comparison of Watermark Detecting Schemes

The graph in Figure 6-5 shows that the time taken for extraction in the case of the developed sine wave algorithm is the same as that taken for embedding and is thus much more efficient than that of the line shift and word shift algorithm. The reason is that the detection procedure in case of the developed algorithm is an inverse of the embedding algorithm. It does not need to analyze all the features of the watermarked image and compare it with the original.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

In this project, a new text watermarking method has been developed after extensive study of existing text watermarking techniques keeping in view their advantages, disadvantages as well as their efficiency. Both **private** (*requires the original*) and **public** (*does not require original*) watermarking algorithms have been developed and implemented successfully.

The watermarking method developed, slightly modifies inter-word spaces so that different lines across a text act as sampling points of a sine wave consequently the variations from one line to the next remain imperceptible.

The watermarking scheme is inherently robust and survives common attacks. The watermark embedded by this scheme remains in the document even in **hard form** and is subsequently retrievable even after the document has been printed, photocopied multiple times and rescanned. The marks are imperceptible and are irremovable because the document marking scheme uses hash tables and secret keys for embedding the watermark.

In addition to embedding simple watermarks, a method of adding information through the watermark has also been developed and implemented. This method uses the PSK-16 (*phase modulation technique*) to embed information to the sine wave which is subsequently sampled by the inter-word spaces of the text lines. PSK -8, PSK -32, PSK -64 techniques have also been experimented for embedding information with the result that the more the number of bits used for phase modulation the greater is the payload but the tradeoff is error-free extraction.

A document watermarking system has been successfully developed as an application of the watermarking scheme and highlights one of its many possible applications i.e. **fingerprinting**. It provides a procedure to discourage illicit

redistribution of documents by marking each document copy uniquely, using the information of the intended recipient, so that the original recipient can be identified from an illicit or illegally distributed copy.

The devised scheme is inherently suitable for many applications of watermarking such as copyright protection, fingerprinting (as demonstrated by the document watermarking system), data monitoring and tracking and content authentication.

7.2 Future Work

The watermarking scheme developed in this project is highly robust whereas there is another class of watermarks called **fragile watermarks** which have different applications. Work in this direction using the devised watermark scheme can be carried out to further enhance the applicability of the scheme.

Another extension to the watermarking scheme is to combine it with different watermarking methods that modify other characteristics of the text document such as line spaces, inter character spaces etc. This will qualify for the **hybrid watermarks** class.

The scheme can also be enhanced so that it enables the embedding of **multiple watermarks** that further increase the robustness.

In addition to **PSK** modulation schemes, **ASK** (amplitude shift keying) and **FSK** (frequency shift keying) modulation techniques can also be used to add information to the watermark. In the ASK schemes the amplitude of the sine wave will be modified according to the number of possible bit combinations and similarly the frequency in the FSK scheme.

In the project a document marking system has been developed that marks multiple copies of a document uniquely with the information of the intended recipient. It can be made into a full-fledged application which also distributes the document to the intended recipient through the network or internet. Another suggested scheme is to enable document marking at the recipient end to lessen the processing load on the

server. In this case a document may be sent to the recipient in encrypted form and subsequently the software at the recipient end would mark it by using the private key of the recipient and convert it into a bitmap.

REFERENCES

[1] Frank Hartung and Martin Kutter, "Multimedia Watermarking Techniques" Proceedings of the IEEE, VOL. 87, NO. 7, JULY 1999

[2] Dr. Mohammed Al-Mualla and Prof. Hussain Al-Ahmad "Information Hiding: Steganography and Watermarking"

[3] Fabien A. P. Petitcolas, Ross J. Anderson and Markus G. Kuhn, "Information Hiding-A Survey", Proceedings of the IEEE, special issue on protection of multimedia content, 87(7):1062{1078, July 1999.

[4] Bernd Girod, Frank Hartung and Jonathan Su, "Digital Watermarking of Text, Image and Video Documents", CGI 98 Hannover June 1998

[5] Fernando P´erez-Gonz´alez and Juan R. Hern´andez "A Tutorial on Digital Watermarking", Work partially funded by CICYT under project TIC-96-0500-C10-10

[6] Allan M. Bruce, "A Review of Digital Watermarking ", Department of Engineering, University of Aberdeen, November 2nd 2001

[7] J. Brassil, S. Low, N. Maxemchuk, and L. O'Gorman, "Electronic marking and identification techniques to discourage document copying," *IEEE J. Select. Areas Commun.*, vol. 13, Oct. 1995.

[8] Young-Won Kim, Kyung-Ae Moon, and Il-Seok Oh, "A Text Watermarking Algorithm based on Word Classification and Inter-wordSpace Statistics", Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003) 0-7695-1960-1/03 $17.00 © 2003 IEEE

[9] Talisman. [Online]. Available WWW: http://www.cordis.lu/esprit/src/talisman.htm

[10] P. Jones. Octalis. [Online]: Available: http://www.cordis.lu/esprit/src/octalis.htm

[11] Hiroshi Nakagawa, Tsutomu Matsumoto and Ichiro Murase, "Information Hiding for Text by Paraphrasing", IT center. University of Tokyo

[12] S. Low, Maxemchuk, N.F., Bassil, J. and O'Gorman, L., 1995. "Document Marking and
Identification using Both Line and Word Shifting", Procedings of Infocom 95

[13] Young-Won Kim and Il-Seok Oh, "A survey on text watermarking techniques," Proc. of Honam-Jeju Korea Information Science Society, August 2002.

[14] J. Brassil and L. O'Gorman, "Watermarking document images with bounding box expansion," in Procedings of 1st International Information Hiding Workshop,pp. 227–235.

[15] Fabien A. P. Petitcolas, Microsoft Research, "Watermarking schemes evaluation", Proceedings of the IEEE, special issue on protection of multimedia content (June 2000)