2004

Data hiding in images based on fractal modulation and diversity combining

Sarat B. Atluri

Follow this and additional works at: http://scholarworks.rit.edu/theses

Recommended Citation
DATA HIDING IN IMAGES BASED ON FRACTAL MODULATION AND DIVERSITY COMBINING

By

SARAT ATLURI

Thesis submitted to the faculty of Rochester Institute of Technology in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

IN

ELECTRICAL ENGINEERING

Approved by

Thesis Advisor
Dr. Raghuveer Rao

Raghuveer Rao

Thesis Committee
Dr. Ferat Sahin

Ferat Sahin

Thesis Committee
Dr. Daniel Phillips

Daniel Phillips

Department Head
Dr. Robert Bowman

Robert Bowman

DEPARTMENT OF ELECTRICAL ENGINEERING, COLLEGE OF ENGINEERING

ROCHESTER INSTITUTE OF TECHNOLOGY, ROCHESTER, NEW YORK

FEB 2004
I, Sarat Atluri, hereby grant the permission to the Wallace Library of the Rochester Institute of Technology to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

Date: 05/24/04

Author Sarat B. Atluri
ACKNOWLEDGEMENTS

My advisor, Prof Raghuveer Rao, has been both critical and supportive. I have developed a fine research methodology under his guidance. I am very thankful to him for providing me such a rich experience.

My sincere thanks to Prof Sohail Dianat, for the numerous interesting discussions, we had. We had fruitful discussions that led to the derivation of minimum energy reconstruction of images, from projections.

Dr. Mathew was very supportive during the initial phases of research. We have done rigorous literature surveys and Matlab implementations. Thank you, Prof Mathew for the efforts towards a better understanding of several watermarking algorithms and the ownership deadlock problem.

Dr. Unnikrishnan was helpful throughout my stay at RIT, particularly with my career plans.

I thank Dr. Phillips and Dr. Sahin for being on my thesis committee and for meticulously reviewing the initial draft.

I thank Ken Snyder, facilities manager for promptly, providing the resources, be it an Ethernet cable or the Matlab version6 upgrade.

Finally, I am highly grateful to my parents for their ever-lasting love and support. I am highly indebted to them for what I am today.
ABSTRACT

The current work provides a new data-embedding infrastructure based on fractal modulation. The embedding problem is tackled from a communications point of view. The data to be embedded becomes the signal to be transmitted through a watermark channel. The channel could be the image itself or some manipulation of the image. The image self noise and noise due to attacks are the two sources of noise in this paradigm. At the receiver, the image self noise has to be suppressed, while noise due to the attacks may sometimes be predicted and inverted. The concepts of fractal modulation and deterministic self-similar signals are extended to 2-dimensional images. These novel techniques are used to build a deterministic bi-homogenous watermark signal that embodies the binary data to be embedded. The binary data to be embedded, is repeated and scaled with different amplitudes at each level and is used as the wavelet decomposition pyramid. The binary data is appended with special marking data, which is used during demodulation, to identify and correct unreliable or distorted blocks of wavelet coefficients. This specially constructed pyramid is inverted using the inverse discrete wavelet transform to obtain the self-similar watermark signal. In the data embedding stage, the well-established linear additive technique is used to add the watermark signal to the cover image, to generate the watermarked (stego) image.

Data extraction from a potential stego image is done using diversity combining. Neither the original image nor the original binary sequence (or watermark signal) is required during the extraction. A prediction of the original image is obtained using a cross-shaped window and is used to suppress the image self noise in the potential stego image. The resulting signal is then decomposed using the discrete wavelet transform. The number of levels and the wavelet used are the same as those used in the watermark signal generation stage. A thresholding process similar to wavelet de-noising is used to identify whether a particular coefficient is reliable or not. A decision is made as to whether a block is reliable or not based on the marking data present in each block and sometimes corrections are applied to the blocks. Finally the selected blocks are combined based on the diversity combining strategy to extract the embedded binary data.
CONTENTS

List of figures

Chapter 1: Introduction ........................................................................................................1
  1.1 The information hiding problem .................................................................................1
  1.2 Investigated approach ..............................................................................................2
  1.3 Thesis organization .................................................................................................4
  1.4 Main contributions ..................................................................................................5

Chapter 2: Information hiding – background and overview ..............................................6
  2.1 Evolution of information hiding .............................................................................6
    2.1.1 Brief history of information hiding .................................................................6
    2.1.2 Recent developments and applications .........................................................7
  2.2 Terminology ...........................................................................................................9
  2.3 Generic information hiding system .......................................................................11
    2.3.1 Definition .......................................................................................................11
    2.3.2 Algorithm requirements and design issues ..................................................13
  2.4 Information hiding – current state of the art .........................................................15
    2.4.1 Introduction ....................................................................................................15
    2.4.2 Spatial domain watermarking .......................................................................15
    2.4.3 Fractal based watermarking ...........................................................................18
    2.4.4 Discrete cosine transform (DCT) domain watermarking ...............................19
    2.4.5 Frequency domain watermarking ..................................................................23
    2.4.6 Wavelet domain watermarking ......................................................................26

Chapter 3: Selected background concepts ......................................................................28
  3.1 Deterministically self-similar signals ....................................................................28
    3.1.1 Energy-dominated bi-homogeneous signals ....................................................30
    3.1.2 Power-dominated bi-homogeneous signals ......................................................35
    3.1.3 Discrete-time algorithms for bi-homogeneous signals ....................................36
LIST OF FIGURES

1.1 Communications approach to data embedding and extraction ..........................3
2.1 Generic data-embedding scheme ...................................................................12
2.2 Generic data-extraction scheme ....................................................................12
3.1 The self-similar basis functions $\theta_0^n(t)$ to $\theta_1^n(t)$ of an orthonormal basis for $E^H$, $H=0$ .................................................................32
3.2 The signal $x(t)$ constructed using equation (3.9a) and the orthonormal basis functions shown in figure 3.1 .................................................................33
3.3 The self-similar basis functions $\theta_0^n(t)$ to $\theta_1^n(t)$ of an orthonormal basis for $E^H$, $H=0$, using the haar wavelet ...........................................34
3.4 The signal $x(t)$ constructed using equation (3.9a) and the orthonormal basis functions shown in figure 3.3 .................................................................35
3.5. The time-frequency portrait of a bi-homogeneous signal of degree $H = -1/2$ ......37
3.6 A communication system for transmitting a continuous or discrete-amplitude data sequence $q[n]$ over a noisy and unreliable continuous-amplitude, continuous-time channel. .........................................................................................38
3.7. The channel model for a typical communications scenario .........................39
3.8. A portion of the time-frequency portrait of the transmitted signal for fractal modulation of a finite-length data vector $q$. In this case $H = -1/2$ ..............43
3.9. A sample 2-d generating sequence ..............................................................47
3.10. The 3 level space-scale portrait constructed using the 2-d generating sequence, $\beta = 1/4$ .................................................................................................48
3.11. The self similar image obtained by using the inverse discrete wavelet transform algorithm on the coefficients in Fig. 3.11, wavelet used is haar. ......................49
4.1: The self-similar basis functions $\theta_0^n(t)$ to $\theta_1^n(t)$ using a haar wavelet...........51
4.2: The corresponding self-similar signal ..........................................................52
4.3: The self-similar basis functions $\theta_0^n(t)$ to $\theta_1^n(t)$ using a daubechies 5 wavelet.53
4.4: The corresponding self-similar signal ..........................................................54
4.5: Self-similar signal constructed using the dwt algorithm.................................55
4.6 Bit-error rate curves at various SNRs ( -35:2.5:20 ) and wavelet levels............56
4.7 Bit-error rate curves at various SNRs ( -100:5:20 ) and wavelet levels..............57
5.1 Bunit - the 30x30 matrix of binary data, the goal is to embed this data into a cover image and then recover as much of it as possible at the receiver end.................................60
5.2 An example cover image (size: 240x240).......................................................61
5.3 The binary stealth data (bunits) are arranged as coefficients to form the wavelet decomposition pyramid. This will be used to synthesize a self-similar image that would serve as the watermark signal.................................................................62
5.4 Using the 2-d dwt algorithm on the pyramid shown in the previous figure, the self-similar (bi-homogeneous) image is constructed, this image serves as the watermark signal that would be imperceptibly dissolved into a cover image.................................63
5.5 The original cover image versus its watermarked version. Note that they are both perceptually identical, however, the error indicates that they are not, numerically and this error accounts for the watermark signal, which holds the embedded binary data........64
5.6 The original cover image versus its linear prediction, into which a fraction of the watermark signal is imperceptibly, dissolved.......................................................65
5.7 A low pass estimate of the original image, based on the watermarked image ......66
5.8 The remnant image after the self-noise has been cancelled out from the watermarked image.................................................................................................67
5.9 Wavelet analysis of the remnant image .........................................................68
5.10 The result of wavelet threshholding ..............................................................69
5.11 The coefficient matrix after intelligent block processing .................................70
5.12 The extracted bunit .........................................................................................71
5.13: A cropped watermarked image .................................................................73
5.14: The extracted binary data (bunit). BER = 0.1556........................................73
5.15: The blurred watermarked image.................................................................74
5.16: The extracted binary data (bunit), BER = 0.1022........................................74
5.17: The translated watermarked image.............................................................75
5.18: The extracted binary data (bunit), BER = 0.0122 .......................................75
5.19: Noise added to the watermarked image ....................................................76
5.20: The extracted binary data (bunit), BER = 0.1167 
5.21: Compressed watermarked image 
5.22: Extracted binary data (bunit), BER = 0.4389 
5.23: The rotated watermarked image 
5.24: Extracted binary data (bunit), BER = 1 
5.25: The zoomed watermarked image 
5.26: Extracted binary data (bunit), BER = 0.4778 
5.27: Sharpened watermarked image 
5.28: Extracted binary data (bunit), BER = 1 
5.29: Stirmark attacked watermarked image. The attack causes a 10-degree rotation. 
5.30: Extracted binary data (bunit), BER = 1 
5.31: Test images 
5.32: Test images continued
CHAPTER 1: INTRODUCTION

1.1 The information hiding problem

Digital media has become immensely popular in the past few decades. The advantages of digital media are well known. Efficient and standardized compression technologies enable cheap storage and transmission. Coupled with the growth of the Internet, these advances made exchange of digital media easier than ever. While all these attributes make digital media attractive, the ability to make exact copies of the original, without any way of distinguishing can lead to theft of content. This is precisely why content developers are hesitant in putting their content on the web. The situation needs a mechanism by which serial numbers or copyright messages can be hidden in digital media. The hidden data can be used to identify and prosecute copyright violators. Data embedding is being investigated as a solution to address the above problem and many others caused by the use of otherwise highly desirable digital media.

Data embedding is accomplished by making data dependent, invisible, (imperceptible) yet robust changes to the host media. Detection or extraction of the hidden data is done by putting together these changes in an intelligent way and making a decision as to whether the extracted bit is a one or a zero.

Although research into information hiding and watermarking started only in the early nineties, it has been growing at a rapid pace. This work aims at developing a new data-embedding infrastructure based on fractal modulation and diversity combining. The field of information hiding and watermarking draws from diverse yet related fields of digital modulation, detection theory, cryptography, perceptual coding and with this work fractal modulation and self-similar signals.
1.2 Investigated approach

Ever since the simplest least significant bit technique [36], there have been plenty of publications reporting much more sophisticated techniques for data embedding. In this work, we investigate methods to use the concept of deterministic self-similar signals to construct the data dependant watermark or modification signal. We look at the problem of information hiding from a communications viewpoint. The situation is similar to communication over a noisy and interfering channel. In our case, the channel is an extraction of the host image itself; the additive noise is the image data and the interference accounts for all the attacks. Attacks are any intentional and unintentional image processing that one may attempt on the watermarked image in order to tamper or completely remove the underlying watermark signal. The extracted data $W'$ is an estimate of the original binary data $W$. It is not the same as $W$ and has bit errors. The bit error rate of this extracted data serves as the performance metric for the algorithm. All these ideas have been clearly sketched in the figure below, although it has lot more details.
The research path taken can be split into three parts. In the first part, we extend the concepts of deterministic self-similar signals to two-dimensional digital images. We use those observations and results to construct the binary stealth, data-dependent, self-similar or fractal signal. With this we have completed the watermark generation stage. The second part deals with imperceptibly adding this watermark signal to the image and detecting and extracting it using diversity combining. In the third part, we have
contemplated using marker or flag bits to predict and invert attacks. We have applied these techniques with limited success in tackling the blurring attack.

1.3 Thesis organization

Chapter 2 begins with some watermarking history, then lists the recent developments and applications of watermarking, followed by a generic description of the watermarking problem and the algorithm requirements, and finally does a literature survey of the current state of the art in area of digital image watermarking.

Chapter 3 introduces self-similar signals and fractal modulation theory. Throughout this thesis, we are concerned with only deterministic self-similar, bi-homogeneous signals. These are signals that satisfy the deterministic scale invariance property for scale values that are integer powers of 2 [43]. The algorithms used for the synthesis of bi-homogenous signals, both the basic one and the one that uses the discrete wavelet transform are explained. The framework for one-dimensional fractal modulation and demodulation is developed. And finally the straightforward extension of the one-dimensional concepts to digital images is developed.

The 1D fractal modulation results are presented in Chapter 4. The performance of the diversity combining strategy for digital data demodulation from a noise corrupted fractal signal is simulated for different sets of parameters.

In Chapter 5, the algorithm for fractal based data embedding and extraction is developed. By using a fractal signal as the watermark or modification signal, we obtain an algorithm that is highly resistant to cropping, under the toughest of constraints imposed. The information-hiding infrastructure developed does not use either the original image or the original watermark signal during extraction. We have a method of identifying useful regions and discarding corrupted regions of the wavelet coefficient pyramid. This strategy together with the diversity combining method has lead to low bit error rates in
the extracted data. The performance of the algorithm is evaluated for different cover images and under different attacks and the results are tabulated.

1.4 Main contributions

1. Review of deterministic self-similar signal theory and simulation of fractal modulation.
2. Simulation and verification of the success of diversity combining strategy at various signal to noise ratios and different sets of parameters.
3. Extension of the one-dimensional concepts to digital images and subsequently, a method for fractal watermark generation. This fractal watermark encapsulates the binary stealth data.
4. Use of image fusion type techniques to dissolve the watermark signal into the host image.
5. Image self-noise suppression by using a predictor, during pre-processing in the extraction stage.
6. Method for identifying useful, information-holding regions of the wavelet coefficient pyramid by using suitable marker or flag bits.
7. Reuse of the same markers to detect and invert any attacks on the marked image, especially in case of blurring.
8. Simulation of different possible attacks on the marked image and investigation of robustness of the current algorithm to them.
9. Design of a binary data embedding system that is highly resistant to cropping and moderately resistant to blurring and other attacks.
CHAPTER 2: INFORMATION HIDING – BACKGROUND AND OVERVIEW

The idea to communicate secretly is as old as communication itself. First stories, which can be interpreted as early records of covert communication, appear in old Greek literature, for example in Homer’s Iliad, or in tales by Herodotus. In this chapter we present some of this history and the evolution and the current state of the art in information hiding and watermarking.

2.1 Evolution of information hiding

2.1.1 Brief history of information hiding

Information hiding is an umbrella term for the more commonly used terms, steganography and watermarking. Steganography is a composite word obtained from the Greek words stegano, meaning “covered” and graphos, meaning “to write”. These days, the word steganography is widely used to refer to covert communication. The term watermark can be traced back to nearly 700 years ago and has its origin in the art of handmade papermaking. Watermarking was the perfect authentication mechanism to avoid confusion among different paper makers, artisans and merchants. Watermarks continue to be integral parts of the papermaking industry. It is a well-known fact that currency notes of many countries carry some form of watermark or the other. In addition to providing a level of security, they are also used for dating paper, authenticating paper and determining its source. Although the level of security is low and mostly preventive in nature, it has prevented duplicating and falsifying documents, for the past 700 years. Watermarks are simple, yet effective ways of copy protection. The same ideas of authentication and copy protection used for paper several hundred years ago, find relevance now in the digital age. Digital data, with all its enviable advantages, currently lacks a trustful and efficient mechanism to prevent unauthorized duplication. Digital watermarks could be the solution for a variety of challenges such as labeling, copyright protection, data authentication, and data monitoring and tracking.
While watermarks were around for a few centuries, the concept of steganography can be traced back to around 440 B.C. History is replete with examples of steganography and secret communication. The head of a trusted slave was shaved and the message was tattooed on the skull. This message becomes inconspicuous after the hair grows back. Thus a message was secretly conveyed to instigate a revolt against the Persians. German spies used the same method in the early 20th century. In his Histories [13], Herodotus tells how a Greek emperor was secretly warned of an invasion, by engraving a message on the wood of a wax tablet. Other techniques reported include letters hidden in messenger’s soles or women’s earrings, notes carried by pigeons and text written on wood tablets and whitewashed. Secret communication was also carried out using text as the cover. Tiny invisible dots were punched above or below the letters of cover text, or the heights of characters in the text were varied to convey some secret information. Later, invisible ink, rather than holes, was used for marking text. A variation of this idea has been used in WitnesSoft, a commercial electronic document-marking product from Aliroo [47]. This product marks the document with barely visible dots during printout, which can only be picked up by high-resolution scanners. Evidence of secret writing in Asia is found in various ancient religious and political literatures. In ancient India, Kautilya’s Arthasastra and the Lalita-Vistara, state explicit formulas for secret writing. The study of many different types of cryptography, not just steganography, flourished in ancient India. [36]. In ancient China, important messages were written on thin sheets of silk or paper and rolled into small balls. These balls were then coated with wax and swallowed by the messengers.

2.1.2 Recent developments and applications

Digital data embedding has numerous applications in the Internet era. Copyright protection is the foremost, with a great deal of commercial potential. The ease with which exact digital copies can be reproduced is a threat to content developers. There has to be a secure mechanism for identifying the owners and preventing unauthorized use, not to mention duplicating. DVD copy and playback protection is another important application that has drawn a lot of attention from the multimedia industry. Algorithms and compliant
multimedia devices are being contemplated to protect video piracy and unauthorized playback. For both the copyright and copy protection problems, digital watermarking has been proposed as a solution.

Data embedding can be used for secret communication. The information to be conveyed can be encrypted and then hidden in some cover image, audio or video file. While encryption provides security through obscurity, data embedding provides an additional layer of security.

Medical imaging systems and the health care industry can have potential benefits as well. Most medical images follow standards like DICOM (digital imaging and communications in medicine), which separate image data from the caption, the name of the patient, the date or the physician. By imperceptibly embedding all this additional data directly into the image, the danger of losing the link or mismatching the patient and his images can be eliminated.[1, 14] Whether such embedding will have an effect on the diagnosis or not remains to be seen. Techniques for hiding information in DNA sequences have been reported in [10].

Interesting data embedding applications have been stated in literature [36], for the multimedia broadcasting industry. The embedded control or reference information can be used for tracking and billing purposes in pay-per-use applications, commercials and Internet broadcast of digital media. Since the information is embedded, there is no overhead associated with maintaining a separate header or history file.

Data embedding can be used to insert extra scenes or multi-lingual tracks in a movie [37]. The user can tailor a video to his or her needs, which means he can select the rating or the language. All this can be done without any overhead on the bandwidth, which means some of the capabilities of the digital versatile disk, can be provided in a broadcast environment, without any extra storage or bandwidth requirements.
Data embedding can be used to embed additional data, like captions to images, subtitles or audio tracks to video (audio in video).

Another application for data embedding is multimedia fingerprinting to enable traitor tracking. This can be done on software too, to identify license violators.

Multimedia database indexing and retrieval can be made more efficient by hiding appropriate data into the database files and using that information during the retrieval. Data embedding is also being researched as a multimedia data authentication and tamper detection solution.

2.2 Terminology

The digital age and the internet era have brought with itself a lot of jargon. The field of information hiding is no exception. Here we list, define and contrast some of the technical terms used in the context of information hiding.

**Steganography** is a term used for all techniques that allow secret communication, by hiding the message information in some unsuspected cover data; say image, text, audio or video. These techniques are used in secret point-to-point communication between trusted parties and rely on the fact that the existence of covert communication is unknown to others. They typically have a high capacity and are less robust to manipulations. Technical steganography, data embedding and data hiding are other synonyms for this term.

**Watermarking** has the additional notion of robustness. Unlike steganography which can be easily defeated if its existence is revealed, watermarking imposes tougher requirements; even with the knowledge of the existence of hidden data and the algorithmic principle used, the attacker cannot defeat the watermark, without destroying the actual data itself. The security of the algorithm should lie, solely in the secret key in accordance with the Kerkhoff’s principle for cryptography [35].
**Bitstream watermarking**, as the name suggests, is used for data hiding in the bit stream of compressed media.

**Visible watermarking** is the digital analogue of paper watermarking. It embeds visual patterns or logos onto the cover media. **Labeling** refers to adding informative data in the form of a header to the parent data. Labels can be easily removed and hence are not secure.

**Fingerprinting** is done to trace illegal usage or traitor tracing. The image sold to a customer carries a unique, hidden fingerprint. If the image has been illegally distributed, the copyright violating customer can be traced based on the hidden fingerprint in the suspect image.

**Monitoring or tracking** refer to processes in which the multimedia with hidden data is tracked and the information regarding its usage logged into a history file. Digimarc’s [48] web spider is an excellent example.

The table below lists some of the variable names used in this thesis during description of algorithms.

**Table 2.1 Variable names**

<table>
<thead>
<tr>
<th>( r, c )</th>
<th>Size of the original image</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I )</td>
<td>Original image</td>
</tr>
<tr>
<td>( I' )</td>
<td>Resultant Image after embedding data</td>
</tr>
<tr>
<td>( I'' )</td>
<td>Image under contention</td>
</tr>
<tr>
<td>W, N</td>
<td>Embedded data sequence and its length</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>α</td>
<td>Parameter that determines embedding strength</td>
</tr>
<tr>
<td>S'</td>
<td>Extracted data sequence from I''</td>
</tr>
</tbody>
</table>

2.3 Generic information hiding system

2.3.1 Definition

All data hiding systems have the same generic building blocks: a data embedding algorithm and a data detection/extraction algorithm.

Figure 2.1 shows a typical data-embedding scheme. The inputs to the system are the binary data, which can be the ASCII encoding of serial numbers or copyright information, the cover or host image and an optional secret or public key. The key can be used to enforce security. The embedding algorithm manipulates the cover image, based on the binary data to obtain the marked image.

Figure 2.2 shows a corresponding data extraction scheme. This partly involves reversing the steps performed during the embedding process. The detector usually works on a suspect image. The presence of the original cover image to the detector is optional and depends on the application. If a key was used during embedding, then the same key is needed during extraction. The data extraction step tries to recover the manipulations performed on the cover image, (during embedding) and use them intelligently to make a binary decision. The output is the extracted binary data. Ideally the number of bits in error between the extracted and inserted data should be zero.
Figure 2.1: Generic data-embedding scheme

Figure 2.2: Generic data-extraction scheme
2.3.2 Algorithm requirements and design issues

Depending on the application at hand, data hiding algorithms have different requirements and different design issues. While imperceptibility is a common requirement, the degree of robustness, capacity and security required is highly application dependent and tough to specify. Whether the original unmarked image is available during extraction or not is another issue, which can heavily influence the robustness, capacity, security and false detection capabilities. The following is a general description of different requirements.

Imperceptibility In this thesis we deal with perceptually undetectable or transparent data embedding and watermarking techniques. The algorithm must embed data without causing any perceptual or statistical damage to the image. Significant artifacts should not appear in the marked use the original signal to decode the embedded data. However, most applications cannot afford image, even if it undergoes some routine signal processing like compression or filtering. A tougher constraint would be to have the two images look similar even under zooming. The perceptual transparency is usually verified by means of blind tests. A subject is presented with several copies of the same image, half of them marked and the other half unmarked and asked to distinguish between the two groups.

Blind or Non-blind recovery In some applications, such as copy tracking and copyright protection, the data extraction algorithms may this privilege and this makes extraction a very difficult task and limits the capacity and robustness as well. Extraction becomes more complicated if there is additional post processing on the marked image.

Bit-rate/Capacity Applications like insertion of serial number or author identification, require that relatively small amounts of information be incorporated repeatedly in the signal. On the other hand, data embedding applications like embedding a smaller image into a larger image and embedding multiple speech signals into a video require a lot of bandwidth. In these cases, the algorithms must be able to embed an amount of data that is a significant fraction of the amount of data in the host signal. The ability to embed large
quantities of data in a host signal depends on how the embedding algorithm can adapt its insertion strategy to the underlying signal.

**Robustness** Data embedding, when used in error-free applications like storage of clean signals in a controlled database, may not need a high level of robustness. However, in many cases, common unintentional signal processing like compression and filtering maybe present in the system. For example image databases use compression to reduce image size and hence storage space. With the availability of image manipulation software like Adobe Photoshop and Microsoft Paint, intentional attacks by third parties, aimed at thwarting detection of embedded data is feasible and must be accounted for to design a robust system. Given below is a list of possible intentional or unintentional signal modifications that must be addressed, of course depending on the application.

- Additive and multiplicative noise (Gaussian, uniform, speckle, mosquito)
- Linear filtering (lowpass, highpass, bandpass filtering)
- Nonlinear filtering (median filtering, morphological filtering)
- Local and global affine transforms (translation, rotation, shearing, scaling)
- Flipping
- Deletion of lines or columns
- Data reduction (cropping, clipping)
- Data composition (logo insertion, scene composition)
- Transcoding (H.263 $\rightarrow$ MPEG-2, GIF $\rightarrow$ JPEG)
- D/A and A/D conversion (print-scan, analog TV transmission)
- Multiple watermarking
- Collusion attacks
- Statistical averaging
- Specially designed watermark removal attacks
Security In many applications, the embedding procedure must be secure in that an unauthorized user must not be able to detect the presence of embedded data, let alone remove the embedded data. Security requirement is more stringent in secret communication scenarios. Algorithms must be designed taking the Kerchoff’s\(^1\) principle into account.

2.4 Information hiding – current state of the art

2.4.1 Introduction

In this section we will describe different data hiding algorithms. We will focus on embedding and extraction of data in images. Selected examples are explained in detail for the main types of algorithms: spatial domain and transform domain (which includes discrete cosine transform (DCT), wavelet and others) based algorithms.

2.4.2 Spatial domain watermarking

Bender et al. [3] propose two methods for data hiding. In the first method, called “Patchwork”, randomly selected pairs of pixels \((a_i, b_i)\) are used to hide 1 bit by increasing \(a_i\)'s by one and decreasing the \(b_i\)'s by one. Provided that the image satisfies some statistical properties, the expected value of the sum of the differences between the \(a_i\)'s and \(b_i\)'s of \(N\) pixel pairs is given by \(2N\).

\[
\sum_{N} a_i - b_i = 2N
\]  

(2.1)

for watermarked pairs. The above summation is zero for non-watermarked pairs.

\(^1\) The security of the algorithm must lie in the secret key, not on the assumption that the hacker has little or partial information about the algorithm or the data in hand. [48]
In the second approach called “Texture Block Coding”, the watermark is embedded by copying one image texture block to another area in the image with a similar texture. To recover the watermark, the autocorrelation function has to be computed. A remarkable feature of this technique is the high robustness to any kind of distortion, since both image areas are distorted in a similar way, which means that the watermark recovery by autocorrelation still works.

Pitas and Kaskalis propose signature casting on digital images [22, 27, 28], which is based on, the same basic idea as the patchwork algorithm proposed by Bender et al. [6]. The watermark \( S = \{s_{m,n}\} \) consists of a binary pattern of the same size as the original image, and where the number of ones is equal to the number of zeros. The original image \( I \), with luminance values \( x_{m,n} \) at location \( m \) and \( n \), is divided into two sets \( A \) and \( B \) of equal size in the following way:

\[
A = \{x_{mn} \in I, s_{mn} = 1\} \\
B = \{x_{mn} \in I, s_{mn} = 0\}
\] (2.2)

The watermark is superimposed by changing the elements of the subset \( A \) by a positive integer factor \( k \), e.g. \( A' = \{x_{mn} + k, x_{mn} \in A\} \). The watermarked image is then given by the union of \( A' \) and \( B \). To verify the presence of a watermark hypothesis testing [11] is applied. The test statistic \( q \) is defined as the normalized difference between the mean \( \bar{a}' \) of set \( A' \) and the mean \( \bar{b} \) of set \( B \):

\[
q = \frac{(\bar{b} - \bar{a}')}{\sigma^2_A + \sigma^2_B}
\] (2.3)

where \( \sigma^2_A \) and \( \sigma^2_B \) defines the sample variance of set \( A' \) and \( B \), respectively. The test statistic is then compared with a threshold to determine if there is a watermark. The method is immune to sub-sampling followed by up-sampling and resistant to JPEG compression up to a compression factor of 4:1.
To increase the performance of the block based spatial watermarking methods, Bruyndonckx et al. [4] suggest the use of pixel classification. Pixels within pseudo-randomly selected blocks are classified into zones (1 and 2) of homogeneous luminance values. The classification is based on three types of contrast between zones: hard contrast, progressive contrast and noise contrast. Each zone is then further subdivided into two categories \( A \) and \( B \) based on a grid defined by the coder. Each pixel is thus assigned to one of four zone/category combinations, e.g. 1/A, 1/B, 2/A, 2/B. A bit \( b \) is embedded by modifying the zone/category mean pixel value as to satisfy the following constraints:

If \( b = 0 \):

\[
\begin{align*}
    m_{1A}' - m_{1B}' &= S \\
    m_{2A}' - m_{2B}' &= S
\end{align*}
\]  

If \( b = 1 \):

\[
\begin{align*}
    m_{1A}' - m_{1B}' &= S \\
    m_{2A}' - m_{2B}' &= S
\end{align*}
\]

where \( m_{1A}' \), \( m_{1B}' \), \( m_{2A}' \), and \( m_{2B}' \) are the modified zone/category mean values and \( S \) the watermark embedding strength. The modification of the mean values is done by applying equal luminance variations to all pixels belonging to the same zone. To increase robustness the authors suggest to perform redundant bit embedding and use error-correcting codes. Good robustness to JPEG compression is reported.

Watermark embedding based on quantization has been proposed by Chen and Wornell [9]. Their method is called quantized index modulation (QIM) and is based on a set of \( N \)-dimensional quantizers. The quantizers satisfy a distortion constraint and are designed such that the reconstruction values from one quantizer are far away from the reconstruction points of every other quantizer. The message to be transmitted is used as an index for quantizer selection. The selected quantizer is then used to embed the
information by quantizing the image data in either the spatial or DCT domain. In the decoding process, a distance metric is evaluated for all quantizers and the index of the quantizer with the smallest distance identifies the embedded information. The authors show that the performance of the resulting watermarking scheme is superior to the standard spread spectrum modulation without watermark weighting.

Besides spatial domain watermarking related to modulation, it was proposed by Maes et al. [21] to modify geometric features of the image. The method is based on a dense line pattern, generated pseudo-randomly and representing the watermark. A set of salient points in the image is then computed, for example based on an edge detection filter. The detected points are then warped such that a significant large number of points are within the vicinity of lines. In the detection process, the method verifies if a significantly large number of points are within the vicinity of lines.

2.4.3 Fractal based watermarking

Related to spatial domain watermarking schemes are methods based on fractal image compression. This approach has first been proposed by Puate and Jordan [32]. In fractal image compression the image is coded using the principles of iterated function systems and self similarity [34]. The original image is divided into square blocks $R_k$ called range blocks. Further let $F$ be a set of mapping functions $f_k$, which are composed of a geometric transformation $g_k$ and a massic transformation $m_k$. The mapping functions work on domain blocks $D_k$, which are larger than range blocks. The geometric transformation consists moving the domain block $D_k$ to the location of the range block $R_k$ and reducing the size of the domain block to the size of the range block. The massic transformation adjusts the intensity and orientation of pixels in the domain block after geometric transformation. Massic transformations include rotation by 90, 180 and 270 degrees, reflection about mid-vertical, mid-horizontal and cross diagonal axis as well as identity mapping. To compress an image, the best combination of domain block $D_k$ and mapping function $f_k$ has to be found for each range block $R_k$ such that the difference
between the range block $R_k$ and the mapped domain block $f_k(D_k)$ is minimal. That means that the encoding includes a spatial search over all possible domain blocks. Decoding is accomplished by iterating over the coded mapping functions using any initial image. To embed a bit into this scheme a range block is pseudo-randomly selected. The corresponding search space $S_k$ for the range blocks is then split up into two sub-search spaces $S_k^1$ and $S_k^2$ of equal size. Each subspace is assigned to a bit value and the current range block is encoded by searching only in the subspace corresponding to the bit value of the current bit. To recover an embedded bit, the watermarked image is again compressed, however this time using the full domain block search space. Then for a marked range block the location of the corresponding domain block reveals the embedded bit value. The algorithm was tested against JPEG compression and showed good robustness down to a compression quality of about 50%. A drawback of this technique is the slow speed due to fractal compression scheme.

A very similar approach has been proposed by Davern and Scott [12]. The only difference is that they do not encode the entire image, but only a user defined range region based on a user defined domain region. Given the two regions, the watermark encoding is equivalent to the system proposed by Puate and Jordan in that the domain region is divided into two parts and depending on the bit value, one or the other region is used for encoding a range block. This idea of watermarking using spatial domain fractal image coding, has been extended to DCT blocks by Bas et al. [2].

2.4.4 Discrete cosine transform (DCT) domain watermarking

Efficient watermarking in the DCT domain was first introduced by Koch, Burgett, Zhao and Rindfrey [8, 17, 18]. As in the JPEG compression scheme, the image is first divided into square blocks of size 8x8 for which the DCT is computed. From a pseudo-randomly selected block, a pair of mid-frequency coefficients is selected from 12 predetermined pairs. To embed a bit, the pairs are modified such that the difference between them is either positive or negative depending on the bit value. In order to accommodate lossy
JPEG compression, the quantization matrix is taken into account when altering the DCT coefficients. This method shows good robustness to JPEG compression down to a quality factor of 50%.

Bors and Pitas [6, 7] suggest a method that modifies DCT coefficients satisfying a block site selection constraint. The image is first divided into blocks of size 8x8. Certain blocks are then selected according to a Gaussian network classifier decision. The middle range frequency DCT coefficients are then modified, using either a linear DCT constraint or a circular DCT detection region, to convey the watermark information. In the watermark recovery process, the algorithm first verifies the DCT coefficient constraint for all blocks followed by the location constraint. The algorithm can accommodate JPEG compression for a compression ratio of 13:1 and 18:1 using the linear DCT constraint or the circular DCT detection region, respectively.

Swanson et al. [38, 39] suggest a DCT domain watermarking technique, based on frequency masking of DCT blocks, which is similar to methods proposed by Smith and Comiskey [40]. The input image is split up into square blocks for which the DCT is computed. For each DCT block, a frequency mask is computed based on the knowledge that a masking grating raises the visual threshold for signal gratings around the masking frequency. The resulting perceptual mask is scaled and multiplied by the DCT of a maximal length pseudo-noise sequence. This watermark is then added to the corresponding DCT block, followed by spatial masking to verify that the watermark is invisible and to control the scaling factor. Watermark detection requires the original image as well as the original watermark and is accomplished by hypothesis testing. The authors report good watermark robustness for JPEG compression, colored noise and cropping.

Tao and Dickinson [41] introduce an adaptive DCT-domain watermarking technique based on a regional perceptual classifier with assigned sensitivity indices. The watermark is embedded in N AC DCT coefficients. The AC coefficients are essentially all the coefficients except the DC or the zero frequency coefficient. The coefficients are selected
such as to have the smallest quantization step sizes according to a JPEG compression table. The selected coefficients $x_i$ are modified as follows:

$$\bar{x}_i = x_i + \max[x_i \alpha_m, \text{sign}(x_i) \frac{D_i}{\kappa}]$$

(2.6)

$$5 \leq \kappa \leq 6$$

where $\alpha_m$ defines the noise sensitivity index for the current block, $D_i$ the quantization step for $x_i$, and $\kappa$ satisfies $5 \leq \kappa \leq 6$. It should be noted that the watermark signal is not generated randomly. Various approaches exist to determine the noise sensitivity by efficiently exploiting the masking effects of the human visual system. The authors propose a regional classification algorithm, which classifies the block in one of six perceptual classes. The classification algorithm exploits luminance masking, edge masking and texture masking effects of the human visual system. Namely the perceptual block classes from 1 to 6 are defined as: edge, uniform, low sensitivity, moderately busy, busy and very busy, in descending order of noise sensitivity. Each perceptual class has a noise sensitivity index assigned to it. Watermark recovery requires the original image as well as the watermark and is based on hypothesis testing. Experiments show that the method resists to JPEG compression down to a quality of 5% and can accommodate random noise with a PSNR of 22.1 dB.

Podilchuk [31, 30] introduces perceptual watermarking using the Just Noticeable Difference (JND) method to determine an image dependent watermark modulation mask. The watermark modulation onto selected coefficients in either the DCT or wavelet transform domain can be described as:

$$I^*_{u,v} = I_{u,v} + \text{JND}_{u,v} \times w_{u,v}$$

if $I_{u,v} > \text{JND}_{u,v}$ and

$$I^*_{u,v} = I_{u,v}$$

otherwise

(2.7)
where \( I'_{u,v} \) are the transform coefficients of the original image, \( w_{u,v} \) are the watermark values and \( JND_{u,v} \) is the computed just noticeable differences based on visual models. For DCT coefficients, the author suggests use of a perceptual model defined by Watson based on frequency and brightness sensitivity as well as local contrast masking. This model provides image dependent masking thresholds for each 8 x 8 DCT block. Watermark detection is based on the correlation between the difference of the original image and the image under inspection and the watermark sequence. The maximum correlation is compared to a threshold to determine whether an image contains the watermark in question. Experiments showed that the watermark scheme is extremely robust to JPEG compression, cropping, scaling, additive noise, gamma correction and printing-photocopying-scanning. For attacks involving a geometrical transformation, the inverse operation has to be applied to the image before the watermark detection process.

Piva et al. [29] describe another DCT based method, which exploits the masking characteristics of the human visual system. The watermark consists of a pseudo-random sequence of \( M \) real numbers with normal distribution \( X = \{x_1, x_2, \ldots, x_M\} \). The coefficients of the \( N \times N \) DCT of the original image \( I \) are reordered into a vector using a “zig-zag” scan. From this vector, \( M \) coefficients, starting at position \( L + 1 \) are selected to generate the vector \( T = \{t_1, t_2, \ldots, t_M\} \). The watermark \( X \) is embedded into \( T \) according to:

\[
t'_i = t_i + \alpha |t_i| x_i
\]

(2.8)

where \( \alpha \) determines the watermark strength. The modified coefficients replace the non-modified coefficients before the watermarked image \( I' \) is reconstructed. In order to enhance the robustness, visual masking is applied as follows:

\[
y'_y = y_y (1 - \beta_y) + \beta_y y'_y = y_y + \beta_y (y'_y - y_y)
\]

(2.9)
where $\beta_y$ is a weighting factor taking into account the characteristics of the human visual system. A simple way of choosing $\beta_y$ is the normalized sample variance at pixel $y$, defined as the ratio between the sample variance for a square block with center at $y$ and the maximum of all block variances. As in most schemes watermark detection is performed by comparing the correlation $z$ between the watermark and the possibly corrupted signed DCT coefficients $T^*$ with a threshold $S_z$. The correlation $z$ is defined as:

$$z = \frac{XT^*}{M} = \frac{1}{M} \sum_{i=1}^{M} x_i f_i^*$$  \hspace{1cm} (2.10)

The threshold $S_z$ is adaptive and is given as:

$$S_z = \frac{\alpha}{3M} \sum_{i=1}^{M} |f_i^*|$$ \hspace{1cm} (2.11)

Experimental results demonstrate that the watermark is robust to several image processing operations (for example JPEG compression, median filtering and multiple watermarking) and geometrical distortions (after applying the inverse geometric transformation).

### 2.4.5 Frequency domain watermarking

Frequency domain watermarking was first introduced by Boland et al. [5] and Cox et al. [11] who independently developed perceptually adaptive methods based on modulation. Cox et al. draw parallels between their technology and spread spectrum communications since the watermark is spread over a set of visually important frequency components. The watermark consists of a sequence of numbers $x = x_1, ......., x_n$ with a given statistical distribution, such as a normal distribution $N(0, I)$ with zero mean and unit variance. The
watermark is inserted into the image $V$ to produce the watermarked image $V'$. Three techniques are proposed for watermark insertion:

$$v'_i = v_i + \alpha x_i$$  \hspace{1cm} (2.12)

$$v'_i = v_i (1 + \alpha x_i)$$  \hspace{1cm} (2.13)

$$v'_i = v_i e^{\alpha x_i}$$  \hspace{1cm} (2.14)

where $\alpha$ determines the watermark strength and the $v_i$'s are perceptually significant spectral components. Equation 2.9 is only suitable if the values $v_i$ do not vary too much. Equation 2.10 and 2.11 give similar results for small values of $\alpha x_i$, and for positive $v_i$'s equation 2.11 may even be viewed as an application of equation 2.9 where the logarithms of the original values are used. In most cases equation 2.10 is used. The scheme can be generalized by introducing multiple scaling parameters $\alpha_i$ as to adapt to the different spectral components and thus reduce visual artifacts. To verify the presence of the watermark the similarity between the recovered watermark vector $X'^*$, given by the difference between the original image data vector $V$ and the possibly tampered image data vector $V^*$, and the original watermark vector $X$ is measured. The similarity measure is given by the normalized correlation coefficient:

$$\text{sim}(X, X^*) = \frac{X^* \cdot X}{\sqrt{X^* \cdot X^*}}$$  \hspace{1cm} (2.15)

Robustness tests showed that the method resists JPEG compression (at a quality factor of 5% and no smoothing), dithering, fax transmission, printing-photocopying-scanning, multiple watermarking and collusion attacks. For the experiments the watermark was of length 1000 with $N(0, I)$, $\alpha$ was set to 0.1 and the watermark was embedded into the 1000 strongest DCT coefficients using equation 2.10. Boland et al. propose a similar
technique based on a hybrid between amplitude modulation and frequency shift keying and suggest the use of different transform domains such as the discrete cosine transform (DCT), Wavelet Transform, Walsh-Hadamard transform and the Fourier transform.

Ruanaidh et al. propose watermarking by modification of the phase in the frequency transform domain [23, 24]. To embed a bit, the phase of the selected coefficient \( F(k_1,k_2) \) of an \( N_1 \times N_2 \) discrete Fourier transform (DFT) is modified by adding a small \( \delta \):

\[
\angle F(k_1,k_2) \leftarrow \angle F(k_1,k_2) + \delta
\]  

(2.16)

In order for the watermarked image to be real, the phase must satisfy negative symmetry which leads to the additional modification:

\[
\angle F(N_1-k_1,N_2-k_2) \leftarrow \angle F(N_1-k_1,N_2-k_2) + \delta
\]  

(2.17)

Coefficients are only modified if their relative power is above a given threshold. If the original image is available the watermark can easily be recovered by comparing the phase. In case the original is not available, Ruanaidh suggests pre-quantizing the original phase prior to modifying it. Then deviations between the quantized states could be used to convey the data.

In another publication Ruanaidh and Pun [25] explicitly design a watermarking technique invariant to translation, rotation and scaling. The method is a hybrid between Discrete Fourier Transforms (DFT) and log-polar mapping. The process is depicted in the figure below. At first, the DFT of the image is computed. One of the DFT properties is that a shift in the spatial domain results in a phase shift in the frequency domain. Keeping only amplitude for further processing makes the image translation invariant. In the second step rotation and scale invariance is achieved by mapping the amplitude from the Cartesian grid to the log-polar grid. Consider a point \( (x, y) \in \mathbb{R} \), then the mapping is defined as:
\[ x = e^\mu \cos \theta \]
\[ y = e^\mu \sin \theta \]  \hspace{1cm} (2.18)

where \( \mu \in \mathbb{R} \) and \( 0 < \theta < 2\pi \). One can easily see that this is a one to one mapping and that rotation and scaling in the Cartesian grid are converted to a translation of \( \mu \) and \( \theta \) coordinates respectively. Computing the DFT of the log-polar map again and keeping only the amplitude results in rotation and translation invariance. Taking the Fourier transform of a log-polar map is equivalent to computing the Fourier-Mellin transform [25]. Hence combining the two steps results in rotation, scale and translation (RST) transformation invariance. The watermark takes the form of a two-dimensional spread spectrum signal in the RST transformation invariant domain. In a test, a 104 bit long watermark was embedded into an image. The watermarked image was then rotated by \( 143^\circ \) and scaled by 75\% along each axis. The embedded watermark was recovered from this image. Furthermore the method resists JPEG compression down to a quality factor of 75\% and cropping to 50\% of the original image size. This approach, which is actually the first one, which was especially designed to resist to geometrical attacks, has interesting aspects and ideas, and might trigger a new way of approaching the design of future watermarking techniques. A variation of this idea based on the Radon transform has been proposed by Wu et al. [44].

\[ \text{2.4.6 Wavelet domain watermarking} \]

Embedding the watermark using a multi-resolution decomposition has first been proposed by Boland et al. [5]. As for schemes working in other transformation domains, the watermark is usually given by a pseudo-random 2-D pattern. Both the image and watermark are decomposed using a 2-D wavelet transform and a weighted version of the watermark is added in each sub-band of the image. Watermark decoding is as usual based on a normalized correlation between an estimate of the embedded watermark and the watermark itself. Various wavelet-based schemes have been proposed [16, 19, 45, 46]. The difference between the schemes usually lies in the way the watermark is weighted in order to decrease visual artifacts.
Although digital watermarking is a quite new field of research, we have shown in this chapter that worldwide research activities have already led to a huge collection of definitions, technologies, concepts and publications. Designing efficient watermarking methods is a very delicate process and requires knowledge from a variety of domains, such as digital modulation, human visual system, cryptography, and general signal processing. Thus it is a very challenging task and for sure interesting.
CHAPTER 3: SELECTED BACKGROUND CONCEPTS

The data-embedding algorithm presented in this work is based on self-similar signals and fractal modulation concepts. In this chapter, we develop this very important theoretical basis, first in the one-dimensional sense and then present the extension to two-dimensional images and use these developments to construct self-similar images.

3.1 Deterministically self-similar signals

A signal $x(t)$ that satisfies the deterministic scale-invariance property [43]

$$x(t) = a^{-H}x(at)$$  \hspace{1cm} (3.1)

for all $a > 0$, is called a homogeneous function of degree $H$.

Homogeneous functions can be regular or nearly regular or generalized. Some examples of homogeneous functions are given below.

$$x(t) = 1 \quad \text{(regular homogeneous function)}$$  \hspace{1cm} (3.2)

$$x(t) = u(t) \quad \text{(nearly regular homogeneous function)}$$

$$x(t) = \delta(t) \quad \text{(generalized homogeneous function)}$$

It has been claimed that homogeneous functions can always be parameterized with only a few constants [15].

Bi-homogeneous functions are waveforms that satisfy the deterministic self-similarity property for values of $a$ that are integer powers of 2, or the so-called dyadic self similarity property

$$x(t) = 2^{-KH}x(2^K t)$$  \hspace{1cm} (3.3)
for all integers \( k \). In this chapter, we explain the properties and characterizations of this general class of homogenous signals of degree \( H \). In all of our current work, we deal with bi-homogeneous signals only. Statistically self-similar signals are a different class of signals and satisfy the scale invariance property in a statistical sense. While we do not deal with these signals in our current work, a good description of these signals and models to construct them can be found in [20].

It is claimed that bi-homogenous signals can be of significant value in some communications-based applications [43]. In this current work, we exploit the diversity strategy to embed information into a waveform “on all time scales”. As these waveforms are intrinsically self-similar, an arbitrarily short duration time-segment is sufficient to recover the entire waveform, and hence the embedded information, given adequate bandwidth. Likewise an arbitrarily low-bandwidth approximation to the waveform is sufficient to recover the undistorted waveform and hence the embedded information, given adequate duration. The bound here is that the time-bandwidth product should be above a certain threshold. It has also been observed that these bi-homogenous waveforms have close to \( 1/f \) type spectral characteristics and fractal properties as well [43]. These properties of the modulation scheme and the diversity paradigm may be well suited for communication over highly unreliable channels of uncertain duration, bandwidth, and SNR [43]. A convenient and efficient mathematical framework for characterizing bi-homogenous signals exists, which we exploit in our work. Bi-homogenous signals that have close to \( 1/f \) type spectral characteristics and fractal properties can be classified as energy-dominated or power-dominated. Roughly, signals that have a \( 1/f \) – like Fourier transform are energy-dominated, while those that have a \( 1/f \) – like power spectrum are power-dominated [43].

Orthonormal “self-similar” bases for bi-homogenous signals, can be constructed based on the orthonormal wavelet basis expansions. It turns out that this representation is natural as well as efficient, similar to the case of statistically self-similar \( 1/f \) – type processes [43].
3.1.1 Energy-dominated bi-homogeneous signals

A bi-homogeneous signal \( x(t) \) is said to be energy-dominated if, when \( x(t) \) is filtered by an ideal bandpass filter with frequency response

\[
B_0(\omega) = \begin{cases} 
1 & \pi < |\omega| \leq 2\pi \\
0 & \text{otherwise}
\end{cases}
\]  

(3.4a)

the resulting signal \( \tilde{x}_0(t) \) has finite energy, i.e.,

\[
\int_{-\infty}^{\infty} \tilde{x}_0(t) dt < \infty
\]  

(3.4b)

Because the dyadic self-similarity property of bi-homogeneous signals is similar to the dyadic scaling relationship between basis functions in an orthonormal wavelet basis, wavelets provide a particularly nice representation for this family of signals. For an energy dominated bi-homogeneous signal \( x(t) \), the expansion in an orthonormal wavelet basis is

\[
x(t) = \sum_{m} \sum_{n} x_n^m \psi_n^m(t)
\]  

(3.5a)

\[
x_n^m = \int_{-\infty}^{\infty} x(t) \psi_n^m(t)
\]  

(3.5b)

\( \psi_n^m(t) \) are dilations and translations of the mother wavelet \( \psi(t) \) and satisfy

\[
\psi_n^m(t) = 2^{m/2} \psi(2^n t - n)
\]  

(3.6)

where \( m, n \) are the dilation and translation indices, respectively.
Since \( x(t) \) satisfies (3.3) and \( \psi_n^m(t) \) satisfies (3.6), it follows from (3.5b) that for bi-

homogeneous signals

\[
x_n^m = \beta^{-m/2} x_n^0
\]

(3.7)

where

\[
\beta = 2^{2H+1} = 2^\gamma
\]

(3.8)

Denoting \( x_n^0 \) by \( q[n] \), (3.5a) then becomes

\[
x(t) = \sum_m \sum_n \beta^{-m/2} q[n] \psi_n^m(t)
\]

(3.9)

From which we see that \( x(t) \) is completely specified in terms of \( q[n] \). We term \( q[n] \) a

generating sequence for \( x(t) \) since, as we will see, this representation leads to techniques

for synthesizing useful approximations to bi-homogeneous signals in practice.

The wavelet based norms for \( E^H \), the Hilbert space representing all energy dominated

signals with degree of homogeneity \( H \), constitute a highly convenient and practical

collection for applications involving the use of bi-homogeneous signals. Each associated

wavelet based inner product leads immediately to an orthonormal self-similar basis for

\( E^H \): If \( x(t) \in E^H \), then

\[
x(t) = \sum_n q[n] \theta_n^H(t)
\]

(3.10a)

\[
q[n] = \langle x, \theta_n^H \rangle
\]

(3.10b)

Where, again, the basis functions
are all self-similar, mutually orthogonal, and have unit norm.

As an example for the case \( H = 0 \), Fig. 3.1 depicts the self-similar basis functions \( \theta_0^H(t) \), \( \theta_1^0(t) \), \( \theta_2^0(t) \), \( \theta_3^0(t) \), \( \theta_4^0(t) \), \( \theta_5^0(t) \), \( \theta_6^0(t) \), and \( \theta_7^0(t) \) corresponding to the Daubechies 5th-order compactly supported wavelet basis.

Figure 3.1 The self-similar basis functions \( \theta_0^H(t) \) to \( \theta_7^0(t) \) of an orthonormal basis for \( E^H \), \( H=0 \)

These functions were generated by evaluating the summation (3.11) over a large but finite range of scales \( m \). We emphasize that \( q[n] \) is only a unique characterization of \( x(t) \)
when we associate it with a particular choice of wavelet $\psi(t)$. In general, every different wavelet decomposition of $x(t)$ yields a different $q[n]$, though all have finite energy.

Figure 3.2: The signal $x(t)$ constructed using equation (3.10a) and the orthonormal basis functions shown in figure 3.1

The self-similar basis functions and the self-similar signal in case of the Haar wavelet are shown in Figures 3.3 and 3.4 respectively.

It may be noted that for an arbitrary non-homogeneous signal $x(t)$, the sequence $q[n] = \langle x, \theta_n^H \rangle$ defines the projections of $x(t)$ onto $E^H$, so that

$$\hat{x}(t) = \int_{-\infty}^{\infty} q[n] \theta_n^H(t) dt \tag{3.12}$$
represents the closest bi-homogeneous signal to \( x(t) \) with respect to the induced norm \( \| \cdot \| \), i.e.,

\[
\min_{v \in \mathbb{R}^n} \| v - x \| \leq (313)
\]
Figure 3.3 The self-similar basis functions $\theta_0(t)$ to $\theta_7(t)$ of an orthonormal basis for $E^H$, $H=0$, using the Haar wavelet.

![constructed signal graph]

Figure 3.4: The signal $x(t)$ constructed using equation (3.10a) and the orthonormal basis functions shown in figure 3.3

3.1.2 Power-dominated bi-homogeneous signals

Energy-dominated bi-homogeneous signals have infinite energy, when not perfectly band-filtered. Most have infinite power as well. However, there are other infinite-power bi-homogeneous signals that are not energy-dominated. Such classes of signals are referred to as power-dominated signals.
A bi-homogeneous signal is said to be power-dominated if when \( x(t) \) is filtered by an ideal band-pass filter with frequency response (3.4a) the resulting signal \( \tilde{x}_0(t) \) has finite power, i.e.,

\[
\lim_{T \to \infty} \frac{1}{T} \int_{-T}^{T} \tilde{x}_0(t) dt < \infty
\]  

(3.14)

The notation \( p^H \) is used to designate the class of power-dominated bi-homogeneous signals of degree \( H \). Thus we note that bi-homogeneous signals can either be energy dominated or power dominated.

### 3.1.3 Discrete-time algorithms for bi-homogeneous signals

Orthonormal wavelet representations provide some useful insights into bi-homogeneous signals. For example, since the sequence \( q[n] \) is replicated at each scale in the representation 3.9 of a bi-homogeneous signal \( x(t) \), the detail signals

\[
D_m x(t) = \beta^{-m/2} \sum_n q[n] \nu_n^m(t)
\]  

(3.15)

representing \( q[n] \) modulated into a particular octave band are simply time-dilated versions of one another, to within an amplitude factor. The corresponding time-frequency portrait of a bi-homogeneous signal is depicted in Figure 3.5, from which the scaling properties are apparent. For purposes of illustration, the signal in this figure has degree \( H = -1/2 \) (i.e., \( \beta = 1 \)), which corresponds to the case in which \( q[n] \) is scaled by the same amplitude factor in each octave band. As always, the partitioning in such time frequency portraits is idealized; in general, there is both spectral and temporal overlap between cells. Wavelet representations lead to some highly efficient algorithms for synthesizing, analyzing, and processing bi-homogeneous signals. These algorithms are obtained by applying the DWT algorithm to the highly structured form of the wavelet coefficients of bi-homogeneous signals. One discrete-time representation for a bi-
homogeneous signal $x(t)$ is in terms of a generating sequence $q[n]$. The generating sequence corresponds to the coefficients of the expansion of $x(t)$ in an orthonormal basis $\{\theta_n^H(t)\}$ for $E^H$ [43].

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$q[0]$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$q[1]$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$q[2]$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$q[3]$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.5. The time-frequency portrait of a bi-homogeneous signal of degree $H = -1/2$. 
3.2 Fractal modulation and demodulation

In this section we develop the process of fractal modulation. In particular we explore the use of bi-homogeneous signals as modulating waveforms in a communication system. Beginning with an idealized but general channel model, we demonstrate that the use of bi-homogeneous waveforms in such channels is both natural and efficient, and leads to a novel multi-rate diversity strategy in which data is transmitted simultaneously at multiple rates. The problem is to design a communication system for transmitting a continuous or discrete valued data sequence over a noisy and unreliable continuous-amplitude, continuous-time channel. We must therefore design a modulator at the transmitter that embeds the data sequence \( q[n] \) into a signal \( x(t) \) to be sent over the channel. At the receiver, a demodulator must be designed for processing the distorted signal \( r(t) \) from the channel to extract an optimal estimate of the data sequence \( \hat{q}[n] \). The overall system is depicted in Figure 3.6 [43]

![Figure 3.6](image)

Figure 3.6 A communication system for transmitting a continuous or discrete-amplitude data sequence \( q[n] \) over a noisy and unreliable continuous-amplitude, continuous-time channel.

The particular channel we consider has the characteristic that it is open for some time interval \( T \), during which it has a particular bandwidth \( W \) and signal-to-noise ratio (SNR). This rather basic channel model is a useful one for a variety of settings, and in particular it can be used to capture both characteristics of the transmission medium and constraints inherent in one or more receivers. When the noise characteristics are additive, the overall channel model is as depicted in Figure. 3.7, where \( z(t) \) represents the noise process.
When either the bandwidth or duration parameters of the channel are known \textit{a priori}, there are many well-established methodologies for designing an efficient and reliable communication system. However we restrict our attention to the case in which both the bandwidth and duration parameters are either unknown or not available to the transmitter. In designing a communication system for such channels, the following key performance characteristics must be satisfied:

1. Given a duration-bandwidth product $T x W$ that exceeds some threshold, we must be able to transmit $q[n]$ without error in the absence of noise, i.e., $z(t) = 0$.
2. Given increasing duration-bandwidth product in excess of this threshold, we must be able to transmit $q[n]$ with increasing fidelity in the presence of noise. Furthermore, in the limit of infinite duration-bandwidth product, perfect transmission should be achievable at any finite SNR.

Figure 3.7. The channel model for a typical communications scenario
The first of these requirements implies that, at least in principle, we should be able to recover $q[n]$ using arbitrarily narrow receiver bandwidth given sufficient duration, or alternatively, from an arbitrarily short duration segment given sufficient bandwidth. The second requirement implies that we ought to be able to obtain better estimates of $q[n]$ the longer a receiver is able to listen, or the greater the bandwidth it has available. Consequently, the modulation must contain redundancy or diversity of a type that can be exploited for the purposes of improving the reliability of the transmission and the use of bi-homogeneous signals for transmission appears to be naturally suited to fulfilling both these system requirements. The modulation strategy is to embed the data to be transmitted into the bi-homogeneous signal. Since the resulting signal has fractal properties the scheme is referred to as fractal modulation. There is a natural redundancy that is built in to the fractal signal due to the fact that it is self-similar to within an amplitude factor at different scales. This enables us to be able to recover the signal even from narrow bandwidths (when the signal is transmitted through channels with very narrow bandwidth). Besides this, the redundancy is also exploited at the receiver’s end by adopting a diversity combining strategy, thus enabling improved transmission reliability.

3.2.1 Transmitter design: modulation

To embed a finite power sequence $q[n]$ into a bi-homogeneous waveform $x(t)$ of degree $H$, it suffices to consider using $q[n]$ as the coefficients of an expansion in terms of a wavelet-based orthonormal self-similar basis of degree $H$ i.e.,

$$x(t) = \sum_n q[n] \varphi_n^H(t) \tag{3.16}$$

When the basis is derived from the ideal bandpass wavelet, as we will generally assume in our analysis, the resulting waveform $x(t)$ is a power-dominated bi-homogeneous signal whose time-frequency portrait has the form depicted in Figure. 3.5. Although this considerably simplifies our analysis, the use of the ideal bandpass wavelet to synthesize the orthonormal self-similar basis in our modulation strategy is impractical due to the
poor temporal localization in the wavelet. In practice, we may replace the ideal bandpass wavelet with one having not only comparable frequency domain characteristics and better temporal localization, but sufficiently many vanishing moments as well. One easy choice would be to use Daubechies wavelets. Another apparent problem with fractal modulation as initially proposed is that it requires infinite transmitter power. Indeed, as Figure 3.5 illustrates, $q[n]$ is modulated into an infinite number of octave-width frequency bands. However, it should be appreciated that in a practical implementation only a finite collection of contiguous bands $\mu$ would, in fact, be used by the transmitter. As a result, the transmitted waveform

$$x(t) = \sum_n q[n] \sum_{m \in \mu} \beta^{-m/2} \psi_n^m(t)$$ (3.17)

would exhibit self-similarity only over a range of scales, and demodulation of the data would be possible at one of only a finite range of bandwidths chosen to cover extremes anticipated by the system.

The fractal modulation transmitter can be implemented in a computationally highly efficient manner, since much of the processing can be performed using the discrete wavelet algorithms mentioned in section 3.1.3.

The transmission of finite length sequences using fractal modulation requires some basic modifications to the scheme. In fact, as initially proposed, fractal modulation is rather inefficient in this case, because successively higher frequency bands are increasingly under-utilized. In particular, we note from the time-frequency portrait if Figure 3.5 that if $q[n]$ has finite length, e.g., $q[n], n<0,n>L-1$, then the $m$th band completes its transmission of $q[n]$ and goes idle in half the time it takes the $(m-1)st$ band and so forth. However, finite length messages may be accommodated rather naturally and efficiently by modulating their periodic extensions $q[n \mod L], [43]$ thereby generating a transmitted waveform.
\[ x(t) = \sum_n q[n \mod L] \theta_n^H(t) \] (3.18)

which constitutes a periodicity-dominated bi-homogeneous signal. If we let

\[ q = \{q[0], q[1], \ldots \ldots , q[L-1]\} \] (3.19)

denote the data vector, then the time-frequency portrait associated with this signal is shown in Figure. 3.8

The final aspect of fractal modulation that remains to be considered in this section concerns the specification of the parameter \( H \). The spectral matching rule suggests that fractal modulation may be naturally suited to channels with additive \( 1/f \) noise whose degree is the same as that of the transmitted signal. The class of \( 1/f \) processes includes not only classical white Gaussian noise (\( H = -1/2 \)) and Brownian motion (\( H = 1/2 \)), but, more generally, a range of rather prevalent nonstationary noises that exhibit strong long-term statistical dependence. [43]
3.2.2 Receiver design: demodulation

The receiver problem is to recover a finite length message $q[n]$ from band-limited, time-limited, and noisy observations $r(t)$ of the transmitted waveform $x(t)$ consistent with the channel model of Figure 3.7. We assume that the noise $z(t)$ is a Gaussian $1/f$ process of
degree $H_x = H$, and that the degree $H_x$ of the bi-homogeneous signal $x(t)$ is chosen to be the same as that of the noise, i.e.,

$$H_x = H_z = H$$

(3.20)

The transmitter can exploit some of the robust and efficient parameter estimation algorithms developed in [43] to measure $H_z$ and use it to perform spectral matching. Let us consider the case of demodulating digital data, where the transmitted message is a random bit stream of length $L$ represented by a binary-valued sequence

$$q[n] \in \{+\sqrt{E_0}, -\sqrt{E_0}\}$$

(3.21)

where $E_0$ is the energy per bit. We develop a receiver that demodulates $q[n]$ so as to minimize the bit-error probability. An efficient implementation of the optimum receiver should process the observations $r(t)$ in the wavelet domain by first extracting the wavelet coefficients $r^m_n$ using the Discrete Wavelet Transform. These coefficients take the form

$$r^m_n = \beta^{-m/2} q[n \mod L] + z^m_n$$

(3.22)

where the $z^m_n$ are the wavelet coefficients of the noise process and based on the assumption that periodic replication of the finite length sequence $q[n]$ has been modulated. To simplify analysis we assume that the ideal bandpass wavelet is used in the transmitter and receiver, although we reiterate that comparable performance can be achieved when more practical wavelets are used [43].

In general, the duration-bandwidth characteristics of the channel affect which observation coefficients, $r^m_n$ are available to the receiver. In particular, if the channel is bandlimited to $2^M\omega$ Hz for some integer $M$, this precludes access to the coefficients at scales corresponding to $m > M$. Simultaneously, the duration constraint in the channel results in
a minimum allowable decoding rate of $2^{M_L}$ symbols/sec for some integer $M_L$, which precludes access to the coefficients at scales corresponding to $m < M_L$. As a result the collection of coefficients available at the receiver is

$$r = \{r^n_m, m \in \mu, n \in N(m)\}$$  \hspace{1cm} (3.23a)

where

$$\mu = \{M_L, M_{L+1}, \ldots, M_U\}$$

$$N(m) = \{0,1,\ldots,L2^{m-M_L} - 1\}.$$  \hspace{1cm} (3.23b)

This means that we have available

$$k = \sum_{m=M_L}^{M_U} 2^{m-M_L} = 2^{M_U-M_L+1} - 1$$  \hspace{1cm} (3.24)

noisy measurements of each of the $L$ non-zero samples of the sequence $q[n]$. The optimal decoding of each bit can be described in terms of a binary hypothesis test on the set of available observation coefficients $r$. Denoting $H_1$ the hypothesis in which $q[n] = +\sqrt{E_0}$ and by $H_0$ the hypothesis in which $q[n] = -\sqrt{E_0}$, we may construct the likelihood ratio test for the optimal decoding of each symbol $q[n]$. The derivation is particularly straightforward because of the fact that, in accordance with the wavelet-based models for $1/f$ processes, under each hypothesis the $z^n_m$ may be modeled as independent zero-mean Gaussian random variables with variances

$$\text{var } z^n_m = \sigma_z^2 \beta^{-m}$$  \hspace{1cm} (3.25)
for some variance parameter $\sigma^2_z > 0$. Consequently, given equally likely hypotheses (i.e., a random bit stream) the likelihood ratio test readily reduces to the test

$$
\begin{align*}
&l > 0, H_1 \\
&l < 0, H_0
\end{align*}
$$

(3.26)

where

$$
\begin{align*}
l &= \sum_{m=M_x}^{M_x} \sum_{l=0}^{2^m-M_z-1} \beta^{m/2} r^{m}_{n+lK}
\end{align*}
$$

(3.27)

is a sufficient statistic to discriminate between whether the received bit is a 1 or -1. The above summation is the so-called diversity combining strategy where we appropriately scale and accumulate all the appropriate coefficients at the receiver.

### 3.3 Extension to two-dimensional images

We now focus our attention on the construction of 2-dimensional self-similar signals (images). The extension is pretty straightforward from the one-dimensional case. We use a two-dimensional sequence $q[m,n]$ as a generating sequence. A sample 30x30 generating sequence is shown in the figure below.
Figure 3.9. A sample 2-d generating sequence

This sequence is used to generate the detail coefficients that are used in a 2-D inverse Discrete Wavelet Transform algorithm to generate the 2-D self-similar signal. Figures 3.10 and 3.11 show the detail coefficients and the corresponding image respectively.
Figure 3.10. The 3 level space-scale portrait, showing the wavelet detail coefficients, constructed using the 2-d generating sequence, $\beta = 1/4$. 
Figure 3.11. The self similar image obtained by using the inverse discrete wavelet transform algorithm on the coefficients in Figure 3.10. with the Haar wavelet.

We have used Matlab and the wavelet analysis functions from the Wavelet Toolbox to implement the algorithm and generate the figures.

Thus we now have a mechanism for synthesizing self-similar images starting with a two-dimensional generating sequence. In Chapter 5, we will show how such an image as the above is embedded into a host image as a watermark signal and then extracted at the receiver's end. The inherent redundancy of these images is exploited by using a diversity combining strategy, thus enabling a more robust extraction.
In this chapter, we present the performance of fractal modulation and demodulation for the one-dimensional case. A binary signal is embedded (modulated) into a self-similar signal. We add Gaussian noise to this signal and then try to recover (demodulate) the embedded binary data. We simulate this under various Signal-to-Noise ratio conditions and using different wavelets and measure the Bit-error rates in each case.

4.1 Synthesizing bi-homogeneous signals

We construct the self-similar signal based on the following equation from chapter 3

\[ x(t) = \sum_{m} \sum_{n} \beta^{-m/2} q[n] \psi^{m}_{n}(t) \]  \hspace{1cm} (4.1)

This equation can also be re-written as

\[ x(t) = \sum_{n} q[n] \theta^{H}_{n}(t) \]  \hspace{1cm} (4.2a)

\[ \theta^{H}_{n}(t) = \sum_{m} \beta^{-m/2} \psi^{m}_{n}(t) \]  \hspace{1cm} (4.2b)

where \( \theta^{H}_{n}(t) \) are the basis functions and are all self-similar, mutually orthogonal and have unit norm and \( m \) and \( n \) are finite positive integers.

4.1.1 Basis functions and the self-similar signal for the haar wavelet

In principle, the number of scales \( m \) and the generating sequence length \( n \) can have infinite values. However, the higher these values, the longer the computational time and storage overhead. Thus in our example, we use computationally practical values of 8 for \( n \), the generating sequence length and 5 for \( m \), the number of scales.
The following values are used for the other quantities involved.

Generating sequence \( q = [1 \ 1 \ -1 \ 1 \ -1 \ -1 \ 1 \ -1] \)

\( H = 0; \ \beta = 2^{(2H+1)} = 2 \)

Wavelet = Haar

We have used Matlab and the Wavelet Toolbox to implement the algorithm and generate figures 4.1 and 4.2.

![Figure 4.1: The self-similar basis functions \( \theta_0^0(t) \) to \( \theta_7^0(t) \) using a Haar wavelet and the other parameters mentioned above](image)

Figure 4.1: The self-similar basis functions \( \theta_0^0(t) \) to \( \theta_7^0(t) \) using a Haar wavelet and the other parameters mentioned above
4.1.2 Basis functions and the self-similar signal in case of Daubechies 5 wavelet

The following values are used for generating the signals using the Daubechies wavelet.

Wavelet = 'db5', scales m = 1, 2, 3, 4, 5

Generating sequence q = [1 1 -1 1 -1 -1 1 -1]

H = 0; β = 2^{(2H+1)} = 2

The self-similar basis functions and the corresponding self-similar signal are shown in figures 4.3 and 4.4 respectively.
Figure 4.3: The self-similar basis functions $\theta_0^0(t)$ to $\theta_7^0(t)$ using a Daubechies 5 wavelet.
4.2 Synthesizing bi-homogeneous signals using the discrete wavelet transform algorithm

Self-similar signals can be synthesized much more efficiently using Discrete-Time algorithms, specifically the DWT routine [43]. This is based on the fact that, in the equation

$$x(t) = \sum_m \sum_n \beta^{-m/2} q[n] \psi_n^m(t)$$  \hspace{1cm} (4.3)

$D_{m,n} = \beta^{-m/2} q[n]$ can be considered to be the detail coefficients of the wavelet expansion of the self-similar signal $x(t)$. 

Figure 4.4: The corresponding self-similar signal.
Using a generating sequence of length 8 (i.e. \( n = 8 \)), 3 decomposition levels or scale values, i.e. \( m = 1, 2, 3 \) and zero for the degree of self-similarity, \( H \), the self-similar signal synthesized using the Daubechies 5 wavelet is shown below in Figure 4.5.

![synthesized bihomogenous signal](image)

Figure 4.5: Self-similar signal constructed using the dwt algorithm

### 4.3 Extraction of the generating sequence (q) from the self-similar signal

The self-similar signal constructed in the previous section has the generating sequence (q) embedded (modulated) into it. This sequence can be recovered by using the diversity combining strategy described in 3.2.2. In the absence of noise the entire sequence is recovered without any errors.
Using a generating sequence of 100 bits, we simulated the above scenario for different wavelet levels and observed that the generating sequence is successfully recovered for each case, in the absence of noise.

4.4 One-dimensional fractal modulation and demodulation in the presence of noise.

We now introduce noise into the above simulation scenario and observe performance of diversity combining as a demodulation strategy. The bit error rates for various Signal to Noise ratio are shown below. Notice that the BERs decrease as you increase the number of wavelet levels used.

![Graph showing bit-error rate curves at various SNRs ( -35:2.5:20 ) and wavelet levels](image)

Figure 4.6 Bit-error rate curves at various SNRs ( -35:2.5:20 ) and wavelet levels
In Figure 4.6, the Bit-error rates are plotted versus the Signal to Noise Ratio (SNR) values in dB. The simulations are carried out for SNR values ranging from -35 dB to 20 dB, with 2.5 dB increments.

Figure 4.7 Bit-error rate curves at various SNRs (-100:5:20) and wavelet levels

In Figure 4.7 above, the Bit-error rates are plotted versus the Signal to Noise Ratio (SNR) values in dB, as well. This time the simulations are carried out over a wider range of SNR values ranging from -100 dB to 20 dB, with 5 dB increments.

In both the above simulations, we note that the curve corresponding to, when diversity combining is not used at all, is on top of all the other curves, meaning higher BERs in this case. Thus clearly diversity combining using any number of wavelet levels results in improved receiver performance. The other four curves below represent the simulation results, when diversity combining is used, using 3, 5, 7 and 9 wavelet levels during
modulation. We note that the BER values decrease (better receiver performance) as more wavelet levels are used in the fractal modulation process. This proves that as the bandwidth available increases, the transmission reliability is improved.

We have thus presented one-dimensional self-similar signals and simulated fractal modulation and demodulation performance. These simulations corroborate the established theoretical concepts. With the observations made from this exercise, we now present two-dimensional self-similar signal synthesis and their use in digital watermarking in the next chapter.
CHAPTER 5: TWO-DIMENSIONAL RESULTS

In this section we present our core watermarking algorithm, represented by the acronym IDEA, an Image Data Embedding Algorithm.

5.1 The algorithm

The watermarking algorithm has the following stages:

1. Construction of the basic watermarking unit, bunit

The basic watermarking unit, which we refer to as bunit, is the two dimensional matrix of binary data that would be modulated using a fractal modulation technique and then imperceptibly dissolved into the cover image. For our simulations we chose the following 30 x 30 matrix of binary data, shown in Figure 5.1. A sample cover image is shown in Figure 5.2.

As one would observe, it is a logo of RIT with marker data superimposed onto it. The marker data is the boundary fence around the logo and the two cross diagonals that run across the logo. This marker data is known to the receiver and is used during watermark extraction, to perform intelligent block processing, to repair blocks damaged by attacks. We note that bunit can be any arbitrary matrix of binary data. The only condition is that the marker data be superimposed on any chosen bunit and that the receiver has complete knowledge of this data. Bunit is translated and scaled to populate a wavelet decomposition pyramid, which is passed to an IDWT routine to obtain the self-similar image. Thus bunit with the appropriate scaling, represents the smallest block of coefficients in the pyramid, whose size is equal to the size of the image divided by two raised to the number of decomposition levels. Thus the size of bunit can be calculated using the following equation, expressed as a line of Matlab pseudo code below.

\[
\text{Size of bunit} = \frac{\text{size of the image}}{2^{\text{number of wavelet levels}}} \quad (5.1)
\]
Although we haven't explicitly verified the BER performance of the algorithm for bunits of arbitrary sizes and values, we do not expect them to have any influence. In our simulations we chose a cover image of size 240x240 and 3 wavelet levels, which results in a bunit size of 30x30.

Figure 5.1: bunit - the 30x30 matrix of binary data, the goal is to embed this data into a cover image and then recover as much of it as possible at the receiver end.
2. Synthesis of the 2-d self-similar watermark signal.

The next step is to use two dimensional fractal modulation on bunit and generate the 2-D self-similar (fractal) watermark signal. This involves two steps. In the first step, we populate the 2-D Wavelet Decomposition matrix, by translating and scaling bunit appropriately. This results in the matrix shown in Figure 5.3.
The 3 level wavelet decomposition matrix

Figure 5.3 The binary stealth data (bunits) are arranged as coefficients to form the wavelet decomposition pyramid, this will be used to synthesize a self-similar image that would serve as the watermark signal.

The second step is to use the 2-D Inverse Discrete Wavelet Transform algorithm and reconstruct the corresponding 2-D self-similar signal. Assuming Daubechies 3 wavelet, the resulting fractal signal is shown in Figure 5.4:
3. Addition of the watermark signal to the cover image.

The watermark signal is imperceptibly inserted into the cover image by linear addition. The exact equation used in the process, can be expressed as the following line of Matlab pseudo code.

\[ \text{wim} = \text{im} + \alpha \cdot \text{watermark} \cdot \text{pim}; \]  

(5.2)

where, im is the original gray scale cover image. It essentially is a matrix with values between 0 and 1. wim is the resulting watermarked image, and has values that extend slightly beyond 0 and 1, depending on the amount of watermark signal added to the cover.
image, alpha is the watermark strength and is equal to 0.001 for the simulations presented here, watermark is the 2-D self-similar watermark signal. pim is a linear prediction of the cover image im and is obtained by low-pass filtering im using the following 3x3 filter, [0 1/4 0; 1/4 0 1/4; 0 1/4 0]. The resulting watermarked image versus its original image is shown below in Figure 5.5, while the original image versus its prediction is shown in Figure 5.6. The error signal, which is essentially wim – im, has values of the orders of 10^-2.

Figure 5.5: The original cover image versus its watermarked version; note that they are both perceptually identical; however, the error indicates that they are not, numerically and this error accounts for the watermark signal, which holds the embedded binary data.
4. Extraction process

This is a 5-stage process. The 5-stages are

1. Image self-noise suppression to obtain an estimate of the bi-homogeneous watermark signal. The first step in the detection process is to obtain an estimate of the original image, by using a prediction filter on the watermarked image. This predicted image, shown in figure 5.7, will be used for Image self-noise suppression. The exact equation used in this process can be expressed as the following line of Matlab pseudo code.

\[
\text{estimate} = \frac{\text{watermarked image} - \text{predicted image}}{\text{predicted image}} \tag{5.3}
\]
This estimate (or the remnant image) is shown in Figure 5.8.

Figure 5.7: A low pass estimate of the original image, based on the watermarked image
Figure 5.8: The remnant image after the self-noise has been cancelled out from the watermarked image.

2. Wavelet decomposition of this estimate
The next step is to analyze the remnant image using the Discrete Wavelet Transform. This reveals the embedded binary logo and is shown in Figure 5.9
3. Threshholding of the Wavelet coefficients obtained above.

We use Wavelet Threshholding on the coefficients from the previous step and clean the matrix up a little bit. Specifically we zero out all the coefficients greater than an empherically obtained threshold of .025. The logo is now more significantly evident in areas where it has survived and is shown in Figure 5.10.
Figure 5.10: The result of wavelet thresholding

4. Intelligent block processing (cleanup) of the (thresholded) Wavelet decomposition matrix. We do intelligent block processing and decide on the usefulness of each block. This involves the following two steps.

1. Check if the watermarked image has undergone blurring and if yes, then undo the effect that blurring would have had on the 30x30 block. This is done by extracting the coefficients along the diagonals (leading and non-leading) and then comparing them with prior-known values.

2. Check if the 30x30 block is reliable or not. If not just discard it (Thus not use it in diversity combining). We do this by extracting the coefficients along the border of the 30x30 Block) and then by comparing them with prior-known values. If the comparisons yield an error-rate greater than a threshold, then the block is discarded.
The result of this process is shown in Figure 5.11

Figure 5.11: The coefficient matrix after intelligent block processing

5. Diversity Combining to extract an estimate of bunit.
Finally we use the "Diversity Combining" strategy to extract an estimate of the original binary data that was embedded. This estimate is shown in Figure 5.12. The BER (Bit error rate) in this case is 0.0111 (~10 bits in 900)
Figure 5.12: The extracted bunit
5.2 The attacks on the algorithm

In this section, we present possible attacks that a watermarked image can be subjected to. This is only a small subset of the several many attacks one may attempt in reality.

We have simulated our algorithm for the following attacks:

1. Cropping
2. Blurring
3. Circular Translation
4. High SNR Gaussian noise
5. Image compression
6. Image Rotation
7. Zooming
8. Sharpening
9. Stirmark attacks

We have excellent results against cropping, blurring and circular translation. We have very good results in the presence of high SNR (~40 dB) gaussian noise). Against the more ambitious rotation, zooming, sharpening and Stirmark attacks, the performance is very poor.

For each of the above attacks, we report the attacked image, the recovered data and the bit-error rate of the recovered data in Figures 5.13 to 5.30.
1. **Cropping** Figure 5.12 shows a sample cropped watermarked image. The data extracted from it is shown in Figure 5.13

![Cropped Image](image1.png)

Figure 5.13: A cropped watermarked image

![Cropped Image](image2.png)

Figure 5.14: The extracted binary data (bunit). BER = 0.1556
2. **Blurring (using a 3x3 window)** Figure 5.14 shows a watermarked image blurred using a 3x3 window. The data extracted from it is shown in Figure 5.15.

![Blurred Image](image.png)

**Figure 5.15:** The blurred watermarked image

![Extracted Data](image.png)

**Figure 5.16:** The extracted binary data (bunit), BER = 0.1022
3. **Translation** Figure 5.16 shows a cyclically translated watermarked image. The data extracted from it is shown in Figure 5.17

![Translated Image](image1)

**Figure 5.17:** The translated watermarked image

![Extracted Binary Data](image2)

**Figure 5.18:** The extracted binary data (bunit), BER = 0.0122
4. **High SNR (40 dB) Gaussian Noise** Figure 5.19 shows a noisy watermarked image. The data extracted from it is shown in Figure 5.20.

![Noisy Image](image1)

*Figure 5.19: Noise added to the watermarked image*

![Extracted Binary Data](image2)

*Figure 5.20: The extracted binary data (bunit), BER = 0.1167*
5. **JPEG Compression** Figure 5.21 shows a JPEG Compressed watermarked image. The data extracted from it is shown in Figure 5.22.

![JPEG compressed image](image1)

**Figure 5.21: Compressed watermarked image**

![Extracted binary data](image2)

**Figure 5.22: Extracted binary data (bunit), BER = 0.4389**
6. Image Rotation Figure 5.23 shows a rotated watermarked image. The data extracted from it is shown in Figure 5.24.
7. **Zooming** Figure 5.25 shows a watermarked image that has been zoomed into and then cropped. The data extracted from it is shown in Figure 5.26.

![Zoomed and cropped image](image1.png)

**Figure 5.25:** The zoomed watermarked image

![Extracted binary data (bunit), BER = 0.4778](image2.png)

**Figure 5.26:** Extracted binary data (bunit), BER = 0.4778
8. **Sharpening** Figure 5.27 shows a watermarked image that has been sharpened.

The data extracted from it is shown in Figure 5.28

![Sharpened Image](image)

Figure 5.27: Sharpened watermarked image

![Extracted binary data](image)

Figure 5.28: Extracted binary data (bunit), BER = 1
9. **Stirmark attacks** Figure 5.29 shows a watermarked image that has been subjected to a stirmark attack. The data extracted from it is shown in Figure 5.30.

![Stirmark attacked image, 10° rotation](image)

**Figure 5.29:** Stirmark attacked watermarked image, the attack causes a 10-degree rotation.

**Figure 5.30:** Extracted binary data (bunit), BER = 1
5.3 Summary of results

We have tested our algorithm on several different images. We have selected images with different characteristics, ranging from highly smooth images to moderately textured to very high texture or edgy images. The following is our test set. Each of these images are shown in Figures 5.31 and 5.32

1. hare
2. man
3. barbara
4. lena
5. bear
6. kids
7. newyork
8. flowers

With each image we performed the following simulations
1. Watermark insertion and extraction in the absence of any attacks.
2. Watermark insertion and extraction in the presence of each of the nine attacks.

We use the Bit error rate (BER) between the inserted binary data unit (bunit) and the extracted binary data as our metric and report the results for each of the images in tables 5.1 and 5.2.
Figure 5.31: Test images
Figure 5.32: Test images continued

Table 5.1: Bit error rates for different images under different attack conditions

<table>
<thead>
<tr>
<th>Test Image</th>
<th>No attack</th>
<th>Cropping</th>
<th>Blurring</th>
<th>Translation</th>
<th>Noise (40dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>hare</td>
<td>0</td>
<td>0.0033</td>
<td>0.0300</td>
<td>0</td>
<td>0.0189</td>
</tr>
<tr>
<td>man</td>
<td>0.0044</td>
<td>0.0978</td>
<td>0.0611</td>
<td>0.0022</td>
<td>0.0489</td>
</tr>
<tr>
<td>barbara</td>
<td>0.0078</td>
<td>0.0133</td>
<td>0.2678</td>
<td>0.0056</td>
<td>0.1389</td>
</tr>
<tr>
<td>lena</td>
<td>0.0111</td>
<td>0.1556</td>
<td>0.1022</td>
<td>0.0122</td>
<td>0.1167</td>
</tr>
<tr>
<td>bear</td>
<td>0.0644</td>
<td>0.0744</td>
<td>0.2133</td>
<td>0.0600</td>
<td>0.2067</td>
</tr>
<tr>
<td>kids</td>
<td>0.0756</td>
<td>0.4678</td>
<td>0.3756</td>
<td>0.1033</td>
<td>0.6078</td>
</tr>
<tr>
<td>newyork</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>flowers</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>
Table 5.2: Bit error rates for different images under more intense attack conditions

<table>
<thead>
<tr>
<th>Test Image</th>
<th>Compression</th>
<th>Rotation</th>
<th>Zooming</th>
<th>Sharpening</th>
<th>Stirmark</th>
</tr>
</thead>
<tbody>
<tr>
<td>hare</td>
<td>0.1833</td>
<td>1.0000</td>
<td>1.0000</td>
<td>0.3789</td>
<td>1.0000</td>
</tr>
<tr>
<td>man</td>
<td>0.3844</td>
<td>1.0000</td>
<td>1.0000</td>
<td>0.3922</td>
<td>1.0000</td>
</tr>
<tr>
<td>barbara</td>
<td>0.4989</td>
<td>1.0000</td>
<td>1.0000</td>
<td>0.3944</td>
<td>1.0000</td>
</tr>
<tr>
<td>lena</td>
<td>0.4389</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>bear</td>
<td>0.4333</td>
<td>0.4522</td>
<td>1.0000</td>
<td>1.0000</td>
<td>0.4933</td>
</tr>
<tr>
<td>kids</td>
<td>0.3867</td>
<td>0.4322</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>newyork</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>flowers</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

In the absence of any attack, the algorithm performs very well with all the images, except the newyork and flowers images. These two images are highly edgy or textured (high frequency content) as opposed to the others, which are either smooth or moderately textured. Thus the results obtained in case of these two images are not very meaningful and are 100% in error. This is expected and is due to the fact that edges are much harder to predict. Our algorithm relies on a prediction filter for image self-noise suppression. The prediction is very good on smooth and moderately textured images, but fails in case of images with lot of edges or texture.

At the receiver’s end, during the intelligent block processing stage in the detection process, with these two images, all the blocks are being discarded, as all of them are pretty much useless. Thus the wavelet decomposition matrix has all zeros. And subsequently, the statistic matrix is also all zeros, and the likelihood ratio test decodes all these to be 1’s. We have documented the BERs as 1 for these two images, to indicate the fact that they lack the conventional meaning, unlike the other image cases.
With the seven smooth and moderately textured images, the performance is very good in case of cropping, blurring, translation and additive noise attacks. Cropping essentially means that a shorter segment (or duration) of the signal is available to work with at the receiver. However, since our signal is self-similar and is present in different wavelet levels (higher bandwidth), we are able to tap into this redundancy, to still extract meaningful estimates of the embedded data. The detection algorithm is able to withstand moderate amounts of blurring, which is being predicted and inverted during the intelligent coefficient block processing stage. In the case of translation, the image is cyclically translated, which results in only a similar translation of the coefficient matrix at the receiver, and not any bit errors. High SNR Gaussian noise does not do much damage and thus the algorithm is able to withstand this as well.

We note that the algorithm performs moderately against Compression and Sharpening, but fails completely against Zooming, Rotation and Stirmark attacks. In case of all these attacks, the wavelet decomposition matrix of the watermark signal estimate, gets completely unsettled and the data that results subsequently is of no value whatsoever to the receiver.
CHAPTER 6: CONCLUSIONS AND FUTURE WORK

In this work, we introduced the related fields of digital watermarking and steganography, which together fall under the bigger umbrella of data hiding. While they are both conceptually similar, a big difference between the two is the additional notion of robustness for digital watermarking. A generic scheme for data embedding and extraction has been presented, along with the algorithm requirements and design issues. Watermark imperceptibility and robustness are for many applications the most important requirements.

In chapter 3 we presented the deterministic self-similar signal theory, on which our data-embedding algorithm is based. We have seen the algorithms used to synthesize bi-homogeneous signals and how these signals can be used as modulating waveforms in a communications system, leading to the term fractal modulation. We then presented the extension of these concepts to images, which involves using a two-dimensional generating sequence and the two-dimensional discrete wavelet transform algorithms. We simulated one-dimensional fractal modulation and obtained results that corroborated the theoretical concepts in chapter 4.

Chapter 5 presented the image data-embedding algorithm that uses fractal modulation in the embedding stage and diversity combining in the extraction stage. Fractal modulation is used to construct the binary data dependent watermark signal. We then tested the algorithm performance with several images and also under different attack conditions and reported and discussed the results. Clearly the algorithm performance depends on the nature of the images. The performance is good in case of smooth and moderately textured images. With highly textured or edgy images, the algorithm fails completely. In case of attacks, the algorithm is able to withstand cropping, blurring, translation and additive noise attacks. However it does not perform very well with Zooming, Rotation and Stirmark attacks. This would be a good starting point for future research, to work on strategies to predict and invert these attacks. Another good extension would be in the area of image analysis. In our work we have assumed a fixed H, the degree of homogeneity
for the watermark signal. It would be interesting if the cover image could be analyzed and its degree of homogeneity determined using some of the models developed in [20]. This value can then be used as the degree of homogeneity for the watermark signal and could lead to better receiver performance.
Missing Page
BIBLIOGRAPHY


USA, November 1996. IS&T, The Society for Imaging Science and Technology and SPIE, The International Society for Optical Engineering, SPIE.


47. [http://www.aliroo.com](http://www.aliroo.com)

48. [http://www.digimarc.com](http://www.digimarc.com)